ИЗГРАЖДАНЕ НА МРЕЖОВА СИГУРНОСТ С ENDIAN FIREWALL CE

Александър Цокев

София, 2015

Българска асоциация за развитие на знанията за информационните и комуникационни технологии

[1] http://creativecommons.org/licenses/by-sa/4.0/

Искам да изкажа специални благодарности на колегите Радослав Милчев и Йордан Йорданов, които ми помогнаха при редактирането на книгата и специално на Вилизара Лишева за проявеното търпение и подкрепата.

Съдържание

Глава 1. Въведение в мрежовата сигурност

Въведение

През 2010 година е открит компютърен вирус, наречен Stuxnet, чиято цел се различава коренно от познатите до момента зловредни програми (вируси, червей и троянски коне). Вместо да атакува най-често използваните потребителски операционни системи и програми Stuxnet се опитва чрез софтуерния пакет Siemens STEP 7 да доведе до проблеми в индустриалните устройства и специализирани системи за управление. Приложеният метод е чрез STEP 7 да се изпращат грешни стойности към центрофуги за обогатяване на уран, като към инженерите се извеждат нормални и предварително зададени коректни данни. Това води до срив на почти 20% от производството на обогатен уран в Иран. В пресата има различни твърдения за това кой е разработил Stuxnet, като една от версиите, е че това са тайните служби или на САЩ или на Израел, а основната цел е Иранската ядрена програма. Въпреки, че това не е първият случай на атака към индустриална комуникационна инфраструктура, това е първият открит случай на директна заплаха за индустриалните контролери и системи за наблюдение.

От програмна гледна точка Stuxnet е изключително интересен и поставя нови правила при разработването на злонамерен код, като използва над 20 пропуска в сигурността и изключително сложен програмен механизъм за анализ, сглобяване на изпълнимия код и прикриване на следите. Разпространението на Stuxnet започва чрез специално разработен товар (payload), чиято единствена цел е софтуера Siemens STEP 7. Използвайки приложението STEP 7 вирусът успява да препрограмира контролерите, и чрез извеждане на заблуждаващи данни към SCADA (системи за автоматизирано наблюдение и управление на производствени процеси) да заблуди инженерите от поддръжката, че всички параметри са в нормите. Различни версии на вируса атакуват пет отделни Ирански организации и въпреки твърдението на Siemens, че вирусът не е нанесъл щети на техни клиенти, от Иран потвърждават проблемите, свързани с Stuxnet. На фиг. 1.1 е показано разпространението на този зловреден код по държави.

Фиг. 1.1 Разпространение на Stuxnet

Въпреки, че Stuxnet е открит и са налични начини за предпазване от него, има още една изключително сериозна опасност – неговият програмен код може да бъде открит в Интернет, което означава, че самият вирус може да бъде променен и е въпрос на време да бъде използван отново. През 2012 от Symantec обявяват за открит злонамерен софтуер, който като принцип на

работа е свързан с Stuxnet и наречен Flame. Flame се разпространява през локалните офис мрежи или USB памети и позволява да се записва звук, работните екрани на компютрите, натиснатите бутони на клавиатурата и мрежовия трафик. Също така като цел на този злонамерен код е Skype, като от него се записват разговорите, а ако компютърната системи има и поддръжка на Bluetooth се прави опит за неговото активиране и свързване с периферни устройства. Според Kaspersky този код е инфектирал над 1000 устройства в правителствени и други организации, свързани с обучение и отбрана, като над 65% от устройствата се намират в Иран, Израел и Палестина.

Flame е типичен пример за изключително сложен злонамерен код, с размер над 20 MB, разработен на C++, с цел събиране на информация и разузнаване. Интересен е и факта, че веднага след обявяването, че Flame е открит той спира да функционира.

От посочените примери ясно се вижда необходимостта от защита на комуникационните канали и данните от неоторизиран достъп и от атаки, а след откриването на Stuxnet вече официално е налична и кибервойна.

Необходимост от защита на комуникацията

От примера за Stuxnet може да се направи извода, че колкото и да е сложна и надеждна дадена система или комуникационна инфраструктура тя винаги може да бъде атакувана, а при налични уязвимости и успешно заобиколени модулите и системите за защита.

Ако направим аналог между защитата на компютърните мрежи и средновековните методи за водене на отбранителни действия може да се каже, че до скоро нещата са били сравнително аналогични. Средновековните замъци са силно укрепени с надеждни стени, ровове и един или няколко добре охранявани и наблюдавани входа, които могат лесно и бързо да бъдат затворени. До преди няколко години защитата на локалните мрежи използва подобен принцип – периметъра на мрежата (най-често маршрутизаторите, които осигуряват свързаност с доставчици на Интернет или WAN) се последните устройства, които отделят вътрешните мрежи от заплахите, идващи от Интернет и други външни устройства. Аналогията със средновековието е, че при този модел винаги “добрите” системи трябва да са скрити от защитни стени и други специализирани устройства и технологии от “лошите”, които биха атакували извън периметъра. Още една подобна аналогия е и троянския кон, който в информационните и комуникационни технологии се явява и отделен вид зловреден програмен код.

С напредъка на технологиите, появата на облачните услуги, смартфоните и мобилния достъп до Интернет концепцията за граница или периметър на мрежата отпада. В момента е нормално потребителите да достъпват корпоративни данни от всяка точка на света, по всяко време от различни по вид устройства, използващи различни операционни системи. Cisco Systems® наричат този модел “мрежа без граници” (borderless network). Този нов подход при комуникацията поставя още по-строги критерии и изисквания към изграждане на надеждна и сигурна технология за защита както на данните, така и на комуникационните канали.

За да се дефинира нуждата от защита на комуникацията и данните, може да се направи един кратък преглед и на разпространението на червея Code Red. През 2001 в рамките на 24 часа от своето пускане в Интернет този зловреден код успява успешно да зарази над 350000 устройства, като освен отказаният достъп до изключително голям брой WEB сървъри се наблюдава и значително намаляване на скоростта на трансфер в много мрежи по цял свят. Това е типичен пример за атака от тип отказ на услуги – Denial of Service (DoS). Ако трябва да се анализират причините за успешната атака на Code Red може да се направи следното обобщение:

- В програмния код на WEB сървърите има уязвими модули, дължащи се на грешки на програмистите (bug) или на лош софтуерен дизайн;
- Ако е имало налични обновявания на софтуерните пакети на WEB сървърите те не са били инсталирани;
- Не е имало политика за защита от подобен тип атаки, въпреки наличните технологии, които биха позволили блокирането на този червей.

Всеки успешен пробив на системата за защита може да доведе до значителни проблеми, свързани с бизнес процесите, като:

- Кражба на важна фирмена информация;
- Отказ на услуги, свързани с банкиране, продажби и др.;
- Проблеми, свързани с производствените процеси;
- Заплаха за здравето на човека (при здравеопазването, индустрията и транспорта);
- Уронване на престижа на компаниите и много др.

Изграждането на защита на мрежовата комуникация предпазва корпоративни данни и системи от неоторизиран достъп, но това изисква добре обучен персонал, технологии за защита и постоянно обновяване, както на знанията на специалистите и потребителите, така и на програмните и хардуерните средствата.

Много често в литературата, свързана с мрежовата и компютърна сигурност се използва определението “Необходимостта е майка на изобретението” (Necessity is the mother of invention). Това се отнася изключително силно за ИТ сигурността – първоначално не се е обръщало голямо внимание на тази проблемна област, липсата на Интернет и сравнително ограничените възможности на компютърните системи не водят до потенциални заплахи.

С развитие на мрежовата комуникация и Интернет първоначално потребителите не са под риск, но с появата на първите мрежови вируси, червей и DoS атаки става ясно, че е необходимо да се вземат сериозни мерки за защита както на работните станции и крайните системи, така и на данните и мрежовите устройствата.

В момента Интернет е на всякъде и всекидневно разчитаме на тази глобална мрежа за неща, свързани с бизнеса, забавлението и достъпа да информация. Почти няма модерни бизнес процеси, които да не разчитат на Интернет и отдалечена комуникация с други партньори. С по-широкото въвеждане на IPv6 и “Интернет на нещата” (Internet of Things) достъпът до глобалната мрежа ще бъде не само от потребителски устройства, но и от изключително голям брой сензори и специализирани системи.

Интересно е и развитието на методите и инструментите за атаки, ако само преди 35 години се изисква злонамерените атаки да се правят от лица, с много добри познания по програмиране и компютърни технологии, то към момента има изключително сложни и мощни инструменти, които се стартират само с едно натискане на бутон от интуитивен графичен интерфейс, като в Интернет може да се намерят множество описания и примери за тяхното използване.

От изложеното до тук може да се направи заключението, че **изграждането на системата за защита на комуникацията и на данните е приоритет и задължително условие за безпроблемното функциониране на бизнес процесите, както на малки, така и на големи компании**.

Исторически факти

В исторически план развитието на технологиите за мрежова защита започва в началото на 80^{те} години на 20^{ти} век. През 1984 година компанията SRI International публично обявява своята технология, наречена “Intrusion Detection System” (IDS). IDS позволява в реално време да

се получи информация за извършвана атака, но не и тя да бъде блокирана. За да се реализира тази функционалност се използва копиране на трафика и анализ чрез сравнение със сигнатури[2]. В последствие през 1988 година от Digital Equipment Corporation (DEC) пускат за публична употреба първата защитна стена (Firewall), която работи на принципа на пакетното филтриране. За разлика то IDS при пакетното филтриране се анализират полетата в хедъра на пакета и на база на получената информация се взима решение дали той да бъде пропуснат или не. Една година по-късно Bell Labs развиват идеята за пакетно филтриране и добавят алгоритми за следене на сесиите, като технологията за защитните стени се развива до т.нар. “stateful firewall”. Аналогично на пакетното филтриране и при тази технология се използват предварително дефинирани правила за ограничаване на трафика през устройството, но се следи и състоянието на сесията между двете системи. През 1991 година DEC успешно реализират защитна стена, която може да анализира трафика до приложното ниво на OSI референтния модел – “Application Gateway Firewall”.

Технологията IDS остава без значителни промени до 1999 година, когато е пусната в експлоатация IPS (Intrusion Prevention System). За разлика от IDS, IPS не използва копие на трафика и анализира данните отново чрез сигнатури, но при открита аномалия или заплаха блокира пакетите в реално време.

Актуалната към момента технология за защитните стени използва подход за филтриране чрез зони, а поради размитите граници на мрежите се преминава и към облачно базирани (cloud-based) системи за защита - например TrustSec[3] на Cisco Systems®.

С развитие на методите и средствата за атаки е логично да се наблюдава и усъвършенстване на средствата за предпазване от тях. **За специалистите, работещи в областта на сигурността на комуникациите е изключително важно да са запознати с най-актуалните тенденции, методи и средства както за атаки, така и за защита.**

Фундаментални принципи

Първоначално по-големият брой заплахи и атаки идват извън границите на локалната мрежа, но в последствие се оказва изключително важно да се подсигурят и анализират за потенциални атаки и пакетите, които се генерират във вътрешните мрежови сегменти.

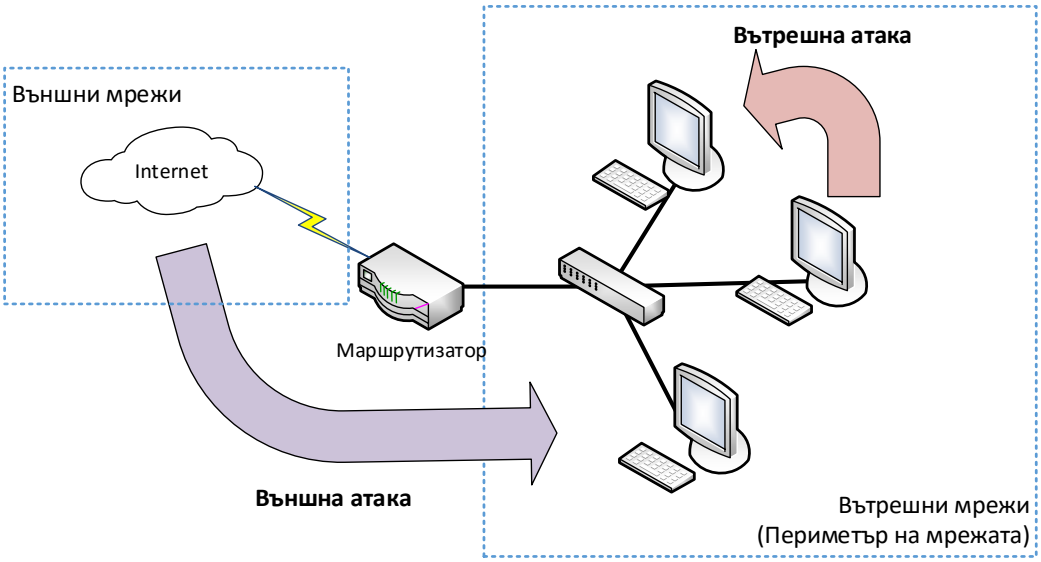
Условно може да се приеме, че атаките се разделят на два типа (фиг. 1.2):

1. **Външни атаки** – източникът на атаката се намира извън периметъра на разглежданата мрежа;
2. **Вътрешни** - атаката е стартирана от мрежови сегмент или система, която се намира в периметъра на дадената мрежа.

Както вече беше споменато в момента има размиване на границата на мрежите – използват се облачни услуги, виртуални частни мрежи (VPN), мобилни устройства и др., като това води до по-трудно дефиниране дали атаката е вътрешна за мрежата или външна.

[2] Виж Глава 8.

[3] http://www.cisco.com/c/en/us/solutions/enterprise-networks/trustsec



Фиг. 1.2 Вътрешни и външни заплахи при мрежовата комуникация

Външните атаки могат да бъдат различни видове, като някой от по-честите са:

- Сканиране на портове, операционни системи, използвани услуги и др.;
- Опити за пробив на системите за сигурност чрез атака на потребителски или административни пароли;
- Използване на технологични или други пропуски (bug) за получаване на неоторизиран достъп до устройства;
- Отказ на услуги - DoS (Denial of Service) и DDoS (Distributed Denial of Service);
- SPAM и др.

Вътрешните атаки са изключително опасни, поради по-трудното защитаване на устройствата и протоколите, които работят на каналното ниво на OSI референтния модел. Много пъти заразено вътрешно устройство (хост) се използва за стартиране на масирани атаки, които могат да включват:

- Сканиране на мрежови ресурси, протоколи и системи;
- Подслушване и събиране на пакети;
- DoS;
- Пренасочване на портове;
- DNS spoofing;
- ARP poisoning;
- Опити за неоторизиран достъп до ресурси, чрез атака на парола или потребителски профил и много др.

Друго разделяне на мрежовите атаките по следните два критерия:

1. **Структурирана** – целенасочена атака, изпълнена от силно мотивирани и опитни лица, насочена към конкретна организация или система;
2. **Не-структурирана** – случайни опити за атака, които нямат точно дефинирана цел.

Ако класификацията на видовете мрежови атаки се отнесе към мрежовата комуникация може да се направи обобщението, показано на фиг. 1.3.

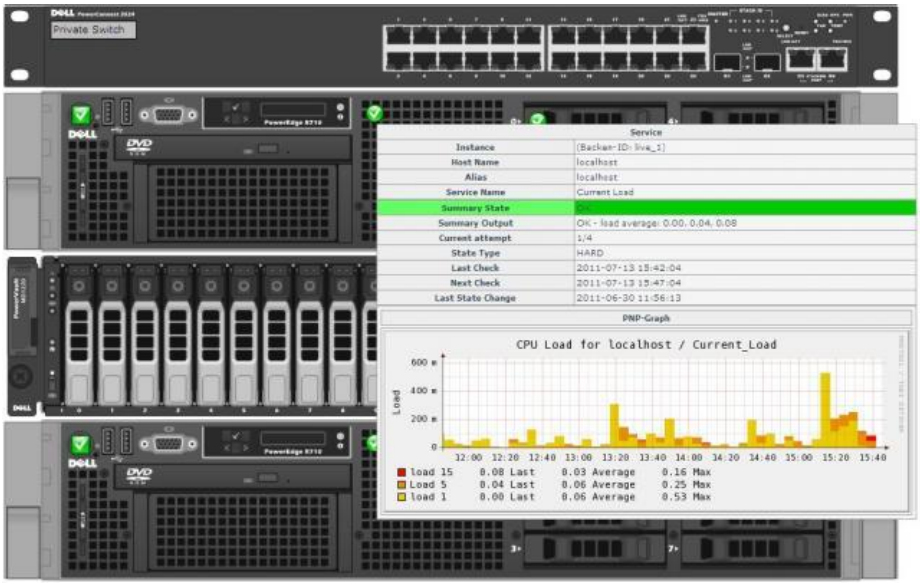
Фиг. 1.3 Обобщено представяне на видовете атаки, насочени към компютърните мрежи

Изграждането на надеждна защита изисква задълбочени познания и внимателно проучване на топологията на мрежата, използваните протоколи и системи, както и непрекъснато наблюдение и подобрение. Като база за проектиране и конфигуриране на защитните системи се използва набор от документи, наречен “фирмена политика за сигурност” (Security Policy), който трябва да имат ясно и еднозначно дефинирани всички аспекти, свързани с подсигуряването на мрежовата комуникация.

Процесът на изграждане, наблюдение и повишаване на степента на защита е цикличен и е показан на фиг. 1.4.

Фиг. 1.4 Цикличен процес на подсигуряване на мрежова комуникация

Фирмената политика за защита, която се изгражда на база на общоприети стандарти и нуждите на съответната организация включва набор от документи, които трябва да бъдат допълнително анализирани за тяхната точност и обхват. Тези документи се използват като база за проектиране, изграждане и конфигуриране на устройствата за защита на комуникационните процеси. Етапът на подсигуряване завършва с конфигуриране на всички необходими системи и описание на направените настройки. Задължително е да се извършва наблюдение на работата на системите за подсигуряване на мрежовата комуникация, като за целта се използва специализиран софтуер за анализ на трафика в реално време, както и периодична проверка на журналите (log).



Фиг. 1.5 Наблюдение на мрежовата комуникация с Nagios[4] и Nagvis[5] (източник Интернет)

Периодично се извършва проверка на текущото състояние на сигурността на мрежата, като за целта се използват специализирани инструменти, а в определени случай проверката може да се извърши от външни организации или трети лица.

Един от най-популярните пакети за анализ на сигурността е Kali Linux[6] – безплатна Linux дистрибуция, която включва стотици специализирани и изключително мощни пакети за анализ на защитата на комуникациите. Kali Linux е наследника на BackTrack, като някой от най-съществените разлики са оптимизираната система за инсталиране и обновяване на включените инструменти, възможността за инсталиране на Kali на мобилни устройства (Kali NetHunter), включените инструменти и др.

[4] http://www.nagios.org
[5] http://www.nagvis.org
[6] http://www.kali.org

Фиг. 1.6 Kali Linux (източник Интернет)

На база на резултатите от проверката на мрежовата сигурност е необходимо да се направи обобщение и да се подобри защитата, като отново след това се премине към етапа на конфигуриране.

Важно е да се отбележи, че тези четири стъпки трябва да се извършват периодично, поради основната причина, че инструментите, използвани от злонамерените лица (хакери и др.) непрекъснато се развиват, както и постоянното откриване на нови пропуски в защитните системи и методи за тяхното използване. Ако е проектирана и конфигурирана защита и не се извършва наблюдение и ново подсигуряване само в рамките на няколко седмици или дни (в зависимост от популярността) може да си получи успешен опит за пробив на сигурността, който да доведе до сериозни последствия за компанията и отделните информационни и бизнес процеси.

Редица институции и организации в световен мащаб работят по проблеми, свързани със защитата на комуникациите, като някой от най-известните са:

- SysAdmin, Audit, Network, Security (SANS) Institute – www.sans.org;
- Computer Emergency Response Team (CERT) – www.cert.org;
- International Information Systems Security Certification Consortium – www.isc2.org;
- ICSA Labs – www.icsalabs.com;
- Forum of Incident Response and Security Teams – www.first.org;
- Center for Internet Security - www.cisecurity.org;
- Information Systems Security Association – www.issa.org и др.

Области за защита на мрежовата комуникация

Защитата на мрежовата комуникация обхваща следните области:

1. **Анализ на риска** – извършва са качествен и количествен анализ и оценка на риска и щетите, ако дадена атака е проведена успешно;
2. **Фирмена политика за защита** – документи, описващи защитата на комуникациите, отговорностите на служителите, използваните средства и дефиниращи типовете данни (включително потоците от информация) и тяхното ниво на поверителност;

3. **Организиране на защитата на информацията** – съдържа обобщен модел на защитата на информацията, на база на отделните бизнес процеси;
4. **Управление на активите** – класифициране на инвентара, информацията и др.;
5. **Защита на служителите** – дефинира процедурите, свързани със защитата на информацията при напускане, преместване или уволнение на служители;
6. **Физическа защита на помещенията** – описва защитата на отделните работни помещения, както и на мрежовата инфраструктура от неоторизиран достъп и намеса;
7. **Управление на комуникационните процеси** – областта дефинира управлението и използваните средства за мрежова защита;
8. **Достъп** – описва изискванията и нивата на достъп до отделните системи, данни и услуги;
9. **Информационни системи** – широка област, която обхваща сигурността на отделните приложения и специализирани информационни системи (съвкупност от хардуерни и софтуерни модули);
10. **Инциденти** – описва процедурите при инциденти, свързани с пробив на системите за защита на комуникациите или при загуба и изтичане на данни;
11. **Непрекъснатост на бизнес процесите** – описват се процедурите за осигуряване на непрекъснати, надеждни и сигурни бизнес процеси;
12. **Съвместимост** – дефинира процесите и изискванията за съвместимост с общоприети стандарти и добри практики.

Тези 12 области се използват като база за проектиране и развитие на комуникационната сигурност и са застъпени във фирмената политика за защита.

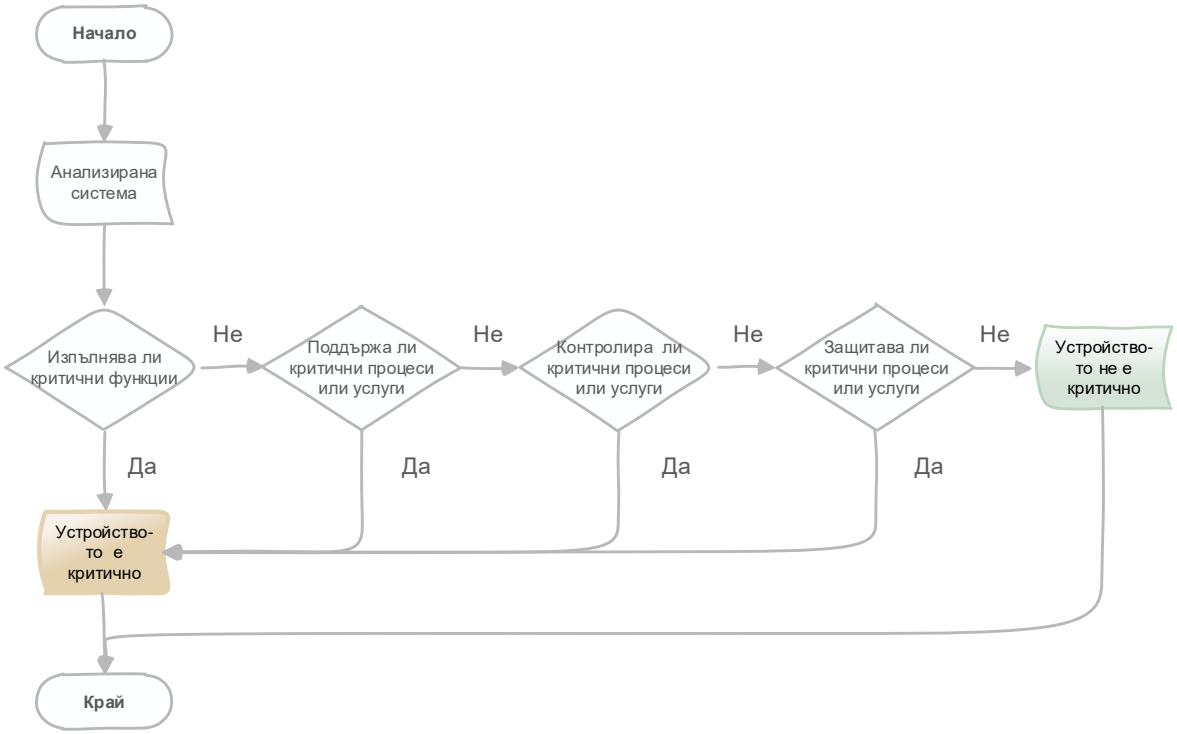
За фирмената политика за защита може да се обобщи, че това е изключително важен документ, обхващащ горепосочените 12 области (при по-малки организации може и само някой от тях), който изисква внимателно проучване и разработване от екипи, които имат необходимата подготовка и познания. Отново е необходимо да се извършва периодичен анализ за текущата актуалност и обхват на тези документи и при необходимост да се нанесат корекции.

Анализ на сигурността и риска

За да се определят критичните зони, процеси и устройства за дадена комуникационна инфраструктура е необходимо внимателно провеждане на задълбочен анализ, който да даде отговор на въпроси като:

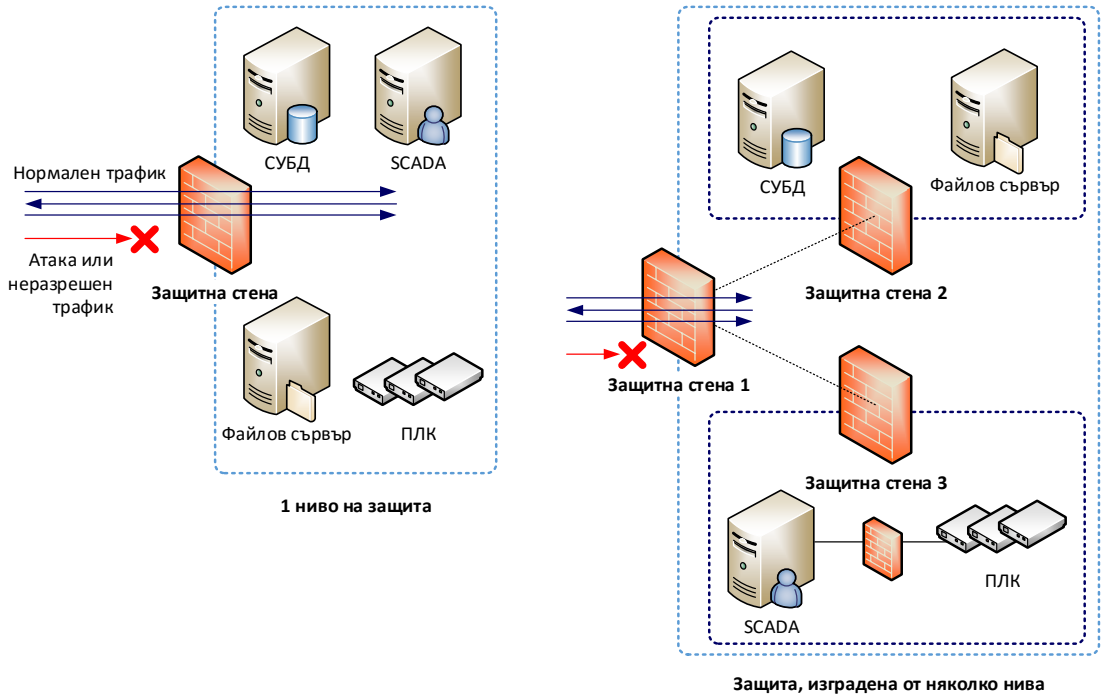
- Какво притежава дадената организация, което може да е от интерес за други лица и до което те нямат оторизиран достъп?
- Кои са критичните бизнес и комуникационни процеси, устройства и приложения?
- Какво може да причини срив и спиране на бизнес процесите?
- Какво би станало, ако системите за защита на комуникационните процеси бъдат заобиколени?
- Колко бързо ще се възстанови нормалното протичане на бизнес процесите при успешна атака от страна на злонамерени лица?

Един от по-важните етапи при проектирането и конфигурирането на мрежовата защита е свързан с определяне на критичните системи и услуги. В зависимост от големината и типа на организацията те могат да се различават, но обобщен алгоритъм за тяхното определяне е показан на фиг. 1.7.



Фиг. 1. 7 Алгоритъм за определяне на критични ресурси

След като бъдат определени критичните устройства е необходимо да се анализира и кои процеси изцяло или частично зависят от тях. По този начин може да се въведе приоритет за отделните критични системи и да се извърши подсигуряване на комуникацията на база на определен брой нива (виж. Фиг. 1.8).



Фиг. 1.8 Изграждане на защита на комуникацията с едно и няколко нива на основно подсигуряване на мрежовия трафик

Изграждането на защита на комуникацията на няколко нива изисква внимателно сегментиране на мрежовата инфраструктура на база на параметри като:

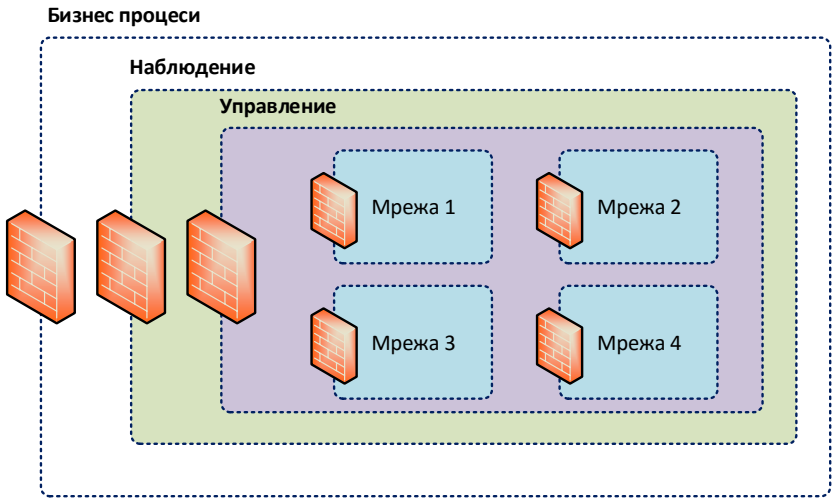
- Тип на използваните услуги;
- Местоположение на устройствата;
- Комуникационни протоколи;
- Адресиране на устройствата;
- Използвани VLAN (брой и предназначение);
- Бюджет и възможности на техническия персонал;
- Наличие на дублирани връзки между мрежови сегменти и др.

Основни модели за подсигуряване на мрежовата комуникация

При изграждане на системата за защита на комуникационни системи много често се използва втория подход (няколко нива), който позволява по-надеждно да се подсигурят отделните мрежови сегменти, работни екипи, информационни системи и бизнес процеси.

Въпреки, че в примера на фиг. 1.8 се използват няколко отделни защитни стени за филтриране на трафика между отделните сегменти, това не винаги е оптимално и в някои случаи е по-целесъобразно да се използва една единствена защитна стена с дефинирани отделни зони (Zone-based Firewall) и VLAN. От своя страна това води до намаляване на разходите, но усложнява конфигурирането и поддръжката. Също така ако се инсталира едно единствено защитно устройство то се явява критично и е необходимо неговото дублиране.

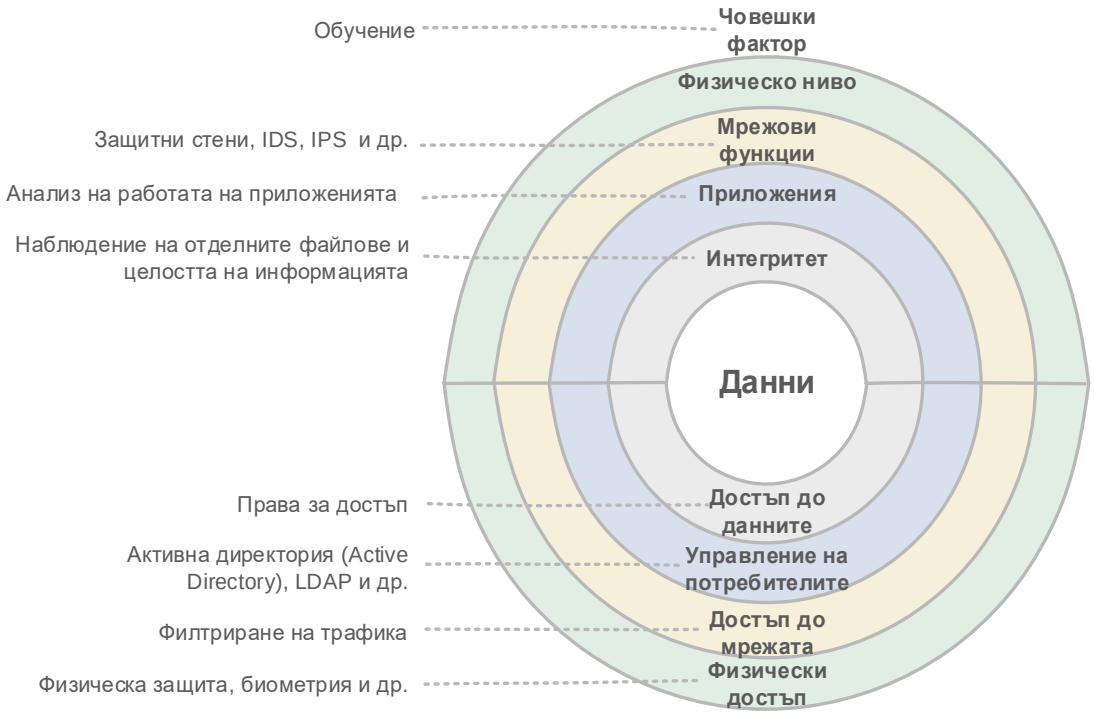
Логически изграждането на сигурността на комуникацията дефинира няколко отделни зони, показани на фиг. 1.9.



Фиг. 1.9 Логически зони при подсигуряване на комуникацията

Важно е да се отбележи, че трафикът през защитната стена трябва задължително да се анализира и филтрира както във входяща, така и в изходяща посока, което води до повишаване на степента на защита и до предотвратяване на възможността за изпращане на конфиденциална вътрешна информация към външни системи или мрежи. Двупосочното сканиране на пакетите допълнително натоварва защитните стени и е необходимо те да разполагат с необходимите системни ресурси за да не се редуцира производителността на мрежовата комуникация.

За да се дефинират необходимите технологии за подсигуряване на мрежовата комуникация е необходимо да се разграничат функционалните нива, което е показано на фиг. 1.10.



Фиг. 1.10 Нива и основни технологии за защита на мрежовата комуникация

Подходът, показан на фиг. 1.10 се базира на анализ на защитата на комуникацията спрямо нивата на OSI референтния модел, като анализираните области са:

- Данни – базовата единица информация, която трябва да бъде подсигурена и защитена от неправомерен достъп и/или промяна;
- Достъп до данните – на база на предварително дефинирани права всеки потребител (човек или софтуерен процес) получава достъп само до определена информация;
- Интегритет – извършва се проверка дали данните са били модифицирани. Най-често се използва криптографско хеширане или подписване с електронен подпис;
- Управление на потребителите – използват се специализирани технологии и сървъри за съхранение на потребителските данни (име, парола, права за достъп и др.). Някой от най-често прилаганите технологии са Microsoft Active Directory, Lightweight Directory Access Protocol (LDAP) и др.;
- Приложения – анализират се изискванията и използваните услуги от всяко приложение. Необходимо е да се провери и за наличие на софтуерни обновявания;
- Достъп до мрежата – задължително е да се ограничи входящия и изходящия трафик към и от мрежовите сегменти. Използват се технологии от типа на пакетно филтриране, “application gateway firewall”, базирани на зони защитни стени, IDS/IPS и др.;
- Мрежови функции – подсигуряването на преноса на данни на мрежовото ниво на OSI референтния модел най-често се извършва чрез интегриране на защитни стени, IDS (Intrusion Detection System) и IPS (Intrusion Prevention System);
- Физически достъп – физическият достъп до устройствата изисква защита от неправомерно посегателство, като най-често за да се подсигури тази функционалност се

инсталира видео наблюдение и контрол на достъпа до помещенията чрез RFID карти и/или биометрия, жива охрана и др.;

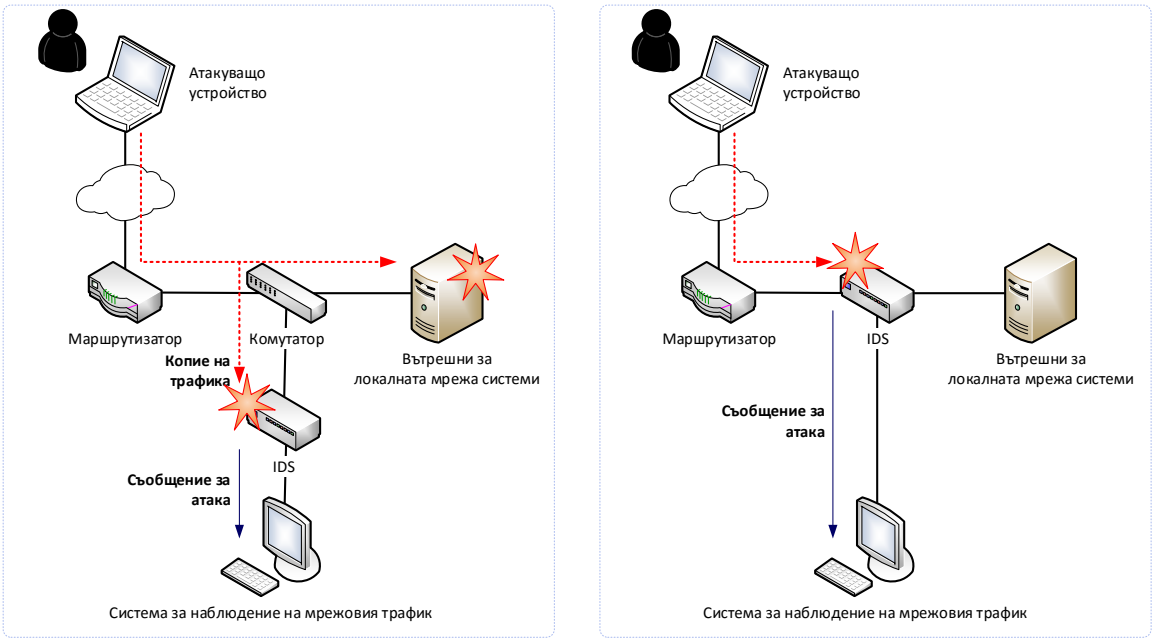
- Физическо ниво – задължително е физическото ниво на OSI референтния модел да бъде подсигурено, като се гарантира, че неоторизиран персонал няма да има пряк физически достъп до системите и преносните среди;
- Човешки фактор – голяма част от успешните пробиви в сигурността се дължат на човешки грешки, свързани с грешки в конфигурирането на системите, не-надеждни (слаби) пароли и др. Задължително е персоналът да бъде обучен и запознат с основните изисквания, свързани с изграждането на мрежовата и комуникационна защита.

Важно е да се отбележи, че защитата на мрежовата комуникация не може да бъде реализирана единствено с технологията на защитните стени, дори ако се използват най-актуалните и варианти – базиран на зони анализ и филтър на трафика до приложното ниво на OSI модела. Много атаки използват открити пропуски и технологични проблеми в софтуерни пакети или протоколи и чрез напълно легитимен трафик могат да осъществят успешна атака. В този случай за предпазване се използват две основни технологии (фиг. 1.11), които анализират поведението на мрежата и приложенията и могат да докладват за откриване на атака или да я блокират в реално време:

1. **IDS** – извършва пасивен анализ на копие на мрежовия трафик чрез сравнение със сигнатури, описващи атаки, вируси и червей. При открити аномалии в трафика се изпраща съобщение от сензора към станцията за наблюдение и администраторите научават за откритата потенциална проблемна ситуация. Поради фактът, че се работи с копие на трафика IDS не оказва съществено негативно влияние върху производителността на мрежата, но пасивният характер на работа означава, че ако е открита атака, то злонамереният трафик е постъпил към мрежата, а администраторите единствено са уведомени за това и трябва да предприемат необходимите превантивни мерки;
2. **IPS** – за разлика от IDS, технологията IPS анализира пакетите, преминаващи през устройството и при открита атака или аномалия блокира трафика в реално време. Това води до по-висока степен на сигурност, но и до значително намаляване на скоростта на обмен на данни, особено при анализ с голям брой сложни сигнатури или търсене на определен символни низове в съдържанието на пакетите.

Въпреки, че IPS технологията е по-надеждна от гледна точка на защита на потребителите, поради забавянето на трафика в много топологии се използва комбинация от IDS и IPS – там където е необходимо да има по-висока скорост на трансфер, но и анализ за атаки се поставят IDS сензори, а при критичните мрежови сегменти – IPS.

На фиг. 1.12 е показан хардуерен IPS сензор на McAfee.



Фиг. 1.11 Сравнение на технологиите IDS и IPS

Фиг. 1.12 McAfee Network IPS (източник Интернет)

Заплахи за комуникацията

Заплахите за комуникацията могат да бъдат разделени на няколко основни групи:

1. Злонамерен код;
2. Сканиране;
3. Атаки с цел достъп до ресурси;
4. DoS и DDoS атаки;
5. SPAM и spoofing.

Вируси, червей и троянски коне

Както вече беше споменато с развитието на комуникационните технологии и създаването на Интернет възможностите за споделяне на информация нарастват значително, но в същото време се повишават и рисковете от заразяване със злонамерен код, получаването на голямо количество ненужни електронни писма и хакерските атаки (както целенасочени, така и случайни).

Като злонамерен код (Malicious code) се класифицират три основни вида програми:

1. **Вируси** (Viruses) – за да се разпространят този тип програми трябва да се прикачат към друга програма или файл (не могат да се пренасят без посредник).
2. **Червеи** (Worms) – злонамерения код се стартира и използвайки дадена слабост или грешка в протоколи или програми се разпространяват без да е необходимо друг файл, който да ги пренася.
3. **Троянски коне** (Trojan horses) – това са програми, които наподобяват нормални приложения, но оставят възможност хакерите да се включат към инфектираната система с цел достъп до ресурси или данни.

Модерните компютърни вируси са представени от Фред Коен през 1983 година. Аналогично на биологичните (които се пренасят чрез определени обекти – течности, повърхности и др.), компютърните вируси се прикачат към определена програма или файл и чрез контролиране на изпълнението на заразения код инфектират други файлове. Първите компютърни вируси се разпространяват чрез информация, записана на дискети (1986 – Brain Virus), което води до бавното разпространяване на зловредния код. Интернет дава възможност за изключително бързото прехвърляне на заразения код по целия свят, а мрежите стават основното място за развитието на червеи, които са вид вируси, притежаващи възможност да се разпространяват сами без да е необходимо да се пренасят от друг код или файл. Отново е налице биологична аналогия, а терминът червей е предложен от Джон Соуч и Джон Хъп по време на техните експерименти с мобилен софтуер в Xerox PARC през 1979 година.

Интернет увеличава възможността да се използват слабости (Vulnerabilities) и пропуски в програмния код на операционните системи, протоколите и програмите, като по този начин злонамерения код може да се разпространява между свързаните в мрежа системи. Разпространението на червея Blaster през 2003 година и SQL Slammer през януари 2003 година показват, че устройствата с достъп до Интернет са изключително уязвими, въпреки използването на антивирусни продукти.

Анализът на кода на вирусите доказва, че тяхната сложност се е повишила многократно през годините. Една от най-важните задачи е вирусите да скрият своето присъствие в заразената система. През 1986 година вирусът Brain се записва в паметта на системата и симулира всички заявки, които операционната система използва за да провери дали има наличие на злонамерен код, като връща отговори вместо DOS (Disk Operating System), които показват, че липсва такъв код. През 2001 година червеят Lion инсталира rootkit[7] наречен t0rn, като по този начин откриването на вируса е изключително трудно. Към момента вирусите и троянските коне се "крият", като атакуват най-популярните алгоритми, използвани от антивирусните продукти (подход, наречен "armoring").

През 1987 година се появява вирусът Christma Exec, като той е един от първите, използващи похвата "social engineering". Christma Exec се разпространява през електронната поща и подлъгва потребителите, че ще изрисува на дисплея "красива" коледна елха. При стартиране потребителят вижда изображение на коледно дърво, но вирусът се прикача към електронната поща на потребителя и се изпраща към всеки контакт, който успее да открие. По този начин получателят вижда, че писмото е изпратено от негов познат и логичното действие е той също да го отвори, което води до ново препращане на зловредния код.

[7] Rootkit или руткит е програма или набор от програми за скрито превземане и удържане контрола над една компютърна система

Фиг. 1.13. Вирус Chrismta Exec

Подходът за заразяване чрез електронна поща е обичайна практика и към момента, която често се използва от вируси, червеи и троянски коне. През януари 1999 година се появява кода Happy99/Ska worm/Trojan, който отново се разпространява през електронната поща на потребителите. Писмото съдържа прикачен файл с име Happy99.exe (изпълним файл за операционна системи Microsoft Windows), при стартирането на който на екрана се появяват анимирани фойерверки, но се подменя системния файл WSOCK32.dll с модифициран. По този начин системата се заразява с троянски кон, който може да използва всички системни процеси, свързани с Интернет и комуникацията в рамките на локалната мрежа.

Фиг. 1.14. Червей Happy99 (Източник Интернет)

Отново през електронната поща и като прикачен файл се разпространява и вирусът PrettyPark (1999 година). За разлика от Happy99 в писмото няма описание на прикачения файл, освен че неговата икона е герой от популярен телевизионен сериал – South Park. При стартиране вирусът се копира в директория %SYSTEM% (системната директория на Microsoft Windows) и модифицира регистрите на заразената система така, че при всяко изпълнение на изпълним файл

от тип EXE се стартира и вирусът. Други известни вируси и троянски коне, които използват подхода “social engineering”, са Anna Kournikova и Gibe.

Първият макро вирус е Concept и е написан за Word for Windows 95. Най-често тези вируси се изпълняват като макроси за отделните продукти от пакета Microsoft Office. Характерно за този тип зловреден код е, че се разработва лесно и, че може да се стартира на всяко устройство, което използва Microsoft Office.

През 1999 година вирусът Melissa достига 100000 хоста в рамките на 3 дни, като поставя и нов рекорд за броя на поразените и спрени сървъри за електрона поща, работещи под управлението на Microsoft Exchange. В изпратеното писмо се посочва адрес от който потребителите могат да изтеглят Word файл с потребителски имена и пароли за еротични сайтове. Към самият файл има прикрепен макро вирус, който се разпространява през Word и Outlook. Интересно е, че към този момент се е считало, че компютърна система не може да се зарази само при отваряне на електронна поща.

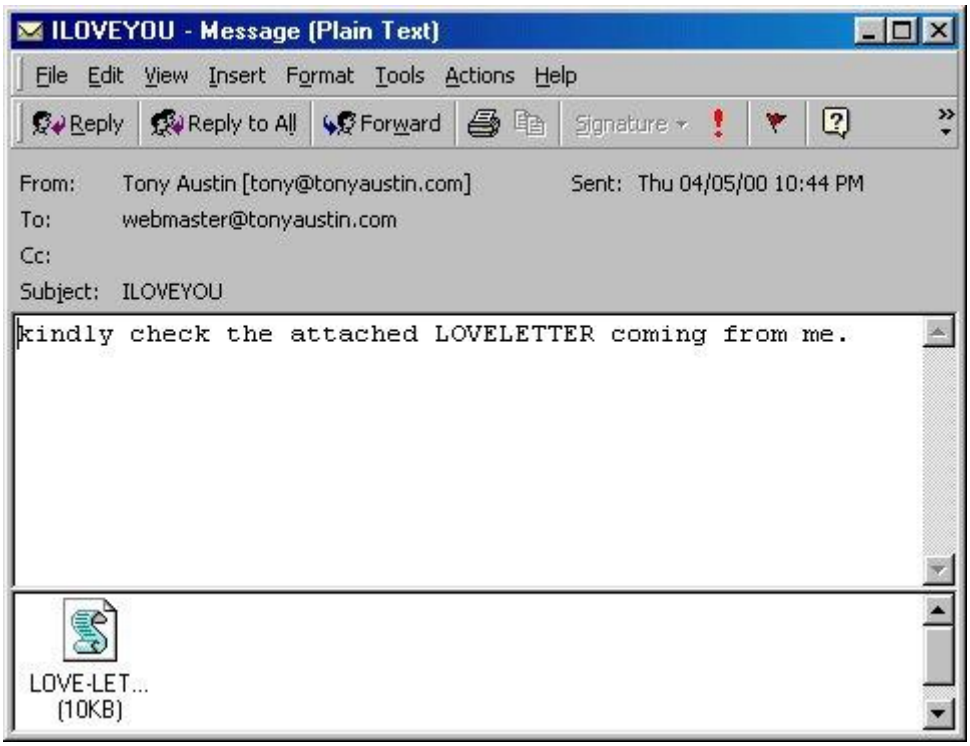
Процесът на заразяване с Melissa е сравнително сложен и е в няколко стъпки:

1. Проверка дали на системата има инсталирана версия на Microsoft Word, която може да бъде заразена;
2. Промяна на настройките за защита на Word, целяща да не се показват съобщения, свързани с изпълнението на макроси;
3. Търсене в регистрите на системата за стойност “Kwyjibo” (популярен момент от сериала “Семейство Симпсън”);
4. Ако този ключ не е открит, вирусът стартира Outlook и се изпраща към 50 от записите, налични в адресната книга на потребителя;
5. Променя се съдържанието на файла NORMAL.DOT (базисен шаблон) чрез Microsoft Visual Basic for Applications (VBA), като всеки записан документ ще пренася и вируса.

Това е още едно потвърждение за степента на сложност на новите вируси и използването на все по-задълбочени и комплексни методи за заразяване.

През месец май 2000 година се повява един от най-популярните до момента червеи – Love Letter. Този код се разпространява като прикачен файл към електронна поща със заглавие “I love you”. Прикрепеният файл е Visual Basic скрипт, който се стартира през Windows Script Host (WSH). При стартиране кодът се копира в системната директория на Windows и модифицира регистрите (стартиране на определени файлове при пускане на компютъра). Също така червеят заразява и файлове с разширение VBS, JPG, MP3 и др., налични на локалните твърди дискове, както и на споделените мрежови ресурси.

Една от задачите на червея е да краде пароли, въведени от потребителите, чрез пренасочване на URL.



Фиг. 1.15. Червей Love Letter

През 2002 година приблизително 90% от вирусите използват подхода “mass e-mailer”. По-интересни са Bugbear и Klez, който използват вградени в техния код Simple Mail Transfer Protocol (SMTP) сървъри. След като се вземат необходимите мерки за защита на SMTP и POP3 сървърите, този подход вече не е популярен и почти не се използва в злонамерения код.

Технологията “полиморфизъм” се базира на идеите за криптиране на данни, като целта е да се скрият т.нар. “patterns”, по които антивирусните продукти разпознават вирусите. Ако се използва само криптиране на кода, антивирусните системи ще намерят вируса, тъй като ще се използват едни и същи ключове за кодиране и декодиране на всички заразени системи. Чрез полиморфизъм кодът на вируса минава през пермутация и по този начин се избягва наличието на еднакви символни низове - “patterns[8]”. Един от първите подобни вируси е открит в Европа през 1989 година - той допълва своя код с псевдослучайни байтове. Тази техника набира широка популярност при създаването на вируси след като известният хакер Dark Angel пуска кода “Mutation Engine”, който е лесен за използване и интегриране към разнороден зловреден код.

Известният червей Morris (1988 година) използва комбинация от няколко атаки за да се разпространи бързо на 6000 UNIX системи само в рамките на няколко часа, като етапите са следните:

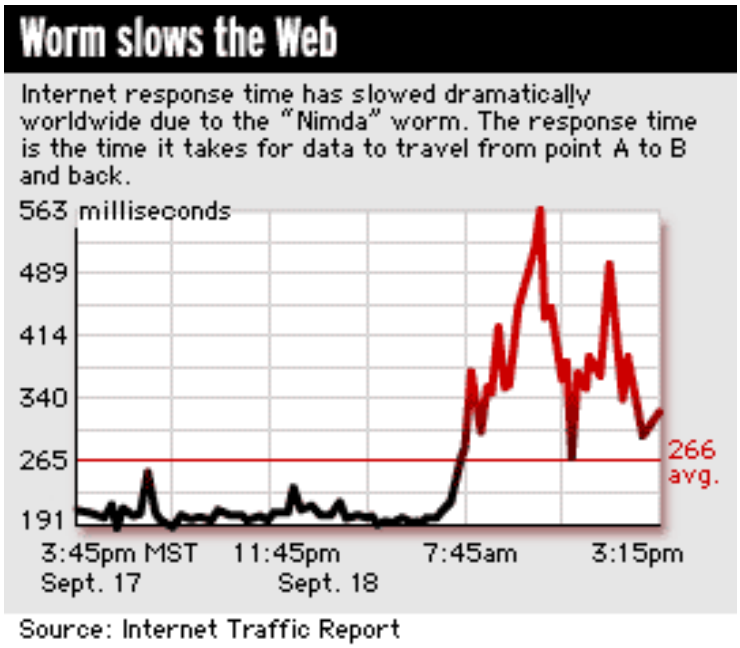
1. Открива се файла, в който се съхраняват паролите и чрез речникова атака се извършва опит за тяхното разчитане;
2. Използва се debug опцията на sendmail програмата за UNIX и Morris се разпраща с електронна поща;
3. Използва се атака с препълване на буфера (buffer overflow) на fingerd демона, за да блокира системата.

Други червеи, използващи комбинация от няколко атаки, са Sadmind/IIS (2001 година), Sircam (2001 година) и Nimda (2001 година). Nimda успява да зарази 450000 хоста за първите 12

[8] От английски - модел или образец

часа от пускането на кода. Това е изключително сложен, както от гледна точка на реализация, така и от гледна точка на провежданата атака злонамерен код:

1. Nimda открива e-mail адресите на заразената система чрез кеша на Messaging Application Programming Interface (MAPI). Изпраща се на откритите адреси със случайни теми и прикачен файл с име readme.exe. На някои устройства прикаченият файл се стартира автоматично;
2. Заразява Microsoft IIS сървъра чрез препълване на буфер – "Unicode Web traversal exploit[9]";
3. Копира се на наличните споделени мрежови ресурси;
4. Добавя Java Script на WEB страниците на сървъра;
5. Търси дали има оставени задни врати (back doors) от вирусите Code Red II и Sadmind.



Фиг. 1.16. Забавяне на Интернет трафика от Nimda (източник CNet)

През ноември 2002 година в кода на червея Winevar са открити функции, който деактивират антивирусния софтуер на системата, чрез търсене на определени процеси в паметта. Тази технология се използва и от червеите Fizzer и Lirva. Lirva се опитва да се свърже с адрес web.host.kz, откъдето да свали софтуерния инструмент BackOrifice, чрез който всеки би имал пълен отдалечен достъп до системата.

Интересен е подходът, който използва Welcha. Този червей се представя като специален инструмент за премахване на друг червей – Blaster, като изтегля специален "Fix" (коригиращ софтуерен пакет) от сайта на Microsoft.

Червеят Slapper, който работи под Linux е един от първите, който се разпространява през peer-to-peer (P2P) протоколите. При успешно заразяване системата става част от P2P мрежа.

Освен P2P мрежите цел на злонамерения код стават и програмите за изпращане на съобщения между потребителите. През 2003 година червеят AimVen заразява America OnLine Instant Messager (AIM), като подменя използвания от потребителя софтуерен клиент. През същата година Fizzer се разпространява през мрежата KaZaa.

[9] https://www.kb.cert.org/vuls/id/111677

Един изключително опасен зловреден код е Cryptolocker. Този троянски кон атакува компютри, използващи Microsoft Windows и се пренася през инфектирани писма от електронна поща. След активиране Cryptolocker използва изключително надеждния алгоритъм RSA с 4096 битов ключ, с който шифрира файловете на потребителите и надеждно изтрива оригиналите. След това потребителите биват изнудвани в определен интервал от време да заплатят сума, срещу която ще получат ключа за дешифриране на техните файлове. Важно е да се отбележи, че декриптиране по метода на грубата сила е безсмислен поради изключително дългия период от време, който е необходим. Cryptolocker поставя началото на т.нар. “ransomware” – софтуер за изнудване.

Фиг. 1.17 Троянски кон Cryptolocker (Източник Интернет)

Защитата от зловреден код може да се извърши чрез софтуерни пакети за антивирусно действие или чрез интегриране на тази функционалност в мрежови устройства. Задължително е потребителите да бъдат наясно за рисковете от изтегляне, инсталиране или стартиране на файлове от неясни източници.

Атаки с цел достъп до ресурси

Много услуги и протоколи използват защита с парола или потребителски профил (комбинация от потребителско име и парола). За да се получи достъп до такива защитени ресурси, често се извършват атаки, насочени към паролите или ключовите фрази, които могат да се разделят на няколко основни вида:

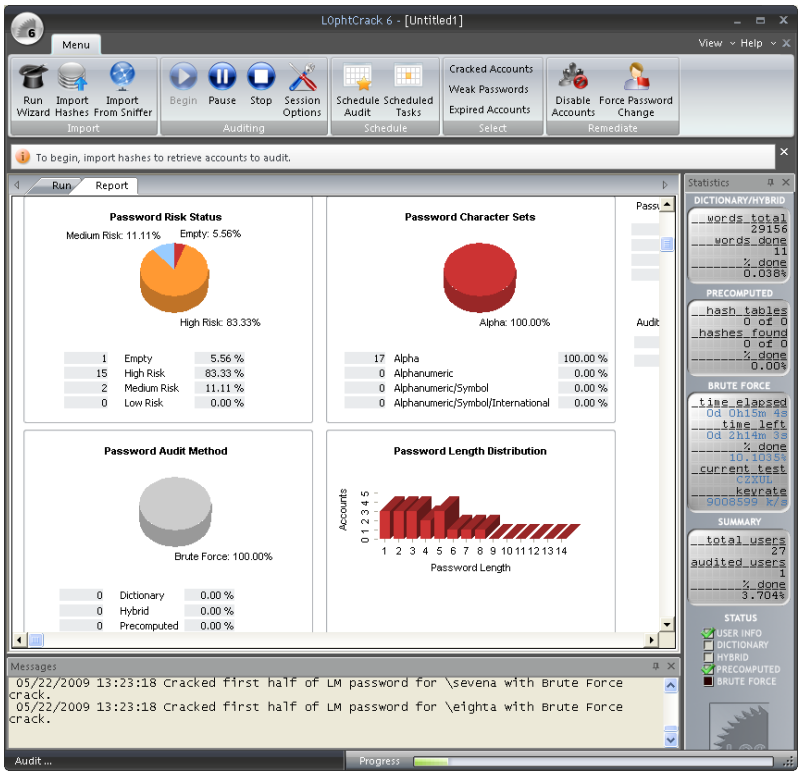
- **Активни** – атакуващото устройство генерира парола или комбинация от данни с цел налучкване и евентуален достъп до ресурса;
- **Пасивни** – на база на предварително събрани данни в пасивен режим на работа (без да се изпращат данни към други системи) се прави опит за изчисляване на паролата или ключовата фраза;
- **Речникови** – за да се минимизира времето за атака се използват специални речникови файлове, които съдържат голям брой често използвани пароли;
- **Метод на грубата сила** – атаката се базира не генериране на всички възможни комбинации от символи и тяхното използване с цел достъп до ресурс. Важно е да се

отбележи, че този метод винаги ще постигне 100% резултат, но необходимото време може да бъде практически огромно – в зависимост от използваните криптографски алгоритми може да надмине десетки хиляди години с текущото ниво на развитие на компютърната техника;

- **“Man-in-the-middle”** – използва се междинна система или софтуер, който следи и записва данните, пренасяни през мрежовата инфраструктура с цел откриване на пароли в чист текст или хешови стойности, отговарящи на парола;
- **Троянски коне** – след инсталиране на троянски кон на даден система може да се изпращат въведените от потребителите пароли към отдалечено устройство или да се съхраняват в скрит файл, до който атакуващото лице да има достъп.

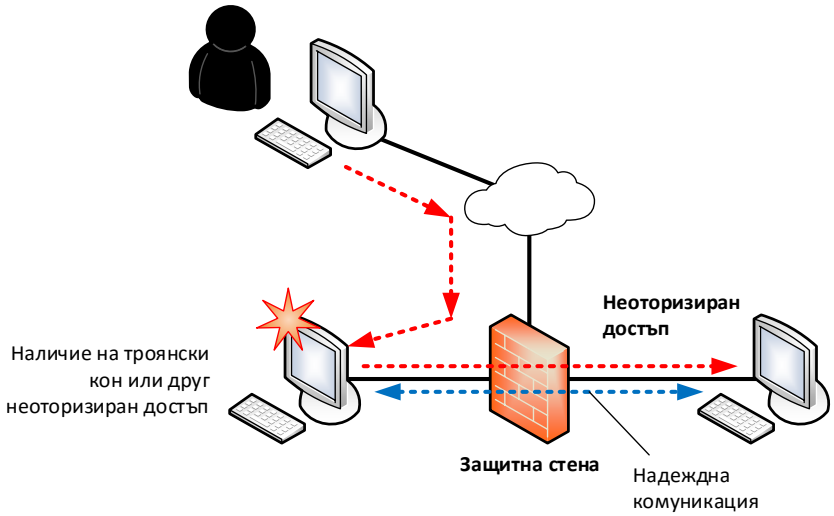
Някои от по-известните инструменти за атаки, насочени към пароли са:

- Hydra;
- Cain and Abel;
- L0phtCrack;
- Burp и др.



Фиг. 1.18 L0phtCrack (източник Интернет)

Други методи за атака с цел достъп до ресурси използват вече изградена надеждна комуникация между две или повече системи и през една от тях получават достъп до другите устройства.



Фиг. 1.19 Атака с използване на изградена надеждна комуникация

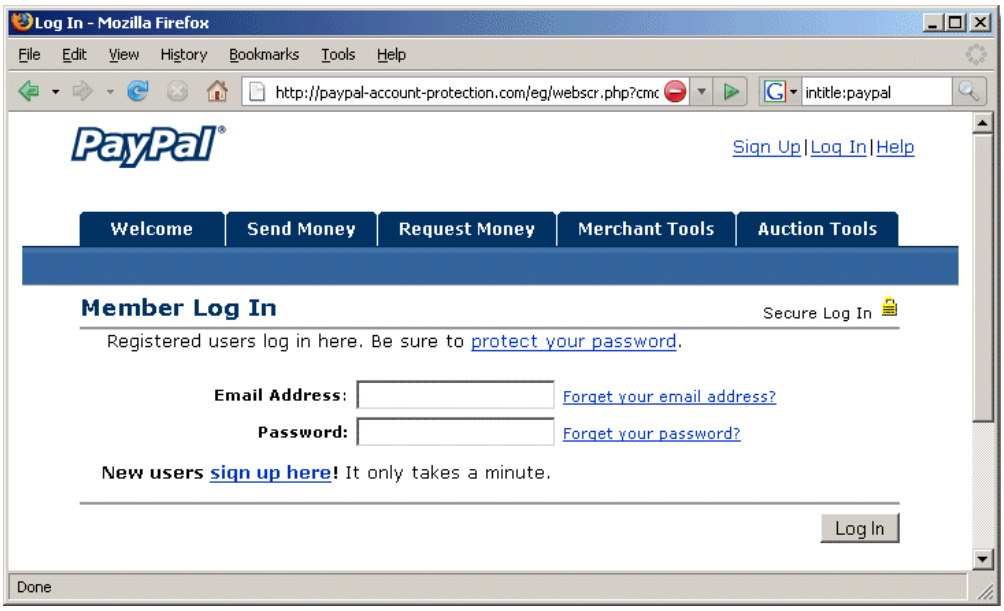
Подобен тип неоторизиран достъп може да се извърши и чрез пренасочване на портове или промяна на маршрутизиращи таблици на мрежовите устройства в топологията. При някои технологични проблеми (бъгове) е възможно след препълване на софтуерен буфер или изпращане на лошо структурирани данни да се получи неоторизиран достъп до ресурс (типичен пример са атаките HeartBleed и ShellShock).

За да се предпазят системите от неоторизиран достъп е възможно да се интегрират IDS/IPS системи, които да следят мрежовия трафик, както и да се използва технологията “sandboxing”, която изолира системните ресурси между отделните софтуерни процеси, изпълнявани от операционната система.

Атаки с цел подвеждане (spoofing)

Атаките с цел подвеждане могат да бъдат различни видове и да са насочени към крайния потребител или към комутирането и маршрутизиране на трафика. Много често се използват фалшиви IP или MAC адреси, което позволява пакетите да преминават през мрежовите устройства към отдалечени сегменти. Този тип атаки могат лесно да бъдат филтрирани от защитните стени и IDS/IPS системите.

Към тези атаки се включват и клонирането на сайтове (phishing) с цел потребителите да въведат своите потребителски данни и най-често номера на кредитни карти, като информацията бива откраднатa и в последствие използвана от трети лица. Най-често се клонират сайтове за електронна търговия, банки и социални мрежи.



Фиг. 1.20 Фалшива страница за включване към PayPal - обърнете внимание на използвания протокол HTTP, а не HTTPS (Източник Интернет)

Предпазването от phishing атаките изисква обучаване на потребителите и повишено внимание при работа със сайтове за електронно банкиране и при въвеждане на лични данни и потребителски профили.

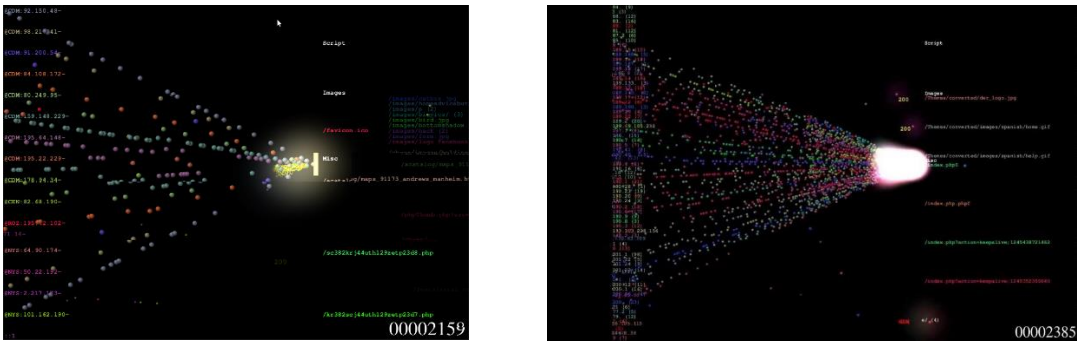
DoS атаки

Атаките, свързани с отказ на услуги на дадено приложение или устройство се наричат DoS (Denial of Service). За тяхното реализиране могат да се използват различни механизми – от генериране на голям обем мрежови трафик до сложни технологии, базирани на пропуски в сигурността. Най-често този тип атаки са насочени към сървърни приложения, като техният принцип на действие попада в една от следните две групи:

1. Хост или приложение не успява правилно да обработи проблемна ситуация, свързана с умишлено изпратени данни с грешен формат или изчерпване на определен системен ресурс;
2. Мрежа, хост или приложение е претоварено от изключително голям обем трафик, което води до блокиране на софтуерни процеси или сериозно забавяне в работата.

На фиг. 1.21 е показан анализ на работата на WEB сървър с безплатния инструмент logstalgia[10], който графично визуализира трафика към устройството, на база на информацията от журналните файлове. Ясно се вежда разликата при нормалната работа на системата и при DoS атака.

[10] http://code.google.com/p/logstalgia/



а) б)

Фиг. 1.21 Анализ на журналите на WEB ісървър с logstalgia
а) Нормален трафик б) DoS атака (източник Интернет)

Някои от първите примери за DoS атаки са:

- Ping of death – изпраща се ICMP echo request с големина над 65535 байта, като при получаването на този грешен пакет някои от по-старите операционни системи могат изцяло да бъдат блокирани;
- TCP SYN flood – изпращат се голям брой пакети, съдържащи SYN флага за начало на процеса на договаряне на TCP сесия (3-way handshake). При получаване на SYN и ACK не се изпраща следващия ACK. Това може да доведе до значително редуциране на системните ресурси на атакуваната система, а в определени случаи и до пълно блокиране.

От гледна точка на анализ на риска DoS атаките се считат за изключително опасни и е задължително да се вземат необходимите мерки за тяхното предотвратяване. Откриването на DoS атаки може да се направи след анализ на поведението на мрежата и на отделните системи, като се следи за:

- Необичайно забавяне в скоростта на обмяна на данни;
- Значително забавяне в работата на отделните софтуерни модули, приложения и операционни системи;
- Отказан достъп до мрежова услуга или сървърно приложение за продължителен интервал от време;
- Изключително голям брой записи за грешки в журналите;
- Повишаване на броя на грешно доставени пакети по мрежата.

Важно е да се запомни, че при откриване на DoS атака е задължително да се провери дали тя не е част от по-голяма целенасочена злонамерена стратегия.

Освен DoS атаките се дефинират и т.нар DDoS (Distributed Denial of Service). DDoS използват предварително заразени със зловреден код системи, наречени зомбита, които при получаване на специални инструкции стартират специфична DoS атака към посочените хостове, като по този начин размерът на атаката става изключително голям, а защитата се затруднява. Типичното протичане на DDoS атака може да се опише със следните етапи:

1. Атакуваната система внимателно се сканира дали е достъпна и дали подлежи на избраният тип атака;
2. Изпраща се заявка към системите зомбита за начало на DDoS;
3. Инициира се DoS атаката от заразените системи към посочените цели.

Предпазването от DoS и DDoS може да се извърши чрез интегриране на защитни стени, IDS/IPS системи, внимателно и периодично преглеждане на файловете със статистика за работата и журналите на устройствата и анализ в реално време на работата на мрежата.

Основни модели за подсигуряване на мрежовата комуникация

Според някои източници основните модели при подсигуряването на мрежовата комуникация са:

1. **Отворен** – при този модел се разрешават всички възможни услуги, протоколи и приложения и се блокират ненужните;
2. **Затворен** – всички услуги, протоколи и приложения са блокирани и се разрешават единствено дефинираните в корпоративната политика за сигурност.

От гледна точка на степента на сигурност затворения модел е по-надежден и повечето защитни стени и системи за анализ и филтриране на трафика се базират на него, като по този начин вероятността да бъде пропуснат нежелан трафик или да се работи с нерегламентирано приложение са сведени до минимум, а отговорността пада върху администраторите.

Още един важен параметър на моделите за подсигуряване на мрежовия трафик и по-конкретно на защитните стени е реда на изпълнение на посочените правила. Най-често първите правила се анализират с по-висок приоритет и ако дадено условие отговаря на критериите, то се извършва посоченото действие а по-долните правила не се анализират.

Както вече неведнъж беше споменато периметъра на мрежата е размит поради използването на мобилни устройства и на облачни услуги. В много редки случай мрежовата топология се изолира изцяло от Интернет и дори не се използват безжични устройства. Това се прави най-често при военната промишленост, специализирани правителствени организации или мрежи между банкомати и фискални устройства.

Използвани инструменти

За да се анализира мрежовата сигурност се използват разнородни инструменти, а голяма част от тях по същество се явяват програмен код използван от хакери, същевременно чрез който може да се анализира дали дадена система може да бъде успешно атакувана.

Съществуват хиляди инструменти, като тяхната функционалност може основно да се групира в следните категории:

1. **Разузнаване** – сканиране на мрежата, сканиране на портове, откриване на активни устройства и услуги, определяне на типа на хардуера и използваните операционни системи и др.;
2. **Анализ за пропуски** – специализирано сканиране на системи за наличие на пропуски, които биха позволили провеждане на определена атака. Най-често този тип програмен код е за специфичен пропуск или използва система от модули (plugins), които позволяват по-общо сканиране;
3. **WEB приложения** – инструменти за атаки, насочени към WEB приложения и сървъри;
4. **Атаки на пароли** – активно и пасивно анализиране на пароли;
5. **Атаки на безжични мрежи** – активни и пасивни атаки на WLAN[11], инструменти за анализ на пароли и WPS[12];

[11] Wireless Local Area Network
[12] Wi-Fi Protected Setup

6. **Разработване на атаки** – атаки на база на потенциални пропуски;
7. **Подслушване** – запис на трафик и следене за определени данни, пренасяни по мрежовата инфраструктура;
8. **Запазване на достъпа** – след успешна атака инструментите позволяват атакуващата система да запази своя контрол над жертвата;
9. **Обратно инженерство** – анализ на начина на работа на програмен код и софтуерни и хардуерни системи;
10. **Социално инженерство** – инструменти за подлъгване на потребителите, клониране на сайтове и др.;
11. **Разследване** – анализ на причините за успеха или неуспеха на дадена атака.

Една от най-популярните специализирани Linux дистрибуции, за анализ на мрежовата сигурност е Kali (наследява BackTrack). Този пакет може де бъде безплатно изтеглен от www.kali.org и съдържа всички необходими инструменти за провеждане на задълбочен анализ на сигурността на комуникациите. В разгледаните тук примери се използват софтуерни пакети, включени в Kali 1.1.0.

Откриване на пропуски и аномалии

Препоръчителните етапи при анализ на сигурността на мрежовата комуникация са:

1. Избор на време за провеждане на тестовете и предупреждаване на администраторите и потребителите за възможни прекъсвания на работата на мрежата;
2. Сканиране на мрежата с цел определяне на активните устройства, техния тип, операционните системи и работещите приложения;
3. Анализ за потенциални пропуски в сигурността на откритите системи;
4. Анализ за възможност за осигуряване на достъп чрез атаки на пароли;
5. Допълнителни атаки от тип DoS и метод на социалното инженерство;
6. Анализ на резултатите и тяхното подробно описание.

Един от най-съществените етапи е първия, при който е изключително важно да се подбере правилното време за провеждане на анализа на сигурността. По никакъв начин останалите етапи не трябва да водят до прекъсване или проблеми с бизнес процесите и най-вече до загуба на данни или до разкриване на корпоративни конфиденциални данни на трети лица.

Сканиране на мрежата

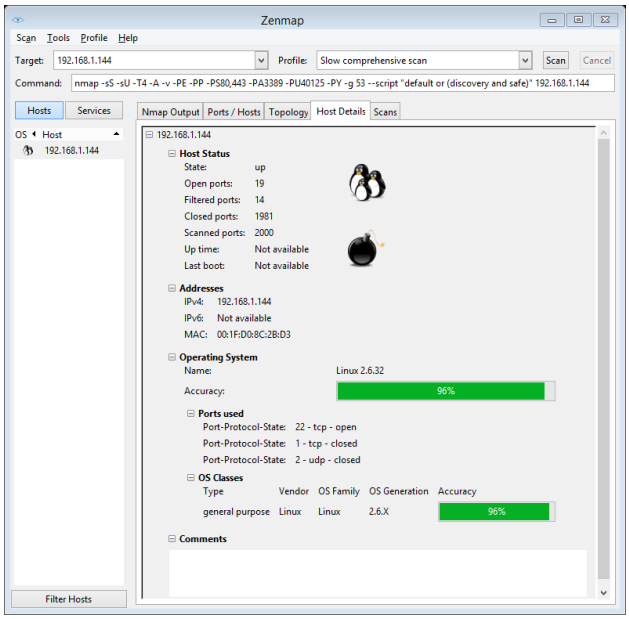
Най-често началото на проверка на мрежовата сигурност, но и на някои видове атаки е сканирането на мрежата с цел да се открият:

- Работещи устройства;
- Адресите на устройствата;
- Типа и хардуера на устройствата;
- Типа и версията на операционната система;
- Отворените портове и работещите приложения;
- Версиите на работещите приложения и др.

Този процес може да трае продължително време и в общия случай изисква активно изпращане на пакети към анализираната мрежа или устройства. За целта може да се използва специализиран софтуер, като най-популярния продукт е nmap.

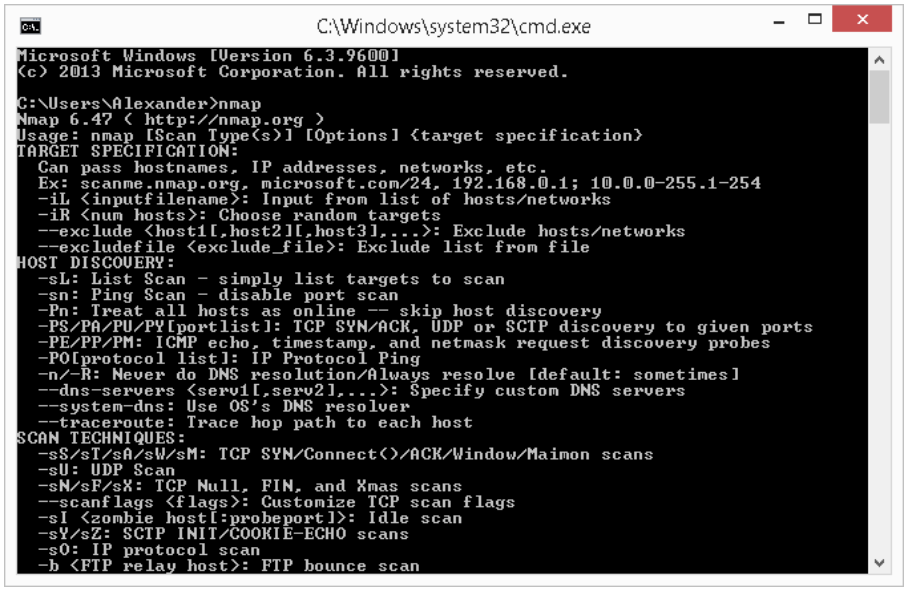
Nmap

Nmap[13] е безплатен инструмент, който работи под Microsoft Windows, Linux и други операционни системи и позволява да се извърши изключително детайлно сканиране на мрежата и отделните устройства, както в активен, така и в пасивен режим. Под активно сканиране се разбира даденият инструмент да генерира необходимите пакети, като по този начин сканиращата система може да бъде открита. При пасивен анализ се анализира събраната по мрежата информация и на нейна база се получават определени резултати. В пакета на nmap е включен и инструмента Zenmap, който предоставя удобен базов графичен интерфейс.



Фиг. 1.22 Zenmap - резултат от сканиране с nmap

Употребата на nmap най-често е през командния ред, като пълния списък с опции може да се види при стартирането на програмата без параметри.



Фиг. 1.23 Стартиране на nmap през команден ред

[13] http://nmap.org/

В таблицата по-долу са показани някой от най-популярните комбинации от параметри при сканиране с nmap (IP адресите и портовете са примерни).

Параметри	Описание
nmap 192.168.1.1	Сканиране на един хост по посочен IP адрес
nmap www.server.bg	Сканиране на хост по DNS име
nmap –v www.server.bg	Сканиране на хост по DNS име и извеждане на разширена информация
nmap 10.0.0.1 10.0.0.10 10.0.0.20 **nmap 10.0.0.1,5,10** **nmap 10.0.0.1-10**	Сканиране на няколко хоста
nmap 192.168.1.0/24	Сканиране на цяла мрежа
nmap –iL ~/hosts.txt	Сканиране на всички хостове и мрежи, записани във файл (hosts.txt)
nmap -6 www.server.bg **nmap -6 2607:f0d0:1002:51::4** **nmap -v A -6 2607:f0d0:1002:51::4**	Сканиране на хостове и мрежи при пренос на данни с IPv6.
nmap -sP 192.168.1.0/24	Сканиране за активни устройства (ping sweep)
nmap –F 192.168.1.1	Бързо сканиране
nmap --reason 192.168.1.1	Извеждане на причината даден порт да е в откритото състояние
nmap --packet-trace 192.168.1.1	Извеждане на информация за всички изпратени и получени пакети
nmap -p 80 192.168.1.1	Сканиране на определен порт
nmap -p T:80 192.168.1.1	Сканиране на определен TCP порт
nmap -p U:80 192.168.1.1	Сканиране на определен UDP порт
nmap -p U:53,111,137,T:21-25,80,139 192.168.1.1	Сканиране на комбинация от TCP и UDP портове
nmap -T5 192.168.1.0/24	Оптимален вариант за сканиране на всички хостове в посочената мрежа и откриване на отворените портове
nmap –O 192.168.1.1	Сканиране с цел определяне на типа и версията на операционната система
nmap -sV 192.168.1.1	Сканиране за активни сървърни приложения
nmap -PS 192.168.1.1 **nmap -PA 192.168.1.1**	Ако защитна стена блокира ICMP Echo може да се стартира сканиране чрез TCP ACK и SYN
nmap -PU 192.168.1.1	Анализ чрез UDP Ping
nmap -sU 192.168.1.1	Сканиране за UDP сървърни приложения
nmap -sO 192.168.1.1	Анализ за поддържаните от отдалеченото устройство IP протоколи
nmap -n -D192.168.1.5,10.5.1.2,172.1.2.4 10.0.0.1	Сканиране чрез добавяне на хостове примамки. По този начин атакуваната система по-трудно може да определи от къде идва трафика
nmap -v -sT -PN --spoof-mac 0 192.168.1.1	Сканиране чрез променен (в примера случайно генериран) MAC адрес
nmap 192.168.1.1 > result.txt	Запис на резултатът от сканирането в файл (result.txt)

Посочените в таблицата примери са само малка част от поддържаните от nmap параметри. Важно е да се отбележи, че има значение дали даденият аргумент е с малка или главна буква (case sensitive).

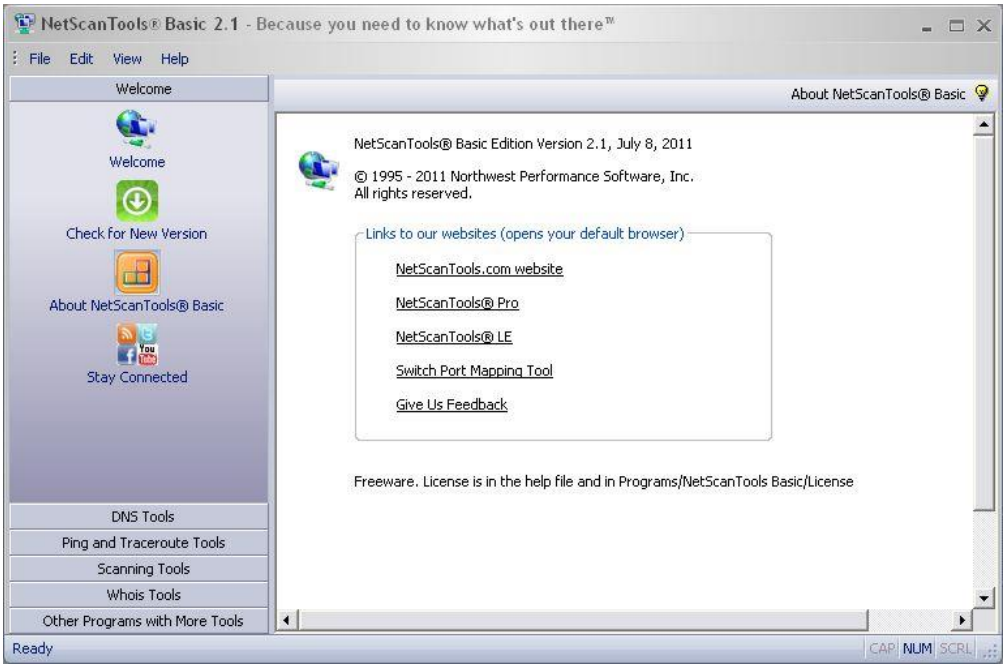
Lanmap2

Lanmap2[14] е пасивен инструмент за анализ на мрежова свързаност, който не изисква SNMP, а в основата е заложен метода на събиране на целия мрежови трафик и филтриране на информацията с цел определяне на устройства, използван софтуер и други полезни данни. Резултатът позволява да се генерира и графично изображение за по-лесно визуализиране на откритата топология.

Предимство на този тип сканиране е пасивния характер на работа на инструмента и липсата на какъвто и да е генериран трафик към отдалечени мрежови сегменти и устройства, което прави засичането му изключително трудно.

NetScanTools

NetScanTools[15] е инструмент за активно сканиране на мрежи, който работи под Microsoft Windows и позволява лесна употреба чрез интуитивен графичен потребителски интерфейс. Включените инструменти са свързани със сканиране на портове, ping sweep (откриване на активни устройства в дадена мрежа), анализ на DNS и други.



Фиг. 1.24 NetScanTools (източник Интернет)

Metasploit

Проблемите, свързани със сигурността на мрежовата комуникация се дължат на няколко основни типа пропуски (vulnerabilities):

- **Конфигурационни** – много често в конфигурацията на устройствата се допускат грешки като потребителски профили с пароли по подразбиране, използват се не-надеждни пароли, не се активират необходимите функции за защита на достъпа и комуникацията и др. Този тип пропуски в сигурността могат лесно да бъдат открити при сканиране с необходимите за целта инструменти и след това да бъдат направени корекции на конфигурациите с цел тяхното отстраняване;

[14] https://github.com/rflynn/lanmap2
[15] http://www.netscantools.com

- **Технологични** – този тип пропуски в сигурността се дължат на грешки в софтуерни функции, библиотеки, приложения, протоколи и др. За да бъдат отстранени трябва да се инсталират всички най-нови версии и обновления на софтуерните пакети и библиотеки;
- **Свързани с политиката за защита** – ако политиката за корпоративна защита на информацията е непълна, непоследователна или в нея липсват важни секции като план за действие в критични ситуации това може да доведе до съществени проблеми при изграждане и поддържане на защитата на комуникациите.

Проектът Metasploit[16] е разработен с цел да се осигури среда за създаване на приложения за анализ и оценка на защитата на информационните и комуникационни технологии, като резултатите могат да се използват за повишаване на степента на сигурност, за разработване на дефиниции за IDS/IPS системи и др.

Една от най-известните функции на Metasploit е безплатният набор от библиотеки (framework) с отворен код, които позволяват да се разработва код за отдалечен анализ на сигурността. Други популярни инструменти, свързани с цялостния проект Metasploit са Opcode Database, архивът shellcode и др. Първоначално Metasploit е разработен от HD Moore през 2003 година, а през 2007 е изцяло пренаписан на езика Ruby. През 2009 година Metasploit е закупен от компанията Rapid7, която стартира нов платен клон на проекта, но запазва и общодостъпния вариант с отворен код. MSF може да се използва в конзолен режим или с графичен интерфейс.

Употребата на Metasploit Framework (MSF) е лесна и изисква само няколко основни стъпки:

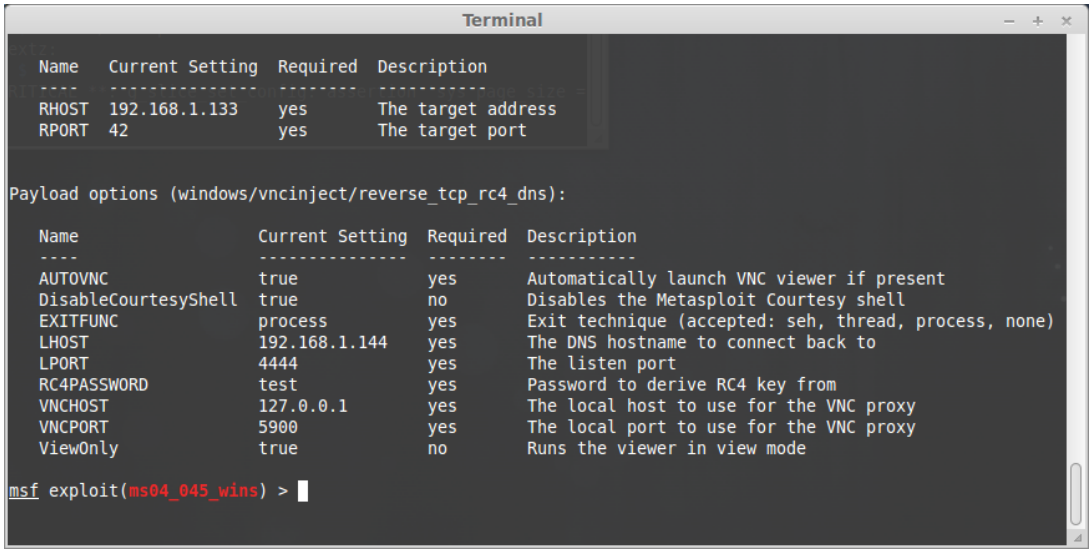
1. Избор и конфигуриране на модул за атака (exploit), който използва определен технологичен или конфигурационен пропуск в отдалечената система;
2. Опционална проверка дали отдалечената система има наличен технологичния или конфигурационен пропуск;
3. Избор и конфигуриране на товар (payload) – код, който се стартира на отдалечената система при успешен пробив в сигурността;
4. Избор на технология за кодиране на данните – използва се с цел заобикаляне на защитни стени и на IDS/IPS системи;
5. Стартиране на процедурата.

При работа в конзолен режим се използват команди, като някои от по-важните са:

- **help** или **?** – извежда списък с наличните команди за приложението msfconsole;
- **show exploits** – извежда информация за модулите за атаки, които могат да бъдат използвани;
- **show payloads** – извежда данни за различните видове товар, който може да бъде пренесен със съответния модул за атака;
- **info exploit [exploit name]** – показва описание на определения модул за атака, включително различните опции, които се поддържат и изискват при неговото стартиране;
- **info payload [payload name]** – показва информация за опциите и необходимите параметри за посочения товар;
- **use [exploit name]** – показва на MSF кой код за атака да се използва. След тази команда се преминава в специален режим за конфигуриране и употреба на посочения модул;

[16] http://www.metasploit.com

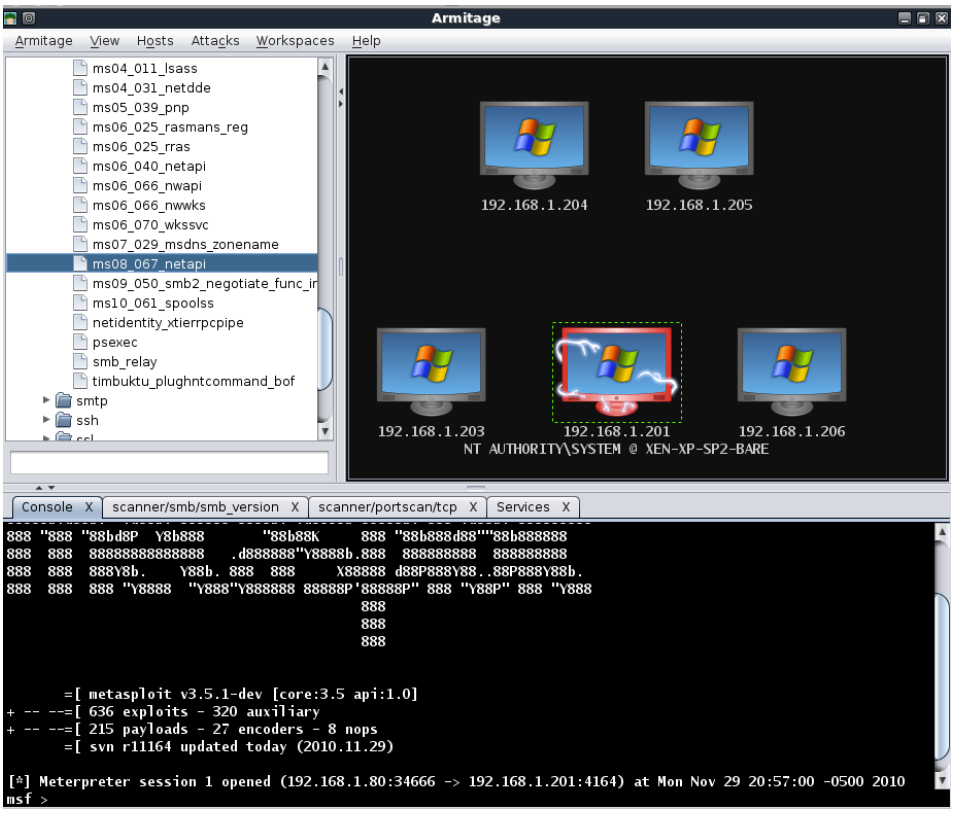
- **show options** – показва различните параметри на активирания код за атака;
- **show payloads** – извежда списък със съвместимите товари за активирания код за атака;
- **set PAYLOAD** – избира товар, който да се използва при проверката на сигурността на отдалеченото устройство;
- **show targets** – извежда информация за възможните цели на атаката – операционни системи, приложения и др.;
- **set TARGET** – посочва целта на кода за атака;
- **set RHOST** – използва се за да се посочи IP адресът на целта (един адрес или група адреси);
- **set LHOST** – задава адресът на локалното устройство, ако ще се извършва обратна връзка след успешно активиране товара на отдалеченото устройство;
- **run** – стартира кода за атака;
- **back** – връща към предишния режим на работа на MSF.



Фиг. 1.25 Работа с MSF в msfconsole

MSF може да се използва и чрез графичният инструмент Armitage[17], който значително улеснява изборът на код за атака и другите параметри, свързани с анализа на сигурността.

[17] http://www.fastandeasyhacking.com



Фиг. 1.26 armitage (източник Интернет)

DoS атаки

Едни от най-опасните мрежови атаки са DoS. Те целят дадено устройство или услуга да е недостъпна за потребителите, което може да доведе до редица проблеми за бизнес процесите. Използваните методи при този тип атаки са най-различни и някой от по-често срещаните са:

- Генериране на голям обем от заявки и/или трафик;
- Изпращане на грешни данни с цел препълване на буфер;
- Генериране на нови записи в маршрутизиращи таблици;
- Изтриване на части от конфигурационни файлове и др.

WEB атаки

Атаките към WEB сървърите и приложенията са сред най-честите, поради изключително високата популярност на този тип софтуер.

Най-честите атаки, към WEB приложенията са:

- DoS;
- Атаки на пароли и потребителски профили;
- Кражба на потребителски профили;
- SPAM;
- Phishing и др.

Един опасен инструменти, които беше разработен в последните години е скриптът slowloris[18]. Кодът е написан на Python от RSnake и реализира специфичен HTTP клиент, който въпреки, че не генерира голям обем от трафик може да доведе до успешна DoS атака на някои

[18] http://ha.ckers.org/slowloris/

от най-популярните WEB сървъри. В своята същност Slowloris не е TCP/SYN flood атака. Скриптът стартира и поддържа частични HTTP заявки към WEB сървър, като периодично изпраща хедър с цел да се поддържа сокета в отворено състояние. По този начин WEB сървъра може бързо да изчерпа своите ресурси и да не може да отговори на заявки на други потребители.

Social Engineering Toolkit

Social Engineering Toolkit[19] (SET) е разработен от TrustSec и представлява Python скрипт с отворен код, който позволява стартиране на различни атаки по метода на социалното инженерство. Някой от по-важните функции на SET са:

- Генериране на заразени файлове с “backdoor”;
- Масово изпращане на електронна поща (Mass mailer);
- Клониране на сайтове и кражба на потребителски акаунти;
- Arduino инструменти;
- Автоматизиране на MSF и др.

SET има текстов интерфейс и въпреки, че първоначално може да изглежда леко объркващ е изключително лесен за употреба.

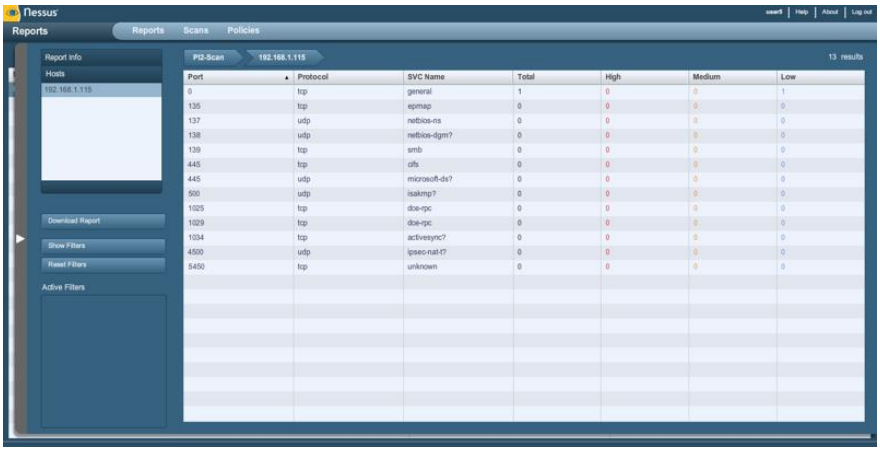
Автоматизирана проверка на защитата на комуникациите

За да се анализира сигурността на мрежовата комуникация и на отделните системи може да се използват специализирани инструменти, като някои от тях обхващат редица модули за анализ и след приключване на дейностите предоставят подробни доклади.

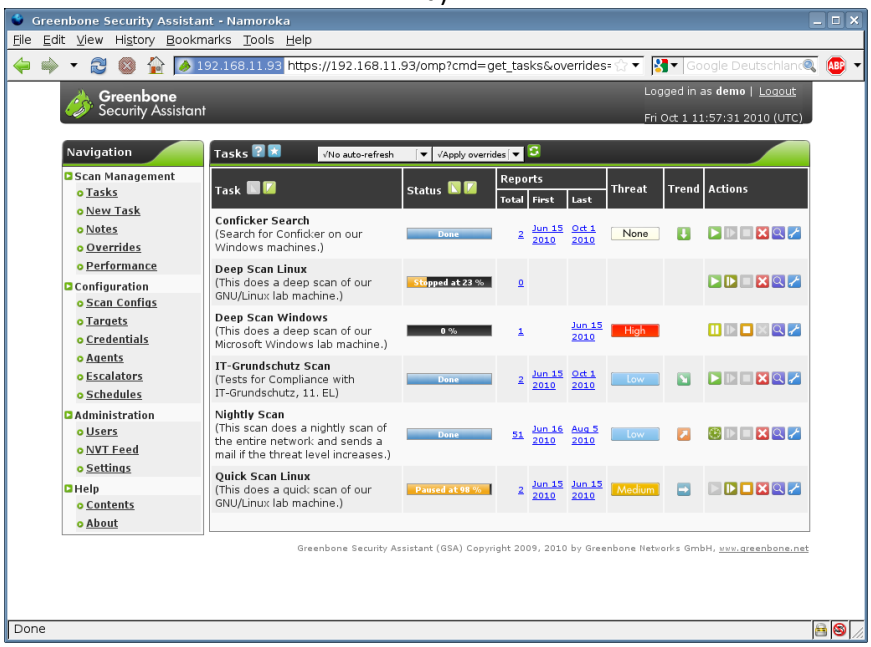
Популярните софтуерни пакети, извършващи този тип проверки са:

- **Tenable Nessus[20]** – използва система от модули (plugins), като потребителя може да избере коя група от тях да се използват за съответния анализ. За да се оптимизира работата на Nessus се използва модела клиент-сървър. Сканирането се извършва от сървърното приложение, като неговото управление е чрез специално разработен и изключително интуитивен софтуерен клиент. Точността на сканиране и определяне на резултатите е висока, като след приключване на анализа се визуализира подробен доклад, съдържащ препратки към откритите пропуски, тяхното описание и начини за предпазване;
- **OpenVAS[21]** – система с отворен код за анализ на мрежовата сигурност, която към момента извършва над 35000 теста (към април 2014 година). По-голямата част от модулите са под GNU GPL лиценз. Отново се използва модела за работа клиент-сървър;
- **BeyondTrust Retina[22]** – продуктът е пуснат на пазара през 1998 година. Поддържа интегриране на MSF и достъп до ExploitDB[23] и други on-line бази данни, свързани с откриването на пропуски в сигурността.

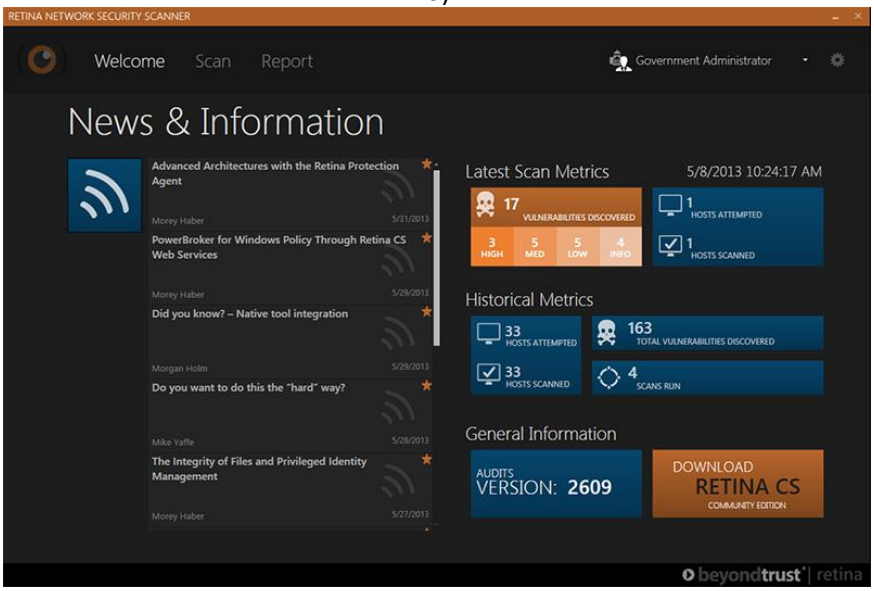
[19] https://www.trustedsec.com/social-engineer-toolkit/
[20] http://www.tenable.com/products/nessus-vulnerability-scanner
[21] http://www.openvas.org
[22] http://www.beyondtrust.com/Products/RetinaCSThreatManagementConsole/
[23] http://www.exploit-db.com



а)



б)



в)

Фиг. 1.27 Автоматизирано сканиране за пропуски в сигурността.
а) Nessus б) OpenVAS в) BeyondTrust Retina (източник за всички изображения - Интернет)

Използването на автоматизираните скенери за проверка на защита на мрежовата комуникация трябва да е периодично, като на база на получените резултати е необходимо да се вземат мерки за подсигуряване на откритите пропуски. Изключително важно е при стартиране на сканирането модулите на съответния продукт да бъдат обновени до най-актуалните им версии, а времето за провеждане на анализа да е съгласувано с техническия персонал и потребителите.

"Дисекция" на stuxnet

В своята книга Ерик Нап описва работата на вируса Stuxnet, като използва термина "дисекция". Обективно погледнато този термин подхожда повече за биологични видове, но поради изключително високата сложност на злонамерения код, в този случай "дисекция" е изключително подходящ. От програмна и архитектурна гледна точка stuxnet е комплексен, като неговата основна цел е да достави товар (payload) до точно определени системи, свързани с управлението на индустриалните процеси. Като принцип на работа този код се явява първият известен rootkit за индустриални мрежи.

Stuxnet може да се обновява, прескачайки някои от най-често използваните технологии за защита на мрежовата комуникация, както и да променя управляващите програми на ПЛК[24] на Siemens, като успешно прикрива своето присъствие изпращайки неверни данни към HMI[25] системите за наблюдение. Интересно е, че stuxnet използва вградени в кода данни за достъп до хардуер, които не са били публично достъпни, както и се прикрива чрез "валидни" цифрови сертификати, генерирани чрез откраднати секретни (private) ключове. Всичко това описва stuxnet не като прост зловреден код, а като изключително сложно кибер-оръжие.

Действието на stuxnet може да се опише със следните основни етапи:

- Инфектиране на Microsoft Windows операционна система, чрез използване на множество атаки от тип zero-day (открити технологични пропуски, но все още не коригирани от производителите със съответни софтуерни обновявания) и кражба на информация за цифрови сертификати. Инсталиране на rootkit на инфектираната система;
- Опит за заобикаляне на IPS функциите, които следят функцията LoadLibrary(). Това се извършва от специален процес, който зарежда необходимите софтуерни библиотеки (DLL) и при необходимост инжектира код към стартирани процеси;
- Най-често инфектирането се извършва като цялата библиотека на stuxnet се включва към съществуващ процес и при необходимост се повикват допълнителни функции;
- Извършва се проверка, дали системата използва поддържана версия на операционната система Windows, както и дали вече системата е била заразена. Преди да се извърши инжектирането на кода се стартира сложна проверка за откриване на наличен и активен антивирусен софтуер;
- Stuxnet се препраща чрез мрежова комуникация, flash памети, файлове с проекти на STEP 7 и др.;
- Търсят се специализираните индустриални системи Siemens WinCC (SCADA[26]). Ако такава система бъде открита се инжектира SQL код, като за да се прескочат системите за

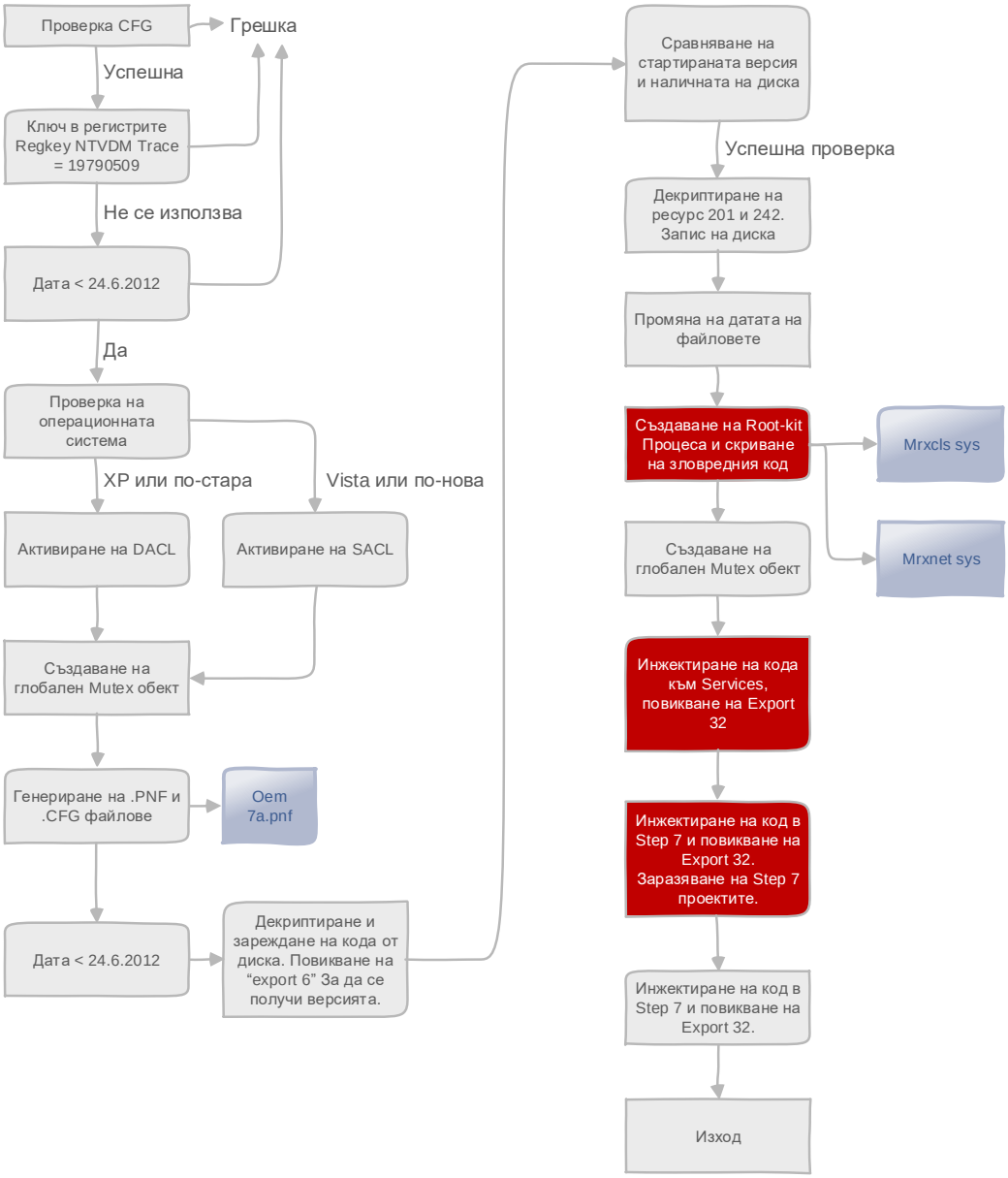
[24] Програмируеми логически контролери – специализирани системи за управление на индустриални процеси

[25] Human-Machine Interface – интерфейс за наблюдение и управление на индустриални процеси

[26] Supervisory Control and Data Acquisition – системи за наблюдение и управление на състоянието на производствени процеси

автентификация се използват заложени в кода на stuxnet данни за достъп (hard-coded authentication credentials). Ако инжектирането е успешно се получава достъп до S7 ПЛК;

- Управляващата програма на ПЛК се заменя, като се цели блокиране на определени процеси, като се генерира трафик през Profibus индустриалната комуникация, активиране на изходи и работа като rootkit, който служи за изпращане на команди към ПЛК;
- Заразените контролери се наблюдават за специфични дейности. Тази проверка се извършва на база на Profibus[27] комуникацията между индустриалните системи;
- Ако са открити точно определени настройки, свързани с управление на честота, stuxnet променя стойността от 14010 на 2 Hz за цикъл;
- Ако системата не е съвместима с stuxnet зловредния код се премахва, като неговите следи изцяло се заличават.



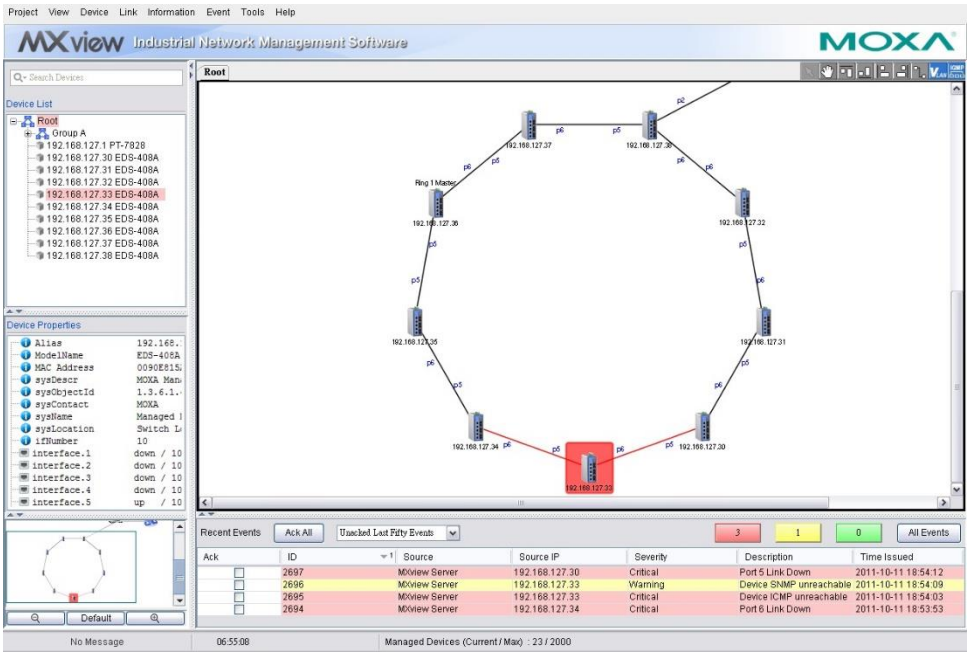
Фиг. 1.28 Основен алгоритъм на работа на stuxnet

[27] Специализиран индустриален комуникационен протокол на Siemens

Наблюдение

Един от задължителните етапи при изграждане на мрежовата сигурност както в офис мрежите, така и при индустриалната комуникация е наблюдението на работата на комуникационната инфраструктура, мрежовите устройства и системите за защита. За целта се използват редица технологии, като най-често те се базират на протокола Simple Network Management Protocol (SNMP), който позволява отдалечено конфигуриране и извличане на информация за работата на дадено устройство. Най-често системите за наблюдение предоставят централизиран интерфейс, чрез който лесно се наблюдава цялата мрежа, а при критични ситуации се визуализират алармени съобщения. Този тип софтуерни системи използват модулна структура и в зависимост от активираните и инсталирани модули могат да анализират различни параметри и да получат достъп до съответните модули. Съществуват комерсиални и решения с отворен код, като някои от най-популярните са:

- **Nagios** – система за мрежово наблюдение с отворен код, висока степен на гъвкавост и сравнително лесно конфигуриране и администриране. Към Nagios могат да бъдат добавени редица допълнителни модули, като Nagvis, Centreon и др., които позволяват графично представяне на мрежовата топология, използване на СУБД за съхранение на конфигурациите и по-лесно конфигуриране на отделните софтуерни пакети;
- **Icinga**[28] – алтернатива на Nagios, базирана на същия основен код, но с разширена поддръжка на допълнителни модули;
- **MXView** – специализиран софтуер на Моха за отдалечено и централизирано наблюдение на техните индустриални комуникационни устройства и др.



Фиг. 1.29 Моха MX View (източник Интернет)

Изборът на система за наблюдение се определя от редица фактори, като някои от по-важните са:

- Съвместимост със системите за комуникация и използваните протоколи;

[28] https://www.icinga.org

- Цена;
- Лекота при използване и поддръжка;
- Автоматично активиране на аларми и изпращане на съобщения до администраторите и инженерните отдели;
- Надеждност;
- Поддържани софтуерни платформи (Windows, Linux);
- Възможност за работа в режим клиент-сървър и др.

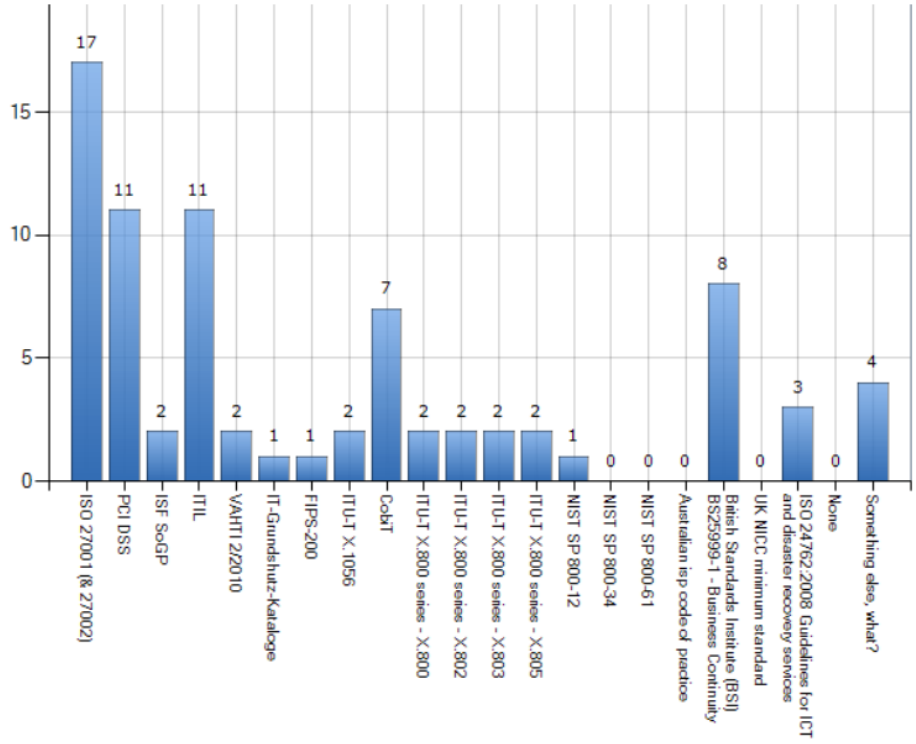
Стандарти и препоръки

Европейската агенция за мрежова и информационна сигурност ENISA (European Network and Information Security Agency) през 2012 година публикува доклад, съдържащ кратък списък с най-важните стандарти, свързани с информационната и комуникационна защита, както и препоръки към администраторите. Докладът се базира на резултати на анализатори и на анкета и интервюта, проведени от независими лица, като най-важните препоръки са описани във втори доклад, със заглавие “Technical Guideline for Minimum Security Measures”. И двата документа са разработени за да се подпомогне по-лесното въвеждане и хармонизиране на чл. 13а на европейска директива 2009/140/EC.

Кратък списък на по-важните стандарти, свързани с изграждането на информационната и комуникационна защита съдържа 20 стандарта:

1. ISO/IEC 27001/2:
2. ISO/IEC 24762:2008 Guidelines for ICT and disaster recovery services;
3. ISO/IEC 27005 Information security risk management;
4. ISO/IEC 27011 Information security management guidelines for telecommunications;
5. BSI BS25999-1 Business Continuity;
6. ITU-T X.1051 (02/2008);
7. ITU-T X.1056 (01/2009);
8. ITU-T X.800 (1991);
9. ITU-T X.805 (10/2003);
10. ISF Standard of Good Practice 2007;
11. CobiT;
12. ITIL Service Support;
13. ITIL Security Management;
14. IT-Grundschutz-Kataloge;
15. KATAKRI (FI);
16. NIST SP 800-34;
17. 17 NIST SP 800-61;
18. FIPS-200;
19. UK NICC Minimum Standard ND1643;
20. PCI DSS 1.2.

Интересни са резултатите от проведената анкета, които обобщават, че 100% от анкетираните използват IOS 27001 като основа за изграждане на информационната и комуникационна защита. 65% използват PCI DSS и ITIL. За да се осигури непрекъснатост на бизнес процесите при 47% от анкетираните се въвежда BS 25999. На фиг. 1.30 са показани отговорите на анкетираните за приложението на отделните стандарти от горепосочения списък.



Фиг. 1.30 Приложение на стандартите при анкетираните лица във връзка с директива 2009/140/ЕС на ЕС

Важно е да се отбележи, че в зависимост от типа на стандарта и областта, която той обхваща неговото приложение за дадени сектори може да е задължително.

Системите за управление на информационната сигурност (СУИС) имат за цел да гарантират конфиденциалността и интегритета на данните на организацията, да управляват надеждния достъп до тях и да оптимизират използваните ресурси по съхраняването им. Необходимостта от създаване и внедряване на СУИС произтича от редица фактори, като някои от по-важните са:

- Наличието на нормативна база и законови изисквания;
- Нарасналите изисквания на клиентите;
- Конкуренцията между компаниите, която води до изискване за повишаване на сигурността и конфиденциалността на фирмените данни и др.

Един от по-важните стандарти е ISO 27001:2005, който е част от групата документи ISO 27000. ISO 27001:2005 дефинира изискванията към системата за управление на информационната сигурност, като за целта дава формално описание (препоръки). Този стандарт съдържа 11 секции (без въвеждащите):

1. Политика за сигурност;
2. Организиране на информационната сигурност;
3. Управление на информационните активи;
4. Защита на служителите;
5. Защита на работната и околната среда;
6. Управление на комуникацията и операциите;
7. Контрол на достъпа;
8. Разработване, внедряване и поддръжка на информационни системи;
9. Управление на инциденти, свързани със сигурността на информацията;

10. Непрекъснатост на бизнес операциите;
11. Съвместимост.

Под термина процесен подход се разбира приложението на система от процеси в рамките на организацията, при точно дефиниране на тяхното взаимодействие. Процесният подход при изграждане на сигурност на информацията спомага за:

- Внедряване на механизми за защита на информацията;
- Ясно определяне на изискванията на организацията, свързани със защитата на информацията;
- Наблюдение и анализ на работата на СУИС;
- Ефективни подобрения на СУИС и др.

ISO 27001 се базира на цикъла на Деминг, който е показан на фиг. 19.24.

Фиг. 1.31 Цикъл на Деминг

Етапът на планиране обхваща създаването на политиката за информационна сигурност, процеси и процедури, свързани със СУИС и др. Изпълнението е свързано с внедряването на СУИС в организацията, а проверката изисква внимателно оценяване на постигнатите резултати след успешното приключване на предходния етап. В етапа за действие се предприемат стъпки, свързани с коригиране подобряване на СУИС, което най-често е на база на вътрешен одит. Целта е да се постигне непрекъснато подобряване на нивото на информационна защита и на качеството на внедрените СУИС.

ISO 27001:2005 дефинира следните задължителни изисквания:

- **Общи изисквания и изисквания към документацията в организацията** – общи условия, внедряване и поддръжка и изисквания към документите;
- **Отговорност на ръководството на организацията** – съпричастност и управление на ресурсите;
- **Вътрешен одит**;
- **Преглед от ръководството** – анализ на входните и изходните данни;
- **Подобряване на СУИС** – непрекъснато подобряване, коригиране на неточности и превантивни действия.

Ключовите процеси, дефинирани в ISO 27001 са следните:

- Управление на документооборота;
- Управление на записите;
- Анализ на риска;
- Управление на инцидентите, свързани с информационната сигурност;
- Анализ от страна на ръководството;
- Вътрешни проверки в областта на информационната сигурност;
- Управление и коригиращи действия;
- Мониторинг на ефективността от използваните механизми за защита в СУСИ;
- Организация на обучение и информираност в областта на информационната сигурност.

Заключение

Сложността и възможностите на инструментите и методите за атаки, насочени към информацията и комуникацията непрекъснато нарастват, което от своя страна води и до непрекъснатото развитие на технологиите за защита от тях. Необходимо е специалистите, които се занимават в тази област да са наясно с най-актуалните заплахи и начините за тяхното отстраняване, което изисква и непрекъснато обновяване на техните знания и умения.

Атаките, насочени към информационните и комуникационни технологии са разнородни, като обхващат зловреден код, атаки с цел достъп до ресурси, подвеждане на потребители, кражба на информация, спам, DoS и много други.

За да се изгради коректна и надеждна защита на мрежовата комуникация е задължително да се подходи систематично, като подхода се базира на бизнес процесите и на фирмената политика за защита. Налични са множество ресурси – както платени, така и свободно достъпни, които дават насоки за оптимизиране на защитата, както и на редица специализирани стандарти.

Източници

1. http://www.insecure.org
2. http://www.caida.org
3. https://github.com/rflynn/lanmap2
4. http://www.tuj.asenevtsi.com/Sec2009/Sec14.htm
5. http://www.metasploit.com
6. http://www.kali.org
7. http://www.tenable.com/products/nessus
8. http://www.openvas.org/
9. https://www.isc2.org/
10. https://www.icsalabs.com/
11. Chen, T., “Trends in Viruses and Worms”, Southern Methodist University, The Internet Protocol Journal - Volume 6, Number 3, September, 2003.
12. “Shortlisting network and information security standards and good practices”, ENISA, v.1.0, January 2012, www.enisa.europa.eu.
13. “Technical Guideline for Minimum Security Measures”, ENISA, v.1.0, December 2011, www.enisa.europa.eu.
14. http://www.tuj.asenevtsi.com/Sec2009/Sec14.htm - последно активен на 25.03.2014 г.

15. Knapp, E., “Industrial Network Security, Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems”, Syngress, ISBN: 978-1-59749-645-2, 2011, USA.

Глава 2. Защитни стени

Въведение

Още от средновековието хората са стигнали до простия и логичен извод, че най-добрият подход да защитят своите градове е да се спазва принципа „добрите“ вътре - лошите „вън“. За целта са били изграждани защитни стени и препятствия с малко надеждно пазени и наблюдавани входове. Тази защита, особено в случаите, когато е била добре реализирана се е оказвала не веднъж сериозно предизвикателство пред атакуващите. Ако се върнем към актуалните за момента технологии и по-специално мрежовата комуникация, може да се направи аналог със средновековието. При мрежови топологии, които имат ясно изразен периметър (гранични устройства между вътрешните мрежи и доставчиците на WAN свързаност и Интернет) изграждането на защита от външни атаки може да стане отново чрез поставяне на защитна стена (Firewall) между вътрешните мрежови сегменти и външните комуникационни линии. Този подход е успешен за дълъг период от време, но към момента е трудно приложим, поради размиването на мрежовия периметър, чрез използване на мобилни устройства, облачни услуги, отдалечен достъп до ресурсите на компанията и безжичните мрежи. Cisco Systems® наричат този тип мрежови топологии „мрежа без граници“ (Borderless Network). Защитата на „мрежите без граници“ изисква нови типове устройства и технологии, но едно от най-важните и към момента си остава защитната стена.

Необходимост от защитни стени

Защитните стени са специализирани системи или софтуерни пакети, които предпазват мрежовия трафик и устройствата от редица атаки сред които:

- Сканиране;
- Неоторизиран достъп до ресурси;
- Атаки, насочени към потребителски профили;
- Изтичане на информация към неоторизирани лица и системи и др.

Защитните стени предоставят някои важни функции като:

- Пакетно филтриране;
- Следене на сесиите (Stateful firewall);
- Дефиниране на зони и филтриране на трафика между тях;
- Прозрачно действие;
- Устойчивост при атаки;
- Налагане на корпоративната политика за защита от гледна точка на това кои устройства имат достъп до мрежата, и кои не;
- Високо бързодействие и производителност;
- Наблюдение в реално време и подробни отчети;
- Маршрутизиране на трафика и много др.

Изключително важно е да се отбележи, че защитните стени сами по себе си не могат да защитят целия трафик от всички видове атаки и често се налага допълване на защитата с IDS/IPS, антивирусни и други специализирани системи и технологии.

История на защитните стени

Технологията за защитните стени започва да се разработва през 80^{те} години на 20^{ти} век. Тогава Интернет е бил все още нова и не толкова популярна комуникационна инфраструктура, като маршрутизаторите са били използвани и за филтриране и защита на трафика.

През 1988 година инженери от Digital Equipment Corporation (DEC) публикуват доклад, в който описват технология за филтриране на пакети, наречена “Packet Filtering”[29], като по същество това се явява първата защитна стена. На база на описанието от този доклад в AT&T Bell Labs разработват работещ прототип на защитна система с пакетно филтриране.

Пакетното филтриране анализира хедъра на пакетите и в зависимост от предварително дефинирани от администраторите критерии или пропуска пакета или го отхвърля. По този начин може да се анализира информация до транспортното ниво (4) на OSI референтния модел, като в общия случай се използва:

- Адрес на изпращащата система;
- Адрес на получател(и);
- Тип на транспортния протокол;
- Порт на изпращащата система;
- Порт на получаващата система.

Пакетното филтриране не предоставя гъвкава възможност за по-комплексен анализ на трафика, а при по-сложни мрежови топологии е трудно за конфигуриране и поддръжка.

Второто поколение защитни стени анализират и сесиите между приложенията. Това са т.нар. “stateful‘ технологии. Идеята за този подход е създадена от трима инженери от AT&T bell Labs, която те наричат “Circuit-level Gateways”. За разлика от първото поколение защитни стени вече се използва по-сериозно анализиране на трафика от гледна точка на транспортното ниво на OSI референтния модел. Проверяват се сегментите и дейтаграмите дали са от активна сесия между двете комуникиращи приложения, която предварително е разрешена от администраторите. По този начин защитната стена може да предотврати и някои от по-старите DoS атаки.

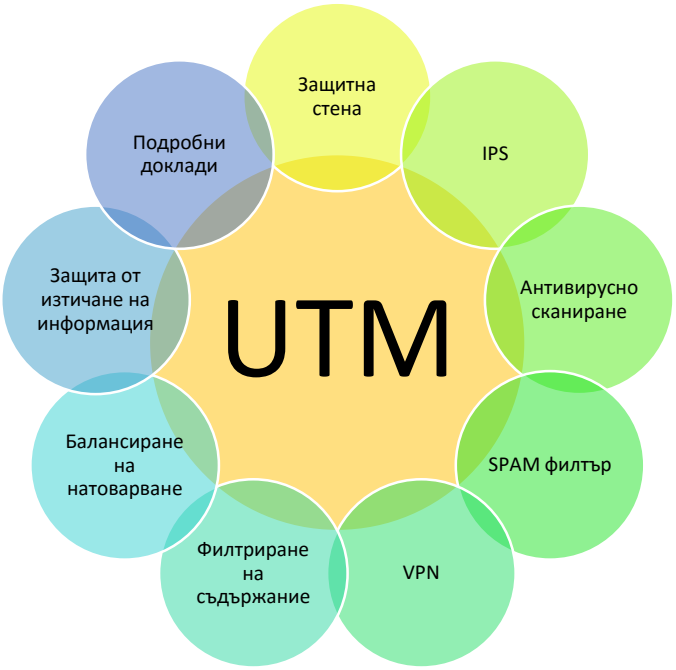
Маркус Ранъм, Уей Ксю и Паутър Чърчярд разработват модел на защитна стена, която да анализира трафика до приложното (7) ниво на OSI референтния модел. В последствие Ксю успешно реализира програмен код, който прави IP филтрирането и следенето на сокети[30] прозрачно, като добавя функционалност в ядрото на използваните операционни системи (UNIX). Trusted Information Systems пускат първата защитна стена от тип “application layer”, която се нарича Gauntlet (в превод – рицарска ръкавица). Едно от най-големите предимства на защитните стени, които инспектират трафика до приложното ниво е възможността за анализ на обмяната на информация между приложенията, като се проверява използвания протокол, например HTTP, FTP, DNS и др. По този начин може TCP порт 80 да е отворен, но вместо очаквания HTTP трафик да се пренася TFTP, и ако защитната стена извършва проверка на приложния протокол веднага ще блокира неоторизирания трансфер.

През 2012 година към функциите на “application layer” защитните стени се допълва и дълбоко сканиране на пакетите, което позволява да се интегрира и допълнителна функционалност от типа на IDS/IPS, автентификация, задълбочен анализ на WEB приложения и др.

Unified Threat management (UTM) е технология, която е разработена през 2004 година и цели да изгради едно устройство, което да интегрира функциите на няколко защитни системи.

[29] Пакетно филтриране

[30] Комбинация от адрес и порт



Фиг. 2.1 Технология UTM

UTM бързо се утвърждава като водеща технология и пазарния дял на този тип устройства през 2007 година (3 години след създаването) достига 1.2 милиарда USD.

Някои от предимствата на UTM са:

- Намаляване на сложността на инсталиране, конфигуриране и поддръжка;
- Опростяване на топологията – не се налага интегриране на няколко разнородни технологии;
- Лесно конфигуриране и наблюдение;
- Намаляване на разходите за обучение на персонал;
- Съвместимост с регионалните изисквания на различните държави и др.

Като по-съществени недостатъци на UTM може да се посочат:

- Устройството се явява критично (single point of failure);
- Потенциално забавяне на трафика, ако се използва система с недостатъчно висока производителност или ограничени ресурси.

Подсигуряване на мрежовата комуникация със защитна стена

Както вече беше споменато защитните стени са едни от най-важните устройства, използвани при подсигуряването на мрежовия трафик, но сами по себе се не са достатъчни за пълна защита, дори ако се използва UTM решение. Изборът на защитна стена е важно и отговорно решение, като под внимание трябва да се вземат редица фактори сред които:

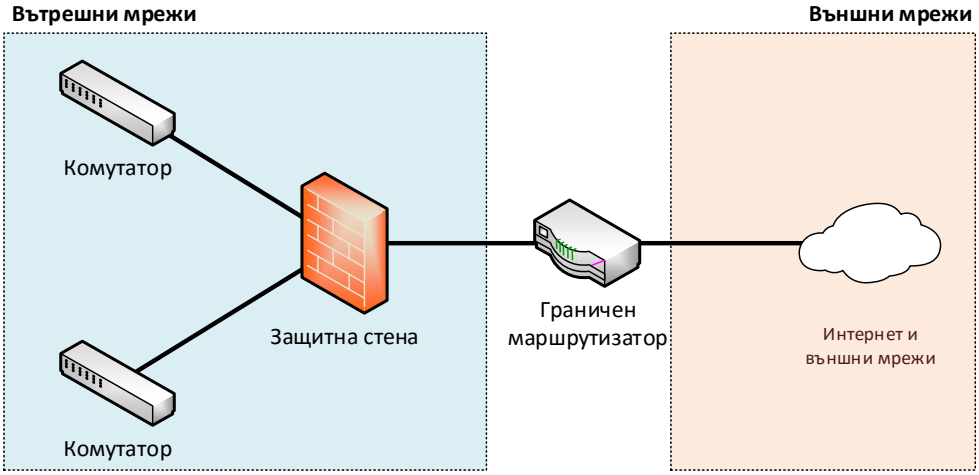
- Къде точно в мрежовата топология ще се поставят защитните стени;
- Какви са използваните протоколи;
- Каква технология за защитна стена би била подходяща;
- Какъв е обемът на трафика (не само като пропускателна способност, но и като брой пакет в секунда);
- Какъв е бюджета с който се разполага;
- Съвместимост с други системи и мрежови устройства;

- Теоретична и практическа подготовка на персонала и др.

Правилно избрани защитни стени могат значително да повишат цялостната сигурност на комуникацията, докато грешен избор би довел до редица проблеми сред които забавяне на трафика, по-лесни атаки, филтриране на грешни данни и др.

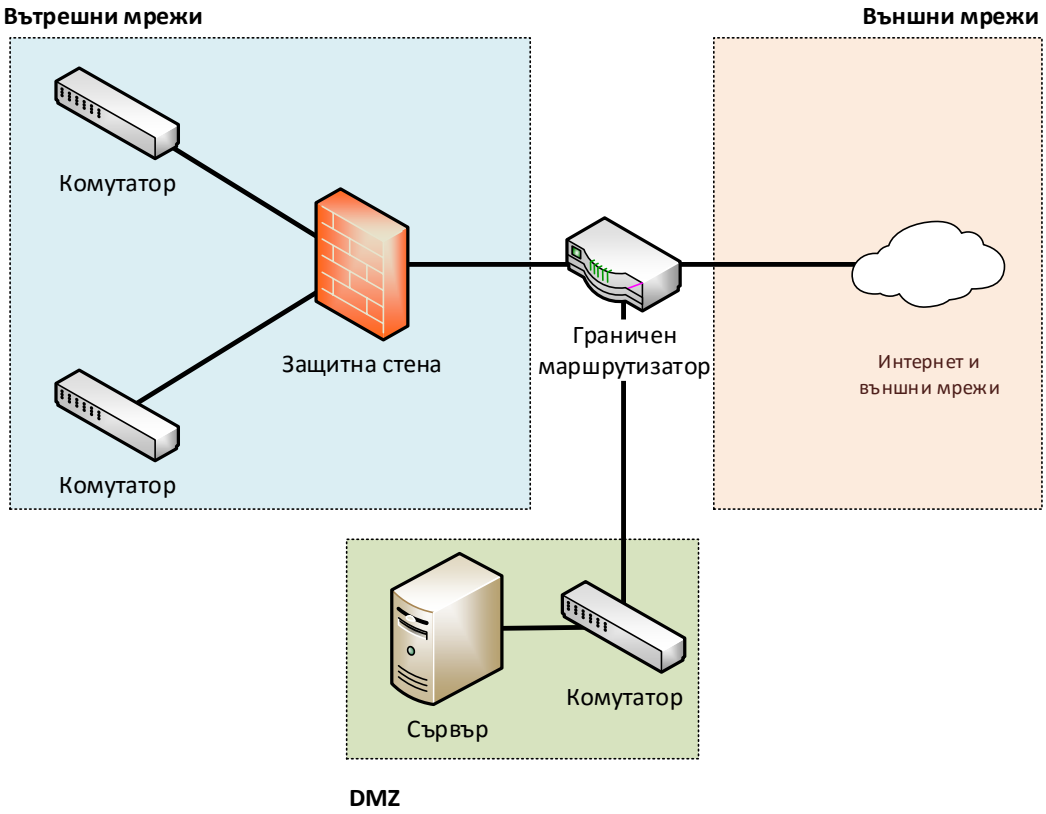
Примерни топологии

При малки мрежови топологии с ясно дефиниран периметър защитните стени могат да се поставят след граничния маршрутизатор. По този начин лесно се подсигурява трафика, ако не се изискват сложни правила за филтриране. Заявките от вътрешните мрежови сегменти се пропускат към външните мрежи само за разрешените протоколи и услуги, а трафикът от вън се разрешава ако е отговор на вътрешна заявка (типичен пример за работа на “stateful” защитните стени). Ако се налага външни заявки да могат да се правят към точно определени вътрешни системи се използва пренасочване на портове (възможни са и други подходи – например статичен NAT).



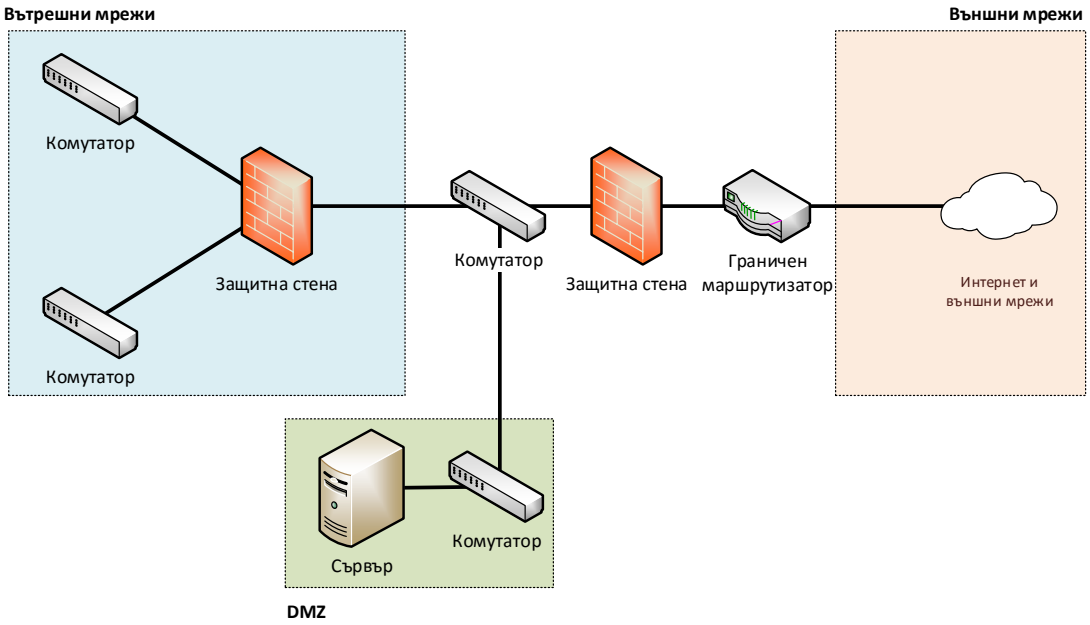
Фиг.2.2 Примерна топология със защитна стена при малки мрежи

При по-сложни мрежови топологии и при наличие на специализирани сървъри, които се поддържат в рамките на вътрешната мрежа е по-целесъобразно да се добави специална зона, наречена демилитаризирана (Demilitarized Zone - DMZ). Най-често DMZ се конфигурира, като нейния мрежови сегмент се свързва към един от интерфейсите на защитната стена. По този начин трафика към DMZ и останалите зони (вътрешна и външна) може лесно да бъде филтриран и по-задълбочено анализиран. Пакетите от вътрешните мрежови сегменти към DMZ и външните мрежи са разрешени за определените в политиката за защита услуги и протоколи. Външните мрежи могат да изпращат заявки към DMZ зоната, а към вътрешните устройства само ако се използва пренасочване на портове. Може да се обобщи, че устройствата от DMZ отговарят на заявки и само при необходимост могат да генерират трафик към външни мрежи или определени вътрешни мрежови сегменти.



Фиг. 2.3 Примерна топология със защитна стена и DMZ

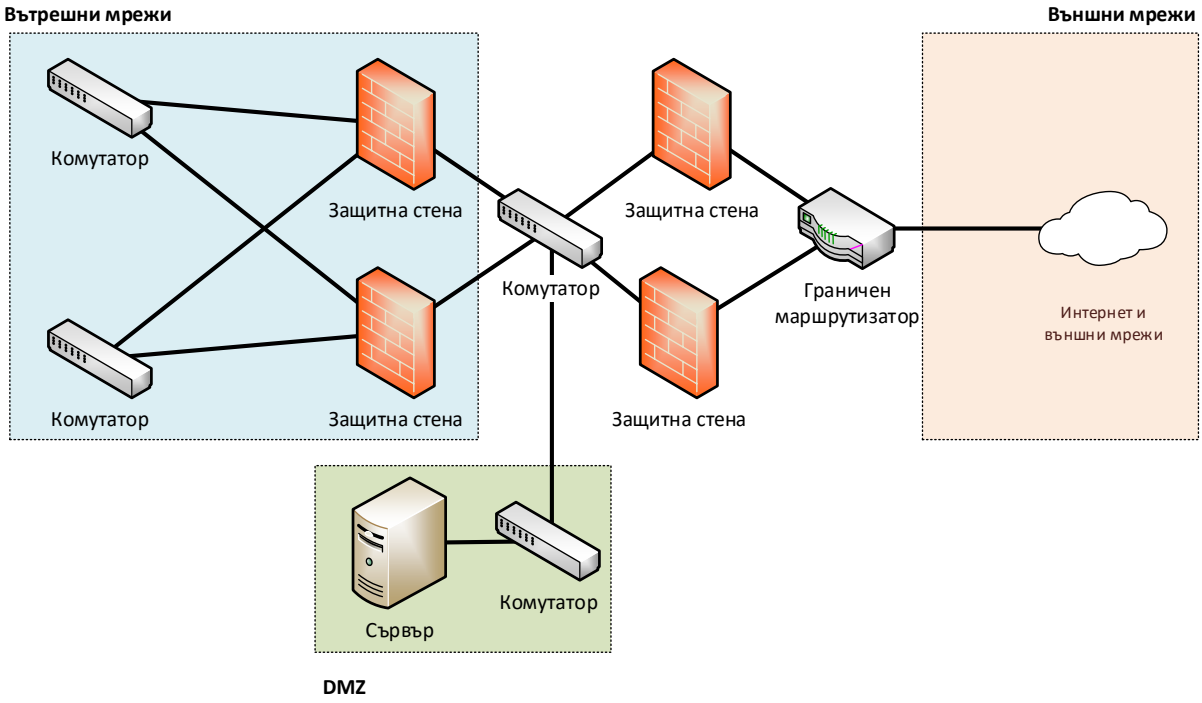
Друг подход за по-висока степен на подсигуряване на DMZ изисква инсталиране и конфигуриране на допълнителна защитна стена.



Фиг. 2.4 Увеличаване на степента на сигурност, чрез добавяне на няколко защитни стени

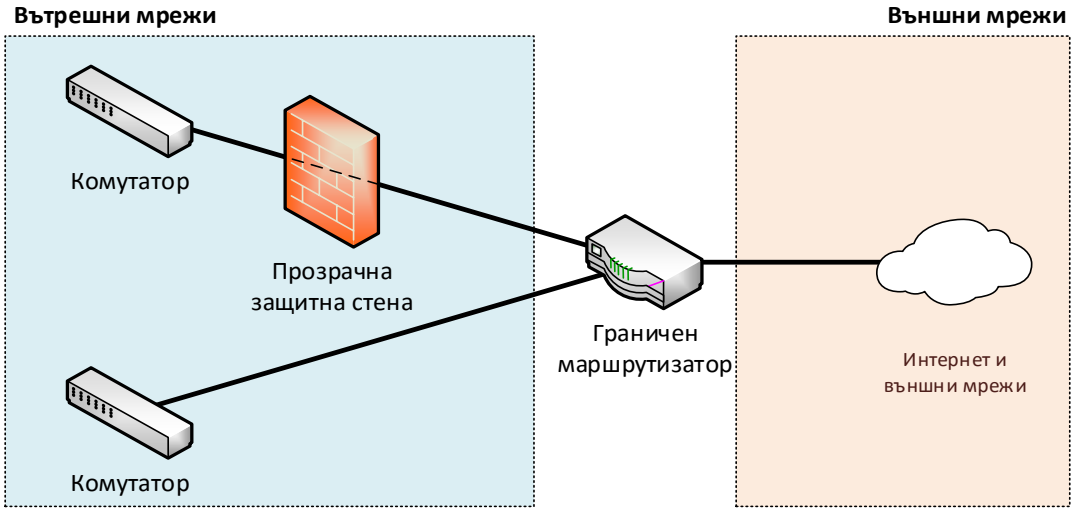
Защитната стена може да се яви критично устройство и често срещана практика е да се използва дублиране на този тип системи. За съжаление, това води до необходимост от конфигуриране на специални допълнителни протоколи, които следят дали активното устройство

е достъпно и при необходимост го заместват с друго, както и да значително нарастване на инвестицията.



Фиг. 2.5 Топология с дублирани защитни стени

Добавянето на защитна стена към мрежови сегмент може да бъде прозрачно (основно работи на каналното ниво на OSI референтния модел), като по този начин устройството не намалява TTL и от гледна точка на потребителите е незабележимо.



Фиг. 2.6 Прозрачна защитна стена

Предимства и недостатъци

Технологиите, използвани при защитните стени имат редица предимства:

- Защитават вътрешните устройства и потребители от злонамерени атаки от външни мрежи;
- Намаляват вероятността потребителите да са изложени на риск от външни атаки;

- Пренасяните протоколи могат внимателно да бъдат филтрирани и по този начин да се редуцира броя на потенциалните заплахи;
- Възможно е да се блокира злонамерен достъп до сървъри, мрежови устройства и хостове;
- Фирмената политика за сигурност се налага със по-строги критерии;
- Намалява се ненужния мрежови трафик;
- Условно потребителите могат да се чувстват по-защитени и др.

Някои от по-важните недостатъци на защитните стени като цяло са:

- В зависимост от топологията могат да се явяват критични устройства;
- При грешна конфигурация могат значително да забавят мрежовия трафик или да блокират легитимен трафик;
- Потребителите могат да се опитват да намерят начин за заобикаляне на защитните стени с цел използване на забранени мрежови приложения или достъп до забранени сървъри, чрез тунелиране на трафика или прокси сървъри;
- Ако към вътрешен хост е изграден криптиран VPN тунел защитната стена не може да анализира съдържанието на шифрираните пакети.

Видове защитни стени

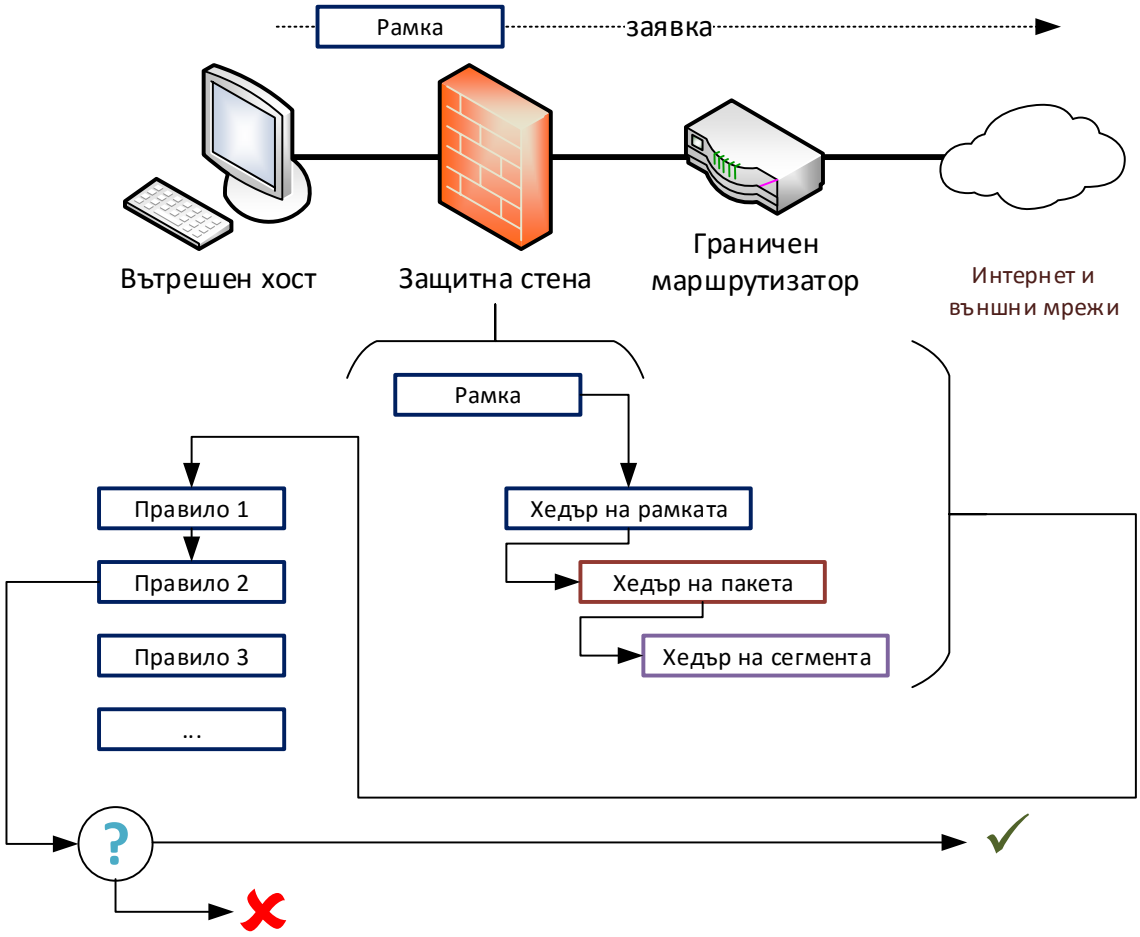
В началото на главата беше направено кратко описание на някои от по-важните видове защитни стени. По-точното им класифициране дефинира следните групи:

- **Пакетно филтриране** – най-често анализа на трафика се извършва на мрежовото (3) и на транспортното (4) нива на OSI референтния модел;
- **“Stateful”** – следят състоянието на сесиите, от къде започва генерирането на заявките, обема на пренасяната информация, както и допълнителни параметри, свързани с транспортния протокол;
- **Прокси или “application gateway”** – анализират трафика на мрежовото (3), транспортното (4), сесийното (5) и приложното (7) нива на OSI. Извършват проверка на използвания протокол за обмяна на данни между приложенията. По-голямата част от тяхната функционалност е програмно реализирана;
- **NAT** – защитна стена, която поддържа NAT и се използва като устройство за транслиране на частни към публични IP адреси. Тази функционалност е налична при повечето актуални защитни системи;
- **Базиран на зонова политика (Zone Policy Firewall)** – при по-сложни мрежови топологии вместо да се използват отделни интерфейси към всеки мрежови сегмент се прави логическо обобщение на зоните, коти има в мрежата (например вътрешна, администратори, външна, DMZ и др.). Дефинират се правилата за трафика между зоните и към всяка една зона се добавят един или повече физически интерфейси.
- **UTM** – устройство, което освен като защитна стена предоставя възможност за използване на IDS/IPS, WEB филтриране, автентификация и други специализирани защитни технологии;
- **Персонални** – най-често софтуерни продукти, които се явяват защитна стена, която предпазва операционната система, файловете и процесите на хостовете, използвани от потребителите.

Пакетно филтриране

Пакетното филтриране е най-старата технология, използвана от защитните стени. Тя предлага единствено базова защита, но има високо бързодействие, при оптимизиран хардуер и софтуер. На фигура 2.7 е показан принципа на действие на защитните стени от този тип:

1. Защитната стена разделя вътрешните устройства от външните мрежи и по подразбиране разрешава заявки, които започват от хост, който се намира във вътрешен сегмент;
2. Вътрешен хост изпраща заявка за комуникация към външно устройство, като рамката се получава в защитната стена;
3. Защитната стена възстановява служебната информация до транспортното ниво на OSI референтния модел, като по този начин се извличат данните от хедърите на рамката, пакета и сегмента (дейтаграма);
4. Алгоритъмът за анализ започва да сравнява данните от рамката, пакета и сегмента с правилата, които са дефинирани от администратора. Първо се извършва сравнение с правило № 1, ако неговите условия за проверка не се изпълняват се преминава към правило № 2 и т.н. Ако нито едно правило не отговаря на условията за проверка рамката се отхвърля;
5. Ако условията на дадено правило са изпълнени, то в зависимост от конфигурираното действие рамката или се пропуска през защитната стена или се отхвърля.



Фиг. 2.7 Пакетно филтриране

Както се вижда от примера (отнася се и за останалите видове защитни стени) подредбата на правилата е от съществено значение. Принципът е да се поставят най-конкретните правила – тези, който извършват проверка по най-много критерии преди по-общите правила. Например ако първото правило, което администраторите са конфигурирали е да се разреши целия трафик, без значение колко последващи проверки са въведени те никога няма да се проверят, а целия трафик ще преминава през защитната стена. Добрата практика е винаги преди да се конфигурират правилата на защитната стена те да бъдат внимателно обмислени, проверени за несъответствия и несъвместимост и при възможност конфигурирани на тестова топология.

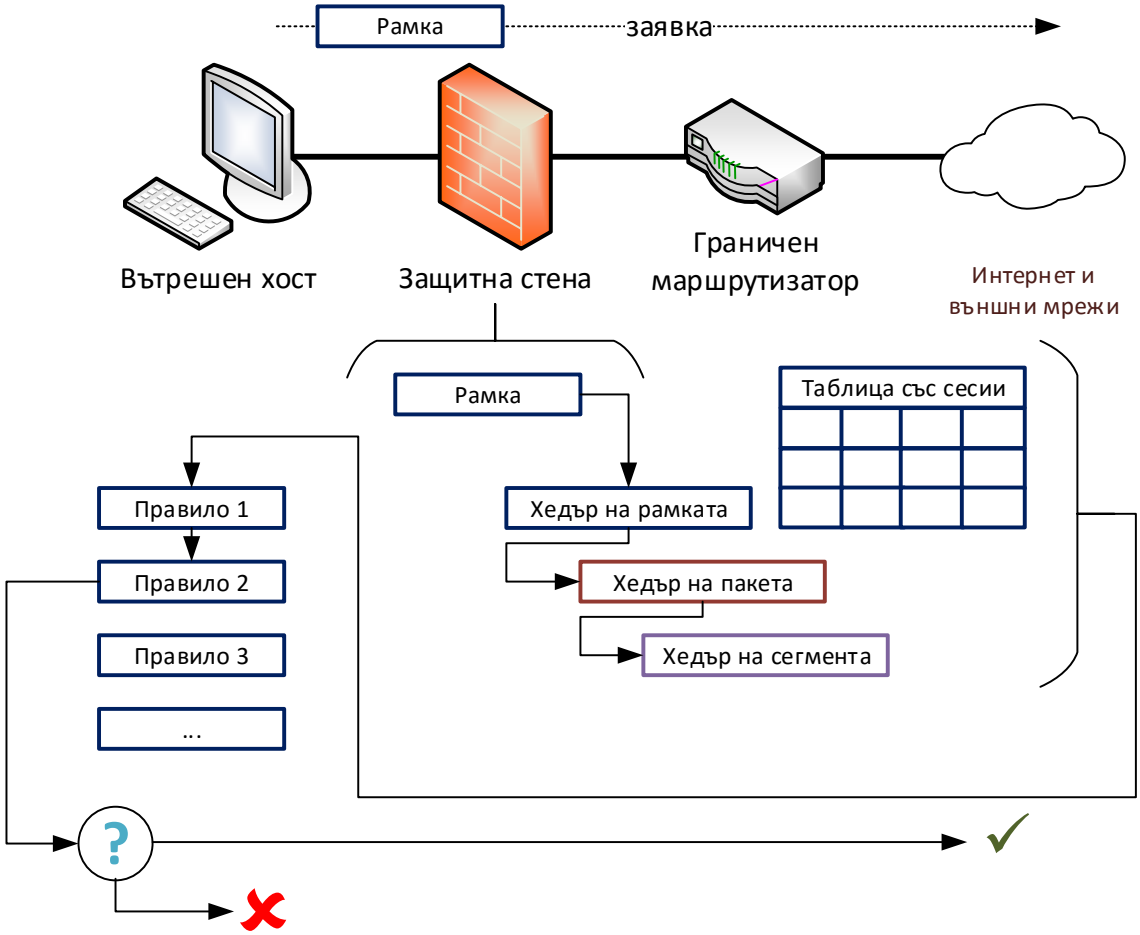
"Stateful" защитна стена

За разлика от пакетното филтриране при технологията "stateful" се добавя разширено анализиране на функциите на сесийното ниво на OSI референтния модел. Това изисква допълнителни системни ресурси, които да подсигурят поддържането на т.нар. „state table" (таблица, описваща състоянието на сесийте, използвана от по-сложното анализиране на трафика). Ако транспортният протокол е TCP следенето на сесията е сравнително лесно, като могат да се използват флаговете SYN, ACK, FIN, RST, PSH, както и други параметри, свързани със сегментите. Не така стои въпросът, ако информацията се транспортира с UDP. UDP е ненадежден протокол и не изисква предварително изграждане на сесия между приложенията. В този случай алгоритъмът, използван от защитната стена може да направи предположение дали дейтаграма е част от "сесия" на база на параметри като адреси на комуникиращите устройства, използвани портове, време от последния транспортиран пакет и др.

Алгоритъмът на работа на „stateful" защитните стени е подобен на пакетното филтриране, но от хедъра на сегмента (дейтаграма) може да се извлече допълнителна информация за сесията:

1. Защитната стена разделя вътрешните устройства от външните мрежи и по подразбиране разрешава заявки, които започват от хост, който се намира във вътрешен сегмент;
2. Вътрешен хост изпраща заявка за комуникация към външно устройство, като рамката се получава в защитната стена;
3. Защитната стена възстановява служебната информация до транспортното ниво на OSI референтния модел, като по този начин се извличат данните от хедърите на рамката, пакета и сегмента (дейтаграма);
4. От хедъра на рамката могат да се получат адресите (ако има такива) на двете устройства, които си комуникират, както и да се определи кой е използвания мрежови протокол. От хедъра на пакета се извличат адресите на изпращащото устройство и на получаващото, както и се дефинира транспортния протокол. При необходимост може да се провери за фрагментирани пакети. В хедъра на транспортния протокол са налични портовете на комуникиращите приложения, като в зависимост от протокола има и флагове, които спомагат за контрола на сесията. Данните за сесията се записват в „state" таблицата, като при необходимост се добавя нов запис или се модифицира съществуващ. Още на този етап може да се провери дали пакета може да премине (част от изградена сесия или нова) през защитната стена;
5. Алгоритъмът за анализ започва да сравнява данните от рамката, пакета и сегмента с правилата, които са дефинирани от администратора. Първо се извършва сравнение с правило № 1, ако неговите условия за проверка не се изпълняват се преминава към правило № 2 и т.н. Ако нито едно правило не отговаря на условията за проверка рамката се отхвърля (принцип по подразбиране при затворен мрежови модел);

6. Ако условията на дадено правило са изпълнени, то в зависимост от конфигурираното действие рамката или се пропуска през защитната стена или се отхвърля.



Фиг. 2.8 “Stateful” защитна стена

Предимство на “stateful” защитните стени е допълнителното ниво на сигурност, което се внася от функциите за проследяване на сесийте. Този модел на работа има редица предимства пред пакетното филтриране, въпреки повечето използвани системни ресурси. Като основен недостатък може да се посочи сравнително по-трудното конфигуриране, поддръжка и наблюдение при сложни топологии или при наличие на множество интерфейси на защитната стена.

„Application layer” защитна стена

Развитието на зловредния код и по специално на червеите води до разработването на допълнителна функционалност, свързана със защитните стени, която цели да се извърши задълбочен анализ на използваните приложни[31] протоколи.

Пакетното филтриране и “stateful” технологията не позволяват да се проследи дали на определен порт се обменя очаквания от администраторите приложен протокол. Например към отдалечен сървър и порт 443 може да не се пренася HTTPS трафик, а RDP пакети. Това нямам как да бъде засечено от по-старите защитни стени, защото в тяхната конфигурация до транспортното

[31] Application protocol – протоколи, работещи на 7 ниво на OSI референтния модел

ниво на OSI референтния модел всичко ще бъде разрешено (адресите, отдалечения порт и транспортния протокол).

Добавените разширени възможности за дълбоко сканиране на пакетите увеличава възможностите на “application layer” защитните стени, като вече може да се следи и за определени атаки (използвайки сигнатури подобно на IDS/IPS) технологията.

Cisco Systems® разработват Context-Based Access Control (CBAC), която позволява да се реализират няколко паралелни нива на защита на трафика:

- Пакетно филтриране;
- Инспекция на приложните и други използвани протоколи;
- Извеждане на подробна информация за открити атаки (alert и audit trail);
- Базова IDS функционалност.

CBAC извършва интелигентно сканиране на TCP и UDP, като анализира сесиите, проверява приложните протоколи и следи за атаки чрез сигнатури. Също така CBAC може да предпази системните ресурси при злонамерен трафик и целенасочена атака. Въпреки предимствата CBAC е сравнително трудна за конфигуриране технология при наличие на голям брой интерфейси и мрежи.

Този тип защитни стени към момента са изместени от базираните на зони и UTM решения.

Базирана на зони защитна стена

С все по-широкото навлизане на Интернет както в бизнес процесите, така и в ежедневието на домашните потребители, а и с непрекъснато нарастващата скорост на трансфер на данни между устройствата технологията на “stateful” защитните стени започва да се явява трудно приложима, особено при сложни топологии.

Базираните на зони защитни стени използват абстрактно представяне на зоните, в които се намират отделните устройства – например вътрешна зона, DMZ, Интернет, бизнес партньори и др. След като са дефинирани отделните зони се описва трафика между всеки две от тях. Това описание съвпада с правилата на по-старите технологии защитни стени и отново цели еднозначно да определи дали даден пакет може да премине или не.

След като се дефинират отделните зони, които се явяват и граници за трафика, към всяка една от тях трябва да се добавят един или няколко физически интерфейса на защитна стена. По този начин свързването на нови мрежови сегменти или VLAN към дадена зона е лесно, а модела става гъвкав от гледна точка на промени в топологията.

Cisco Systems® дефинират няколко основни правила при реализирането на защитна стена, използваща зони, които са в сила и за решения на други производители:

- Зоните трябва да бъдат конфигурирани преди да се добавят интерфейсите към всяка от тях;
- Даден интерфейс може да се намира само в една единствена зона;
- Всичкия входящ и изходящ трафик към и от определен интерфейс по подразбиране е блокиран. Това се отнася, ако източника на трафика и получателя се намират в две различни зони, а действието се извършва в момента в който интерфейса стане част от зона;
- Целият трафик между два интерфейса, които се намират в една и съща зона е разрешен;

- Трафик не може да преминава между интерфейс, който се намира в дадена зона и друг интерфейс, който не е член на никоя зона;
- За да може трафика да преминава между две зони е необходимо внимателно да се конфигурира политика, която да дефинира какъв тип анализ на пакетите ще се извършва и съответните действия;
- Интерфейсите на устройството, които не са включени в дадена зона работят самостоятелно.

Абстрактното представяне на зони има множество предимства, особено при сложни фирмени политики за сигурност, множество виртуални LAN мрежи (VLAN) и тунелиране. Това е най-разпространената към момента технология, която се използва както от хардуерни защитни стени, така и от софтуерни и персонални решения.

Персонални защитни стени

Персоналните защитни стени са последното ниво на предпазване на работните станции на потребителите от атаки и нежелан трафик. Най-често те са софтуерни приложения, които работят на принципа на “application layer” и са специално оптимизирани и съобразени с начина на работа на конкретната операционна система.

Някои от по-важните функции на персоналните защитни стени са:

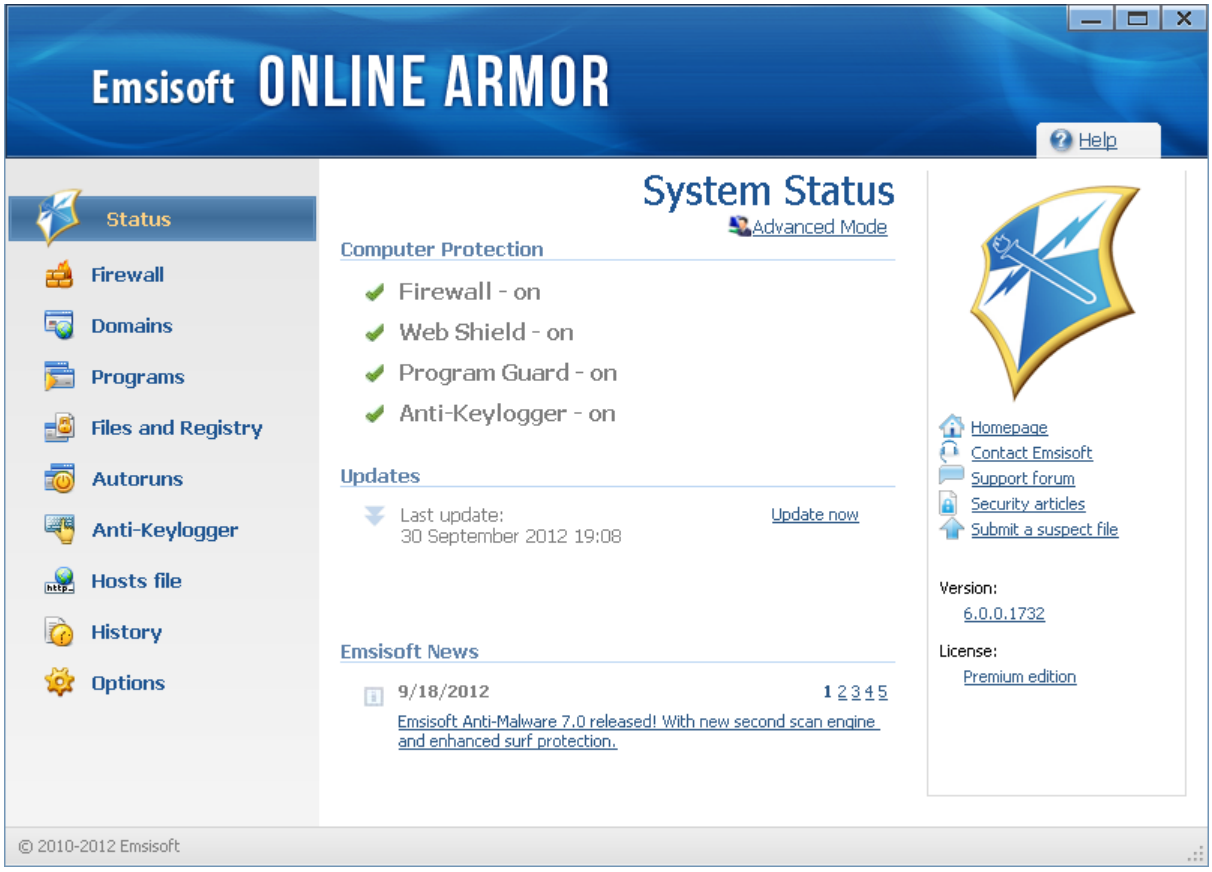
- Интегриране в операционната система;
- Защита на потребителя от нежелан мрежови трафик;
- Предоставяне на възможност на потребителя да конфигурира, кои приложения да имат достъп до мрежовите ресурси, и кои – не;
- Блокиране и своевременно известяване на потребителя за опити за свързване към забранени портове на използваната от него система;
- Наблюдение на работата на приложенията и на отворените портове;
- Постоянна проверка на постъпващите и напускащите системата пакети;
- Блокиране на нежелан трафик, генериран от локално инсталирани приложения;
- Подробни доклади, статистика и др.

Като недостатъци може да се посочат:

- Ако системата е била успешно заразена със зловреден код, е възможно той да заобиколи филтрирането и анализа на персоналната защитна стена. Важно е да се отбележи, че това е изключително трудно и е възможно само за много малък процент от червейте, като при откриването на тяхната функционалност защитните стени биват обновени и зловредния код се блокира;
- Ако конфигурацията на защитната стена е грешна или е неточна е възможно да се филтрира нормален трафик, а нежелани пакети да бъдат пропуснати;
- Този тип защитни стени не винаги могат да открият някои от по-новите и сложни атаки, които лесно се идентифицират от мрежовите защитни системи от тип IDS/IPS;
- Въпреки, че дадена системи има подсигурена комуникация с Интернет, тя все още може да е уязвима от атаки, извършени от вътрешните мрежови сегменти;
- Прекалено много информация за потенциални атаки и въпроси за действие правят някои защитни стени нежелани от потребителите и често част от тяхната функционалност се изключва;
- Ако в операционната система е наличен технологичен пропуск е възможно той да се използва за да бъде заобиколена функционалността на персоналната защитна стена.

В последно време разработчиците на специализиран защитен софтуер разширяват възможностите на своите продукти, като интегрират персонална защитна стена, антивирусно сканиране, “sanboxing[32]” функции, родителски контрол, защита на личните данни и др. По този начин предлаганите решения стават по-сложни от програмна гледна точка, но потребителите биват по-добре защитени. Във всяка модерна операционна система има вградена защитна стена, която предоставя висока производителност и надеждност. Въпреки това някои от популярните допълнителни продукти за персонални защитни стени са:

- Zonealarm;
- Comodo Firewall;
- TinyWall;
- Online Armor Free;
- Outpost Firewall;
- Firestarter (Linux);
- Guarddog (Linux) и др.



Фиг. 2.9 Персонална защитна стена Emsisoft ONLINE ARMOR (източник Интернет)

Други видове защитни стени

Освен изброените видове защитни стени се срещат и:

- **Сървърни защитни стени** – софтуерни продукти, които работят под управлението на специализирани сървърни операционни системи (Linux, Microsoft Windows Server и др.). Най-често този тип защитни стени предпазват потребителите и вътрешните

[32] Изолиране на системните ресурси, използвани от процесите

мрежови сегменти от заплахи и нежелан трафик, постъпващ от Интернет и външни зони;

- **NAT защитни стени** – този термин се използва в документите на Cisco Systems® и описва защитна стена, която предоставя възможност за NAT. Повечето актуални мрежови защитни стени към момента имат пълна поддръжка на NAT и попадат в тази категория;
- **Хибридни** – комбинация от няколко технологии за защитни стени;
- **Индустриални защитни стени** – специализирани системи, които анализират мрежовия трафик, използван в индустрията и предпазват управляващите системи и контролните модули от заплахи и атаки.

Фиг. 2.10 Индустреална защитна стена на adstec (източник Интернет)

Място на защитната стена в мрежовия дизайн

Интегрирането на защитната стена във вече изградена мрежова топология или определянето на оптималното местоположение по време на проектиране изисква внимателен и систематичен анализ, който да обхваща:

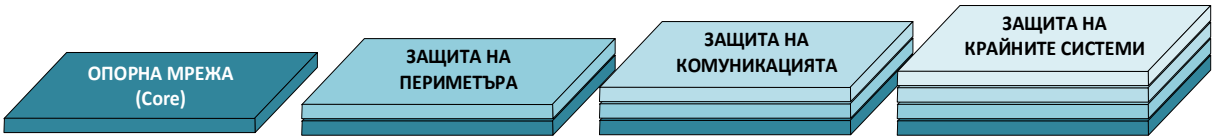
- Използвани мрежови, транспортни и приложни протоколи;
- Адресиране на устройствата;
- Обем на трафика (данни и пакети в секунда);
- Брой крайни и мрежови устройства;
- Необходимост от допълнителна функционалност (антивирусно сканиране, IDS/IPS, PoE, QoS и др.);
- VPN тунели – изпилвани технологии, брой и тип;
- Наличие на DMZ и др.

Липсата на ясно дефиниран мрежови периметър изиска абстрактно представяне на нивата на защита, което да обхваща вътрешните и външни зони, мрежови сегменти и опорната мрежова комуникация. Оптимално поставени и конфигурирани защитни стени биха предпазили мрежовите сегменти и устройства, като в същото време от икономическа гледна точка биха оправдали своята инвестиция (която в определени случаи може да е значителна). Лошо определено място на защитна стена може да доведе до редица проблеми, свързани с филтриране на нормален трафик, пропуск на нежелани пакети, забавяне на работата на мрежата и др.

Нива на защита

Защитата на мрежовата комуникация може да се обобщи в 4 нива:

1. **Защита на опорната мрежа** – предпазване от зловреден код, аномалии в трафика, налагане на фирмената политика за защита и гарантиране на устойчивост на атаки;
2. **Защита на периметъра** – дефиниране, конфигуриране и подсигуряване на границите между зоните;
3. **Защита на комуникацията** – подсигуряване на преноса на данните и защита от изтичане или модифициране на информация;
4. **Защита на крайните системи** – идентифициране на системите и потребителите, налагане на фирмената политика за защита и проверка за съвместимост с нея.



Фиг. 2.11 Основни нива на мрежова защита

Място на защитните стени в политиката за защита

Моделът, базиран на нива и описващ приложението на защитните стени, сам по себе си не е достатъчен за да се изгради надеждна и адекватна защита на мрежовата комуникация. Дори да се използват най-актуалните технологии, те отново изискат внимателен анализ и често добавяне на допълнителни устройства и системи за защита.

При проектиране на фирмената политиката за сигурност относно защитните стени трябва да се вземе под внимание:

- Голяма част от нежелания трафик, както и по-опасните маржови атаки започват от вътрешни за мрежата системи;
- Защитните стени трудно могат да предпазят мрежата от добавяне на неоторизирани точки за безжичен достъп (AP[33]), както и от специализирани системи за събиране на мрежови трафик (анализатори на протоколи);
- Защитната стена не може да замени периодичното създаване на резервни копия и на плана за действие в критични случаи;
- Защитната стена не може да предпази мрежата от човешки грешки.

Някои от препоръките при включването на защитни стени към фирмената политика за сигурност са:

- Защитната стена трябва да се постави в тази част на мрежовата топология, където ще има на-съществен принос към сигурността и/или към нейното повишаване;
- Защитните стени са основни мрежови устройства за подсигуряване, но не са единствените, които са необходими за цялостна защита;
- По подразбиране целия трафик е забранен, което изисква администраторите да конфигурират разрешения (това твърдение се отнася за затворения модел на сигурност);
- Задължително е да се подсигурят както отдалечения достъп за конфигуриране и наблюдение на защитната стена, така и физическия достъп на неоторизирани лица до устройството;
- Задължително е периодично наблюдение на работата на защитните стени;
- Не трябва да се забравя, че технологията на защитните стени е създадена да предпазва вътрешните сегменти от външни атаки.

[33] Access Point

Хардуерни и софтуерни решения

Ако се разгледат мрежови защитни стени се вижда, че на пазара към момента се предлагат следните основни типове:

1. Специализиран хардуер, използващ специално разработен софтуер;
2. Софтуерни пакети, които могат да бъдат инсталирани на стандартен хардуер;
3. Облачно-базирани решения.

И трите технологии имат своите предимства и недостатъци, като правилният избор зависи от заложените изисквания във фирмената политика за сигурност, знанията на администраторите и внимателното проучване на възможностите на избраните продукти.

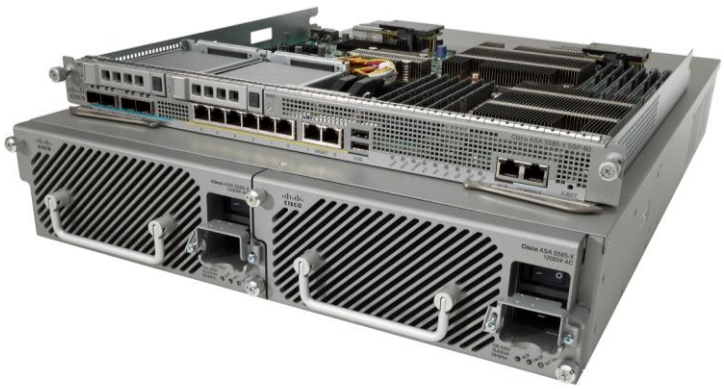
Въпреки, че хардуерните защитни стени имат висока производителност и използват специално оптимизиран софтуер както за операционна система, така и за функциите за защита, много често е по-удачно да се инсталира софтуерна защитна стена, поради по-ниските разходи, както и поради някои от специфичните услуги, които те поддържат. Налични са редица Linux базирани защитни стени, които имат безплатен и свободно достъпен вариант, който е с ограничена или пълна функционалност. Този тип софтуерни решения първоначално изискват задълбочени познания, свързани с Linux, но в момента повечето от тях поддържат и лесни и интуитивни потребителски графични интерфейси, които правят сложното конфигуриране през скриптове и файлове сравнително лесно и бързо.

Хардуерни защитни стени

От гледна точка на производителност хардуерните защитни стени предоставят най-високо бързодействие. Една от водещите серии хардуерни устройства от този тип е Adaptive Security Appliance, серия X, на Cisco Systems®. Едно устройство от серия 5585-X може да поддържа:

- 40 Gbps филтриран трафик (80 Gbps при два модула в общо шаси и до 640 Gbps при клъстер с 16 модула);
- 350000 връзки (connections) в секунда;
- 10000000 едновременни сесии;
- 250 отделни политики;
- 1024 VLAN и др.

Въпреки изключително високата производителност и големия брой поддържани технологии конфигурирането на ASA е сравнително лесно благодарение на Adaptive Security Device Manager – специализиран инструмент с графичен интерфейс.



Фиг. 2.12 ASA 55xx на Cisco Systems® (източник Интернет)

Checkpoint е световно известна компания, която предлага на пазара редица решения, свързани с технологията на защитните стени. Checkpoint X80-S е хардуерна защитна стена, която има следните основни параметри:

- 16 Gbps филтриран трафик, с максимална стойност от 140 Gbps за шаси;
- До 100000000 едновременни сесии;
- До 600000 инспектиране сесии по метода “stateful”;
- Максимален брой интерфейси – 64.

Фиг. 2. 13 Checkpoint® X80-S (източник Интернет)

Компанията Endian предлага специализирани хардуерни защитни стени, които обхващат пазарния сегмент от малки и домашни офиси до средни и големи фирми. Също така са налични и специални устройства за анализи, филтриране и подсигуряване на трафика в индустриални мрежи и за облачни услуги.

Macro R е най-мощната система, предлагана от Endian, която има следните базови параметри:

- Оптични или Ethernet интерфейси със скорост до 10 Gbps;
- Многоядрен процесор;
- UTM технология, обхващаща “stateful” защитна стена, IPS, Web филтър, антивирусно сканиране, VPN, QoS, Hotspot и др.;
- 10 Gbps филтриран трафик;
- 5000000 паралелни сесии;
- До 800000 едновременни сесии;
- До 2500 потребителя.

Фиг. 2.14 Endian Macro R (източник Интернет)

Изборът на хардуерна защитна стена е важна задача, която може да има дълготраен ефект както върху мрежовата сигурност, така и върху бизнес процесите на фирмата.

Всеки производител изтъква предимствата на своя продукт, което прави подбора на устройство още по-труден. Icsalabs[34] провеждат независими тестове на различни технологии, свързани с мрежовата защита, VPN, антивирусното сканиране и други. Списък с всички тествани от тях защитни стени, които успешно са преминали тестовете е свободно достъпен на техния сайт.

Софтуерни защитни стени

Софтуерните мрежови защитни стени предоставят възможност да бъдат инсталирани на стандартен хардуер, което намалява тяхната цена, но и в определени случай може да редуцира производителността. Най-често този тип софтуер използва Linux или BSD дистрибуция като базова операционна система (с възможни модификации, включително и на ядрото), а различните производители добавят своята софтуерна защитна стена, както и специален графичен потребителски интерфейс за по-лесно и удобно конфигуриране и наблюдение. Този подход има редица предимства като интуитивно инсталиране и бързо пускане в експлоатация, лесно задаване на параметрите и поддръжка, ниски хардуерни изисквания и др. Основния недостатък е, че ако в операционната система са налични технологични пропуски злонамерени атаки могат да се възползват от тях и да заобиколят функциите на защитната стена.

Към момента повечето софтуерни защитни стени поддържат UTM модела, което дава широки възможности за анализ и подсигуряване на мрежовия трафик. Съществуват решения както със затворен, така и с отворен код.

Някои от популярните софтуерни защитни стени със затворен код са:

- Microsoft Forefront Unified Access Gateway;
- Checkpoint Firewall;
- Sophos UTM;
- Kerio Control и др.

Често използвани софтуерни защитни стени с изцяло отворен или налична версия със свободен код са:

- Endian Firewall;
- Untangle;
- M0n0wall;
- pfSense и др.

Checkpoint Firewall Software Blade е софтуерна архитектура, която позволява компаниите да изградят надеждна защита срещу мрежовите заплахи. Всички софтуерни пакети и функции се управляват централизирано, което значително редуцира сложното конфигуриране, администриране и поддръжка. Предоставените защитни функции могат лесно да бъдат разширени, без да е необходимо да се прави нова инвестиция в хардуер. Налични са предварително дефинирани пакети със защитна функционалност:

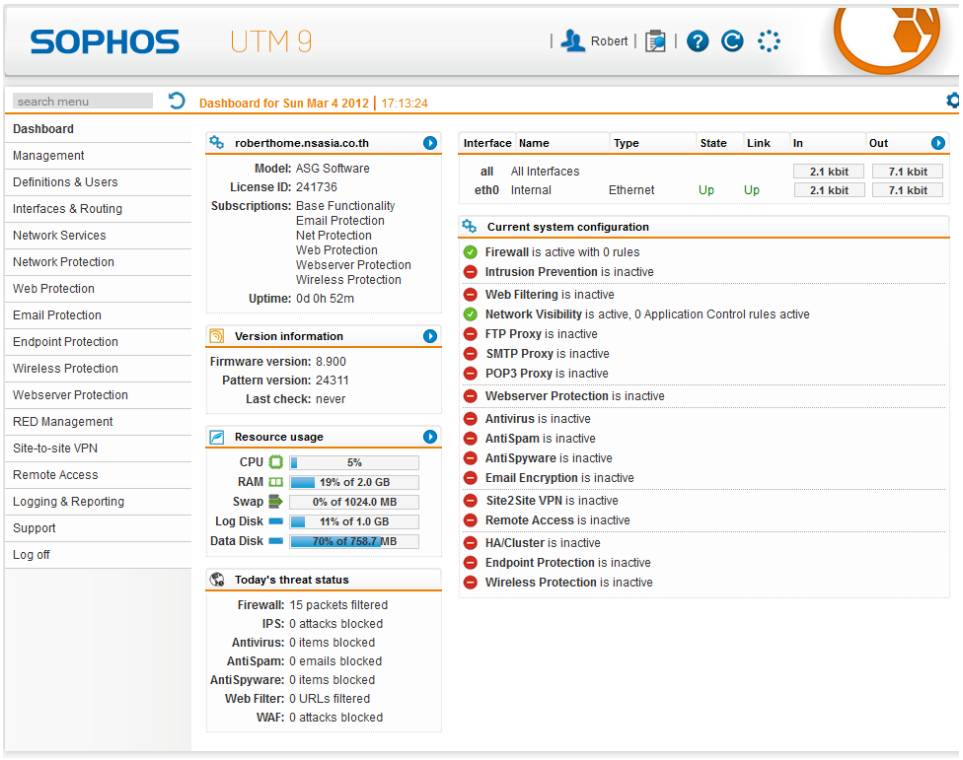
- “Next Generation Firewall” – защитна стена с разширени възможности, IPS, както и контрол и защита на приложните протоколи. Филтрирания трафик е до 110 Gbps с възможности за наблюдение и подробни доклади;

[34] www.icsalabs.com

- “Next Generation Threat Prevention” – защитава фирмената комуникация от зловреден код и филтрира достъпа до WEB страници и услуги. В този пакет са включени антивирусно сканиране, “anti-bot” модул, контрол и филтриране на приложните протоколи, IPS, URL филтър и модул за автентификация;
- “Next Generation Secure Web Gateway” – този пакет е оптимизиран за WEB филтриране и съдържа контрол и подсигуряване на приложните протоколи, антивирусно сканиране, URL филтър и опционални IPS и “anti-bot” функции;
- “Next Generation Data Protection” – този пакет е оптимизиран за DLP (Data Loss Prevention) и съдържа DLP модул, контрол и подсигуряване на приложните протоколи, автентификация и разширен мониторинг.

Sophos Unified Threat management е специализирана софтуерна защитна стена (базирана на Linux), изградена на модулен принцип, като отделните модули могат да се лицензират при необходимост. Sophos UTM поддържа:

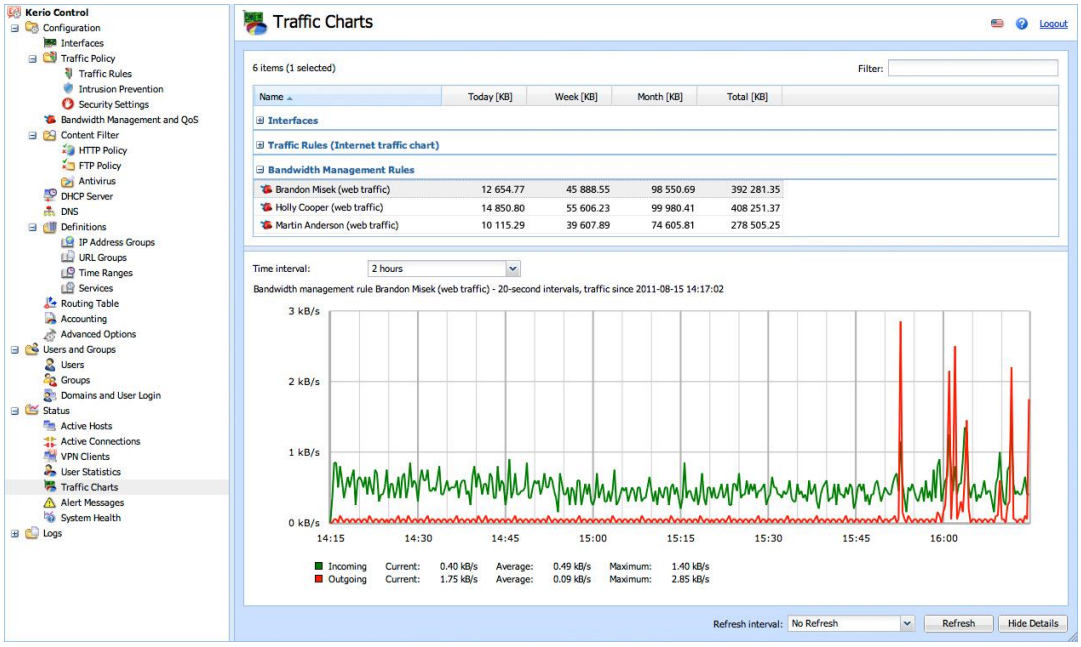
- Мрежова защитна стена;
- IPsec и SLL VPN;
- Защита от DoS и DDoS атаки;
- Защита на електронна поща чрез DLP, spam филтриране и карантина;
- Защита на безжични мрежи;
- Web защита – напълно прозрачни потребителски правила, филтриране на приложения и приложни протоколи, модул за автоматична проверка на направената конфигурация и др.;
- Специализирана защита на WEB сървъри чрез “reverse proxy” и специални алгоритми за защита на банкови транзакции;
- Защита на крайните устройства.



Фиг. 2.15 Графичен потребителски интерфейс на Sophos UTM (източник Интернет)

Kerio Control е софтуерна защитна стена, работеща на база на UTM модела, като някои от по-важните функции са:

- Дълбоко инспектиране на пакетите;
- Анализ на сесиите;
- Поддръжка на NAT;
- Поддръжка на IEEE 802.1q;
- Подробни правила за филтриране на трафика;
- Повече от един IP адрес на интерфейс (multi-home);
- Reverse-proxy;
- IPv6 RA;
- VPN – IPsec, SSL, L2TP over IPsec и др.;
- Подробни доклади и разширено наблюдение;
- Поддръжка на SNMP;
- Балансиране на трафика и QoS;
- URL филтриране;
- Proxy сървър;
- Политики за анализ на Web съдържанието;
- Автентификация – Active Directory, Open Directory, Kerberos, NTLM, Web login и др.;
- IPS система и много др.



на

фиг. 2.16 Графичен потребителски интерфейс на Kerio Control (източник Интернет)

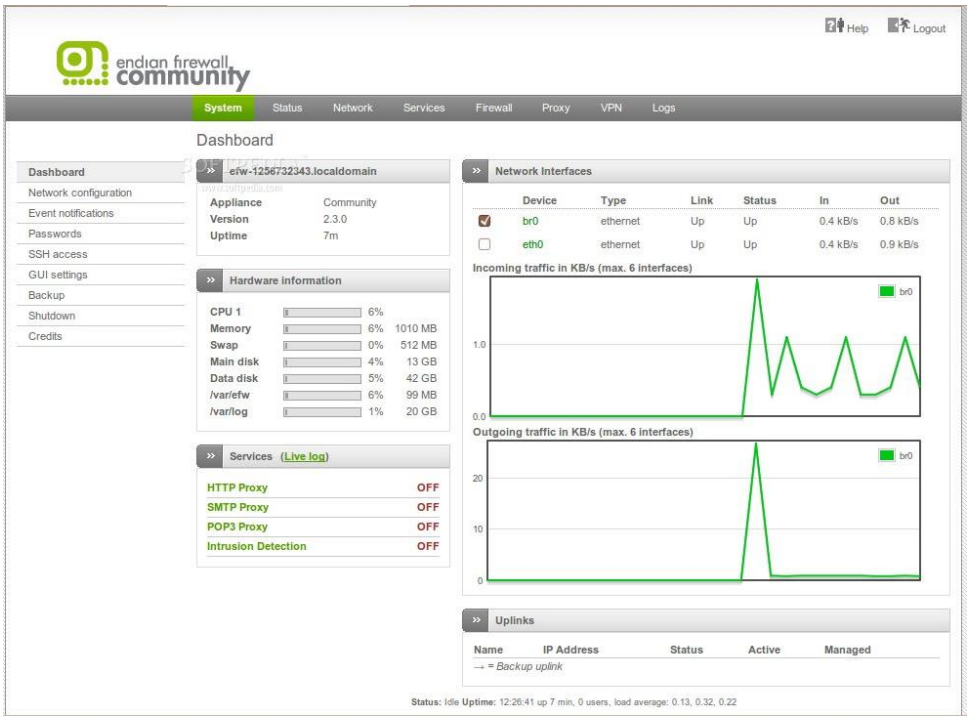
Една от целите на компанията Endian е „*превърнете всеки PC в UTM*”. Въпреки, че звучи амбициозно софтуерният продукт Endian Firewall доказва, че това е напълно възможно и дори компютър със сравнително слаби ресурси може да бъде надежден UTM за малка мрежа. Наред с предлаганите от Endian хардуерни защитни стени тяхната UTM платформа е достъпна и като софтуер с отворен код – Endian Firewall Community Edition (EFW CE).

По-важните характеристики на EFW са:

- „stateful” инспекция на трафика;

- Защита на електронната поща чрез spam филтри, black-list списъци, антивирусно сканиране и др.;
- IPS базиран на snort[35];
- Hotspot (само при платена версия);
- QoS;
- Хардуерно дублиране (High Availability);
- WEB защита с прозрачно прокси, URL филтри, антивирусно сканиране и др,;
- SSL и IPsec VPN;
- Базиран на ClamAV антивирусен модул;
- “multi WAN“ поддръжка;
- Централизирано наблюдение на множество EFW системи (само при платена версия);
- Подробни доклади и съобщения в реално време;
- Достъп до специализирана системна конзола, която предоставя и функции, свързани с наблюдението на използваната операционна система (Linux).

EFW се конфигурира и администрира с изключително лесен и интуитивен графичен интерфейс, който е изцяло WEB базиран.



фиг. 2.17 Графичен потребителски интерфейс на EFW CE версия 3.0

Подобно на Endian компанията Untangle разработва както софтуерни, така и хардуерни защитни стени от последно поколение, като тяхната софтуерна платформа има и безплатна свободно достъпна версия.

По-важните функции на Untangle NG Firewall Free са:

- Web филтър;
- Анализ и филтриране на приложни протоколи (с ограничена функционалност, в сравнение с платените версии);

[35] www.snort.org

- Антивирусно сканиране;
- Проверка за “phishing” на сайтове;
- IPS;
- “stateful” защитна стена;
- Spam филтър;
- “Captivate portal”, блокиране на Интернет реклами и др.

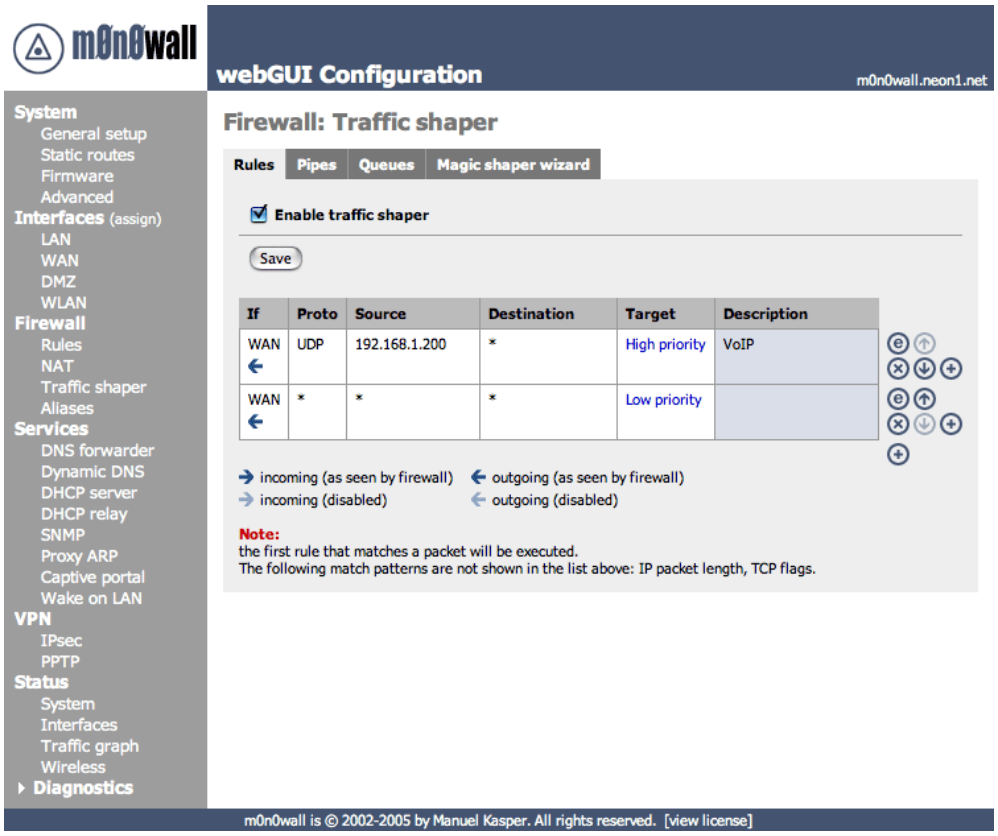
Използваният графичен интерфейс е интуитивен и лесен за употреба.

фиг. 2.18 Графичен потребителски интерфейс на Untangle NG Firewall (източник Интернет)

Endian, Sophos и други софтуерни защитни стени използват като база Linux. За разлика от тях проектът m0n0wall е базиран на FreeBSD. Някои от по-важните функции на m0n0wall са:

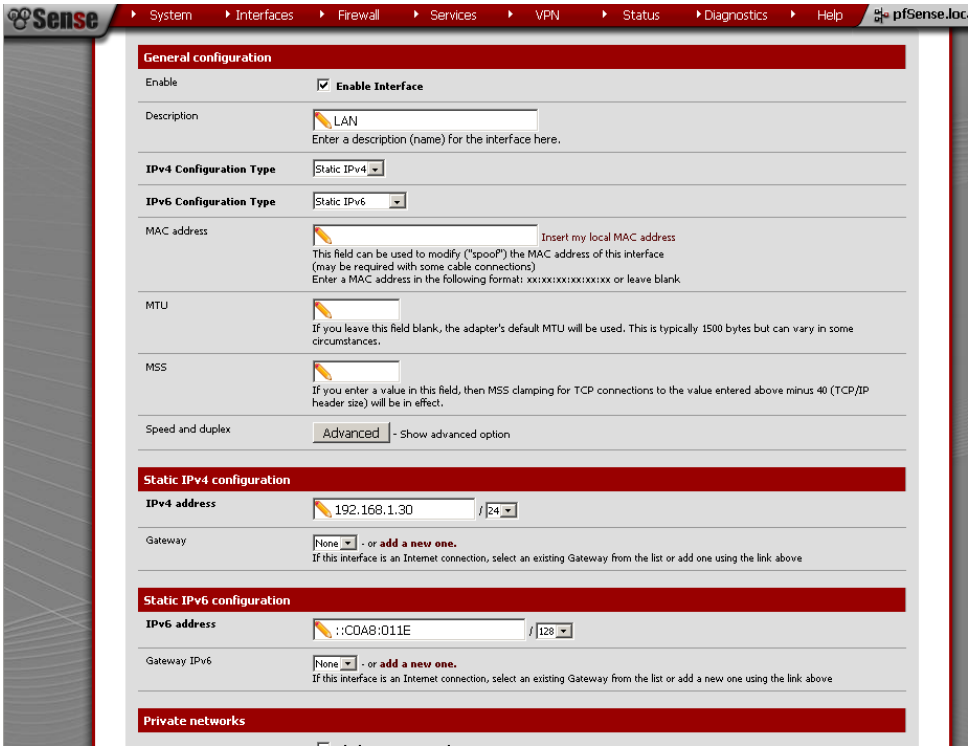
- Конфигуриране и администриране с графичен интерфейс и конзола през серийна комуникация;
- Поддръжка на интерфейси от тип IEEE 802.11 и работа на системата в режим на точка за достъп (AP);
- IPv6 съвместимост;
- Поддръжка на стандарт IEEE 802.1q;
- NAT;
- IPsec VPN тунели;
- DyDNS клиент и обновяване на записите по RFC 2136;
- Интегриран SNMP агент и др.

Предимство на m0n0wall е, че може да работи и на вградени системи с ограничени ресурси.



фиг. 2.19 Графичен потребителски интерфейс на m0n0wall (източник Интернет)

pfSense е друга популярна защитна стена, която използва FreeBSD, но с модифицирано ядро и редица допълнителни софтуерни пакети. Конфигурирането на системата е през WEB базиран графичен интерфейс.

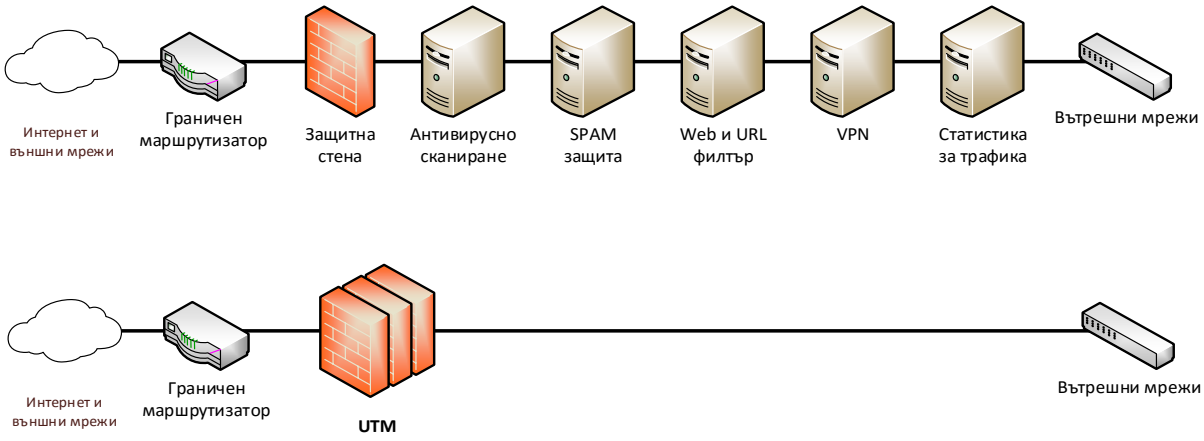


фиг. 2.20 Графичен потребителски интерфейс на pfSense (източник Интернет)

Технология UTM

UTM технологията се явява логичното надграждане на функционалността на защитните стени с нови модули за защита, като мрежова IPS, антивирусно сканиране на трафик и електронна поща, VPN, WEB филтриране по URL и съдържание, QoS и балансиране на трафика, защита от изтичане на конфиденциална информация и др. Терминът UTM е представен в маркетингово проучване от International Data Corporation (IDC).

Още от своята поява през 2004 този тип системи намират изключително широко приложение поради редуцирането на разходите и значително по-лесното интегриране, конфигуриране и администриране. От гледна точка на системна поддръжка едно от най-големите предимства на UTM е, че администраторите трябва да се грижат за едно устройство с разширена функционалност, а не за няколко отделни системи.



фиг. 2.21 Обобщена UTM функционалност

Друга причина за възникването на UTM е развитието на методите и инструментите за мрежови атаки. За да се предпази фирмена мрежа от сложни атаки, които използват няколко различни подхода се налага интегриране на редица защитни устройства, като най-слабото от тях определя нивото на цялостната сигурност (принципа на най-слабото звено). Също така използването на SAN[36] и центрове за съхранение на данни с отдалечен достъп изисква да се подсигури както достъпа до информацията, така и обратния поток – към кого се изпращат данните.

Интегрирането на няколко типа устройства като функционалност в едно гарантира високата степен на сигурност, гъвкавост, лесно конфигуриране и поддръжка.

Някои от по-важните предимства на UTM са:

- Намаляване на сложността на интегриране, конфигуриране и поддръжка;
- Премахване на необходимостта от няколко защитни системи и заменянето им с една;
- Намаляване и оптимизиране на разходите;
- Лесно управление, наблюдение и поддръжка, най-често чрез WEB интерфейс;
- Съвместимост с почти всички актуални мрежови протоколи и технологии;
- Съвместимост със законови мерки и изисквания в различни държави и др.

[36] Storage Area Networks – специализирани мрежи, използвани в центрове за данни при съхранение на информация

Като недостатъци на UTM могат да се посочат:

- UTM устройствата се явяват критични (т.нар. “single point of failure”) и често се налага да бъдат дублирани;
- Ако защитата на UTM устройството бъде преодоляна от злонамерени лица, в общия случай няма други защитни системи, които да блокират атаките;
- Потенциално забавяне на скоростта на обмен на данни, ако не е избран правилен модел с достатъчно системни ресурси и/или е направена грешна конфигурация.

Препоръки

При проектиране на интегрирането на защитни стени и в мрежови топологии, както и при тяхното конфигуриране е желателно винаги да се взима под внимание:

1. Защитната стена сама по себе си не е достатъчна за надеждно подсигуряване на мрежовия трафик, което води до необходимост да се интегрират и други технологии като IDS/IPS, URL филтриране, антивирусно сканиране и др.;
2. Мрежовите защитни стени не могат да предпазят крайните устройства, ако атаката преминава през криптиран VPN тунел, който завършва на крайното устройство;
3. UTM технологията интегрира няколко защитни техники в едно устройство, но то може да се яви критично в дадената мрежова топология;
4. Без значение дали се използва защитна стена или UTM ако устройството е критично е необходимо то да бъде дублирано;
5. Най-често защитните стени блокират целия трафик и администраторите разрешават само и единствено дефинираните във фирмената политика за защита приложения и протоколи;
6. При сложни мрежови топологии е целесъобразно да се използват защитни стени, които работят на принципа на зони и анализират трафика до приложното ниво на OSI референтния модел;
7. Задължително е да се извършва периодично наблюдение на работата на защитните стени и UTM системите;
8. При използване на IDP/IPS е необходимо сигнатурите винаги да бъдат периодично обновявани (през защитена комуникация) и внимателно да се подберат най-важните от тях, в зависимост от конкретната мрежова топология и изискванията на трафика;
9. UTM може да доведе до значително забавяне на мрежовия трафик;
10. Изборът на защитна стена и UTM е важна задача, която изисква предварителен анализ и проучване;
11. Препоръчително е да се инсталират персонални защитни стени дори при интегрирани мрежови или UTM системи.

Заключение

Защитните стени са изключително важни устройства, свързани с подсигуряването на мрежовия трафик и на крайните устройства. Тяхната технология непрекъснато се развива – започвайки от пакетно филтриране, през “stateful” анализ, добавяне на проверка на приложните протоколи до най-актуалната за момента UTM.

Необходимо е администраторите да са запознати с всички възможности на използваните от тях защитни системи за да могат да направят максимално точна конфигурация, която ще има оптимално ниво на защита и минимално негативно влияние върху мрежовия трафик.

Източници

1. www.cisco.com
2. www.endian.com
3. www.m0n0.ch
4. www.pfsense.org
5. www.untangle.com
6. www.sophos.com
7. www.zonealarm.com
8. www.comodo.com
9. tinywall.pados.hu
10. www.emsisoft.com
11. www.agnitum.com
12. www.checkpoint.com
13. www.fs-security.com
14. www.simonzone.com/software/guarddog

Глава 3. Основи на Linux

Забележка: Целта на тази глава не е да представи на курсистите операционната система Linux в дълбочина, а да им предостави необходимите знания за успешна работа и поддръжка на EFW CE. Препоръчваме да се запознаете подробно с възможностите на Linux, поради изключително широкото приложение на тази операционна система при изграждане на сървърни приложение и при мрежовата комуникация.

Вместо въведение – за катедралата и базарът

Фиг. 3.1 Катедрала и базар (източник Интернет)

На 27 май 1999 година на конференцията “Linux Kongress”, Ерик Реймънд представя своето есе, наречено „Катедралата и базарът“, в което той описва методите на софтуерното инженерство и опита на Реймънд със софтуера fetchmail.

„Катедралата и базарът“ описва и дефинира разликите между двата модела на разработка на свободен софтуер:

- **„Катедрален модел“** – изходният код е достъпен при всяко издание на софтуера, но кодът, който се разработва между отделните издания е с ограничен достъп и е наличен само за определена група разработчици. Като пример за този модел са посочени изключително популярния редактор GNU Emacs[37] и компилатора GCC[38];
- **„Базарен модел“** – кодът на приложението е публичен и разработката се извършва от множество програмисти, които най-често комуникират помежду си през Интернет. Като откривател на този метод Реймънд посочва Линус Торвалдс (водещия разработчик на ядрото на операционната система Linux).

В есето се описва и идеята, че „Ако има достатъчно очи, всички грешки се виждат“[39] което още е наречено Закон на Линус. Погледнато под друг ъгъл твърдението означава, че ако

[37] www.gnu.org/software/emacs
[38] gcc.gnu.org
[39] „Given enough eyeballs, all bugs are shallow“

програмния код е свободно достъпен и с него работят голям брой разработчици откриването и отстраняването на грешки ще е много по-бързо. При катедралният модел откриването на бъгове и тяхното коригиране изисква много повече време и усилия, особено, ако екипът от програмисти е малък.

Влиянието, което есето оказва е съществено за някой от разработваните към момента софтуерни проекти. GNU Emacs и GCC преминават към „базарен модел“. Един от най-важните резултати е, че компанията Netscape Communications Corporation оповестява изходния код на Netscape Communicator, което поставя началото на проекта Mozilla.

Пълният текст на есето “Катедралата и базарът“ е свободно достъпно в Интернет и преведено на български език на адрес: http://catb-bg.sourceforge.net/index.html

Терминът „отворен код“ дефинира софтуер, чийто изходен код е публично достъпен за преглед, редактиране и в определени случай ново приложение (включително след модификации). Много често отворен код се използва като синоним на свободен софтуер, но това е спорно, защото има редица свободно разпространявани софтуерни продукти, чийто изходен код е затворен (не е публичен).

Open Source Initiative[40] (OSI – съкращението съвпада с Open System Interconnection) определя 10 правила, които разграничават дали продукт е с отворен код или не:

1. Свободно разпространение - лицензът, под който се разпространява програмата, не трябва по никакъв начин да забранява продажбата или свободното ѝ даване като компонент от друг софтуер, съдържащ множество отделни компоненти. Този лиценз не трябва да изисква такси за подобни продажби;
2. Изходен (сорс) код - програмата трябва да съдържа изходния код и трябва да позволява разпространението му, включително и в компилирана форма (ако има такава възможност). Ако под някаква форма продуктът не се разпространява заедно с кода му, то трябва да има инструкции откъде може да се изтегли (от Интернет) без заплащане. Изходният кодът трябва да бъде в такава форма, че всеки да може да го променя съобразно своите нужди. Доставянето на маскиран (обфускиран) код или на шифриран код не се допуска;
3. Допълнителни изисквания - лицензът трябва да позволява промени на кода и дописването му, а също трябва да разрешава те да бъдат разпространявани под същия лиценз, под който е публикуван и оригиналния софтуер;
4. Цялостност на авторския код – лицензът може да забранява разпространението на изходния код в модифициран вид само ако се позволява добавянето на специални “patch” файлове заедно със сорс кода, с цел модифициране на програмата по време на изпълнението ѝ (компилирането). Лицензът също трябва да позволява разпространението на софтуера, създаден по описания начин. Лицензът може да изисква версия с допълнително създадения код да носи различно име или номер от тази на оригиналния продукт;
5. Без дискриминация на лица или групи - лицензът не трябва да дискриминира хора или групи от хора;
6. Без дискриминация на области на прилагане - лицензът не трябва да ограничава използването на програмата в различни области;

[40] www.opensource.org

7. Разпространение на лицензa - правата, зададени за всяка програма, трябва да са задължителни за всеки, който я ползва, без нужда от допълнителни лицензи;
8. Лицензът не трябва да бъде за определен продукт - правата, които носи лицензът, не трябва да зависят от това, че програмата принадлежи към дистрибуция (или пакет). Ако тя се изключи от дистрибуцията или се разпространява отделно, всички нейни части зависят от условията на лицензa на оригиналната дистрибуция;
9. Лицензът не трябва да ограничава друг софтуер - лицензът не трябва да налага ограничения на друг софтуер, който се разпространява заедно с лицензирания софтуер;
10. Лицензът трябва да бъде технологично независим - нито една клауза на лицензa не трябва да се отнася към индивидуална технология или определен интерфейс.

Linux е общото название, което се дава на всички операционни системи, основаващи се на ядрото „Линукс“ и системните инструменти и библиотеки от проекта GNU. Различните варианти на този тип операционни системи се наричат Linux дистрибуции, като те се различават по това с какъв друг софтуерни пакети са окомплектовани.

Като цяло Linux е един от най-известните представители на свободния софтуер.

История на Linux

Проектът и движение GNU (GNU's Not Unix), чиято цел е създаване на нова операционна система свободен софтуер е основан от Ричард Столман през 1984 г. Самата система съдържа голям брой инструменти и програми, като компилатори, текстови редактори и сървърни приложения. Този софтуер се разпространява с лицензa GNU GPL[41], което гарантира бъдещата му свободна форма. През 1991 г. към почти завършената операционна система е добавено ядрото Linux, написано от Линус Торвалдс, с което GNU става напълно работеща операционна система и това спомага за бързото ѝ разпространение. Целта на проекта GNU е да се създаде изцяло нова безплатна операционна система. Ричард Столман иска компютърните потребители да бъдат отново „свободни“, както е било през 1960-те и 1970-те — свободни да изучават изходния код на програмите, които ползват, да имат свободата да споделят софтуера с други хора, свободата да модифицират програмите и в последствие да разпространяват своите версии. Тази философия е публикувана като GNU Manifesto[42] през март 1985.

Развитие

Някои от по-важните моменти в хронологията на развитието на Linux са:

- 1991 г. – ядрото на Linux е публично обявено на 25 август от 21 годишния студент Линус Торвалдс;
- 1992 г. – ядрото Linux преминава под лицензa GNU GPL. Дефиниран е термина Linux дистрибуция;
- 1993 г. – над 100 програмиста активно участват в разработката на ядрото. За първи път е пусната най-старата за сега дистрибуция Slackware, а по-късно през същата година е обявена и дистрибуцията Debian, която в момента има най-много последователи (community);
- 1994 г. – спрямо критериите на Торвалдс ядрото е напълно функциониращо и версия 1.0 е свободно достъпна. Проектът XFree86 разработва графичен потребителски интерфейс, а компаниите Red Hat и SUSE обявяват своите първи дистрибуции;

[41] www.gnu.org/copyleft/gpl.html
[42] www.gnu.org/gnu/manifesto.html

- 1995 г. – Linux е компилиран да работи на хардуерни платформи DEC Alpha и Sun SPARC, като в следващите години поддържаните хардуерни устройства непрекъснато се разширяват;
- 1996 г. – обявена е версия 2.0 на ядрото, която включва и мултипроцерона работа;
- 1998 г. – много от големите ИТ компании обявяват своята поддръжка на Linux, като сред тях са IBM, Oracle, Compaq и др. Стартира проекта KDE;
- 1999 г. – стартира проекта GNOME, който се явява свободен заместител на KDE (KDE силно зависи от библиотеките Qt toolkit);
- 2000 г. – от Dell обявяват, че са №2 доставчик на Linux базирани системи в световен мащаб;
- 2004 г. – екипът на XFree86 се разделя и част от него се приобщава към проекта X Org Foundation, което води до по-бърза и успешна разработка на сървъра X;
- 2005 г. – създава се проекта openSUSE, а OpenOffice обявяват версия 2.0 на своя продукт;
- 2006 г. – Oracle пускат своя версия на Red Hat;
- 2007 г. – Dell започва да предлага свои продукти с предварително инсталираната дистрибуция Ubuntu;
- 2011 г. – Ядрото на Linux достига версия 3.0;
- 2012 г. – Linux делът на сървърните приложения надхвърля този на Unix;
- 2013 г. – Google Android (базиран на Linux) достига 75% пазарен дял при смартфоните;
- 2014 г. – Ubuntu се използва от над 22 милиона потребителя.

Приложение

От своето създаване Linux намира широко приложение в редица области сред които:

- Сървърни приложения;
- Облачни услуги;
- Комуникационни услуги;
- Работни станции;
- Специализирани научни приложения и обучение;
- Смартфони;
- Вградени системи и много др.

Фиг. 3.2 Заглавие в пресата за приложението на Linux в ЦЕРН при LHC

Популярни дистрибуции

Сайтът distrowatch[43] извършва задълбочен анализ и определя най-популярните към момента Linux дистрибуции, а статистиката към месец септември 2014 година е следната:

1. Mint – създадена през 2006 г. тази дистрибуция е базирана на Ubuntu. Някои от предимствата са лесната процедура за инсталиране, големия брой специално разработени инструменти за Mint, както и подобренията в интерфейса и функционалността, добавените библиотеки за кодиране и декодиране на аудио и видео (Codecs) и др. Недостатъците са, че не винаги са включени най-новите версии на някои от софтуерните пакети, както и липсата на съвети за повишаване на сигурността. Като цяло тази дистрибуция е насочена към работни станции и ежедневна употреба от потребители, включително с малко опит с Linux;
2. Ubuntu – през 2004 г. обявената нова дистрибуция, наречена Ubuntu поставя своеобразен рекорд, като списъците със желаещи да разработват тази версия нараства изключително бързо. Една от причините е, че Ubuntu е идея на Марк Шътелуорт – милиардер от ЮАР, който е участвал в разработката на Debian. Ubuntu стартира, като вече известните грешки на някои проекти са взети под внимание и са успешно избегнати. Изпращането на безплатни CD до потребители в цял свят спомага за по-широкото разпространение на тази дистрибуция. Като основни предимства могат да се посочат предварително известния цикъл на обновяване на версиите и отпадането на поддръжката, LTS (Long Term Support) – поддръжка в срок от 5 години, подробна документация и удобен графичен интерфейс. Основните недостатъци са несъвместимост с Debian, чести генерални промени, а Unity графичния интерфейс е със спорна степен на одобрение и др.
3. Debian – една от най-старите дистрибуции, която е създадена през 1993 година. Има три основни направления при разработката – “unstable” която съдържа нови пакети с потенциални налични програмни неточности, “testing” която се използва за одобрение от потребителите и “stable” която е с най-надеждна и сигурна работа. За съжаление цикълът на разработка е по-продължителен от този при други дистрибуции. Предимствата на Debian са неговата изключително висока степен на стабилност, стриктния контрол на качеството, наличните над 20000 пакета, както и поддръжката на най-голям брой различни компютърни архитектури от всички Linux дистрибуции. Като недостатъци могат да се посочат сериозната консервативност на проекта, дългия цикъл на разработка на нова версия (1 до 3 години), како и сравнително трудната дискусия между потребителите с блогове и списъци с електронна поща (mailing list);
4. Mageia – дистрибуция, насочена към работни станции, която е лесна за използване от потребители с малко опит с Linux и лесна инсталация. Като недостатък се посочва, сравнително ниската популярност (която в последните месеци расте), както и някои съмнения, че екипът ще успее да поддържа успешно проект в дългосрочен аспект;
5. Fedora – тази дистрибуция е под контрола на Red Hat и много често се използва като тестова база за нововъведения. Това я прави и една от най-иновативните дистрибуции, които са налични към момента. Предимствата на Fedora са високата степен на защита, иновативните технологии и нови пакети, стриктно спазване на идеята за свободен софтуер и др. Недостатъци са насочеността към корпоративно приложение и интегрирането на някои от най-новите графични потребителски интерфейси често се явява спънка пред потребителите;

[43] www.distrowatch.com

6. openSUSE – през 1992 година група немски Linux ентусиасти стартират проекта SuSe (Software und System Entwicklung). Novell закупуват SuSe през 2003 година и променят модела на лицензиране. Напълно безплатната версия openSuSe е налична от 2005 година. Предимства на тази дистрибуция са мощни инструменти за конфигуриране, голям брой налични пакети и отличен сайт с информация. Като недостатъци се посочват някой от инструментите, които са усложнени от гледна точка на начин на използване.
7. Arch – при разработката на Arch е заложен метода KISS (Keep It Simple, Stupid), чиято цел е да се спазва правилото за максимално опростен дизайн и разработка. Това води до по-качествен код и продукт, което е едно от предимствата на тази дистрибуция. Като основен недостатък може да се каже, че някои от приложенията понякога са нестабилни;
8. Elementary – изцяло безплатна дистрибуция за работни станции, която предлага един от най-изчистените и стилни графични потребителски интерфейси;
9. Zorin – базирана на Ubuntu, Zorin е предназначена за начинаещи потребители. Като предимство може да се посочи специално разработените от екипа приложения, които подобряват функционалността и улесняват потребителите;
10. CentOS – създаденият през 2003 година проект CentOS цели да се разработи операционна система, която като база използва изходния код на Red Hat. Към момента това е една от най-предпочитаните специализирани дистрибуции за сървъри, като основни нови версии се обявяват в цикъл от 2 до 3 години, а версии с обновения на всеки 6 до 9 месеца. Предимствата на CentOS са изключително високата стабилност и надеждност, стриктното тестване на кода и модулите, напълно свободното изтегляне и приложение, както и 5 годишен период от безплатно обновяване. Недостатъците са, че най-новите технологии често се интегрират със забавяне и неспазването на срока за публично обявяване на новата версия;
11. Lubuntu – базирана на Ubuntu дистрибуция, която замества интерфейса Unity с по-лекия и бърз LXDE;
12. Puppy – специализирана дистрибуция за администратори, която цели лесно използване и която има малка големина (приблизително 100 MB). Най-често puppy се използва при отстраняване на проблеми с други Linux или Windows системи;
13. Kali – Специална дистрибуция, която наследява популярния BackTrack. Kali включва голям брой специално подбрани инструменти за анализ на мрежовата сигурност, както и проверка на защитата на отдалечени устройства;
14. LXLE – базирана на Ubuntu, тази дистрибуция предлага на потребителя графичния интерфейс LXLE;
15. Manjaro – базирана на Arch дистрибуция, която е оптимизирана за работни станции.

Изборът на дистрибуция е сравнително сложен, поради големия брой налични варианти, субективните фактори, както и разликите във всяка една от тях. Най-често под внимание се взимат:

- Дали дистрибуцията е платена или напълно безплатна;
- Дали дистрибуцията е оптимизирана за сървър или за работна станция;
- Какъв е начина (мениджъра) за управление на пакетите;
- Какъв е графичния интерфейс (ако се използва такъв);
- Дали да се избере максимална стабилност или най-нови технологии;
- Съвместимост с хардуерни модули;
- Поддръжка, описание, документация и др.

Във своя статия порталът Lifehacker[44] предлагат няколко дистрибуции, които потребителите да опитат и да оценят като функционалност и приложимост:

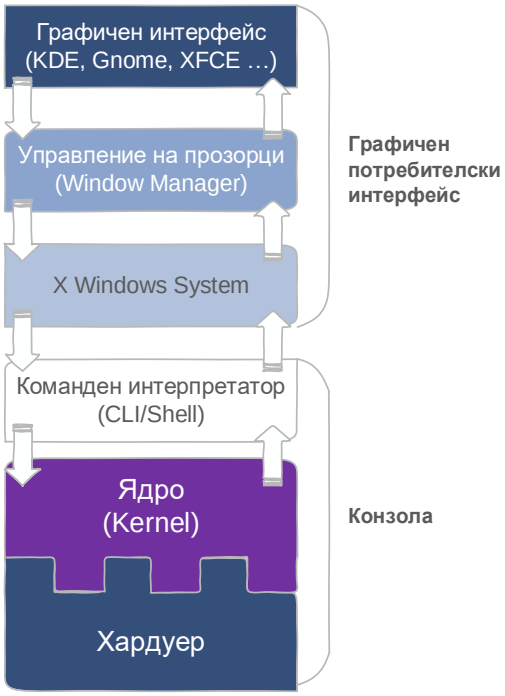
- Ubuntu – функционална и подходяща за ежедневна употреба, както и за сървър;
- Mint – за начинаещи потребители, поради лесната инсталация, конфигуриране и поддръжка;
- Fedora – за тестване на най-новите технологии;
- Debian – за най-предпазливите потребители които търсят максимална стабилност;
- OpenSUSE – за всички, които искат да експериментират с различни настройки;
- Arch – за най-запалените, които искат да инсталират своята операционна система стъпка по стъпка от команден ред и по този начин да получат силно-оптимизиран софтуер.

Гореописаните варианти са само малка част от популярните в момента Linux дистрибуции и е въпрос на личен избор коя да се използва.

Обобщен модел на архитектурата на Linux

Архитектурата на Linux операционната система може да се обобщи по следния начин - хардуерните компоненти на системата взаимодействат с ядрото на Linux, от своя страна ядрото предоставя интерфейси за достъп до неговите функции на командния интерпретатор (Command Line Interface) – т.нар. “shell”, а конзолата на Linux осигурява на потребителя възможност да използва в текстов режим функциите на командния ред и на ядрото.

Много често към Linux дистрибуциите се добавя и “X Window System” в комбинация с мениджър на прозорците (Window Manager). По този начин потребителите могат да използват графичен потребителски интерфейс за достъп до функциите на операционната система. От моделът ясно се вижда, че конзолата и графичните интерфейси предлагат еднаква функционалност, но достъпна по различен начин.



Фиг. 3.3 Обобщен модел на архитектурата на Linux

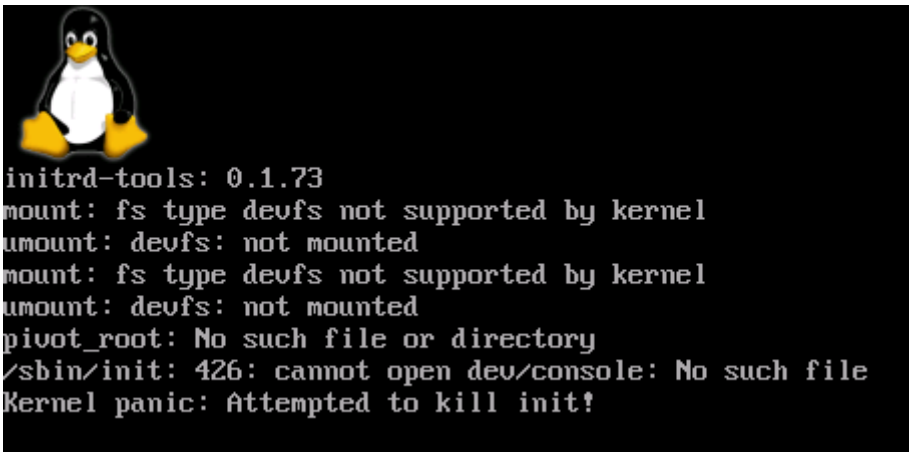
[44] www.lifehacker.com

Като правило може да се посочи, че повечето от административните задачи и най-вече конфигурирането на сървъри се извършва чрез конзолен достъп, докато ежедневната работа на потребителите предполага използване на графичен интерфейс.

Ядро

Ядрото на Linux е монолитно – цялата операционна система работи в адресното му пространство и не се използва режим на супервайзор, като по този начин ядрото предоставя услуги на по-високо ниво през виртуални интерфейси за достъп до хардуерните модули. Най-често графичните интерфейси не работят в пространството на ядрото, за разлика от същите при Microsoft Windows. Също така за разлика от традиционните монолитни ядра Linux предоставя възможност драйверите да бъдат активирани и деактивирани като модули по време на работата на системата.

Въпреки изключително високата степен на надеждност е възможно да възникнат необратими грешки при работата на ядрото – “kernel panic”.



Фиг. 3.4 Пример за грешка (Kernel panic) в работата на Linux ядрото

Потребителски програми

За да се получи напълно функционална от гледна точка на потребителите операционна система около Linux ядрото се добавят редица потребителски програми, които могат да се изпълняват паралелно. При стартиране потребителските програми изпълняват своята функционалност и в общия случай извеждат резултат. При работа в конзола много от приложенията могат да използват един от следните стандартни потоци за данни (Standard Streams):

1. **Standard input (stdin)** – поток от символи, който се изпраща към програмата, като за целта най-често се използва специална програмна инструкция. Важно е да се отбележи, че не винаги се налага към програмата да се изпращат данни. По подразбиране stdin използва въведената информация от клавиатурата;
2. **Standard output (stdout)** – този поток съдържа информацията, която програмата извежда към потребителя. Отново не всички програми могат да използват тази функционалност, като по подразбиране stdout е насочен към текстов терминален прозорец;
3. **Standard error (stderr)** – поток, чрез който програмите извеждат съобщения за възникнали грешки или диагностични данни. Аналогично на stdout отново най-често се използва текстов терминален прозорец.

В дадени случаи определен поток може да се използва като входни данни за друга програма.

Възможно е да се извърши пренасочване на даден стандартен поток, като при командния интерпретатор bash (Bourne Again Shell) за stdout се използват операторите „>“ или „>>“. Разликата в принципа на работа е, че “>” изтрива съдържанието на файла, към който е направено пренасочването, а “>>” запазва неговото съдържание и добавя новите данни.

Пример за пренасочване на stdout:

```
ls > dir_listing.txt
```

Командата ls извежда съдържанието на текущата директория по подразбиране към stdout, а чрез „>“ информацията се записва във файл с име “dir_listing.txt”. Ако файлът не съществува се създава, а ако е бил наличен на мястото на неговото съдържание се записва резултата от ls.

За да се пренасочи stderr при bash се използва “2>”, а пренасочването на stderr и stdout едновременно най-често се извършва по следния начин:

```
ls > directory_listing.txt 2>&1
```

Демони

При Linux е възможно няколко програми да се изпълнявате едновременно (multitasking), като някой от тях могат да работят на заден план (background) и без потребителска намеса - този тип програми се наричат демони (daemon). Типичен пример за демон е sshd – приложение, което стартира SSH сървър и очаква заявка за връзка от отдалечен или локален клиент.

Демоните също използват стандартните потоци, които предоставят възможност за извеждане на съобщения към потребителя, както и за визуализиране на информация за възникнали грешки. За комуникация между няколко демона най-често се разчита на технологията “pipe” – начин на пренасочване на изходните данни на програма към входните параметри на друга.

Много често “pipe” се използва и от потребителите в конзола, като между отделните програми се добавя символа “|”, например:

```
ls | grep "direcotory_listing.txt"
```

В примера командата ls извежда съдържанието на директорията, но нейните изходни данни се предават към grep. От своя страна grep извежда само редовете, които съдържат текста “directory_listing.txt”.

При по-сложни команди е възможно е да се използва последователност от няколко “pipe”.

Стартиране на програми в терминал

За различните дистрибуции командния ред в конзолата може да изглежда по различен начин, но най-често се използва потребителското име, последвани от символа “@” и името на хоста, текущата директория и символа “$”:

```
user@ badkict ~/$
```

В примерът текущата директория е посочена от “~”, което съвпада с /home/user.

Някой полезни команди, които често се използват при работа с конзолата са:

- ls – извежда съдържанието на текущата или на директорията, която е посочена като аргумент;
- pwd – показва пътя към текущата директория;
- cd – сменя текущата директория;
- rm – изтрива един или няколко файла;
- rmdir – изтрива празна директория;
- mkdir – създава нова директория;
- ps – извежда информация за работещите към момента процеси;
- cp – копира файл или файлове;
- mv – премества един или няколко файла, същата команда се използва и за преименуване на файл;
- grep – използва се за търсене във файл и пренасочване на stdout към друга програма;
- find – търси файл или файлове по файловите системи. 100% точност, но е със сравнително бавно действие;
- locate – търси файл или файлове в специален кеш, базиран на съдържанието на файловите системи. Бързо действие, но не се гарантира 100% точност;
- man – извежда документацията на посочената програма (ако е налична);
- clear – изчиства конзолата;
- less – показва съдържанието на файл;
- nano – универсален текстов редактор, с богати възможности и лесен за употреба от потребители с малко опит.

Повечето команди изискват да се използват един или няколко аргумента, като информация за отделните параметри може да се види чрез man. При Linux аргументите са или отделен символ (има разлика между малки и големи букви), който се посочва чрез “-“ или цяла дума, предхождана от “--“. Например за извеждане само на директориите чрез ls може да се използва:

```
ls –d
```

или

```
ls --directory
```

За да се стартира програма в терминал на Linux е необходимо даденият файл да има права, включително за изпълнение (execute), които да са налични за текущия потребител.

Стартиране на потребителски програми

При Linux командите са или вградени в “shell” или се стартират от изпълним файл. Например ls е команда, която е част от “shell” и нейното местоположение винаги е известно. От друга страна потребителските програми са или компилиран изходен код до изпълним файл или скриптове. Винаги когато в “shell“ се въведе текст той се третира като команда и се прави опит за нейното стартиране, като ако бъде открита тя се изпълнява, а в противен случай се извежда съобщение за грешка към stderr.

В общия случай потребителите работят или в тяхната основна директория (home) или в някоя нейна поддиректория. Една от важните системни променливи е PATH. Тя позволява “shell” да използва въведените директории с цел откриване на командата, програмата или скрипта. Най-често home директорията не е въведена в PATH и при стартиране на програма (дори от

текущата директория) се извежда грешка. За да се избегне този проблем се използва комбинацията от символи “./”, Последвани от името на програмата:

```
user@debian:~/backup$ backupFTP

backupFTP: command not found

user@debian:~/backup$ ./backupFTP

Starting full data backup to FTP

...
```

За да се избегне използването на “./” потребителя може да модифицира променливата PATH или да използва символни връзки към програми, инсталирани в нестандартни директории.

Списък с изпълняваните процеси

Аналогично на другите операционни системи и при Linux стартираните програми се изпълняват в контекста на процеси. Въпреки, че ядрото се грижи за управлението на процесите много често се налага администраторите да могат да наблюдават и при необходимост да спират някои от тях.

Най-лесният начин да се видят стартираните процеси е чрез командата top., която извежда и обобщена статистика, последвана от списък с отделните процеси.

```
top - 11:35:09 up 2 min,  2 users,  load average: 0.34, 0.19, 0.08
Tasks: 112 total,   1 running, 111 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1034588 total,   295276 used,   739312 free,    25724 buffers
KiB Swap:  1748988 total,        0 used,  1748988 free,   162304 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 2239 root      20   0 53508  17m 5420 S   0.3  1.8   0:01.51 Xorg
 3319 alex      20   0  149m  13m  10m S   0.3  1.4   0:00.20 gnome-terminal
    1 root      20   0  2196  720  616 S   0.0  0.1   0:00.72 init
    2 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S   0.0  0.0   0:00.05 ksoftirqd/0
    4 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kworker/0:0
    5 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kworker/u:0
    6 root      rt   0     0    0    0 S   0.0  0.0   0:00.00 watchdog/0
    7 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 cpuset
    8 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 khelper
    9 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kdevtmpfs
   10 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 netns
   11 root      20   0     0    0    0 S   0.0  0.0   0:00.00 sync_supers
   12 root      20   0     0    0    0 S   0.0  0.0   0:00.00 bdi-default
   13 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 kintegrityd
   14 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 kblockd
   15 root      20   0     0    0    0 S   0.0  0.0   0:00.04 kworker/0:1
   16 root      20   0     0    0    0 S   0.0  0.0   0:00.00 khungtaskd
   17 root      20   0     0    0    0 S   0.0  0.0   0:00.03 kswapd0
   18 root      25   5     0    0    0 S   0.0  0.0   0:00.00 ksmd
```

Фиг. 3.5 Програма top и списък с процеси

От примерът се вижда, че има общо 112 процеса (total), от които 1 е активен (running) и 111 са в спящо състояние (sleeping), 0 са спрени (stopped) и 0 зомбита (zombie).

Зомбитата са процеси, които са приключили своето изпълнение, но все още се намират в процесната таблица на системата, което най-често се дължи на “child” процес, чийто състояние не може да се определи от базовия процес.

Общата статистика за системата позволява да се види какво е количеството на използваната и на свободната памет, както и аналогична информация за swap файловата система.

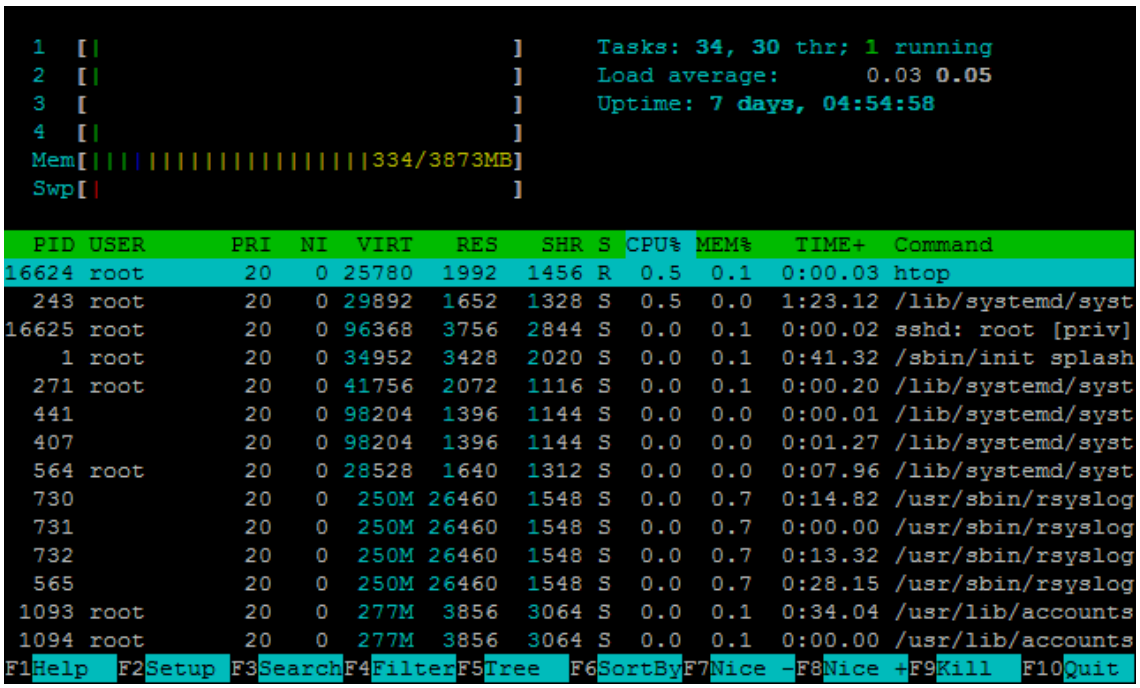
Друг важен параметър от обобщената информация е “load average”. Изведените 3 стойности в примера на фиг. 3.5 са “0.34 0.19 0.08”. Първата стойност (0.34) е средното натоварване в последната 1 минута, 0.19 – осреднената стойност за последните 5 минути и 0.08 – за последните 15 минути. При система с един процесор с едно ядро натоварване от 1 означава 100% използване на изчислителната мощ на процесора. При процесор с две ядра натоварване от 1 показва 50% от общата изчислителна мощ, а 2 – 100%.

След общата статистика командата top извежда детайлни данни за отделните процеси, които включват:

- PID – идентификатор на процеса;
- USER – потребителя, който е собственик на процеса;
- PR – приоритет на процеса;
- NI – стойността “nice” за процеса;
- VIRT – използваната от процеса виртуална памет;
- RES – използваната физическа памет от процеса;
- SHR – използваната споделена памет от процеса;
- S – дефинира статуса (S – sleeping, R – running, Z – zombie);
- %CPU – процентното натоварване на процесора;
- %MEM – процент използвана RAM памет;
- TIME+ - кумулативното време на работа на процеса;
- COMMAND – име на процеса.

За изход от top се използва клавиша “Q” или комбинацията “Ctrl+C”.

Алтернатива на top е htop – програма, която предоставя псевдо-графичен интерфейс и е по-лесна за начинаещи потребители.



Фиг. 3.6 Анализ на стартираните процес с htop

Top и htop са полезни инструменти, но не винаги са достатъчно гъвкави и много често администраторите предпочитат да използват командата ps. Ако ps се стартира без допълнителни аргументи се извежда кратък списък на процесите, стартирани от текущия потребител, което в общия случай не е достатъчно като информация. Най-често се добавя комбинацията от аргументи aux, чрез която се визуализират всички процеси (не само на текущия потребител), а данните се форматират в удобен за анализ вид.

```
alex@debian:~$ ps
  PID TTY          TIME CMD
 3327 pts/0    00:00:00 bash
 4169 pts/0    00:00:00 ps
alex@debian:~$
alex@debian:~$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   2196   720 ?        Ss   11:32   0:00 init [2]
root         2  0.0  0.0      0     0 ?        S    11:32   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    11:32   0:00 [ksoftirqd/0]
root         6  0.0  0.0      0     0 ?        S    11:32   0:00 [watchdog/0]
root         7  0.0  0.0      0     0 ?        S<   11:32   0:00 [cpuset]
root         8  0.0  0.0      0     0 ?        S<   11:32   0:00 [khelper]
root         9  0.0  0.0      0     0 ?        S    11:32   0:00 [kdevtmpfs]
root        10  0.0  0.0      0     0 ?        S<   11:32   0:00 [netns]
root        11  0.0  0.0      0     0 ?        S    11:32   0:00 [sync_supers]
root        12  0.0  0.0      0     0 ?        S    11:32   0:00 [bdi-default]
root        13  0.0  0.0      0     0 ?        S<   11:32   0:00 [kintegrityd]
root        14  0.0  0.0      0     0 ?        S<   11:32   0:00 [kblockd]
root        15  0.0  0.0      0     0 ?        S    11:32   0:00 [kworker/0:1]
root        16  0.0  0.0      0     0 ?        S    11:32   0:00 [khungtaskd]
root        17  0.0  0.0      0     0 ?        S    11:32   0:00 [kswapd0]
root        18  0.0  0.0      0     0 ?        SN   11:32   0:00 [ksmd]
root        19  0.0  0.0      0     0 ?        S    11:32   0:00 [fsnotify_mark]
root        20  0.0  0.0      0     0 ?        S<   11:32   0:00 [crypto]
root        91  0.0  0.0      0     0 ?        S    11:32   0:00 [khubd]
root        99  0.0  0.0      0     0 ?        S<   11:32   0:00 [ata_sff]
```

Фиг. 3.7 Изход от командата ps

Прекратяване на процес

При Linux всеки процес има уникален идентификатор (process ID), наречен за краткост PID. Чрез този параметър ядрото може да следи работата на процесите и ефективно да управлява ресурсите. Най-лесният начин да се определи PID за дадено приложение е чрез pgrep:

```
alex@debian:~$ pgrep bash

22432

alex@debian:~$
```

При стартиране на Linux първият процес (PID=1) е init, а неговата основна функционалност е да се грижи за създаването на всички останали процеси. Поради тази причина всеки нов процес има номер по-голям от 1.

```
alex@debian:~$ pgrep init

1

alex@debian:~$
```

Даден процес може да стартира няколко нови процеса, и при тази ситуация се явява т.нар. родител (parent), а новите процеси – деца (child). Номерът на родителя се дефинира с PPID (parent PID).

Всеки процес при Linux отговаря на изпратените към него сигнали (signals) – механизъм чрез който операционната система прекратява (terminate) или модифицира поведението на процеса.

За да се прекрати изпълнението на даден процес е възможно да се изпрати сигнала TERM чрез командата kill, последвана от аргумент PID или PPID.

```
alex@debian:~$ kill 22432
```

Ако въпреки изпратения TERM сигнал приложението все още е активно може да се изпрати сигнала KILL:

```
alex@debian:~$ kill –KILL 22432
```

За разлика от TERM, сигналът KILL не се изпраща към приложението, а към ядрото, което прекратява дадения процес.

Всеки сигнал има предварително дефиниран номер – например TERM=15, а KILL=9.

За да се изведе информация за възможните сигнали към kill трябва да се добави параметъра “-l”:

```
alex@debian:~$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
...
```

За да се изпрати сигнал към процес не по PID, а по име е възможно да се използва pkill:

```
alex@debian:~$ pkill gedit
```

Действието на pkill е еквивалентно на добавяне на pgrep към командата kill:

```
alex@debian:~$ kill $(pgrep –f gedit)
```

Прекратяването на всички стартирани процеси от дадено приложение най-лесно се осъществява чрез командата killall:

```
alex@debian:~$ killall gedit
```

Промяна на приоритета на процес

Linux управлява приоритета на процеса чрез параметъра “niceness”, стойността на който може да се види чрез top (колона - NI). Niceness е в интервала -19/-20 (най-висок приоритет) до 19/20 (най-нисък приоритет).

За да се стартира програма с определен приоритет трябва да се използва nice, последвана от аргумент, задаващ приоритета и командата:

```
alex@debian:~$ nice –n 10 mc
```

За да се промени приоритет на вече стартиран процес се използва renice, с аргументи стойност за приоритет и PID:

```
alex@debian:~$ renice -10 3329
```

За да се стартира процес на заден план (background) и управлението веднага да се върне към конзолата накрая на реда се добавя символа “&”:

```
alex@debian:~$ gedit&

[2] 3507
```

Работа с файлове и директории

Едно твърдение, което първоначално е дефинирано за UNIX е в сила и за Linux – “При Unix всичко е файл, а ако дадено нещо не е файл то е процес“. Работата с файлове е основна дейност при всяка операционна система, като при Linux освен чрез графичен интерфейс (ако е наличен) пълен контрол върху файловете и директориите може да бъде осъществен и от конзолата.

При Linux голяма част от файловете са “просто файлове”, които съдържат данни или са изпълними програми, но въпреки, че може да се приеме, че цялото съдържание на файловите системи са файлове има и няколко изключения:

- Директории – съдържат файлове или други директории;
- Специални файлове – механизъм, използван за входно изходни операции, най-често се намират в /dev;
- Връзки (links) – механизъм, чрез който файл или директория се вижда от няколко места спрямо файловата система;
- Socket – специален тип файлове, които наподобяват TCP/IP сокет, и които се използват за комуникация между процеси;
- “named pipe” – подобно на сокет осигуряват комуникация между процеси, но не използват мрежов пренос на данни.

Чрез командата “ls –l” може да се провери типа на файловете, който се определя от първия символ на всеки ред.

```
alex@debian:~/Basics$ ls -l
total 204
-rwxrwxrwx 1 alex alex     52 Sep 19 11:18 backupFTP
-rw-r--r-- 1 alex alex   3101 Sep 18 19:46 calc.zip
-rw-r--r-- 1 alex alex 191830 Sep 18 19:45 Debian_6.0.2.1.png
drwxr-xr-x 2 alex alex   4096 Sep 19 21:16 directory1
drwxr-xr-x 2 alex alex   4096 Sep 19 21:16 Temp
alex@debian:~/Basics$
```

Фиг. 3.8 Изход от команда ls -l

Използваните идентификатори са:

- “-“ – обикновен файл;
- “d” – директория;
- “l” – връзка;
- “c” – специален файл;
- “s” – сокет;
- “p” – „named pipe”;
- “b” – блоково устройство (block device).

От примерът на фигура 3.8 се вижда, че има три нормални файла и две директории (Temp и directory1).

За да бъде скрит файл или директория при Linux името трябва да бъде предхождано от “.”.

Всяка отделна операционна система използва един или няколко дяла на диска (partitions). При Linux най-често се използват два основни типа дялове:

1. Дял за данни (data partition) – съдържа системните файлове, root дяла и др.
2. Дял за временна памет (swap partition) – разширение на паметта на системата със специално заделено място на диска.

Всеки отделен дял се прикача към файловата система чрез т.нар. “mount point”., като обикновено всички дялове са прикачени към “root” дяла (“/”).

За да се види информация за дяловете по време на работата на Linux може да се използва командата df (disk free). Добавяйки аргумента “-h” информацията се представя в по-удобен за потребителя вид. Df извежда информация за размера на дяла, заетото и свободното място, процентното съотношение и ‘mount point”.

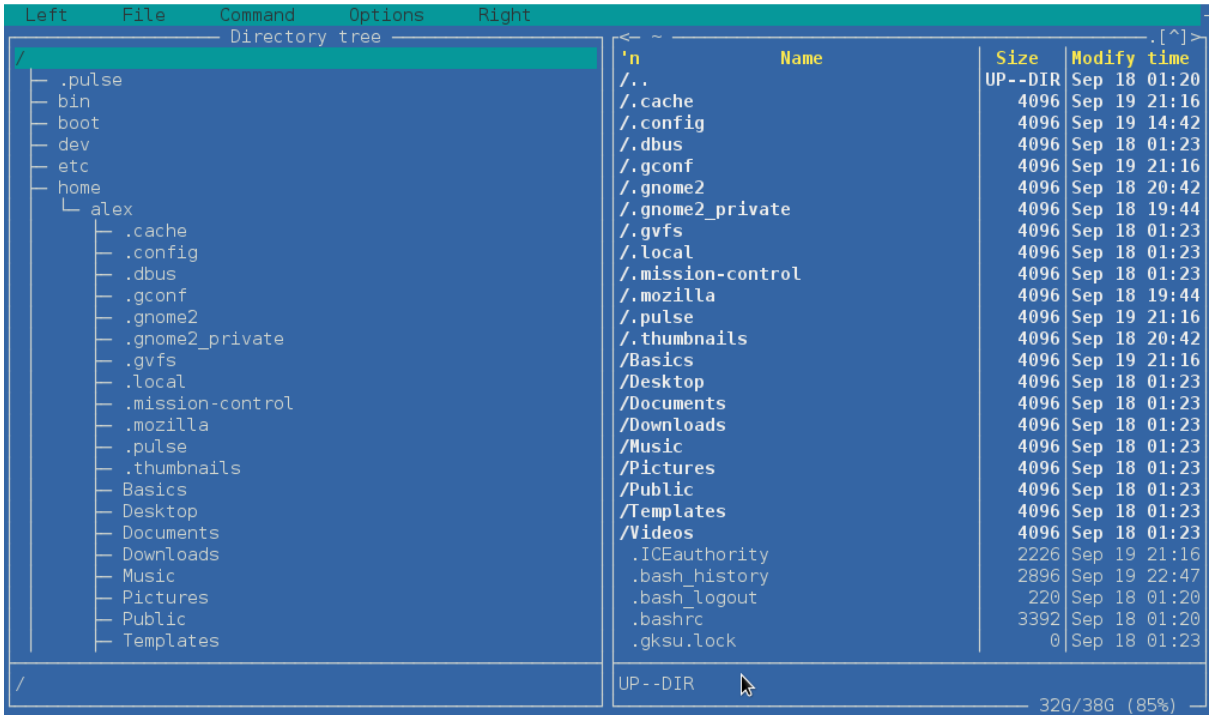
```
alex@debian:~/Basics$ df
Filesystem                                          1K-blocks    Used Available Use% Mounted on
rootfs                                               39559592 3779276  33770768  11% /
udev                                                    10240       0     10240   0% /dev
tmpfs                                                  103460     600    102860   1% /run
/dev/disk/by-uuid/8b704eb0-bce4-48a1-ae8b-247fbc30929e 39559592 3779276  33770768  11% /
tmpfs                                                    5120       0      5120   0% /run/lock
tmpfs                                                  556700     372    556328   1% /run/shm
/dev/sr0                                                63252   63252         0 100% /media/cdrom0
alex@debian:~/Basics$
alex@debian:~/Basics$
alex@debian:~/Basics$
alex@debian:~/Basics$ df -h
Filesystem                                          Size  Used Avail Use% Mounted on
rootfs                                               38G  3.7G   33G  11% /
udev                                                 10M     0   10M   0% /dev
tmpfs                                               102M  600K  101M   1% /run
/dev/disk/by-uuid/8b704eb0-bce4-48a1-ae8b-247fbc30929e  38G  3.7G   33G  11% /
tmpfs                                               5.0M     0  5.0M   0% /run/lock
tmpfs                                               544M  372K  544M   1% /run/shm
/dev/sr0                                             62M   62M     0 100% /media/cdrom0
alex@debian:~/Basics$
```

Фиг 3.9 Команда df

Структурата от директории може да варира в зависимост от дистрибуцията, но някои от по-важните от тях са:

- /bin - основни програми на операционната система;
- /boot - ядрото на операционната система;

- /dev - файловете (устройства), отговарящи на хардуера на системата;
- /etc - глобални конфигурационни файлове;
- /home - директории на потребителите;
- /lib - най-важните системни библиотеки;
- /mnt или /media - монтирани външните устройства и файлови системи;
- /opt - инсталации на големи софтуерни пакети, като KDE и Gnome;
- /proc - достъп до различни параметри на операционната система;
- /root - директория на root потребителя;
- /sbin - основни програми използвани от root;
- /tmp - временни файлове;
- /usr - повечето инсталирани програми, помощната информация и др.;
- /var - данни на системните програми.



Фиг. 3.10 Структура на директориите на Debian дистрибуция, визуализирана в mc

Създаване и изтриване на директория

Преминаването от една директория в друга става чрез командата cd.

За да се създаде директория се използва mkdir (make directory), а аргумента е името и опционално местоположението на новосъздаваната директория:

```
mkdir ~/Backups
```

Изтриване на празна директория се извършва чрез командата rmdir (remove directory):

```
rmdir ~/Backups
```

Ако директорията не е празна се извежда съобщение за грешка:

```
rmdir ~/Backups

rmdir: failed to remove '/home/user/Backups/': Directory not empty
```

Един от методите за изтриване на директория и нейното съдържание, който е потенциално опасен е да се използва командата за изтриване на файл rm с аргументи –rf:

```
rm –rf ~/Backups
```

Комбинацията от аргументи –rf изтрива всички файлове и директории от посочената, без да се иска потвърждение от потребителя.

```
alex@debian:~$ pwd
/home/alex
alex@debian:~$
alex@debian:~$ mkdir Backup
alex@debian:~$
alex@debian:~$ ls
Backup  Basics  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
alex@debian:~$
alex@debian:~$ rmdir Backup
alex@debian:~$
alex@debian:~$ ls
Basics  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
alex@debian:~$
alex@debian:~$ mkdir Backup
alex@debian:~$ cp Basics/*.* Backup/
alex@debian:~$ rmdir Backup/
rmdir: failed to remove `Backup/': Directory not empty
alex@debian:~$
alex@debian:~$ rm -rf Backup/
alex@debian:~$
alex@debian:~$ ls
Basics  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
alex@debian:~$
```

Фиг. 3.11 Команди за работа с директории

Копиране на файлове и директории

При Linux копирането на файлове и директории се осъществява чрез командата cp (copy). Аргументите, които се подават към cp са източник и цел, например за да се копира файл file1.txt към file2.txt се използва:

```
alex@debian:~$ cp file1.txt file2.txt
```

При копиране на файлове се поддържат и т.нар. “wildcard” символи - например за да се копират всички файлове с разширение doc от текущата директория в директория /tmp е необходимо да се стартира:

```
alex@debian: ~$ cp *.doc /tmp
```

При копиране новия файл се създава с информация за дата и час, както и други служебни параметри към момента на извършване на операцията. За да се запазят оригиналните такива е необходимо да се добави аргумента “-p”:

```
alex@debian: ~$ cp –p file1.txt file2.txt
```

Много често се налага да се копира цяла директория, включително нейните поддиректории и файлове в друга директория. Това действие може да се извърши чрез включване на параметъра “-R“ към командата cp:

```
alex@debian: ~$ cp –R ~/Dir1/* /tmp/Backup
```

```
alex@debian:~/Basics$ ls
backupFTP  calc.zip  Debian_6.0.2.1.png  directory1  Temp
alex@debian:~/Basics$ cp calc.zip calc1.zip
alex@debian:~/Basics$ cp -p calc.zip calc2.zip
alex@debian:~/Basics$
alex@debian:~/Basics$ ls -lah
total 220K
drwxr-xr-x  4 alex alex 4.0K Sep 20 19:08 .
drwxr-xr-x 23 alex alex 4.0K Sep 20 19:06 ..
-rwxrwxrwx  1 alex alex   52 Sep 19 11:18 backupFTP
-rw-r--r--  1 alex alex 3.1K Sep 20 19:08 calc1.zip
-rw-r--r--  1 alex alex 3.1K Sep 18 19:46 calc2.zip
-rw-r--r--  1 alex alex 3.1K Sep 18 19:46 calc.zip
-rw-r--r--  1 alex alex 188K Sep 18 19:45 Debian_6.0.2.1.png
drwxr-xr-x  2 alex alex 4.0K Sep 19 21:16 directory1
drwxr-xr-x  2 alex alex 4.0K Sep 19 21:16 Temp
alex@debian:~/Basics$
```

Фиг. 3.12 Пример за копиране на файлове

Изтриване на файлове и директории

Изтриване на файлове при Linux се извършва чрез командата rm (remove), която поддържа “wildcard” символи, както и няколко изключително важни аргумента.

За да се изтрие единичен файл (например calc1.zip) се използва следния синтаксис:

```
alex@debian: ~$ rm calc1.zip
```

Важно е да се отбележи, че при предходния пример не се изисква от потребителя потвърждение на операцията за изтриване на файла. За да се активира това запитване трябва да се добави параметъра “-i”.

Приложението на “wildcard” символите е аналогично на това при копиране на файлове – например за да се изтрият всички файлове с разширение doc е необходимо да се използва:

```
alex@debian: ~$ rm *.doc
```

Изтриването на всички файлове рекурсивно от директория изисква и параметъра “-r”.

Добавяйки аргумента “-f” (force) операционната система не извежда съобщение за грешка пи несъществуващ файл, и се опитва да изтрие всички посочени.

Изтриването на празна директория може да се извърши чрез командата rmdir, а ако има съдържание чрез опасния вариант rm –rf.

```
alex@debian:~/Basics$ ls
backupFTP  calc1.zip  calc2.zip  calc.zip  Debian_6.0.2.1.png  directory1  Temp
alex@debian:~/Basics$
alex@debian:~/Basics$ rm calc1.zip
alex@debian:~/Basics$
alex@debian:~/Basics$ rm -i calc2.zip
rm: remove regular file `calc2.zip'? yes
alex@debian:~/Basics$
alex@debian:~/Basics$ ls
backupFTP  calc.zip  Debian_6.0.2.1.png  directory1  Temp
alex@debian:~/Basics$
alex@debian:~/Basics$
```

Фиг. 3.12 Примери за изтриване на файлове

Местене (преименуване) на файлове и директории

При Linux преместването и преименуването на файлове се осъществява чрез командата mv (move). Ако оригиналния файл и новия са с едно и също име, но се намират в различни директории се извършва преместване, а ако имената са различни – преименуване, например за да се премести file1.txt в директория /tmp трябва да се използва следния синтаксис:

```
alex@debian: ~$ mv file1.txt /tmp/
```

Аналогично на cp и командата mv може да извежда запитване към потребителя за потвърждение на всяко действие. Тази функционалност изисква да се добави допълнителния аргумент “-i”.

```
alex@debian:~/Basics$ ls
backupFTP  calc.zip  Debian_6.0.2.1.png  directory1  file1.txt  Temp
alex@debian:~/Basics$
alex@debian:~/Basics$ mv file1.txt /tmp
alex@debian:~/Basics$
alex@debian:~/Basics$ ls
backupFTP  calc.zip  Debian_6.0.2.1.png  directory1  Temp
alex@debian:~/Basics$
alex@debian:~/Basics$ ls /tmp
file1.txt           pulse-qtc8F7qflcb7  ssh-MxBZUg97MxZQ
pulse-PKdhtXMmr18n  pulse-XSMzY8AmwTHW  tracker-alex
alex@debian:~/Basics$
alex@debian:~/Basics$
```

Фиг. 3.13 Примери за преместване на файл

Работа с устройства

При актуалните Linux дистрибуции добавянето на сменяеми носители автоматично включва техните дялове към активните, най-често в директория /mnt или /media. Ако е необходимо да се добави или премахне дял могат да се използват командите mount и umount. Основните параметри на командата mount са „-t”, последван от типа на файловата система, устройство и директория, в която да се осъществи включването - например ако е необходимо да се добави файловата система от CD_ROM в директория /mnt, се използва следния синтаксис:

```
alex@debian: ~$ mount –t iso9660 –o ro /dev/cdrom /mnt
```

Аргументът iso9660 определя типа на файловата система, използвана при записа на CD-ROM, “-o ro” конфигурира достъпа да е само за четене, а устройството е достъпно през /dev/cdrom.

За да се види списък с всичко добавени файлови системи може да се стартира командата mount без допълнителни параметри:

```
alex@debian: ~$ mount
/dev/sda5 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/fuse/connections type fusectl (rw)
…
```

Друг подход за получаване на информация за активните към момента файлови системи е да се стартира командата df, която беше разгледана в началото на тази глава.

Всички файлови системи, които са описани във файла /etc/fstab се активират при стартиране на системата и ако е необходимо някои от тях могат да бъдат деактивирани чрез командата umount:

alex@debian: ~$ umount /mnt

```
alex@debian:~/Basics$ ls /media
cdrom  cdrom0
alex@debian:~/Basics$
alex@debian:~/Basics$ mount -t iso9660 /dev/sr0 /media/cdrom0
mount: only root can do that
alex@debian:~/Basics$
alex@debian:~/Basics$ su
Password:
root@debian:/home/alex/Basics#
root@debian:/home/alex/Basics# mount -t iso9660 /dev/sr0 /media/cdrom0
mount: block device /dev/sr0 is write-protected, mounting read-only
root@debian:/home/alex/Basics#
root@debian:/home/alex/Basics# ls /media/cdrom0
32Bit        cert                  VBoxSolarisAdditions.pkg
64Bit        OS2                   VBoxWindowsAdditions-amd64.exe
AUTORUN.INF  runasroot.sh          VBoxWindowsAdditions.exe
autorun.sh   VBoxLinuxAdditions.run  VBoxWindowsAdditions-x86.exe
root@debian:/home/alex/Basics#
root@debian:/home/alex/Basics# ls /media/cdrom0
```

Фиг. 3.14 Пример за работа с командата mount

Както вече беше описано някои файлове в Linux не са просто файлове, а процеси или директории. Директорията /dev съдържа необходимата информация за достъп до различните устройства (видими като файлове), а съдържанието на /dev зависи от редица параметри, сред които версията на ядрото, потребителя, инсталираните пакети и много други. Някои от по-важните устройства (не всички могат да се открият в дадена дистрибуция) са:

- alarm – достъп до аларми;
- ashmem – Споделена памет при Andorid;
- autofs – автоматично активирани файлови системи;
- binder – виртуално устройство, което използва споделена памет за надеждна комуникация при Inter-Process Communication (IPC);
- block – устройство, което може да поддържа блоково съхранение на данни (например твърд диск);
- bsg – Block SCSI устройства;
- bus – устройства, които са налични и достъпни през системната шина;
- cdrom – най-често препратка към друго устройство, например /dev/sr0;
- char – устройства с поток от символи;
- disk – връзки към устройствата за съхранение на данни;
- loop0 – използва се за активиране и работа с ISO или IMG файлове;
- mapper – пренасочване на едно блоково устройство към друго;
- mem – физическа памет на системата;

- null – директно изтриване на насочения към него стандартен поток;
- port – комуникационни портове (например паралелен порт);
- tty – текущо използвания терминал;
- tty1 – първата достъпна виртуална конзола;
- ttyS – сериен порт;
- ttyUSB – сериен порт през USB.

Потребители и пароли

Аналогично но почти всяка модерна операционна система Linux поддържа потребителски профили и групи, като всеки потребител се определя еднозначно от името си (username) и идентификатор (user ID – UID). UID е положително число и в общия случай е със стойност над 500 за потребители и над 100 за системни профили. Всеки потребител използва своя парола, която може да се съхранява в шифриран вид. Локалните пароли се записват в /etc/passwd или /etc/shadow. При включване към системата потребителя въвежда или избира своето потребителско име и попълва своята парола, която се криптира и получената стойност се сравнява със съхранените профили.

Най-често администраторите използват подхода, при който в /etc/passwd се записват имената на профилите, а в /etc/shadow – шифрираните пароли.

```
alex@debian:~$ more /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
```

Фиг. 3.15 съдържание на файлове /etc/passwd и /etc/shadow

За да е по-лесно администрирането потребителите се разпределят в групи. Когато дадени права се конфигурират на определена група, нейните членове автоматично ги наследяват. Потребителските групи могат да са първични (primary) и вторични, като всяка една от тях се идентифицира с име и групов идентификатор (group ID – GID).

Всеки потребител може да използва собствена домашна директория (home), която се създава в /home, и която е най-лесно достъпна през конзолата чрез символа „~”.

Създаване на потребител

Добавянето на потребител при Linux чрез конзолна команда се извършва с adduser (Debian) - например за да се създаде нов потребител с име ivan е необходимо да се използва следния синтаксис:

```
root@debian:/# adduser ivan
```

Инструментът adduser добавя потребителя и изисква въвеждане на парола и допълнителна (незадължителна) информация за него.

```
root@debian:/home/alex# adduser ivan
Adding user `ivan' ...
Adding new group `ivan' (1001) ...
Adding new user `ivan' (1001) with group `ivan' ...
The home directory `/home/ivan' already exists.  Not copying from `/etc/skel'.
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ivan
Enter the new value, or press ENTER for the default
        Full Name []: Ivan Ivanov
        Room Number []:
        Work Phone []: 102
        Home Phone []: 359 2 999 87 65
        Other []:
Is the information correct? [Y/n] y
root@debian:/home/alex#
```

Фиг. 3.16 Добавяне на потребител с adduser

Изтриване на потребител

За да се изтрие потребител от конзола може да се използва инструмента deluser (Debian):

```
root@debian:/# deluser ivan
```

Важно е да се отбележи, че не се изисква потвърждение на действието, което изисква и по-внимателна работа и проверка на правилно въведеното потребителско име.

```
root@debian:/home/alex# deluser ivan
Removing user `ivan' ...
Warning: group `ivan' has no more members.
Done.
root@debian:/home/alex#
```

Фиг. 3.17 Изтриване на потребител с deluser

Промяна на парола на потребител

Смяна на парола на съществуващ потребител се извършва чрез passwd с аргумент потребителско име:

```
root@debian:/# passwd ivan
```

Ако не се въведе потребителското име като параметър се сменя паролата на текущия потребител.

```
root@debian:/home/alex# passwd ivan
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@debian:/home/alex#
root@debian:/home/alex#
```

Фиг. 3.18 Смяна на потребителска парола през конзолата на Linux

Групи

Както вече беше споменато за да е по-лесно управлението на потребителите и техните права е възможно да се създадат групи, като това става чрез командата groupadd:

```
root@debian:/# groupadd HRusers
```

Информацията за групите се съхранява в /etc/group, а за подсигурените групи в /etc/gshadow.

При добавянето на нов потребител е възможно да бъде посочена и групата, в която да бъде включен чрез аргумента “-G”:

```
root@debian:/# useradd –G HRusers petar
```

Ако потребителският профил е бил създаден преди да бъде добавена групата е необходимо де се използва usermod:

```
root@debian:/# usermod –g HRusers ivan
```

Изтриване на група се извършва чрез groupdel:

root@debian:/# groupdel HRusers

```
alex@debian:~$ su
Password:
root@debian:/home/alex# groupadd HRusers
root@debian:/home/alex#
root@debian:/home/alex# useradd -G HRusers petar
root@debian:/home/alex#
root@debian:/home/alex# usermod -g HRusers ivan
root@debian:/home/alex#
root@debian:/home/alex# groupdel HRusers
groupdel: cannot remove the primary group of user 'ivan'
root@debian:/home/alex#
root@debian:/home/alex# usermod -g ivan ivan
root@debian:/home/alex# groupdel HRusers
root@debian:/home/alex#
```

Фиг. 3.19 Пример за работа с групи с потребители

Права на достъп

Управлението на параметрите за достъп до файлове е една от най-важните задачи при подсигуряването на операционните системи, сървърите и данните.

Всеки файл или директория има три параметъра, свързани с достъпа:

1. **Собственик** (owner);
2. **Група** (group);
3. **Всички останали** (all users).

За всеки един от параметрите има три опции, които могат да бъдат настройвани:

1. **Read** (четене);
2. **Write** (запис);
3. **Execute** (изпълнение);

Най-често при проверка на правата за достъп се извежда информация във вида:

_rwxrwxrwx 1 owner:group

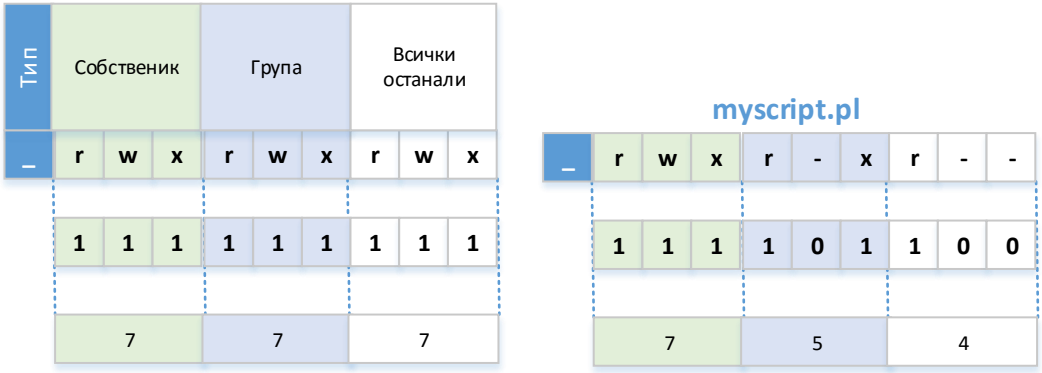
Първият символ показва дали обекта е файл или директория, в случая символа "_" е за файл. Следващите три символа определят правата за собственика, от четвъртия до шестия символ дефинират правата за групата, а последните - правата за всички останали. Цифрата показва броя на връзките (hardlinks) към файла. Последният параметър е комбинацията от собственик и група.

Задаването на правата може да стане чрез определяне с буква или чрез преобразуване на двоични стойности в десетични (спрямо позицията). За да се получи числената стойност, която отговаря на желаните права се приема, че:

- **r** = 4;
- **w** = 2;
- **x** = 1;

Например, ако желаните права са пълен контрол за собствените, само четене и изпълнение за групата и само четене за всички останали се получава стойност 754:

- Собственик: 4+2+1=**7**;
- Група: 4+1=**5**;
- Всички останали: **4**;



Фиг. 3.20 Задаване на права за запис, четене и изпълнение на файл

Промяната на правата става чрез командата chmod:

```
alex@debian:~$ chmod 754 myscript.pl
```

Освен записът, използващ цифри правата могат да се посочат и чрез символи (r, w, x), например:

```
alex@debian:~$ chmod u+rwx myscript.pl

alex@debian:~$ chmod g+rx myscript.pl

alex@debian:~$ chmod o+r myscript.pl
```

Промяната на собственика и групата за даден файл или директория се конфигурират чрез chown:

```
alex@debian:~$ chown ivan:HRusers myscript.pl
```

В примерът на фиг. 3.28 комбинацията ivan:HRusers променя собственика на файла myscrip.pl да бъде потребител ivan и група HRusers.

```
root@debian:/home/alex/Basics# ls -lah myscript.pl
-rw-r--r-- 1 alex alex 0 Sep 23 09:17 myscript.pl
root@debian:/home/alex/Basics#
root@debian:/home/alex/Basics# chown ivan:HRusers myscript.pl
root@debian:/home/alex/Basics#
root@debian:/home/alex/Basics# ls -lah myscript.pl
-rw-r--r-- 1 ivan HRusers 0 Sep 23 09:17 myscript.pl
root@debian:/home/alex/Basics#
```

Фиг. 3.21 Промяна на собственик на файл

Конфигуриране на мрежови интерфейси

Мрежовите функции са силно застъпени в Linux и тяхната конфигурация може да се извърши през конзолата или чрез графичен интерфейс. Още от своето създаване Linux е проект, който разчита на Internet като основна платформа за колаборация между разработчиците, което е още един силен мотив за дълбокото интегриране на TCP/IP.

Една изключително добра книга, за въведение и навлизане в мрежовата администрация на Linux е “Linux Network Administrator’s Guide (NAG)”, чиито първи две издания са достъпни безплатно[45].

Мрежови интерфейси в Linux

За да се получи абстракция от разнородните хардуерни модули TCP/IP стека дефинира обект интерфейс, който се използва за достъп до хардуера на системата. Интерфейсите се явяват набор от операции, които са еднотипни за всеки тип хардуер и които участват в обмяната на пакети по мрежовата инфраструктура. Важно е да се отбележи, че за всеки физически мрежови интерфейс в ядрото на Linux е необходимо да бъде включен софтуерен модул, например ако системата разполага с две мрежови карти, които работят по стандарта Ethernet техните интерфейси са eth0 и eth1 (имената могат да варират в зависимост от дистрибуцията).

За да се види списък с наличните интерфейси може да се използва командата:

```
alex@debian:~$ ip link show
```

Друг начин за визуализиране на данните за мрежовите интерфейси е чрез ifconfig:

```
root@debian:/# ifconfig
```

[45] Второто издание може да бъде изтеглено от адрес http://www.tldp.org/LDP/nag2/index.html123456

```
root@debian:/home/alex/Basics# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 08:00:27:62:b1:14 brd ff:ff:ff:ff:ff:ff
root@debian:/home/alex/Basics#
root@debian:/home/alex/Basics#
root@debian:/home/alex/Basics# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:62:b1:14
          inet addr:192.168.1.248  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe62:b114/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:18151 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6446 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:26418582 (25.1 MiB)  TX bytes:442682 (432.3 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:228 errors:0 dropped:0 overruns:0 frame:0
          TX packets:228 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:13680 (13.3 KiB)  TX bytes:13680 (13.3 KiB)

root@debian:/home/alex/Basics#
```

Фиг. 3.22 Команди “ip link show” и ifconfig

Интерфейс от тип Bridge

Много често в различни мрежови топологии се налага да се извърши т.нар. “bridging”, при който мрежовият сегмент, свързан към една мрежова карта се достъпва от втори сегмент, на друга карта. Например, ако към един от интерфейсите е наличен достъп до Интернет, то втори интерфейс позволява споделяне на връзката.

За да се създаде мостов адаптер (bridge) е необходимо да са инсталирани и работещи инструментите bridge-utils, след което да се стартира:

```
root@debian:/# brctl addbr bro
```

По този начин се създава нов интерфейс – br0. Следващата стъпка е към br0 да се добавят интерфейсите, които ще бъдат включени в моста:

```
root@debian:/# brctl addif bro etho eth1
```

За да се направи конфигурацията постоянна е необходимо да се редактира файла с мрежовите интерфейси /etc/network/interfaces (Debian).

Няколко полезни команди, свързани с мостовите интерфейси са:

- bridge_stp off - деактивиране на STP;
- bridge_waitport 0 – без забавяне при активиране на интерфейса;
- bridge_fd 0 – изключване на забавянето при предаване (forwarding delay);
- bridge_ports regex eth* # - използване на регулярен израз[46] за описване на интерфейсите.

[46] Regular expression

```
root@debian:/home/alex# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 08:00:27:62:b1:14 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 08:00:27:e4:23:10 brd ff:ff:ff:ff:ff:ff
root@debian:/home/alex#
root@debian:/home/alex# brctl addbr br0
root@debian:/home/alex#
root@debian:/home/alex# brctl addif br0 eth0 eth1
root@debian:/home/alex#
root@debian:/home/alex# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP mode DEFAULT qlen 1000
    link/ether 08:00:27:62:b1:14 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP mode DEFAULT qlen 1000
    link/ether 08:00:27:e4:23:10 brd ff:ff:ff:ff:ff:ff
5: br0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT
    link/ether 08:00:27:62:b1:14 brd ff:ff:ff:ff:ff:ff
root@debian:/home/alex#
root@debian:/home/alex#
```

Фиг. 3.23 Пример за създаване на мостов интерфейс под Debian

IPv4 конфигурация

За да се извърши конфигуриране на статичен IPv4 адрес и другите необходими настройки се използва командата ifconfig, например за да се зададе IP адрес 10.0.0.1 с префикс /8 на интерфейс eth1 синтаксиса е:

```
root@debian:/# ifconfig eth1 10.0.0.2/8
```

Добавянето на шлюз (gateway) е чрез командата route:

```
root@debian:/# route add default gw 10.0.0.1
```

Ако е необходимо адресът да се получава чрез DHCP се използва dhclient:

```
root@debian:/#dhclient eth1
```

Добавянето на DNS изисква редактиране на файла /etc/resolv.conf (Debian) и описването на един или няколко реда от типа:

```
nameserver X.X.X.X[47]
```

Направената по този начин конфигурация ще се запази да рестартирането на системата. Ако е необходимо настройките да са перманентни, те трябва да се опишат във файла /etc/network/interfaces.

За предходния пример конфигурацията на eth1 трябва да бъде:

```
auto eth1

iface eth1 inet static

        address 10.0.0.2

        netmask 255.0.0.0

        gateway 10.0.0.1
```

[47] X.X.X.X е адресът на DNS сървър

За да се визуализират IPv4 пътищата, включени в маршрутизиращата таблица (routing table) се използва командата:

root@debian:/#route -n

```
root@debian:/home/alex# ifconfig eth1 10.0.0.2/8
root@debian:/home/alex# route add default gw 10.0.0.1
root@debian:/home/alex#
root@debian:/home/alex# service networking restart
[warn] Running /etc/init.d/networking restart is deprecated because it may not re-enable some interfaces ... (wa
rning).
[ ok ] Reconfiguring network interfaces...done.
root@debian:/home/alex#
root@debian:/home/alex# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:e4:23:10
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::a00:27ff:fee4:2310/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:171 errors:0 dropped:0 overruns:0 frame:0
          TX packets:208 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15514 (15.1 KiB)  TX bytes:21531 (21.0 KiB)

root@debian:/home/alex#
```

Фиг. 3.24 Пример за IPv4 конфигурация при Debian

IPv6 конфигурация

Аналогично на IPv4 настройките на IPv6 адресите могат да бъдат извършени през конзола или чрез графичен инструмент, но използваната команда за добавяне на статичен IPv6 адрес е с добавен аргумент -6, например:

*root@debian:/#ip **-6** addr add 2001:aaaa:bbbb:cccc::1/64 dev eth1*

Една от опциите на IPv6 е поддръжката на множество адреси на един и същи интерфейс, което често изисква и изтриване на един или няколко. Изтриването на определен IPv6 адрес отново е чрез командата ip, например:

root@debian:/#ip -6 addr del 2001:aaaa:bbbb:cccc::1/64 dev eth1

Аналогично на IPv4 направената конфигурация се описва в /etc/network/interface.

```
root@debian:/home/alex# ip -6 addr add 2001:aaaa:bbbb:cccc::1/64 dev eth1
root@debian:/home/alex#
root@debian:/home/alex# ip -6 add show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 fe80::a00:27ff:fe62:b114/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2001:aaaa:bbbb:cccc::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee4:2310/64 scope link
       valid_lft forever preferred_lft forever
root@debian:/home/alex#
```

Фиг. 3.25 Пример за конфигуриране на IPv6 при Debian

Активиране и деактивиране на интерфейси

Активирането и деактивирането на интерфейсите на системата се извършва или от конзола или чрез специализиран инструмент с графичен потребителски интерфейс. В терминал

се използва командата ifconfig, последвана от интерфейса и опцията down за деактивиране и up за активиране, например:

root@debian:/#ifconfig eth1 down

```
root@debian:/home/alex# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:e4:23:10
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::a00:27ff:fee4:2310/64 Scope:Link
          inet6 addr: 2001:aaaa:bbbb:cccc::1/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1104 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1056 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:84075 (82.1 KiB)  TX bytes:67339 (65.7 KiB)

root@debian:/home/alex#
root@debian:/home/alex# ifconfig eth1 down
root@debian:/home/alex#
root@debian:/home/alex# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:e4:23:10
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:1104 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1056 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:84075 (82.1 KiB)  TX bytes:67339 (65.7 KiB)

root@debian:/home/alex#
```

Фиг. 3.26 Пример за деактивиране на мрежови интерфейс при Debian

Полезни клавишни комбинации

Всяка отделна Linux дистрибуция може да оптимизира и промени част от включените софтуерни пакети, в това число и начина на използване на съкратените клавишни комбинации.

Някои от по-важните съкратени клавишни комбинации при работа с конзола са:

- Стрелка нагоре/стрелка надолу – преместване в списъка с последно въведените команди;
- Ctrl+стрелка на ляво/дясно – преместване между аргументите на програмата;
- Home и End – преместване на курсора в началото, респективно в края на реда;
- Ctrl+U – изтриване на целия ред;
- Ctrl+K – изтриване на реда от текущата позиция на курсора до края на съдържанието;
- Ctrl+W – изтриване на думата преди курсора;
- Ctrl+R – извеждане на опция за търсене в списъка с въведените команди;
- Tab – дописване на команда и др.

Допълнителни източници

Най-добрият начин ако сте начинаещи с Linux да започнете да използвате тази изключително мощна операционна система е да я инсталирате (препоръчително първоначално във виртуална машина) и да работите с нея.

Самата идея на Linux е да бъде разработван от множество програмисти, които да комуникират през Интернет и това е само една от многото причини глобалната мрежа да е основното място за всякакви въпроси, описания и съвети, свързани с инсталирането, работата и поддръжката на Linux.

Някои препоръчителни допълнителни източници са:

- Linux NAG - http://www.tldp.org/LDP/nag/nag.html;
- Ресурсите, поместени на сайта www.linux.com;
- Книгата “Introduction to Linux – A Hands on Guide”;
- Книгата “Linux Command Line Cheat Sheet”;
- Книгата “The GNU/Linux Advanced Administration”;
- Книгата “Linux Shell Scripting Cookbook” и др.

Заключение

Linux е един от най-важните проекти с отворен код, който през годините се развива от просто ядро до мощна операционна система, която се налага на пазара на сървърни решения, и все повече при работните станции.

За мрежовите администратори е малко вероятно да не се наложи поне веднъж в своята практика да използват за определени цели Linux, което прави запознаването с него силно препоръчително.

Освен удобните среди с графичен интерфейс Linux може да се използва и да бъде изцяло конфигуриран чрез инструменти с команден ред в конзола.

В тази глава бяха описани някой от най-важните програми и команди при работа с конзола, но съдържанието е ориентирано към изграждане на мрежова сигурност с EFW и не е достатъчно за конфигуриране и поддържане на пълна сървърна платформа или мрежови устройства, базирани на Linux.

Източници

1. http://bg.wikipedia.org/wiki/ГНУ
2. http://www.lifehacker.com
3. http://www.thegeekscope.com/io-redirection-of-stdin-stderr-and-stdout-in-linux/
4. http://www.linfo.org/dot_slash.html
5. https://www.digitalocean.com/community/tutorials/how-to-use-ps-kill-and-nice-to-manage-processes-in-linux
6. https://wiki.debian.org/NetworkConfiguration#The_resolv.conf_configuration_file
7. https://wiki.debian.org/DebianIPv6

Глава 4. Въведение в Endian Firewall CE

Въведение

Компанията Endian[48] е създадена през 2003 година в град Апиано, Италия от екип от мрежови специалисти и Linux ентусиасти. Основната цел, която екипът си поставя е да разработи UTM система, базирана на софтуер с отворен код. Две години по-късно е пусната първата версия на Endian Firewall, която е налична като платен продукт и като свободно достъпно решение. Към момента на писането на книгата Endian Firewall (EFW) има над 1.6 милиона изтегляния. Решенията, които компанията предлага са специализирани хардуерни защитни стени и софтуерния продукт EFW/EFW Community Edition. Endian не спира да анализира и интегрира нови технологии в EFW, като едно от последните нововъведения е Hotspot, което прави продукта приложим в хотелиерския бизнес, здравеопазването, образованието и др.

Успехът на EFW на пазара на офис защитни мрежови решения подтиква екипа да реализира и технология за подсигуряване на индустриална комуникация, която да се базира на EFW, но и която да отговаря на строгите критерии на индустрията. В момента комуникацията между отделните индустриални системи използва както специализирани протоколи, така и Ethernet (Industrial Ethernet). Интегрирането на IP в индустрията за отдалечено наблюдение и управление води до редица рискове за сигурността, като типичен пример за изключително опасен зловреден код е вируса Stuxnet. През 2012 Endian стартират продажбата на индустриалната защитна стена Endian 4i.

През 2013 година след сключване на договор за партньорска дейност с ntop от Endian подготвят специализирана система за VPN тунелиране – т.нар. VPN switchboard.

При всички продукти на компанията се спазва принципа “easy to buy, easy to own”.

Endian Firewall Community Edition

Специализираните хардуерни устройства на Endian използват операционната сиситема EFW, която е достъпна като отделен продукт, но и като софтуер с отворен код – Endian Firewall Community Edition (EFW CE). EFW CE е UTM система, базирана на Linux, която е проектирана с цел максимална степен на защита, но с лесна инсталация, конфигуриране и наблюдение. Лекотата при работа с EFW CE по никакъв начин не е за сметка на възможностите или предоставяната степен на сигурност. EFW CE е класическо приложение с отворен код (изходния код е свободно достъпен в Интернет), като компанията Endian подкрепя и стимулира разработчиците при интегрирането на нови технологии и при отстраняването на откритите пропуски (сравнително малко).

Описание на продукта

Най-актуалната версия на EFW CE към момента на писане на книгата е 3.0, като изтегляниятa към месец октомври 2014 година са над 1398000. Инсталационен ISO файл, както и пълния изходен код са достъпни от основната страница на Endian и могат да бъдат свободно изтеглени.

EFW CE 3.0 предлага следната основна функционалност:

- Защитна стена от тип “stateful firewall”;
- IPS функционалност;
- QoS;

[48] www.endian.com

- Защита на WEB трафик – HTTP прокси и филтриране;
- Защита на електронна поща – SMTP прокси, спам филтър, антивирусно сканиране, конфигуриране на SMTP статуса на доставените съобщения;
- IPsec VPN– методи за шифриране и хеширане Null, 3DES, CAST-128, AES 128/192/256, Blowfish 128/192/256, Twofish 128/192/256, Serpent 128/192/256, Camellia 128/192/256, хеширане с MD5, SHA1, SHA2 256/384/512, AESXCBC и поддръжка на IKEv2;
- OpenVPN – поддръжка на тунелиращ режим (TUN) и специална страница за автентификация на потребителите;
- Управление на потребителите и автентификация – потребители за OpenVPN, вграден CA (Certificate Authority), поддръжка на външен CA, поддръжка на 2-факторен режим на автентификация;
- Журнали и доклади – наблюдение на мрежовия трафик в реално време (използва се ntopng[49]), съхранение на данните за системните графики дори след рестартиране на системата, детайлна статистика и др.;
- Поддръжка на серийна конзола през RS-232;
- Процедура за промяна на забравени пароли през конзолата и др.

Подробно описание на промените и функциите на текущата версия на EFW CE може да бъде изтеглена от сайта на Endian.

Разлики с останалите версии на Endian Firewall

Разликите между функциите, поддържани от хардуерните устройства на Endian и софтуерните версии на EFW са подробно описани на сайта на компанията.

При EFW CE липсват следните опции, които са налични при другите решения на Endian:

- Контрол на потребителските приложения (application control);
- Автентификация на отдалечени VPN потребители чрез специален портал;
- Hotspot;
- High availability;
- Журнал за събития (event logging);
- Специализирана комерсиална поддръжка;
- Заявка за проблем с “ticket”;
- Поддръжка по телефона;
- Поддръжка чрез отдалечен достъп до устройствата;
- Подмяна на хардуерни модули;
- Обновяване на хардуерни модули.

От списъка се вижда, че при EFW CE основно липсва поддръжка, в сравнение с платените варианти. Това е важно най-вече при приложение на технологията за подсигуряване на фирмена комуникация, която при наличие на определени проблеми или открити неточности би следвала да се обърне за съдействие и към производителя.

За малки мрежи (SOHO) EFW CE предоставя всички необходими функции и чрез него може да бъде изградена надеждна защита на мрежовия трафик и на работните места и данните на потребителите.

[49] www.ntop.org/products/traffic-analysis/ntop

	Hardware Appliances							Software Appliances		
	Mini	Mercury 50	Mercury	Macro X1	Macro X2	Macro R1	Macro R2	5-10 Users	25+ Users	Community
❶ Network Security	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Application Control	✓	✓	✓	✓	✓	✓	✓	✓	✓	x
❷ Web security	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
❸ Mail security	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
❹ Virtual Private Neworking	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
❺ WAN Failover	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
❻ User Authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Local User Authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HTTP Remote User Authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VPN Remote User Authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓	x
❼ Hotspot	✓	✓	✓	✓	✓	✓	✓	x	✓	x
❽ Network Address Translation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
❾ Routing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
❿ Bridging	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
⓫ High Availability	✓	✓	✓	✓	✓	✓	✓	✓	✓	x
⓬ Extra Services	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
⓭ Logging and Reporting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Live Network Monitoring	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Event Reporting	✓	✓	✓	✓	✓	✓	✓	✓	✓	x
⓮ Management	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
⓯ Updates and Backup	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓ = Yes, x = Not available, o = Optional

Фиг. 4.1 Сравнение на версиите на EFW (източник http://www.endian.com/en/products/security-gateways-utm/features)

Изисквания към хардуера

Изискванията към хардуерните компоненти за EFW CE са сравнително ниски, но е важно да се отбележи, че цялостната производителност на системата силно зависи от типа на процесора, наличната памет и типа на мрежовите интерфейси:

- Процесор – Intel® x86 съвместим с минимална работна честота 1GHz. Препоръчително е многоядрен процесор с честота 2GHz. Поддържа се симетрична мултипроцесорна работа (SMP);
- RAM – минимум 512 MB, препоръчва се 1 GB или повече;
- Твърд диск – SCSI, SATA, SAS, IDE с минимален размер от 8 GB, препоръчително е размерът да е над 20 GB;
- CD-ROM – не е необходим, ако инсталационната процедура е стартирана от USB Flash памет;
- Мрежови интерфейси – поддържат се почти всички масово разпространени Ethernet и оптични мрежови карти;
- Монитор и клавиатура – необходими са за процеса на инсталиране. Работата на EFW CE може да бъде без тези периферни устройства;
- Операционна система – EFW CE съдържа оптимизирана Linux дистрибуция.

Изтегляне на EFW CE

Както вече беше споменато EFW CE е свободно достъпен като инсталационен ISO файл и като изходен програмен код. Адресът, от който може да бъдат изтеглени необходимите файлове е:

http://www.endian.com/community/download/

За да се получи достъп до обновяванията на EFW CE е необходимо да се направи безплатна регистрация на посочения в конфигурационния интерфейс адрес.

След приключване на процедурата за регистриране инструкциите за обновяване се изпращат по електронна поща.

Документация

Документацията за EFW CE е достъпна от сайта на Endian на адрес:

http://docs.endian.com/3.0/utm/index.html

Документите съдържат пълно описание на функционалността, предлагана от софтуера EFW и някои от конфигурационните параметри или технологии не са налични в EFW CE. Описанието на EFW е направено в HTML формат и е оформено в следните основни секции:

- Предговор (Preface);
- Въведение (Getting Started);
- Меню “Система” (The System Menu);
- Меню “Статус” (The Status Menu);
- Меню “Мрежа” (The Network Menu);
- Меню “Услуги” (The Services Menu);
- Меню “Защитна стена” (The Firewall Menu);
- Меню “Прокси” (The Proxy Menu);
- Меню “VPN” (The VPN Menu);
- Меню “Switchboard” (The Switchboard Menu);
- Меню “Hotspot” (The Hotspot Menu);
- Журнали и доклади (The Logs and Reports Menu);
- Съкращения (Glossary);
- Бърз справочник (Quicksheet);

- GNU лиценз (GNU Free Documentation License);
- Индекс (Index).

Концепция за зони

Една от най-важните концепции, която е заложена не само в EFW, но и в актуалните защитни стени и UTM системи е разделянето на мрежовата комуникация на отделни зони. Трафикът, напускащ или постъпващ в дадена зона, както и обменяната информация между две зони се анализира и филтрира на база на правила, зададени от администраторите.

При EFW има четири базови зони, които могат да обединяват няколко мрежови сегмента в една зона:

1. **Червена зона** (Red) – Интернет/WAN/външни мрежи;
2. **Зелена зона** (Green) – LAN/вътрешни мрежови сегменти;
3. **Оранжева зона** (Orange) – DMZ (демилитаризирана зона, съдържаща защитени сървъри и общодостъпни устройства);
4. **Синя зона** (Blue) – Безжични мрежи/Hotspot.

Червената зона включва несигурните мрежови сегменти, които се явяват външни за мрежата, защитена от EFW – например Интернет или някои от използваните WAN комуникационни линии. Това е единствената зона, която не може да бъде с разширени настройки и за която единствено може трафика да се разрешава или забранява.

Зелената зона дефинира мрежовите сегменти, които са вътрешни и които се считат за надеждни – например LAN (VLAN). Зелената зона не може да бъде директно достъпвана от червената. По подразбиране това е единствената зона, която има позволен достъп до интерфейса за конфигуриране на EFW.

Оранжевата зона най-често се използва за DMZ[50] – включва сървърите, които са достъпни от Интернет (Web, електронна поща и др.). Препоръчително е това да е единствената зона, която може директно да бъде достъпна от червената. По този начин ще се получи изолиране на атаките, а при успешен пробив няма да са застрашени вътрешните LAN сегменти.

Синята зона се използва за безжичен достъп (Wireless LAN или Hotspot). Устройствата от тази зона имат достъп до червената, но най-често нямат до зелената и оранжевата.

Минимално се изисква да се конфигурират зелена и червена зона.

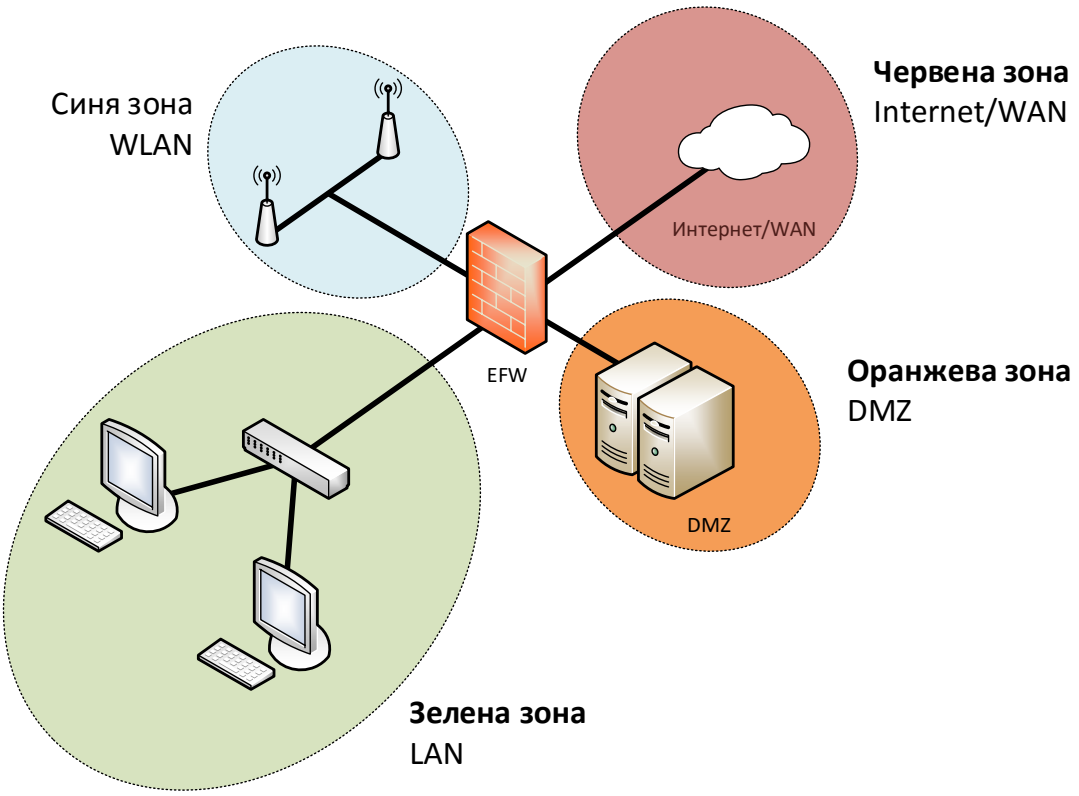
Всяка зона трябва да използва отделни IPv4 или IPv6 мрежи или подмрежи, например:

- Зелена зона – 192.168.1.0/8;
- Червена зона – 81.128.0.0/18;
- Синя зона – 172.16.0.0/24.

Препоръчително е за зелената, синята и оранжевата зона да се използват IPv4 адресите, дефинирани в RFC 1918[51]:

- 192.168.1.0/24;
- 192.168.10.0/24;
- 172.16.0.0/16.

[50] Demilitarized Zone
[51] tools.ietf.org/html/rfc1918

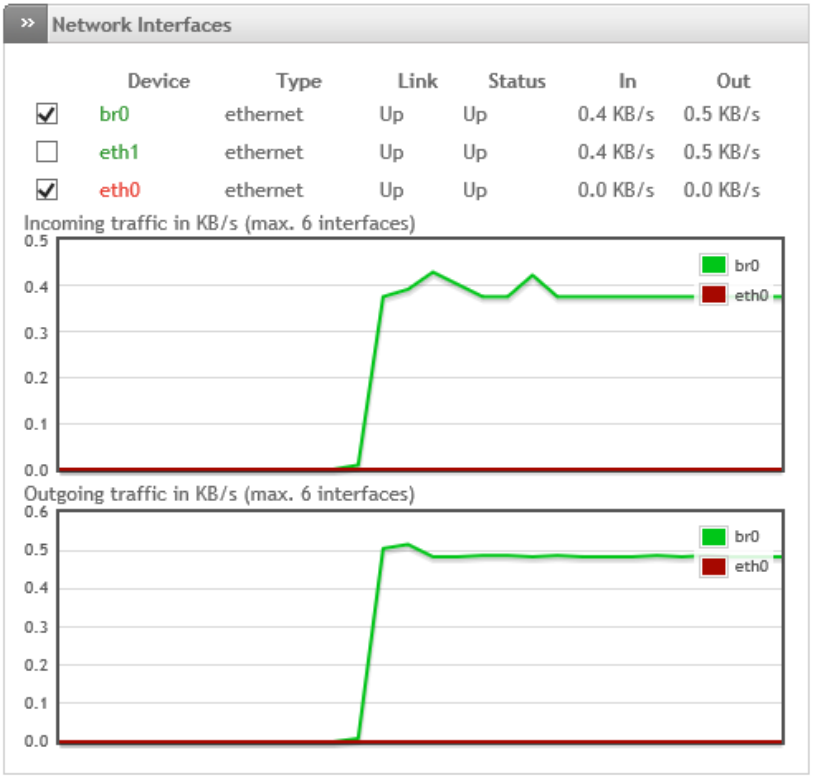


Фиг. 4.2 Зони при EFW

Аналогично на повечето UTM системи EFW CE има предварително дефинирани правила, описващи трафика между зоните, които са рестриктивни и забраняват преминаването на определени типове пакети. Освен четирите основни зони се поддържат и други две, които са налични при по-сложни топологии – зона за OpenVPN клиенти и т.нар. HA (High Availability) зона.

OpenVPN зоната за клиенти (виолетова зона) включва всички устройства, свързани през OpenVPN към EFW и по подразбиране използва IPv4 адреси 192.168.15.0/24. HA зоната е за технологията High Availability и аналогично на виолетовата зона по подразбиране има конфигурирани IPv4 адреси 192.168.177.0/24.

Към всяка една зона има свързан интерфейс, който трябва да има зададен съответен хост IP адрес (възможно е повече от 1 адрес от дадена мрежа). Под интерфейс се разбира Ethernet или безжичен адаптер, през който преминава трафика от и към дадената зона. Логично е всеки интерфейс да има име, отговарящо на цвета на зоната, например RED (червен) е за червената зона и т.н. Конфигурацията на EFW спазва концепцията <зона>IP адрес – ако червената зона има адрес 199.15.1.1/24, то интерфейсът RED ще има адрес 192.168.1.1 – срещан като REDIP.

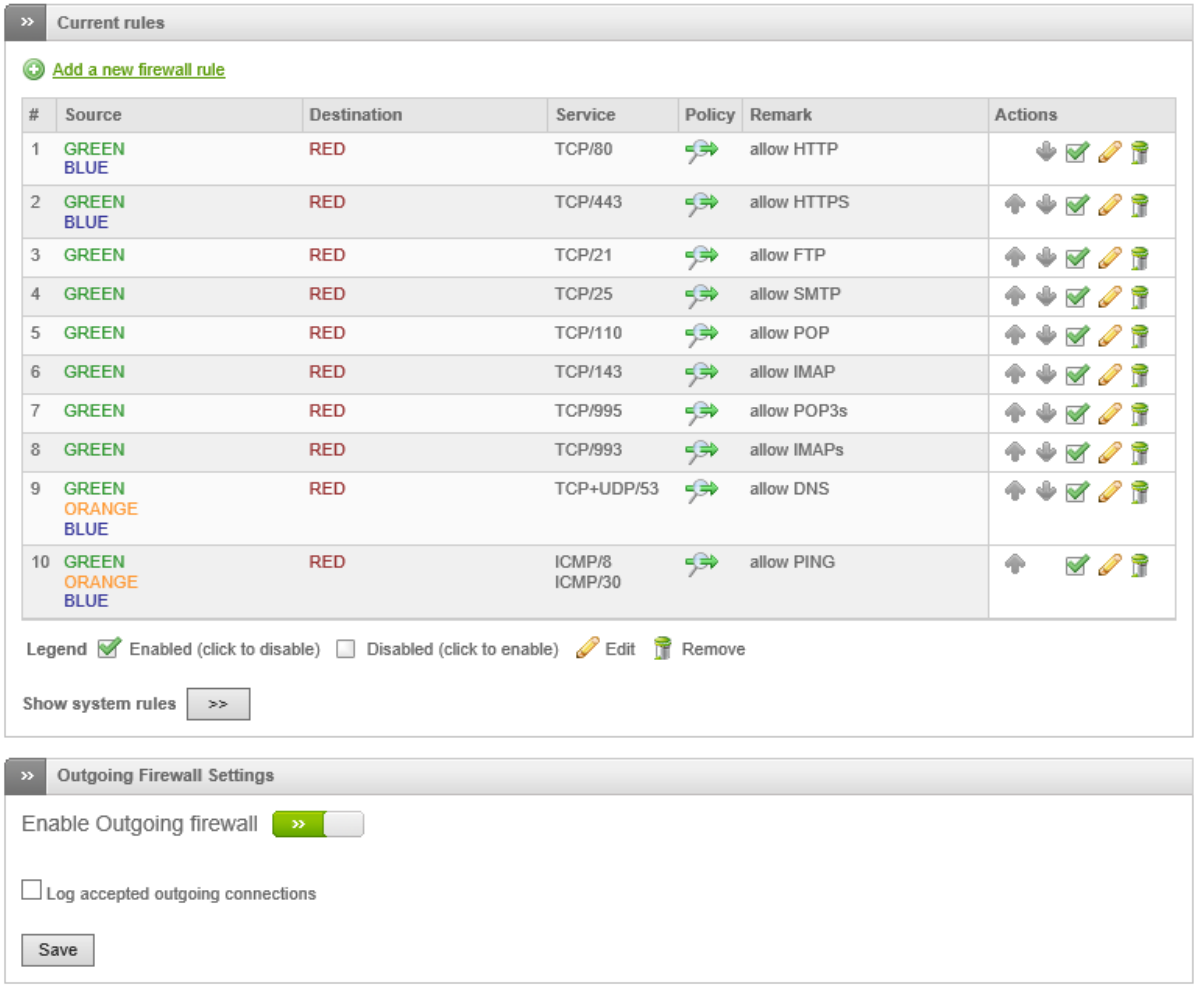


Фиг. 4.3 Интерфейси и зони при EFW CE

Предварително дефинирани правила за трафик

След инсталиране на EFW автоматично се конфигурират правила, свързани с трафика между зоните и с изходящите пакети (към червената зона). По подразбиране през EFW могат да преминават пакети, със следните параметри:

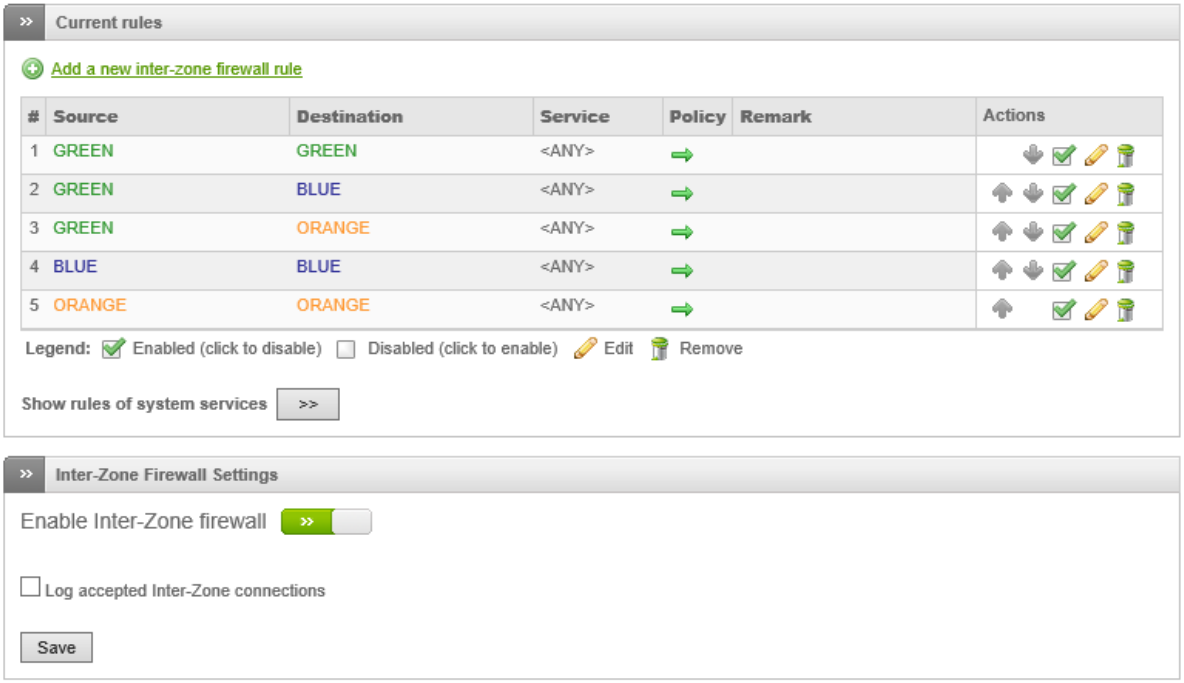
- Източник на трафика в зелената или синята зона, получател в червената зона, използван транспортен протокол TCP и отдалечен порт 80;
- Източник на трафика в зелената или синята зона, получател в червената зона, използван транспортен протокол TCP и отдалечен порт 443;
- Източник на трафика в зелената зона, получател в червената зона, използван транспортен протокол TCP и отдалечен порт 21;
- Източник на трафика в зелената зона, получател в червената зона, използван транспортен протокол TCP и отдалечен порт 25;
- Източник на трафика в зелената зона, получател в червената зона, използван транспортен протокол TCP и отдалечен порт 110;
- Източник на трафика в зелената зона, получател в червената зона, използван транспортен протокол TCP и отдалечен порт 143;
- Източник на трафика в зелената зона, получател в червената зона, използван транспортен протокол TCP и отдалечен порт 995;
- Източник на трафика в зелената зона, получател в червената зона, използван транспортен протокол TCP и отдалечен порт 993;
- Източник на трафика в зелената, оранжевата или синята зона, получател в червената зона, използвани транспортни протоколи TCP и UDP и отдалечен порт 53;
- Източник на трафика в зелената, оранжевата или синята зона, получател в червената зона, протокол ICMP тип 8 (ping) и 30 (traceroute).



Фиг. 4.4 Правила по подразбиране за филтриране на трафика към червената зона

Аналогично на трафика към червения интерфейс при EFW има предварително зададени политики за филтриране на потока от пакети между другите зони:

- Източник на трафика в зелената зона, получател в зелената зона – разрешени са всички протоколи и услуги;
- Източник на трафика в зелената зона, получател в синята зона – разрешени са всички протоколи и услуги;
- Източник на трафика в зелената зона, получател в оранжевата зона – разрешени са всички протоколи и услуги;
- Източник на трафика в синята зона, получател в синята зона – разрешени са всички протоколи и услуги;
- Източник на трафика в оранжевата зона, получател в оранжевата зона – разрешени са всички протоколи и услуги.



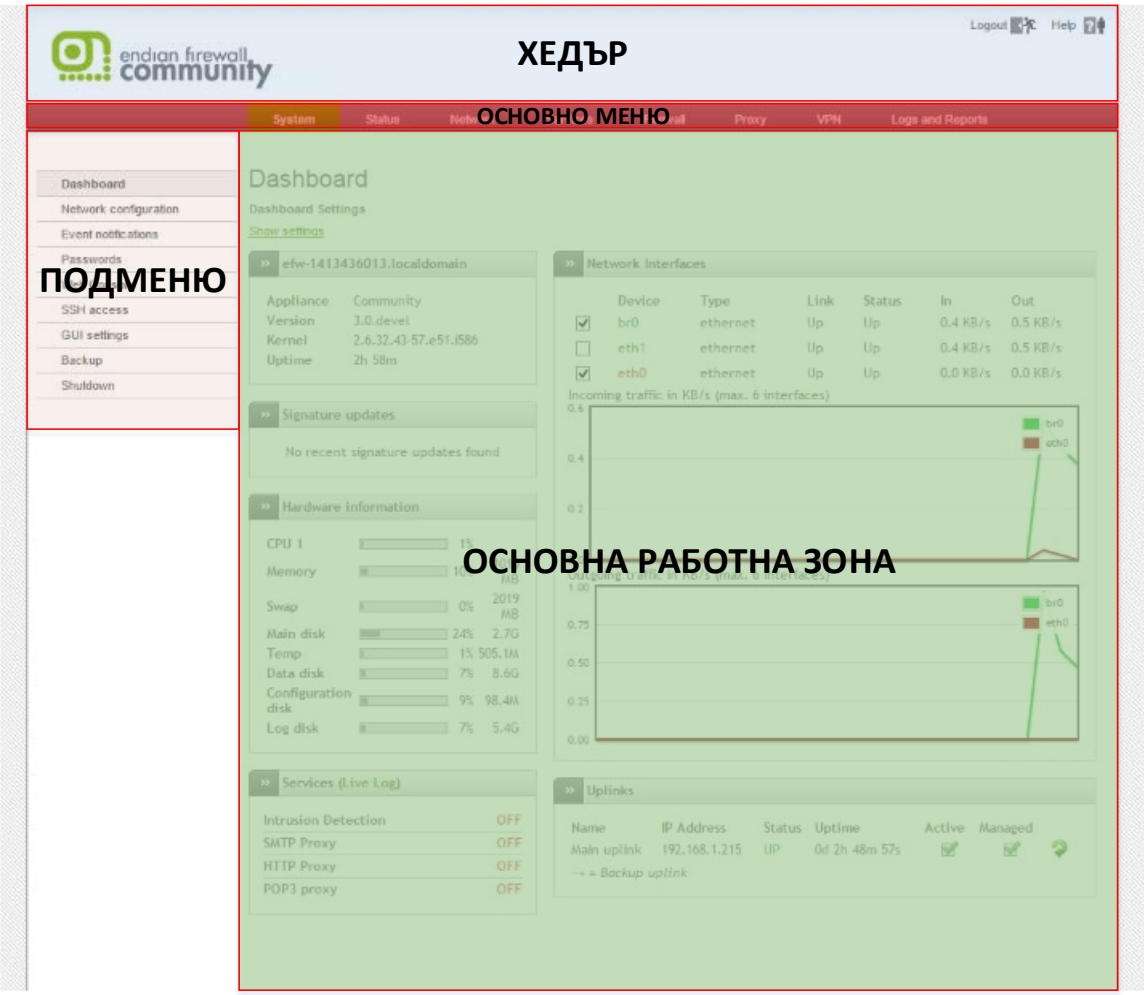
Фиг. 4.5 Правила по подразбиране за филтриране на трафика между зоните

Потребителски интерфейс на EFW CE

EFW CE може да бъде конфигуриран през специално разработения за целта Web интерфейс, който е интуитивен и лесен за употреба. Също така се поддържа и специална текстова конзола, която има предварително дефинирани функции, но позволяваща и достъп до shell на използваната Linux операционна система.

Графичният интерфейс е разделен на пет основни части:

1. **Хедър** – най-горната част на страницата, с логото на Endian, и линковете за изключване (logout) и помощ;
2. **“Footer”** – най-долната част на страницата, която дава кратка информация за продукта и текущия статус. Тази част от интерфейса може да не е налична при EFW CE;
3. **Основно меню** – намира се непосредствено под хедъра и дава достъп до основните групи с функции на EFW;
4. **Подменю** – в най-лявата част на страницата са налични подменютата за избрания от основното меню модул;
5. **Основна работна зона** - по-голямата част от страницата, която съдържа информация опции за настройки и др. Всички настройки са обединени в отделни групи (tab).



Фиг. 4.6 Разположение на отделните компоненти в графичния интерфейс на EFW CE

Hotspot функциите се конфигурират от допълнителен интерфейс (Hotspot Administration Interface), който няма footer и при който подменюто е поставено под основното.

Използвани икони и означения

При работа с менютата на EFW се използват няколко стандартизирани икони, показани на фигурата.

	Бутонът изцяло активира или деактивира дадена услуга. Ако означението на бутона е в сиво функцията е изключена, при зелен индикатор е включена. При активиране е необходим интервал от време за стартиране на необходимите софтуерни компоненти и демони и в зависимост от хардуера до няколко секунди следва услугата да е налична. Аналогично при изключване се извършва спиране на софтуерните процеси и демони.
	В конфигурираната политика достъпа се разрешава безусловно.
	Политиката разрешава достъп, единствено след успешна проверка от IPS системата.
	Пакетите се блокират и отхвърлят.
	Пакетите се отхвърлят и към източника на трафика се изпраща съобщение.
	Показва, че част от правилата имат забранителен характер. Тази икона най-често е поставено в началото на политиките за достъп и има информационен характер.
	Отваряне и затваряне на съдържанието на панел.

	При конфигурация, която изисква точно определяне позицията на реда чрез стрелките може да се извършва промяна на местоположението (например при правилата на защитната стена).
	Редактиране на посочената настройка.
	Изтриване на посочения ред или обект.
	Изтегляне на файл или друг обект.
	Рядко използвана икона за проверка на свързаност с отдалечен сървър.
	При IPS конфигурират пакета дали да бъде блокиран или пропуснат след определяне на активно правило.

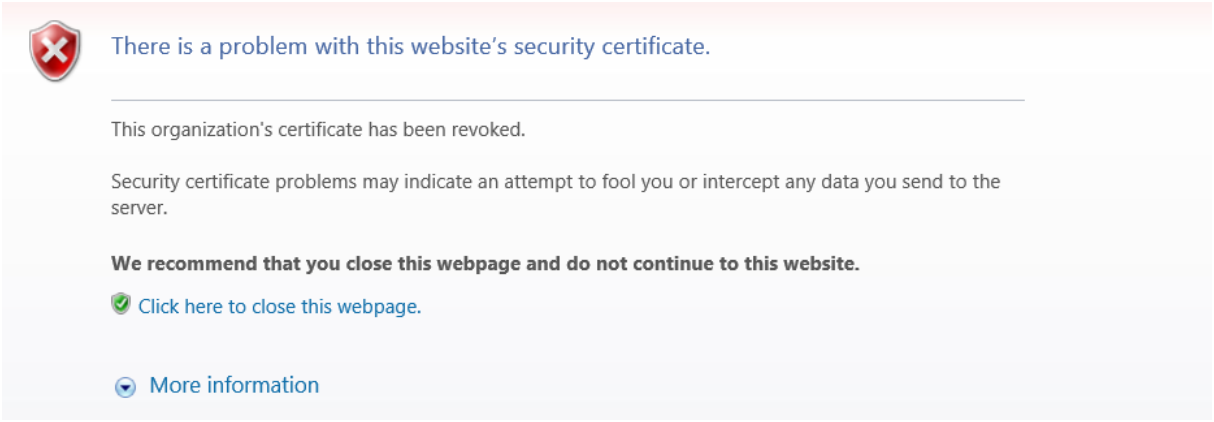
Достъп до EFW CE

Конфигурирането на EFW CE може да се извърши по няколко различни начина, като най-интуитивния и лесен е чрез Web базирания интерфейс.

Други методи за достъп до EFW CE с цел наблюдение и промяна на настройките са специалната конзола през серийна връзка и чрез отдалечен и подсигурен SSH достъп. Тези методи се препоръчват да се използват само от напреднали потребители и администратори, които имат необходимите познания за Linux и EFW.

Web интерфейс

Web интерфейсът по подразбиране може да бъде достъпен на зеления интерфейс на EFW и порт 10443 през HTTPS. Използваният сертификат е “self signed” и браузъра може да изведе съобщение за потвърждаване на зареждането на сайта при първоначалното свързване.



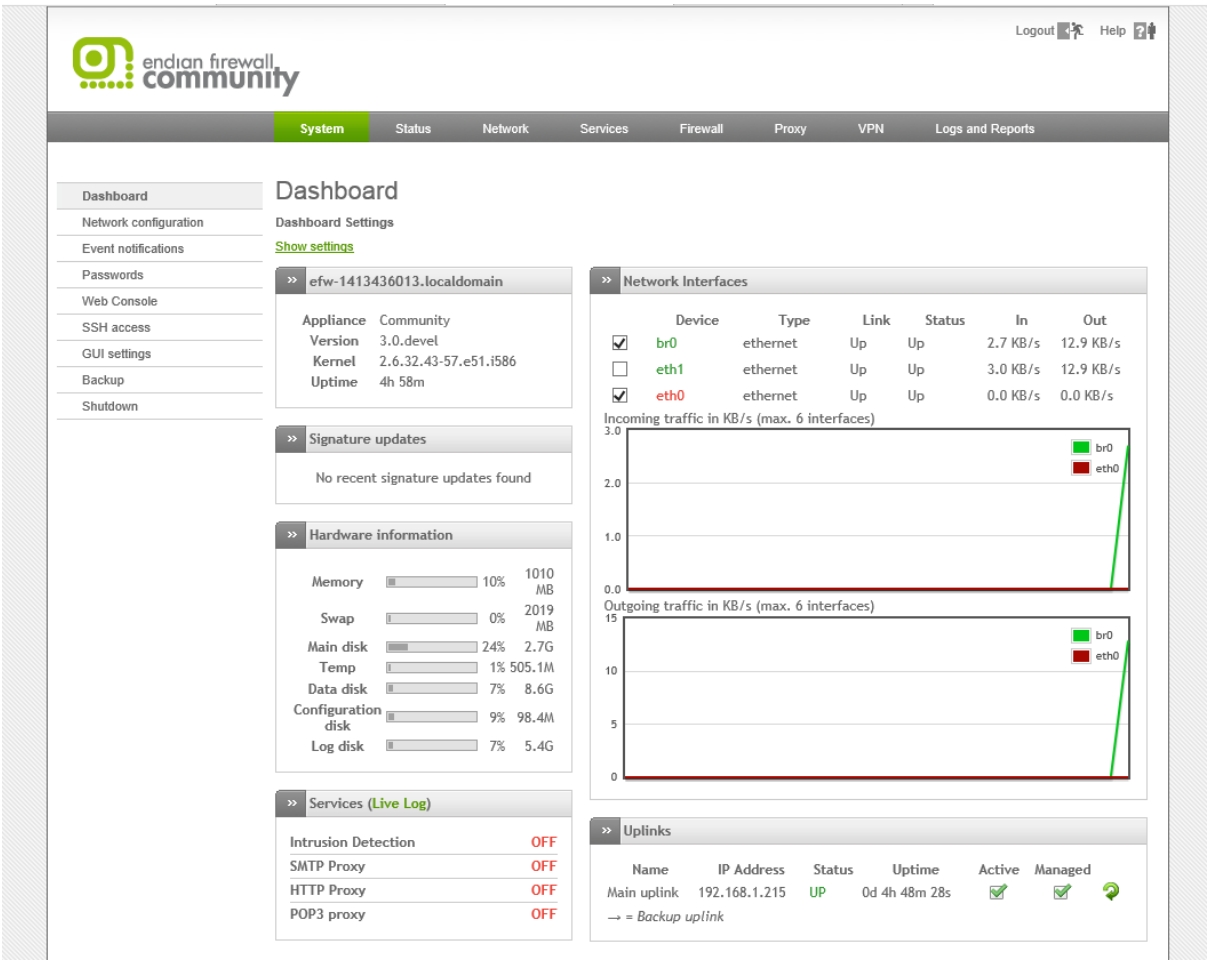
Фиг. 4.7 Предупреждение за “self signed” сертификат от Internet Explorer 11

При успешно зареждане на сайта е необходимо да се въведат потребителско име и парола, които по подразбиране са:

- Потребител – **admin**
- Парола – **endian**

Забележка: възможно е в процеса на инсталиране да са зададени и други комбинации от потребител и парола за достъпа до Web интерфейса.

Ако са въведени правилните данни за потребителя се зарежда т.нар. “Dashboard” страница, която дава обобщен изглед на най-важната информация за системата. Налични са и всички основни елементи – менюта, подменюта и др.



Фиг. 4.8 “Dashboard” страница на EFW CE

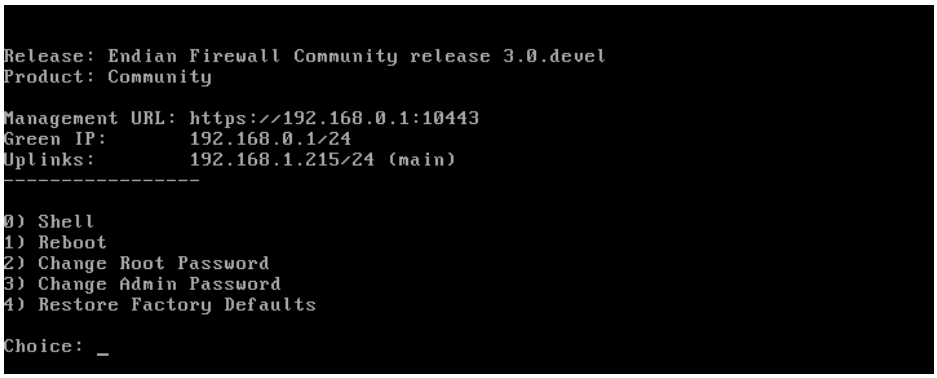
Конзола

Запознатите с Linux потребители могат да използват конзолния интерфейс на EFW за наблюдение и конфигуриране на системата. За да се получи достъп е възможно да се използва SSH или серийна връзка. Важно е да се отбележи, че по подразбиране SSH достъпа е забранен, но серийната конзола е активна със следните параметри:

- Порт – ttyS0 (COM1);
- 8 data bits;
- 1 parity bit;
- No flow control;
- 1 stop bit;
- Скорост на трансфер 115200 (при версия на EFW преди 3.0 скоростта е 38400).

За да се получи достъп до конзолата през RS-232 е необходимо да се използва софтуер за терминална емулация като minicom (Linux) или TeraTerm (Microsoft Windows). Физически връзката се осъществява с null-modem[52] кабел.

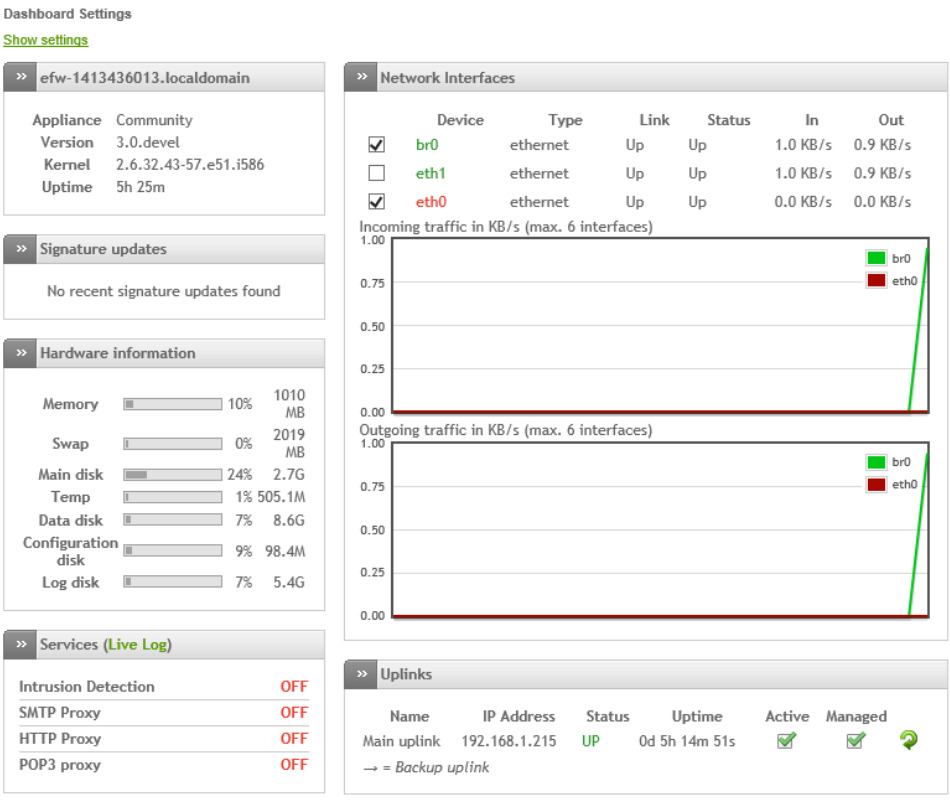
[52] en.wikipedia.org/wiki/Null_modem



Фиг. 4.9 Конзола на EFW CE

Dashboard

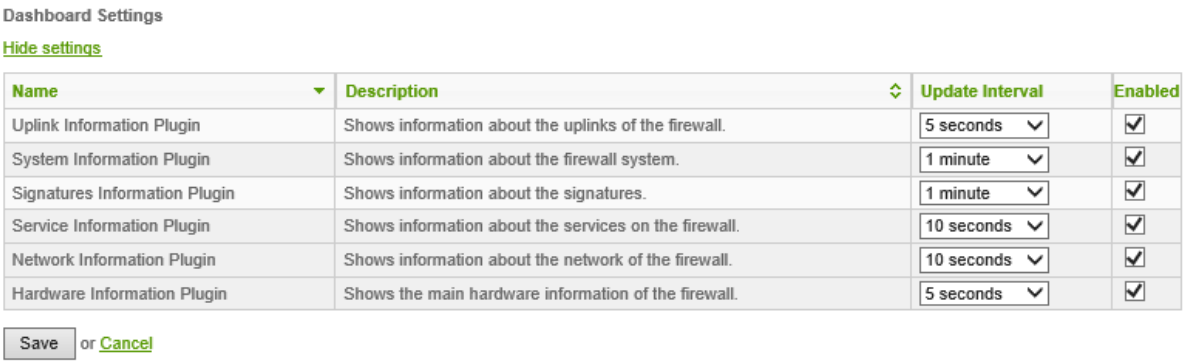
Dashboard е първата страница, която се зарежда при стартиране на Web интерфейса на EFW и която съдържа няколко основни групи с данни (plugins). Групата за основна информация описва типа на системата (в случая Community), версията на EFW, версията на ядрото на използваната операционна система, както и времето на работа до момента. Под основната информация е поместена групата, която описва кога за последно е извършено успешно обновяване на сигнатурите. В групата за хардуерни данни се виждат състоянията на натоварване на процесора, използваната RAM памет и swap и свободното дисково пространство (както и на отделните файлови системи). Групата за услуги (Services) визуализира състоянието на най-важните услуги. На фигура 4.10 се вижда, че SMTP, HTTP и POP3 прокситата, както и IPS са изключени (настройка по подразбиране). Данни за натоварването на мрежовите интерфейси е поместено в “Network interfaces”. Също така има и графично представяне в реално време на отчетените стойности. Последната група “Uplinks” съдържа информация за червения интерфейс.



Фиг. 4.10 Данни, включени в страница Dashboard на EFW CE

Настройките на Dashboard могат да бъдат променени след натискане на връзката “Show settings”. Възможно е да бъдат зададени интервалите на отчитане на отделните параметри, които се визуализират в отделните секции на Dashboard:

- “Uplink Information Plugin” – интервал на отчитане на червения интерфейс (стойност по подразбиране 5 s);
- “System Information Plugin” – времето за опресняване на данните за системата (стойност по подразбиране 60 s);
- “Signatures Information Plugin” – време за отчитане на състоянието на обновяването на сигнатурите (стойност по подразбиране 60 s);
- “Service Information Plugin” – период за отчитане на състоянието на услугите (стойност по подразбиране 10 s);
- “Network Information Plugin” – интервал за отчитане на състоянието на интерфейсите на системата (стойност по подразбиране 10 s);
- “Hardware Information Plugin” – време за опресняване на данните за хардуера (стойност по подразбиране 5 s).



Фиг. 4.11 Интервали на отчитане на данните за Dashboard

Статус на интерфейсите

Статусът на интерфейсите при EFW, който се извежда като информация на страницата Dashboard може да бъде:

- Stopped – интерфейсът не е свързан;
- Inactive – интерфейсът не е активен;
- Connecting – процес на изграждане на свързаност;
- Connected или Up – свързаността е успешно изградена и интерфейсът работи;
- Disconnecting – извършва се процес на прекратяване на изградената свързаност;
- Failure – възникнала е грешка при изграждането на свързаност;
- Failure, reconnecting – поради възникнала грешка се извършва нов опит за свързване;
- Dead link – връзката е осъществена, но не може да се достигнат хостове или да се извърши успешна проверка на състоянието.

Заключение

Компанията Endian предлага специализиране хардуерни UTM устройства, които работят с Endian Firewall – UTM система, изградена на база на Linux дистрибуция, която предлага удобен и интуитивен потребителски интерфейс и мощни защитни функции.

Софтуерът на EFW е наличен и като продукт с отворен код наречен Endian Firewall Community Edition. EFW CE се отличава от пълната версия по поддръжката на hotspot, по-подробните отчети, high availability функциите и др.

EFW работи на принципа на зони и е необходимо като минимални параметри да бъдат зададени конфигурациите на червена (несигурна) и зелена (сигурна) зона.

Страницата Dashboard предоставя цялата необходима информация за базово наблюдение на EFW от едно място.

Глава 5. Инсталиране и основно конфигуриране на EFW CE

Въведение

Въпреки сравнително ниските изисквания към хардуера EFW CE предоставя изключително мощни функции за защита и филтриране и чрез него може лесно да бъде конфигурирана надеждна UTM защитна система.

Аналогично но конфигурирането и наблюдението на EFW CE и в процеса на неговото инсталиране е спазен принципа за максимална лекота и интуитивни процеси. Процедурата по инсталирането отнема не повече от 5 минути и след първоначалното конфигуриране (в рамките на 15 до 20 минути) UTM системата има базова функционалност и е готова за експлоатация. Прецизирането на настройките, най-вече на IPS система изисква допълнително време, което зависи от топологията на мрежата и използваните услуги.

Избор на хардуер

Минималните системни изисквания за инсталиране и работа на EFW CE са процесор, който е Intel® x86 съвместим с минимална работна честота 1GHz, поне 512 MB RAM памет, 20 GB дисково пространство и две (една при по-специфична конфигурация, използваща VLAN или за експерименти) мрежови карти. Поддържаните хардуерни модули са почти всички масово разпространени, като при необходимост може да се направи справка на сайта на Endian, дали даденото устройство се поддържа или не. При конфигуриране на DMZ е необходимо да са налични 3 мрежови карти или да се зададат VLAN.

Препоръчително е използвания хардуер да е надежден за да се избегнат проблеми при продължителната му експлоатация без изключване.

Инсталиране на EFW CE

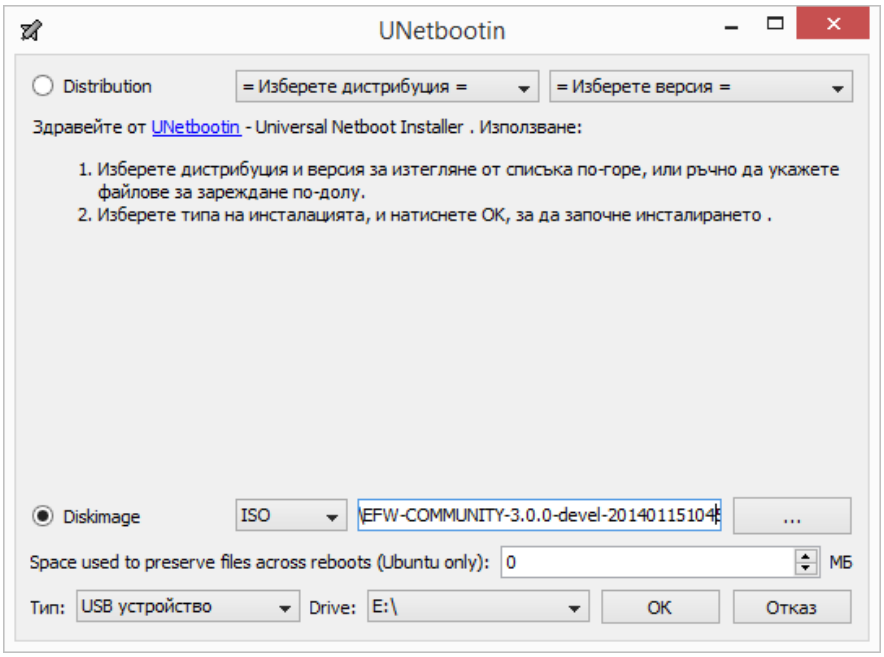
За да инсталирате EFW CE е необходимо да изтеглите актуалната (препоръчително стабилна версия) от сайта на Endian и да запишете ISO файла на CD/DVD или да подготвите USB Flash памет от която да стартирате инсталацията

Въпреки, че първият вариант за инсталиране през CD/DVD е по-лесен има случай, когато на системата няма оптично устройство и тогава се налага да се стартира boot процеса от USB.

Един от най-лесните начини да подготвите bootable ISO за стартиране от USB под Microsoft Windows е да използвате програмата UNetbootin[53], като процедурата за създаване на инсталационен USB носител за EFW CE е:

1. Стартира се UNetbootin (възможно е да се изисква потвърждение на административния достъп до програмата);
2. Посочва се опцията “Diskimage”;
3. Въвежда се пътя до ISO файла или чрез бутона “…” се посочва ISO файла на EFW CE;
4. Типа на устройството трябва да бъде USB, а в полето “Drive” се избира буквата (например E:\);
5. Натиска се бутона OK.
6. **Внимание: след натискане на OK не се изисква повторно потвърждение!**

[53] unetbootin.sourceforge.net



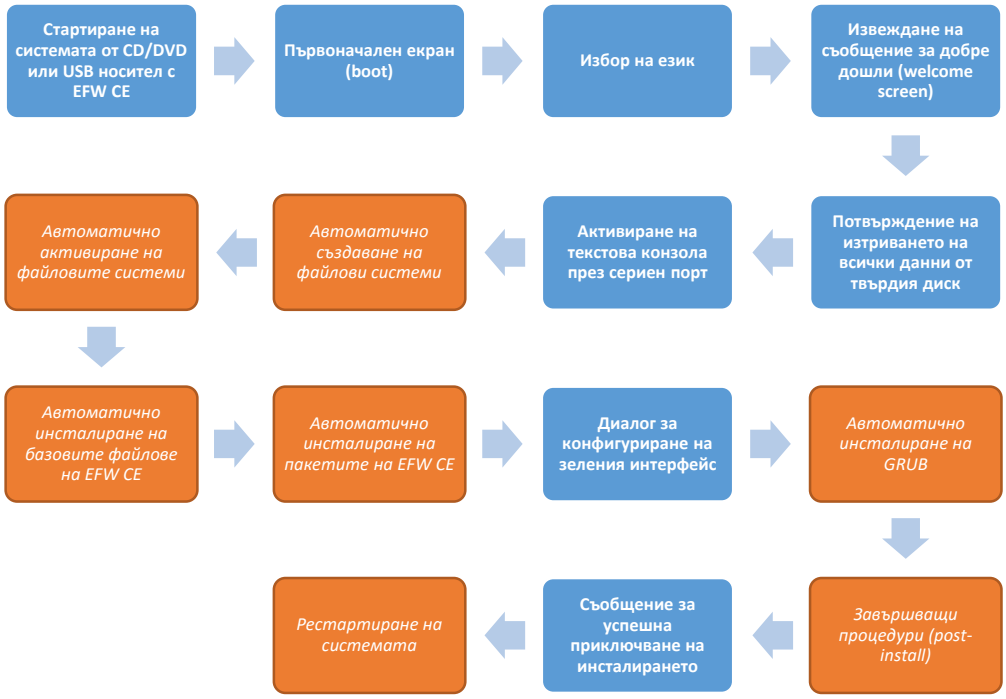
Фиг. 5.1 Интерфейс на Unetbootin

Създаването на инсталационния USB носител отнема няколко минути в зависимост от производителността на системата.

Основни етапи на инсталиране на EFW CE

Основните етапи при инсталирането на EFW CE са показани на фиг. 5.2 и включват:

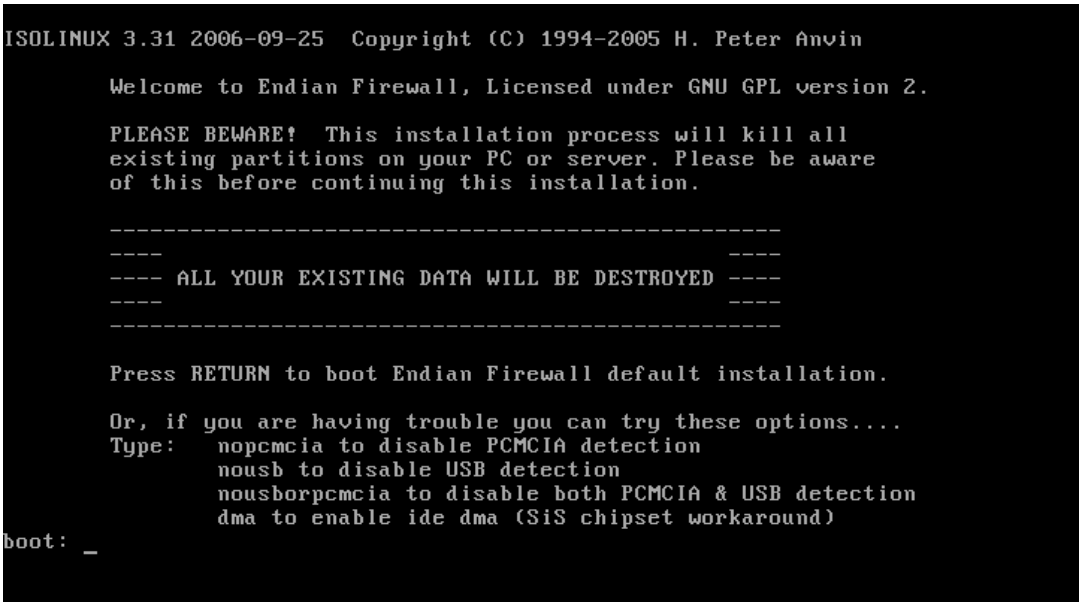
1. **Стартиране на системата от CD/DVD или USB носител с EFW CE;**
2. **Първоначален екран (Boot);**
3. **Избор на език за инсталационната процедура;**
4. **Извеждане на съобщение за добре дошли (Welcome Screen);**
5. **Потвърждение на изтриването на всички данни от твърдия диск;**
6. **Въпрос за активиране на текстова конзола през сериен порт;**
7. *Автоматично създаване на файлови системи;*
8. *Автоматично активиране на файловите системи;*
9. *Автоматично инсталиране на базовите файлове на EFW CE;*
10. *Автоматично инсталиране на пакетите на EFW CE;*
11. **Диалог за конфигуриране на зеления интерфейс;**
12. *Автоматично инсталиране на GRUB;*
13. *Завършващи процедури (post-install);*
14. **Съобщение за успешно приключване на инсталирането;**
15. *Рестартиране на системата.*



Фиг. 5.2 Етапи на инсталиране на EFW CE. В син цвят са показани етапите, при които е необходимо въвеждане на данни или потвърждаване, а в оранжев – автоматизираните стъпки

Първоначален екран (boot)

След зареждане на EFW CE от CD/DVD или от USB носител се извежда първоначалния екран, на който има предупреждение, че инсталационната процедура ще изтрие всички данни от диска.



Фиг. 5.3 Начален екран на инсталационната процедура на EFW CE

При натискане на бутонът “Enter” инсталационната процедура продължава. Опционално може да се изключи автоматичното търсене на PCMCIA модули чрез въвеждане на “nopcmcia” и натискане на “Enter”. Ако се въведе “nousb” не се прави проверка и разпознаване на свързаните към системата USB модули. Командата “nousborpcmcia” деактивира разпознаването както на USB, така и на PCMCIA модулите. Последният наличен опционален вариант за стартиране на

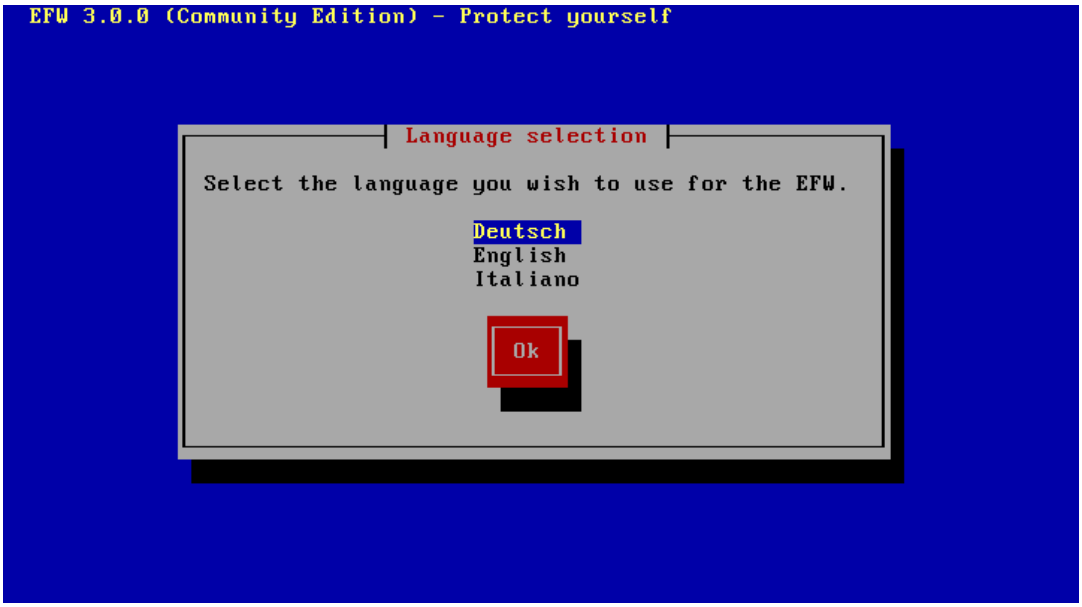
инсталацията на EFW CE е чрез “dma”, което задава специфична функционалност за някой чипове на компанията SiS.

Най-често инсталационната процедура се стартира без допълнителни параметри чрез натискане на “Enter”.

Избор на език

Ако инсталационните модули на EFW CE се заредят успешно се извежда текстово меню за избор на език за инсталацията, като към момента опциите са немски, английски или италиански език.

Чрез стрелките от клавиатурата може да се посочи желаната опция и с “Enter” да се премине към следващия етап от инсталацията. При натискане на бутонът “Tab” или комбинацията ”Alt+Tab” се преминава между отделните елементи на менюто (списък с езици и “OK”). При натискане на “Enter” или “Space” направеният избор се потвърждава.

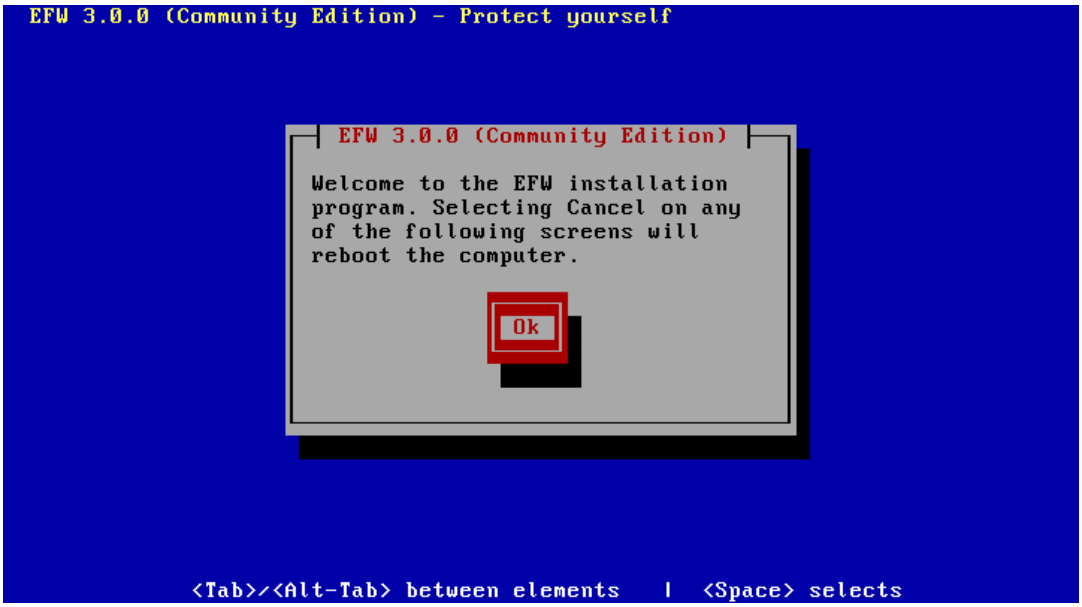


Фиг. 5.4 Избор на език

Извеждане на съобщение за добре дошли (welcome screen)

След изборът на език за инсталацията се извежда кратко съобщение, че при избор на “Cancel” от всяка възможна позиция компютърът ще се рестартира.

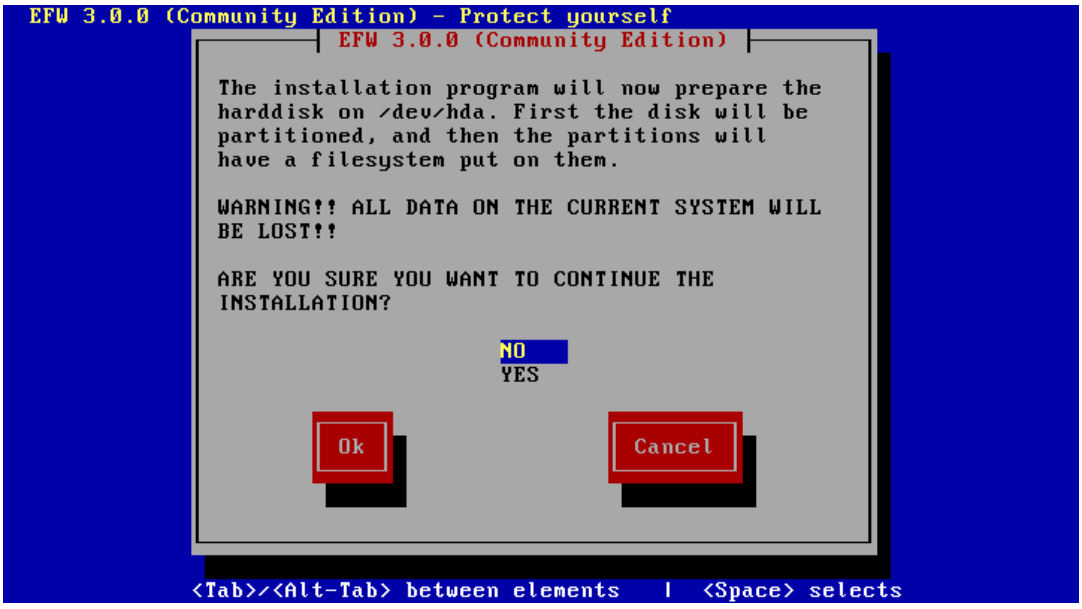
В зависимост от етапа на инсталацията, на който е избрана опцията “Cancel” е възможно да не са нанесени никакви промени по файловата система или вече съдържанието на твърдия диск да е изтрито.



Фиг. 5.5 Начално съобщение на инсталационната процедура на EFW CE

Потвърждение на изтриването на всички данни от твърдия диск

Една от особеностите на EFW CE е, че по време на инсталационната процедура всички файлови системи от твърдия диск се изтриват и се създават нови. Предупреждение за това се извежда непосредствено при стартирането на инсталацията, както и след съобщението за добре дошли.

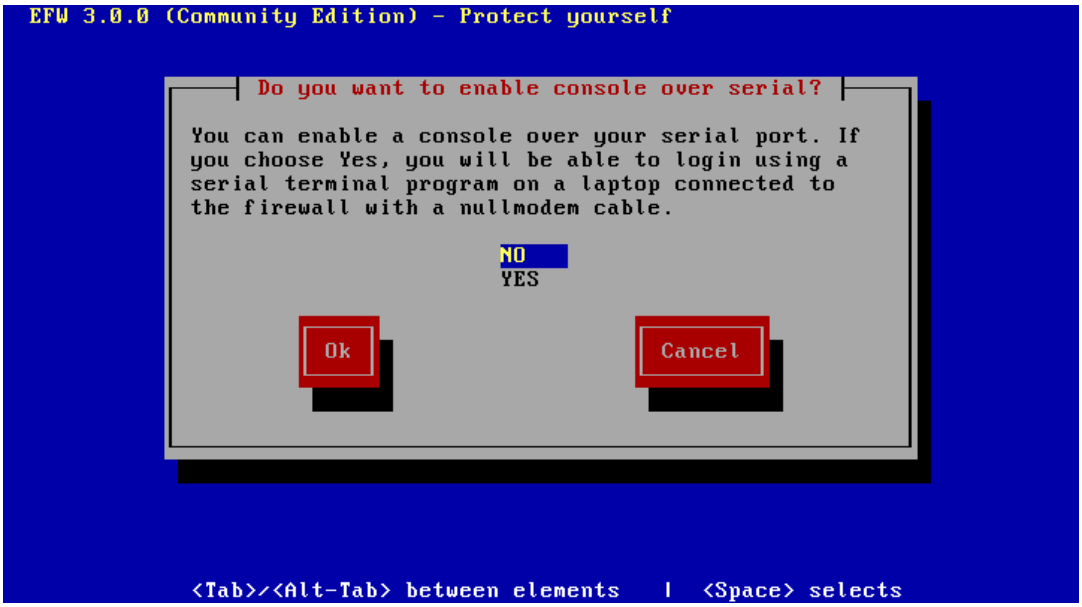


Фиг. 5.6 Предупреждение за изтриване на файловите системи от твърдия диск

Необходимо е де се потвърди изтриването на съдържанието на диска и създаването на новите файлови системи. Ако се избере опцията “No” инсталационната процедура се прекратява и не се извършват промени. Ако се посочи “Yes” се генерират новите файлови системи и инсталацията продължава.

Активиране на текстова конзола през сериен порт

По време на инсталацията на EFW е възможно да бъде активирана специална конзола, която чрез пренос на данни през сериен порт с RS-232 позволява да се наблюдава процеса на инсталиране.



Фиг. 5.7 Активиране на серийна конзола през RS-232

Тази опция е подходяща за специализираните хардуерни устройства на Endian, и при инсталиране на стандартен компютър или сървър рядко е необходима.

За да се активира посочената функционалност трябва да се избере опцията “Yes”, в противен случай при избор на “No” инсталационната процедура продължава без серийната конзола. Ако се избере “Cancel” инсталацията на EFW се прекратява.

След този диалог EFW създава новите файлови системи и започва да опира необходимите системни файлове и пакети. Това отнема няколко минути, като необходимото време зависи от използваните хардуерни компоненти.

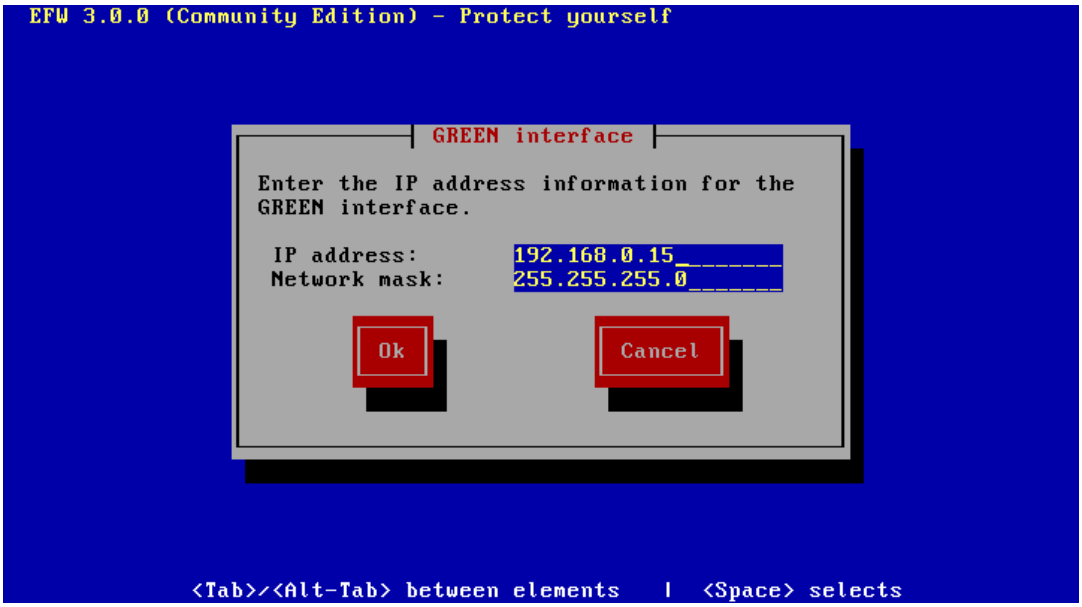
Диалог за конфигуриране на зеления интерфейс

След диалогът за активиране на серийна конзола инсталационната процедура продължава с етапа на копирането на системните файлове и пакети. Ако това действие премине успешно се извежда диалог за задаване на IPv4 адрес на зеления интерфейс (GREENIP).

Важно е да се отбележи, че по подразбиране след приключване на инсталирането по подразбиране графичният Web базиран интерфейс на EFW може да бъде достъпен само от зелената зона.

По подразбиране адресът е 192.168.0.15 с маска 255.255.255.0, като неговата стойност може да бъде променена в зависимост от логическото адресиране, използвано в конкретната мрежа.

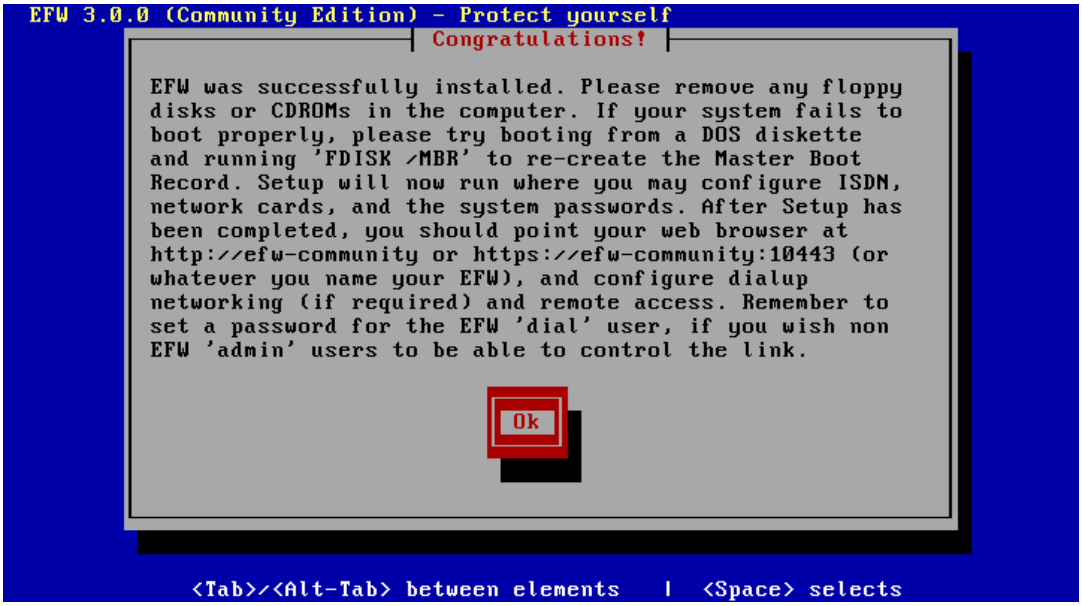
След потвърждаването на GREENIP инсталационната процедура конфигурира и записва GRUB и стартира няколко окончателни скрипта за завършване на инсталацията.



Фиг. 5.8 Задаване на адрес на зеления интерфейс (GREENIP)

Съобщение за успешно приключване на инсталирането

Ако всички етапи при инсталационната процедура на EFW CE са преминали без да възникнат грешки се извежда съобщение за успешно приключване. При натискане на "OK" системата се рестартира и се зарежда EFW CE.

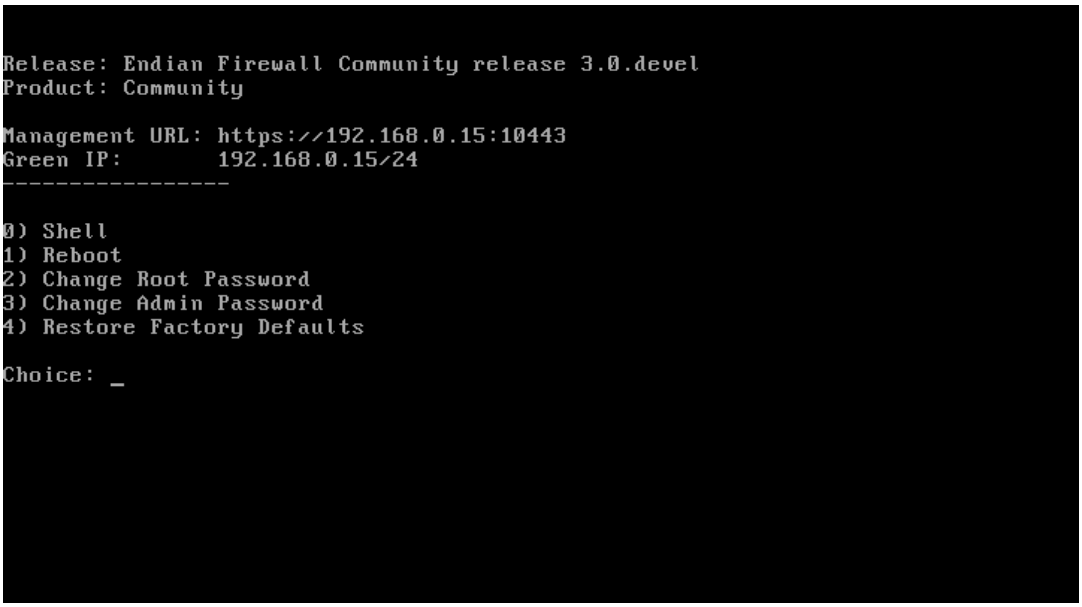


Фиг. 5.9 Съобщение за успешно приключване на инсталационната процедура на EFW CE

Първоначално конфигуриране на EFW CE през GUI

След приключване на инсталационната процедура и рестартирането на системата се зарежда EFW CE.

Ако към системата е свързан монитор се визуализират данните, за версията на EFW, типа, както и адресът, от който е достъпен графичния интерфейс.



Фиг. 5.10 Конзолно меню на EFW CE

От информацията на фиг. 5.10 може да се види версията на EFW – 3.0 developer, типа на продукта - CE, както и GREENIP и IP адресът и порта за достъп до графичния интерфейс за конфигуриране - по подразбиране https://192.168.0.15:10443.

Допълнителни опции, които са достъпни само през клавиатурата на системата са:

- **0** – shell достъп (разгледан е по-късно в книгата);
- **1** – рестартиране на EFW системата;
- **2** – промяна на паролата на потребителя root;
- **3** – промяна на паролата на потребителя admin;
- **4** – възстановяване на настройките по подразбиране.

При физически достъп до EFW е възможно неоторизирано лице да промени паролите на най-важните потребители. Препоръчително е функциите на това меню да бъдат редуцирани (начина е описан на сайта на Endian)

Стартиране на помощника

За да се извърши първоначалното конфигуриране на EFW CE е необходимо след определянето на GREENIP и завършването на инсталацията да се отвори браузър и страницата да се пренасочи към:

https://GREENIP:10443

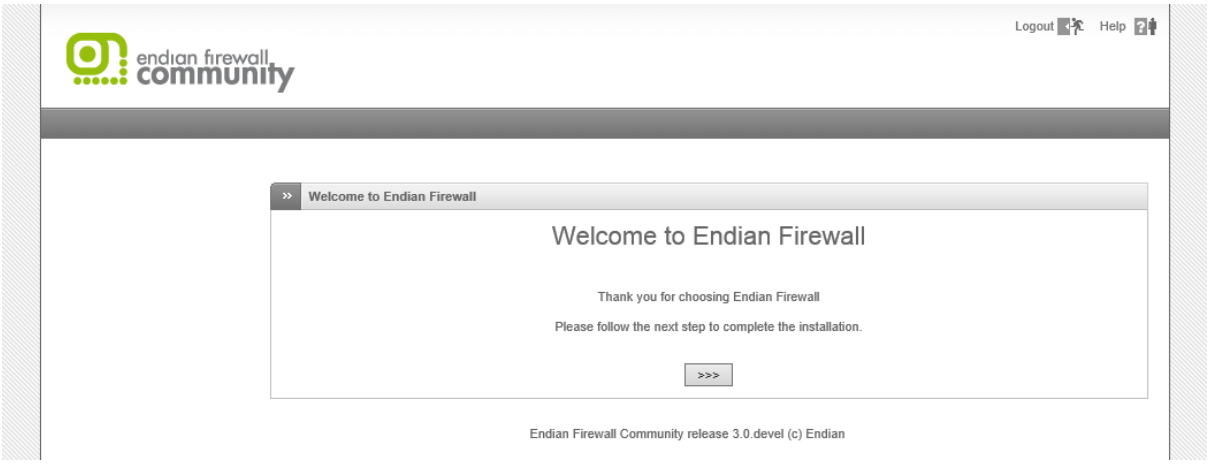
По подразбиране адресът е:

https://192.168.0.15:10443

Необходимо е да се потвърди зареждането на страницата, поради използваният цифров сертификат не е издаден от надеждно CA, а е генериран от Endian (self signed).

Съобщение за добре дошли

След потвърждението на сертификата, страницата, която се визуализира от помощника за първоначално конфигуриране извежда кратко благодарствено съобщение и след натискане на бутона в долната и част се преминава към следващия етап.

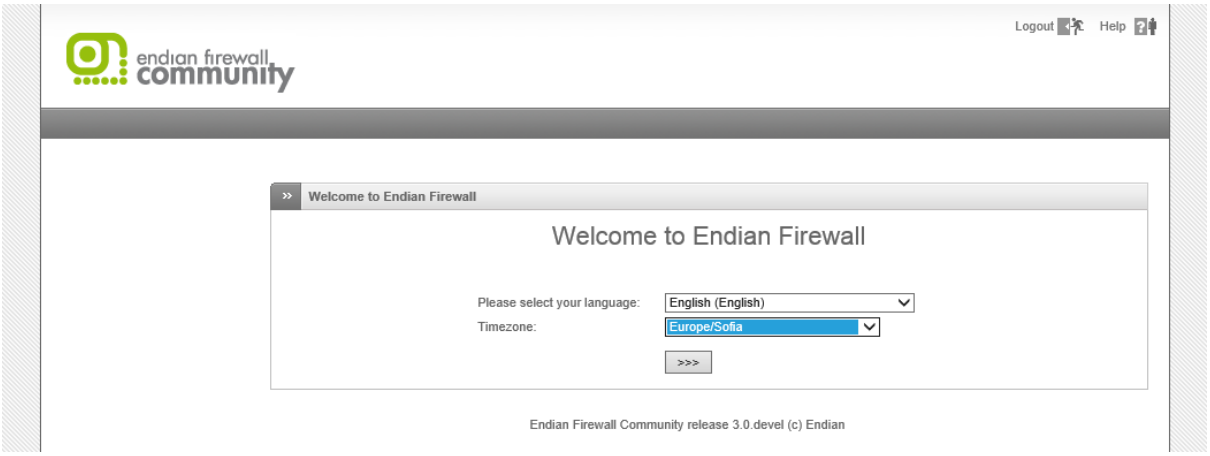


Фиг. 5.11 Начална web страница на помощника за първоначално конфигуриране на EFW CE

Избор на часова зона

Вторият етап от първоначалното конфигуриране на EFW CE е изборът на език и часова зона. За разлика от инсталационната процедура, при която има три възможни езика административният интерфейс е преведен на редица езици, сред които към момента не е наличен български.

След като се изберат необходимите параметри чрез натискане на бутона се преминава към следващата стъпка.



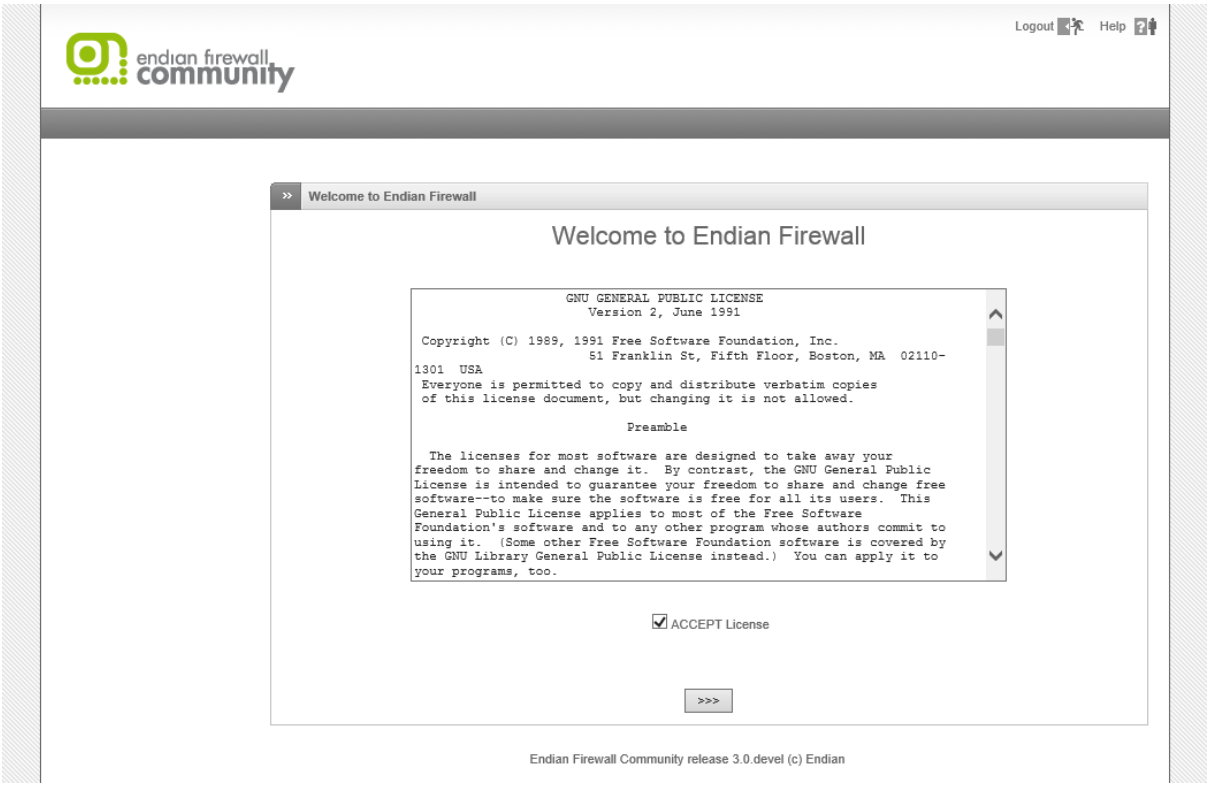
Фиг. 5.12 Избор на език на интерфейса на EFW CE и часова зона

Лицензионно споразумение

Необходимо е да бъде прието лицензионното споразумение, което е на база на GNU GPL[54].

Отметката под текста на споразумението показва, че то е било прието и след натискане на бутона се преминава към следващия етап.

[54] www.gnu.org/licenses/gpl-3.0.en.html

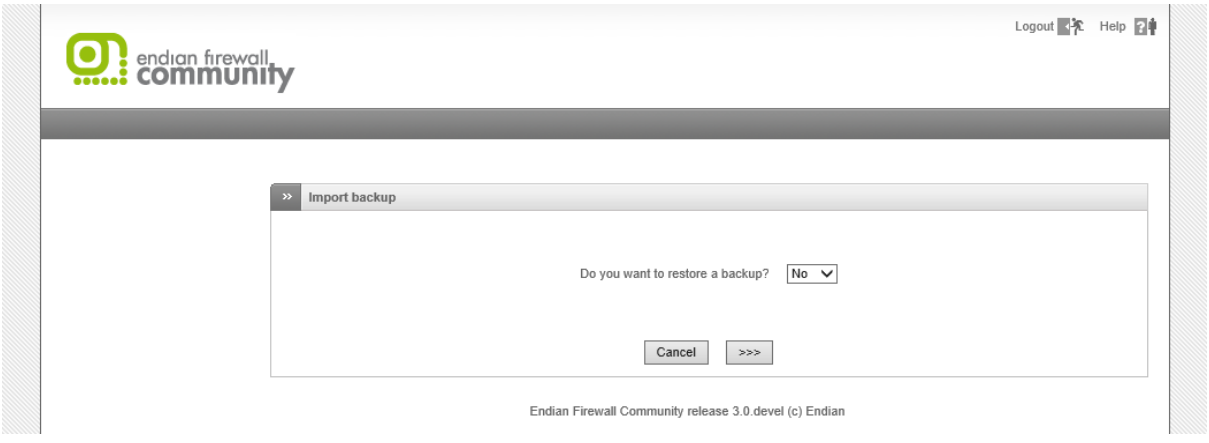


Фиг. 5.13 Лицензионно споразумение за употреба на EFW CE

Възстановяване на настройки

EFW предоставя функционалност, чрез която всички системни настройки да бъдат експортиране като резервно копие, което да бъде записано във файл. След инсталацията на продукта от интерфейса за първоначално конфигуриране има възможност да бъде посочен файл, съдържащ настройките от друга система, които да бъдат импортирани. Това най-често се налага при подмяна на хардуерно устройство или при подготовка на тестове.

Ако се избере опцията “Yes” (възстановяване на настройки) следващите етапи са свързани с тази процедура. За да се конфигурира нова EFW CE система се избира “No” и се натиска бутона в долната част на страницата.



Фиг. 5.14 Възстановяване на настройки от интерфейса за първоначално конфигуриране

Въвеждане на пароли

Следващият етап от първоначалното конфигуриране на EFW CE е да бъдат попълнени паролите за потребителите root и admin. **Задължително е да се използват сигурни пароли, въпреки, че системата може да използва и ненадеждни!**

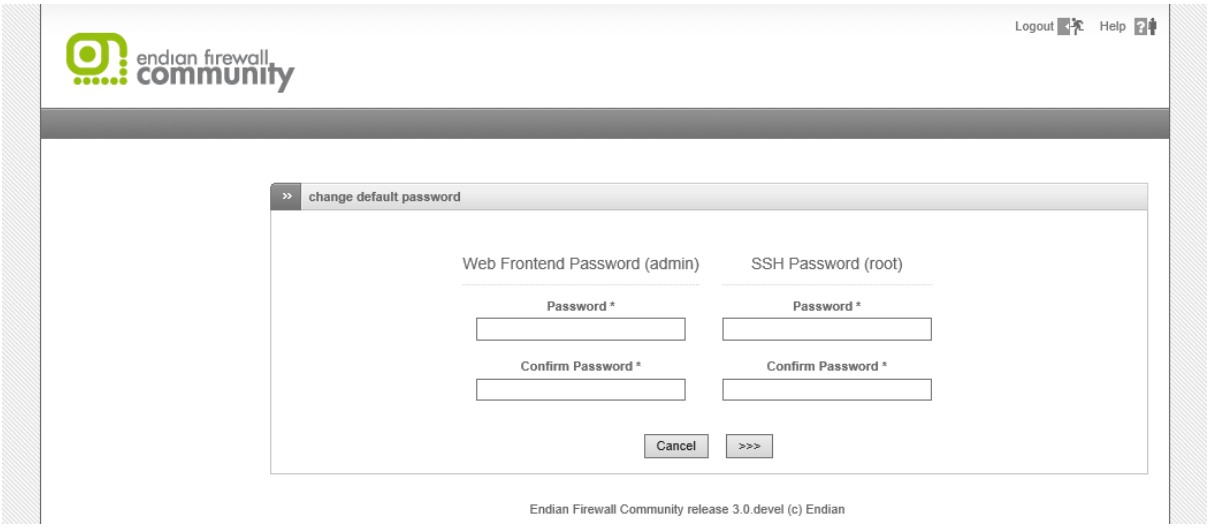
Сигурността на дадена система е толкова силна колкото е нейното най-слабо звено. За съжаление в много случаи това се оказват пароли за административен достъп, като изборът на надеждна парола е труден и е препоръчително да се спазват следните правила:

- Паролата да е с дължина поне 8 символа;
- Да не съдържа потребителското име, истинското име на потребителя, рождени дати, името на компанията, цяла речникова дума и друга лесна за предполагане информация;
- Ако текущата парола се променя – да бъде със значителни разлики спрямо предходните;
- Да съдържа елементи от следните групи – малки и големи букви, символи и цифри, а при възможност и интервали.

Дори дадена парола да отговаря на гореописаните препоръки пак може да се окаже сравнително проста и лесно да бъде открита чрез речников подход, например “pAssw0rd_1”.

Необходимо е да се зададат пароли за следните потребители:

1. admin – осигуряване на достъп до графичния интерфейс на EFW CE;
2. root – използва се за достъп до конзолата и Linux операционната система.



Фиг. 5.15 Въвеждане на пароли за потребители admin и root

Потвърждаване или промяна на конфигурацията на интерфейсите

Един от най-важните етапи на първоначалното конфигуриране на EFW е да се зададат правилните настройки на мрежовите интерфейси, като този етап е в 8 отделни стъпки:

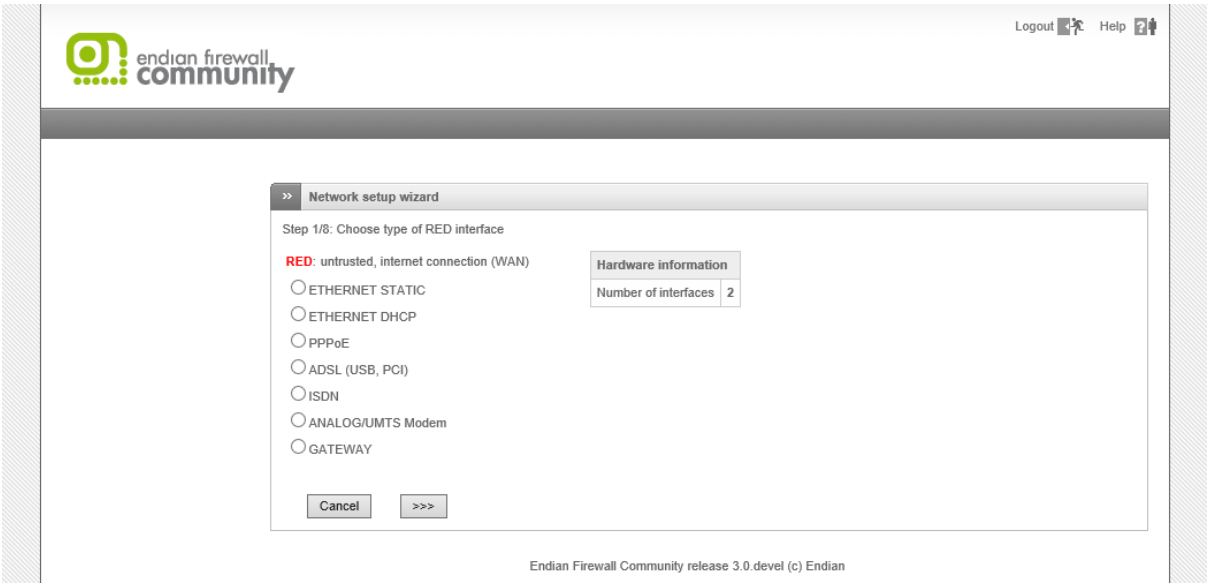
1. Конфигуриране на червения интерфейс;
2. Избор на зони;
3. Конфигурация на GREENIP;
4. Конфигуриране на достъпа до Интернет (REDIP);
5. Конфигуриране на DNS сървъри;
6. Електронна поща на администраторите;
7. Активиране на конфигурацията;

8. Обобщение.

При първият етап конфигурирането на червеният интерфейс изисква първоначално да се определи използвания протокол, като опциите са:

- Ethernet static – Ethernet протокол и статичен IPv4 адрес;
- Ethernet DHCP – Ethernet протокол и IPv4 адрес, получен от DHCP сървър;
- PPPoE – енкапсулиране на PPP рамки в Ethernet рамки;
- ADSL (USB, PCI) – свързаност през DSL модем, включен на PCI или USB порт;
- ISDN свързаност;
- Analog/UMTS Modem – аналогов или 3G модем;
- Gateway – рядко използвана специална опция, при която защитната стена има само един единствен интерфейс.

В таблицата в дясната част на страницата се вижда броя на наличните за системата мрежови интерфейси.



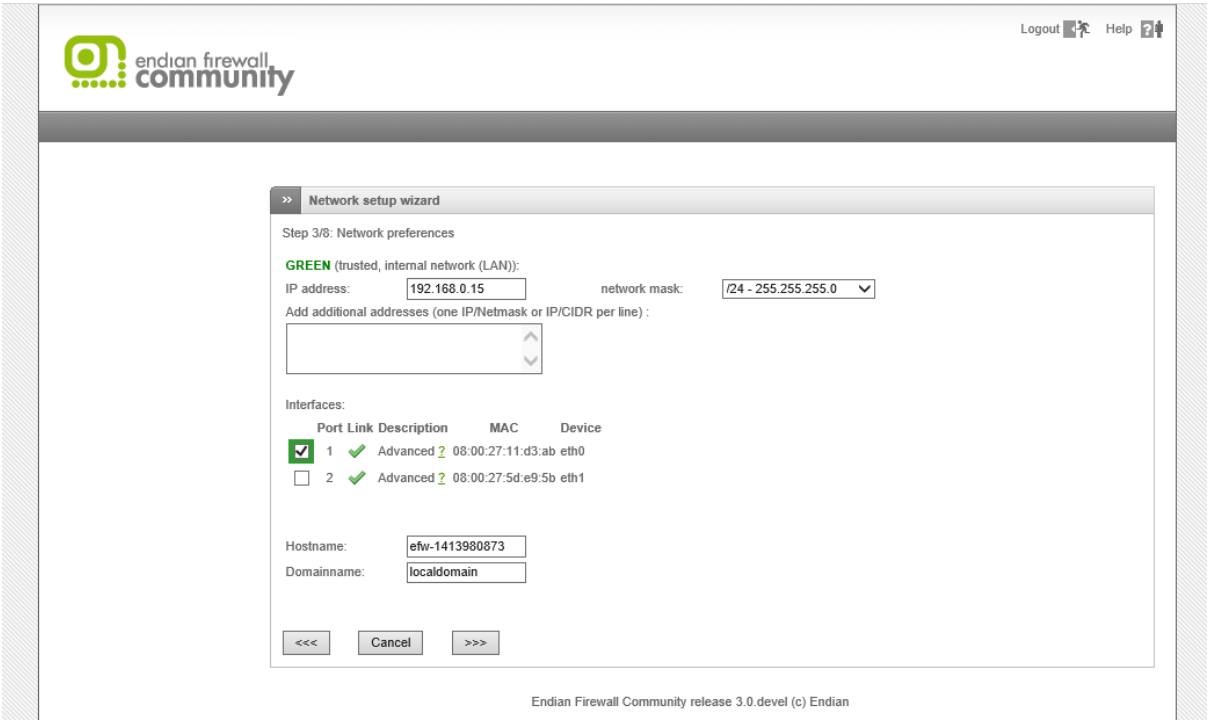
Фиг. 5.16 Конфигуриране на мрежови интерфейси – стъпка 1 от 8. Избор на типа на адресиране на REDIP

Вторият подетап при първоначалното конфигуриране на EFW CE е да се определят свързаните към системата зони. При наличие на два мрежови интерфейса е възможно да се зададат зелена и червена зона, а при повече карти и синя и оранжева.

Фиг. 5.17 Конфигуриране на мрежови интерфейси – стъпка 2 от 8. Определяне на зони

След определянето на наличните зони се преминава към подетапа за задаване на конфигурацията на зеления интерфейс, като за целта се попълват:

- IP address – IPv4 адрес на интерфейса, който по подразбиране е 192.168.0.15 (зададен по време на инсталацията на EFW);
- Network mask – мрежовата маска за адресът, аналогично е зададена в процеса на инсталиране и по подразбиране е 255.255.255.0;
- Допълнителни адреси – опционално поле за допълнителни адреси, като всеки ред съдържа комбинацията IPv4 адрес и префикс[55] или IPv4 адрес и маска;
- Interface – посочва се към кой физически интерфейс е свързана логическата зелена зона. За ориентиране при избора на интерфейса може да се използва посочения MAC адрес;
- Hostname – името на устройството;
- Domainname – името на домейна.



Фиг. 5.18 Конфигуриране на мрежови интерфейси – стъпка 3 от 8. Конфигуриране на адреса на GREENIP

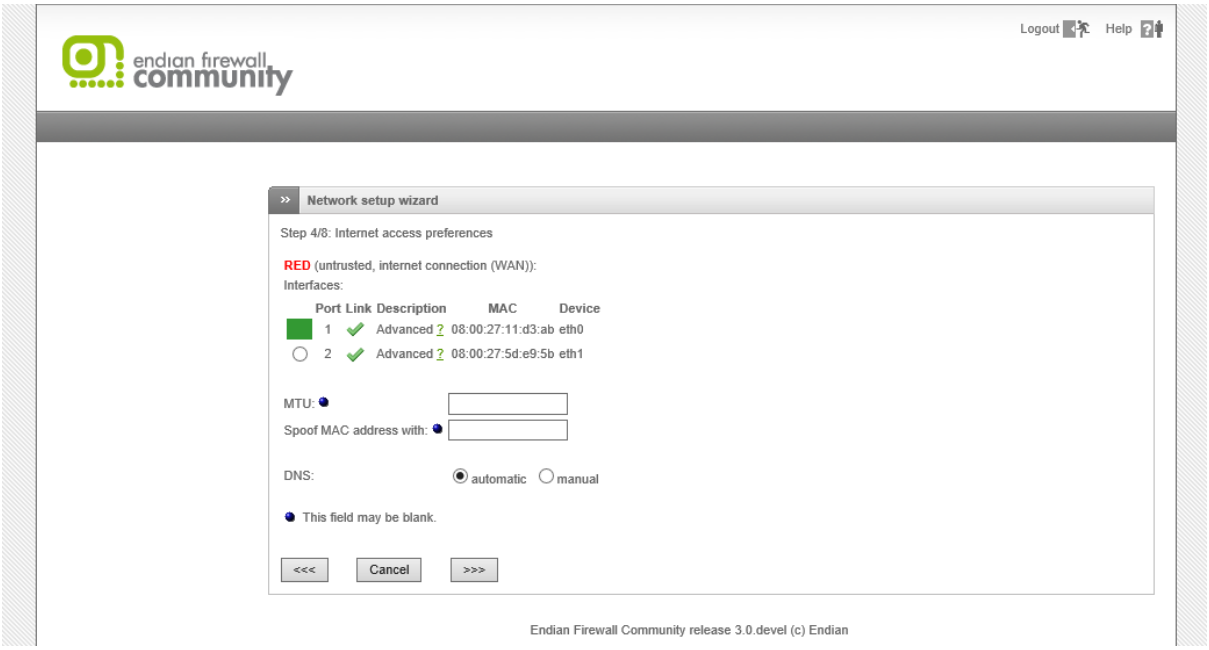
След приключване на конфигурацията на зеления интерфейс се преминава към определяне на комуникационния адаптер за достъп до външните мрежи и Интернет.

Избира се мрежовият адаптер или устройството, към което е свързана червената зона и опционално се попълват MTU (Maximum Transmission Unit) и дали да се използва подменен (Spoofed) MAC адрес.

Конфигурирането на DNS сървърите може да стане автоматично или с посочени адреси.

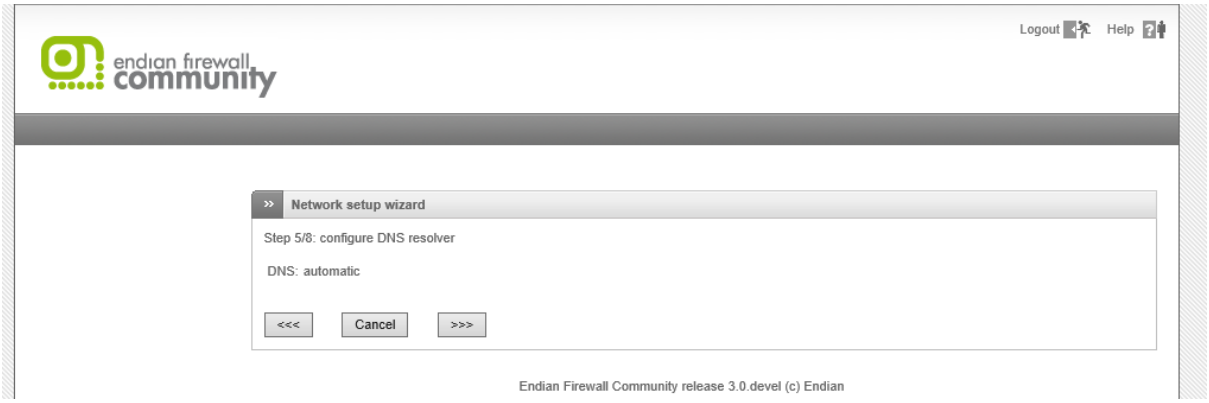
[55] Записване на маската чрез комбинация “/” и брой битове, например 255.255.255.0 се представя като CIDR префикс /24.

Важно е да се отбележи, че съдържанието на тази страница зависи от предходния избор на конфигурацията на червения интерфейс и на фигурата е посочен пример при използване на DHCP.



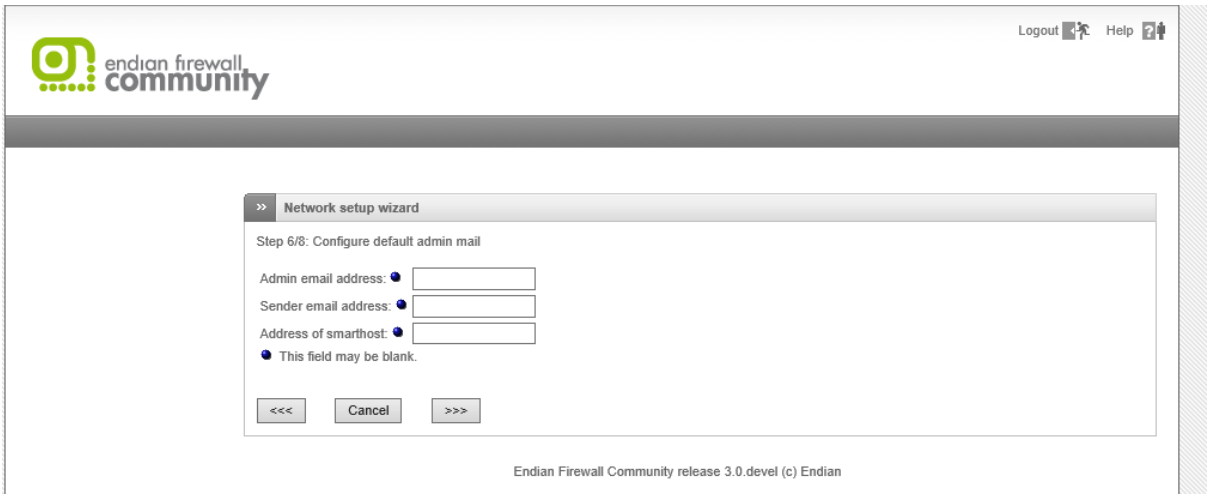
Фиг. 5.19 Конфигуриране на мрежови интерфейси – стъпка 4 от 8. Конфигуриране на адреса на REDIP

Петият подетап е въвеждане на информация за DNS сървъри (възможно е тази конфигурация да е налична от предходната стъпка).



Фиг. 5.20 Конфигуриране на мрежови интерфейси – стъпка 5 от 8. Конфигуриране на DNS

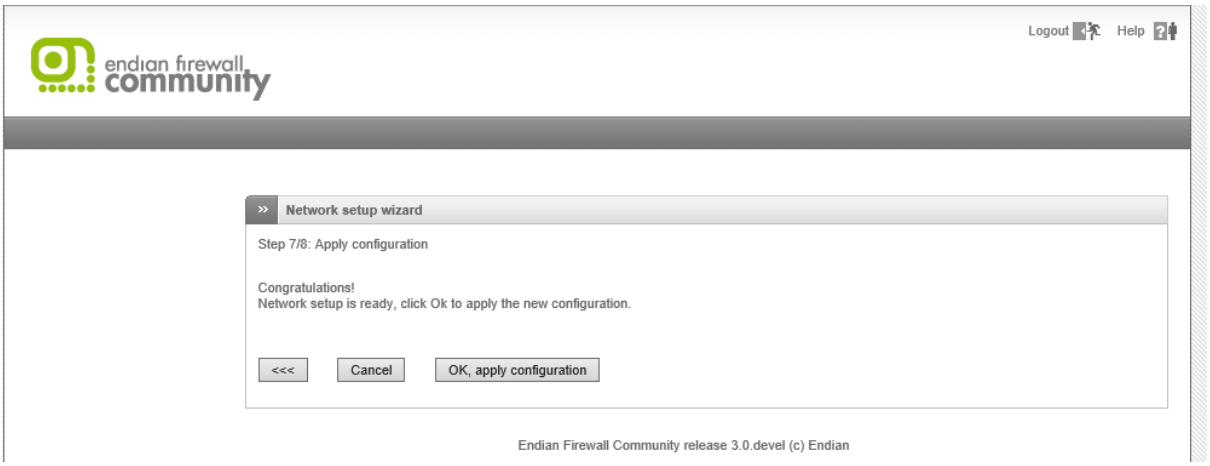
Шестият подетап е опционален и предоставя възможност за задаване на адрес за електронна поща на администратора и smarthost.



Фиг. 5.21 Конфигуриране на мрежови интерфейси – стъпка 6 от 8. Конфигуриране на адреса за електронна поща за изпращане на съобщения към администратора

Седмата стъпка е направената мрежова конфигурация да бъде потвърдена и активирана. Това става при натискане на бутона “OK, apply configuration”.

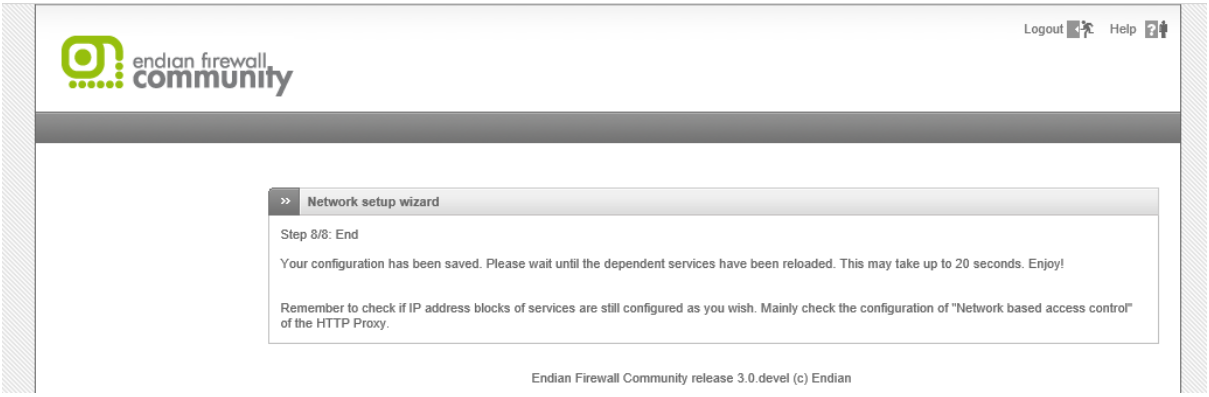
Ако направената конфигурация не е точна може да се натисне бутона “Cancel”, който отхвърля всички параметри или да се премине към предходен етап чрез бутона “<<<”.



Фиг. 5.22 Конфигуриране на мрежови интерфейси – стъпка 7 от 8. Потвърждение за използване на зададените параметри и активиране на настройките

След потвърждаването на конфигурацията е необходимо да се изчака (20-30 секунди) за да се рестартират и заредят отделните системни услуги, които EFW използва. Времето за рестартирането на демоните отново зависи от производителността на системата.

След този етап процедурата по първоначално конфигуриране се счита за завършена.



Фиг. 5.23 Конфигуриране на мрежови интерфейси – стъпка 8 от 8. Завършване на конфигурирането на мрежовите интерфейси на EFW и на първоначалните настройки

Проблем с конфигурация на пароли

По време на първоначалното конфигуриране на EFW CE ако зададените пароли са по-къси от 8 символа не се правят промени и паролата на потребителя (root или admin) остава подразбиращата се “endian”.

При тази ситуация, ако след приключване на помощника за първоначално конфигуриране администратора няма достъп до интерфейса на EFW CE е необходимо паролата да бъде променена от системната конзола.

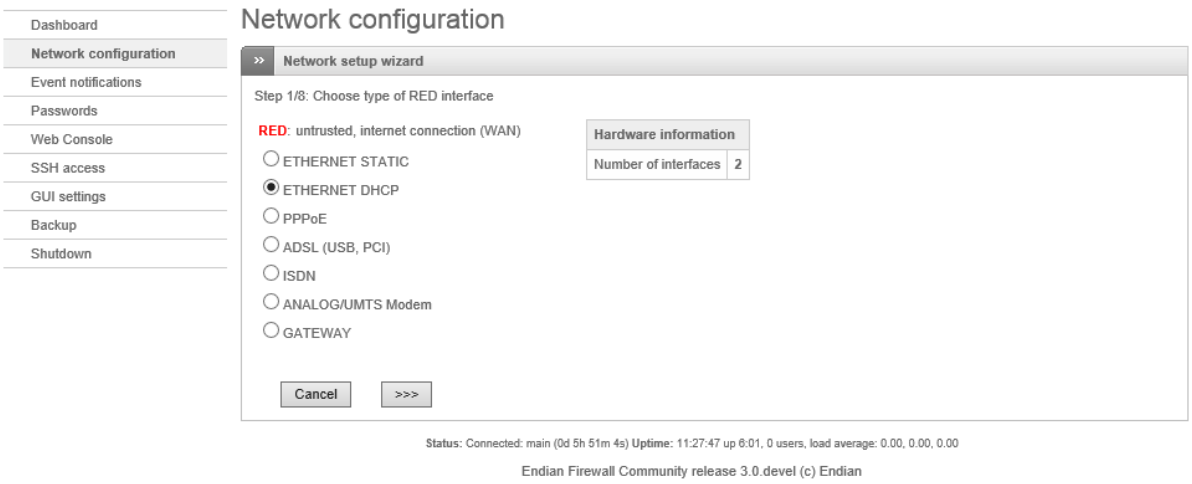
Меню “System”

След успешното приключване на процедурата по първоначалното конфигуриране на EFW CE страницата в браузъра се пренасочва към т.нар. “Dashboard” (виж фиг. 4.8), която е част от менюто “System”, към което са включени:

- Dashboard – основна информация за системата;
- Network configuration – помощник (wizard) за конфигуриране на мрежовите интерфейси, аналогичен на използвания в инсталационната процедура;
- Event notification – известяване за настъпили събития;
- Passwords – конфигуриране на паролите за достъп до конфигурационните интерфейси на EFW;
- Web Console – специална web конзола която предоставя достъп до функциите на операционната система;
- SSH access – конфигуриране на достъпа до EFW през SSH;
- GUI settings – параметри на Web интерфейса;
- Backup – настройки и създаване на резервни копия;
- Shutdown – рестартиране или изключване на устройството.

Подменю “Network configuration”

В подменюто “Network configuration” е възможно да се стартира помощника за конфигуриране на интерфейсите на системата, който е аналогичен на използвания в процеса на първоначално конфигуриране.



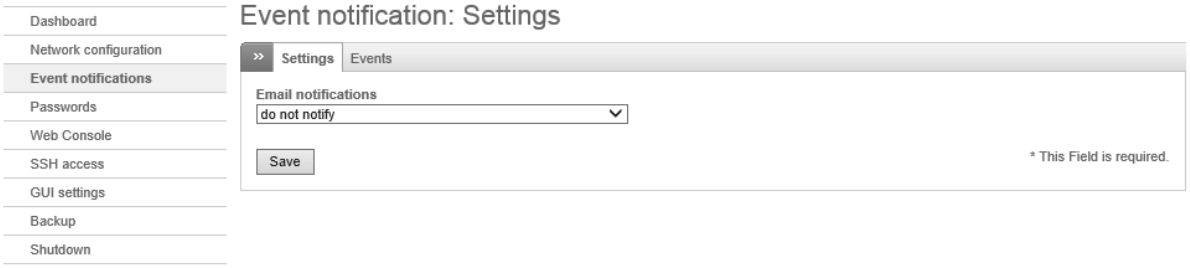
Фиг. 5.24 Помощник “Network configuration”, стартиран от “Dashboard”

Подменю “Event notification”

Изпращането на съобщения при настъпване на критични ситуации или на грешки по време на работата на системите за защита е изключително важно. Уведомяването на администраторите чрез електронна поща е често срещан подход, който позволява максимално бързо да се вземат необходимите мерки и да се коригират проблемните ситуации.

В подменюто “Event notification” са включени възможностите за избор на метод за уведомяване при проблем, който може да бъде:

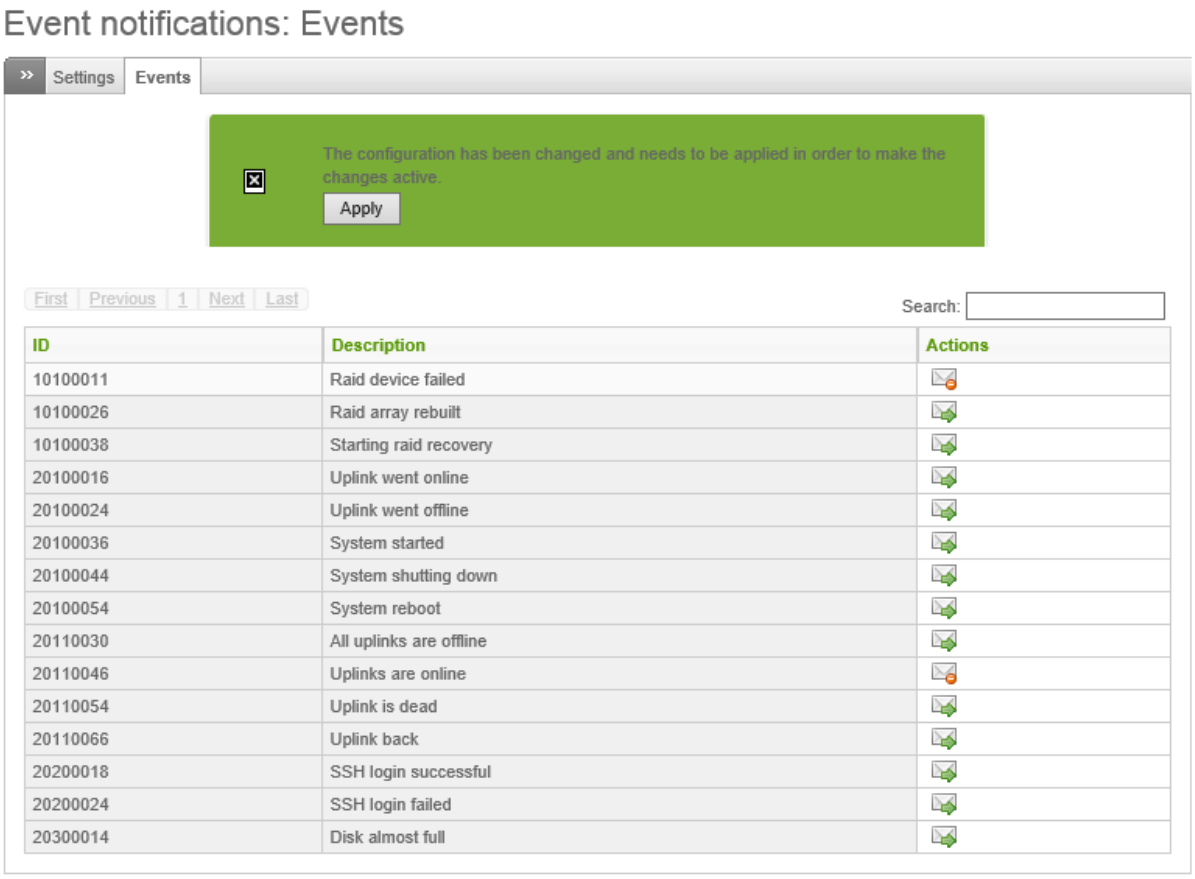
1. Do not notify – не се изпращат съобщения;
2. Notify using default email address – при проблем се изпраща съобщение на посочените в първоначалното конфигуриране на системата адреси за електронна поща;
3. Notify using custom email address – при проблем съобщението се изпраща на посочените на страницата на това подменю адреси.



Фиг. 5.25 Конфигуриране на начина на уведомяване на администраторите при възникване на проблеми по време на работата на EFW CE

Втората част на страницата от менюто “Event notification” е “Events” и от нея може да се посочат, кои събития да генерират съобщение за уведомяване на администраторите. За да се промени действието е необходимо да се натисне върху иконата, в колоната Actions. Когато са конфигурирани всички необходими параметри се натиска бутона “Apply” (в горната част на страницата) и промените се активират.

Бутонът “Apply” е наличен само ако са били извършени конфигурационни промени.



Фиг. 5.26 Конфигуриране на събития, които да генерират съобщения за уведомяване на администраторите

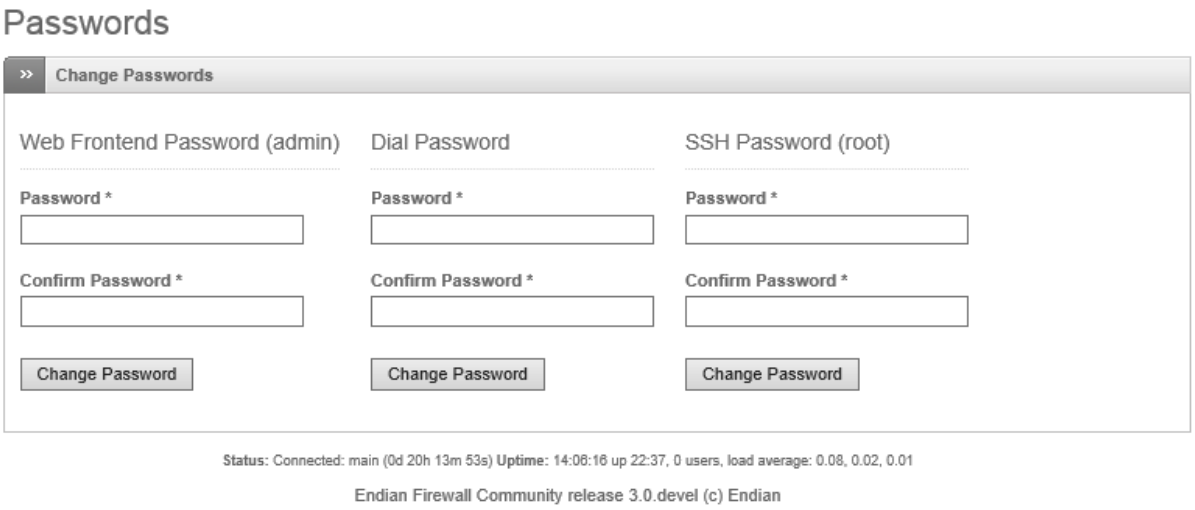
Подменю “Passwords”

Промяната на паролите на основните потребители на EFW CE (admin, SSH, Dial) може да бъдат извършена от подменюто “Passwords”. Новата парола на дадения потребител се въвежда и потвърждава. След натискане на бутона “Change Password” паролата се променя.

Възможно е да се конфигурират следните пароли:

1. Web Frontend Password – парола за потребителя admin, която осигурява достъп до графичния интерфейс за наблюдение и конфигуриране на EFW CE;
2. Dial password – парола, използвана при достъп през модем или друг специализиран канален протокол;
3. SSH Password – парола за достъп до EFW CE през протокола SSH, която променя тази на потребителя root.

Препоръчително е винаги да спазвате съветите за надеждни пароли!

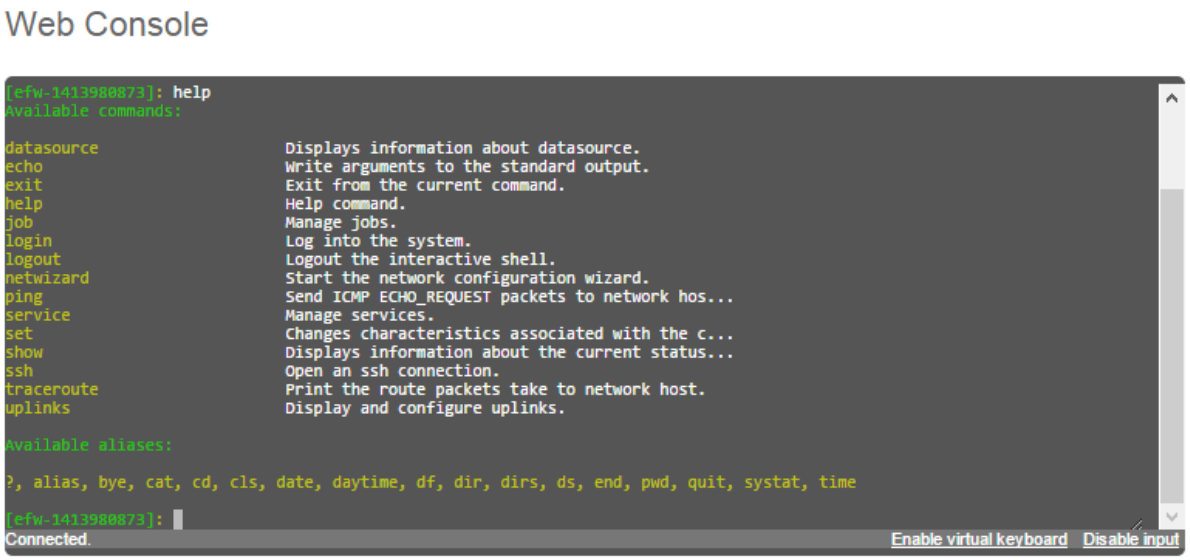


Фиг. 5.27 конфигуриране или промяна на пароли за admin, dial и root

Подменю “Web console”

Една полезна функционалност, която е достъпна от подменюто на страницата “Dashboard” е “Web console”. Тя позволява през браузър да се работи със системната конзола на EFW CE. По този начин може лесно да се анализира работата на системата, както и да се получи достъп до Linux shell. Достъпни са всички команди от конзолата, която EFW CE предоставя при директна работа с клавиатура и монитор.

При необходимост може да се активира виртуална клавиатура (Enable virtual keyboard), която да се използва за въвеждане на текста, както и да се забрани входния поток от данни към web конзолата (Disable input). Важно е да се отбележи, че опцията “Disable input” не се отнася за виртуалната клавиатура.



Фиг. 5.28 Достъп до “Web console” от страницата “Dashboard”

При някои браузъри може да се получи проблем с извеждането на текста[56] и е препоръчително или да използвате тази страница в съвместим режим (compatibility mode) или да проверите с кой браузър няма проблеми.

Подменю "SSH Access"

От това подменю може да се конфигурира отдалечен достъп до EFW CE през подсигурения криптиран протокол SSH. По подразбиране SSH достъпът е изключен и може да бъде активиран чрез натискане на бутона "Enable Secure Shell Access".

Фиг. 5.29 Конфигуриране на отдалечен SSH достъп до EFW CE

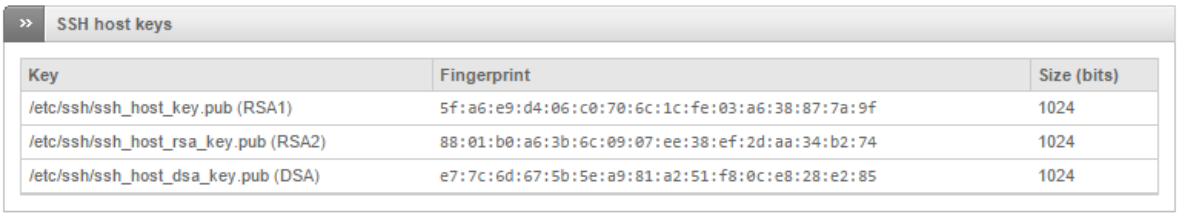
Допълнителните опции, които могат да бъдат зададени са:

1. Support SSH protocol version 1 – активиране на поддръжката на SSH v1, което **не се препоръчва** поради факта, че тази версия вече не се обновява и има редица технологични пропуски в сигурността. Към момента се използва SSH v2;
2. Allow TCP forwarding – разрешава други протоколи да бъдат тунелирани през SSH. Тази функционалност е удобна в определени ситуации, например, ако дадено устройство, което се намира в зелената зона трябва да бъде достъпно от отдалечена система в червената зона през несигурния протокол Telnet. Разрешаването на Telnet от червената зона към устройството не е препоръчително, поради липсата на шифриране на трафика. Ако пакетите се тунелират през SSH ще се получи необходимата функционалност, като цялата обмяна на данни от външното устройство до EFW CE ще е защитена;
3. Allow password based authentication – автентификацията през SSH е чрез комбинация от потребителско име и парола (зададена в подменюто "Passwords");
4. Allow public key based authentication – активира се автентификация с публични ключове, които трябва да са налични в директория /root/.ssh/authorized_keys на EFW CE.

За да се запишат и активират направените настройки се използва бутона "Save".

[56] Застъпващи се редове или размазани символи

В най-долната част на страницата е таблица “SSH host keys”, съдържаща генерираните от OpenSSH ключове.



Фиг. 5.30 Списък с генерираните от OpenSSH RSA ключове

Подменю “GUI Settings”

Изборът на език за графичния потребителски интерфейс може да се извърши от подменюто “GUI Settings”. Опцията “Display hostname in window title” активира извеждането на host името на EFW CE в заглавната част на прозореца на браузъра.

Фиг. 5.31 Избор на език за графичния интерфейс на EFW CE

Подменю “Backup”

Една изключително важна задача при поддръжката на сървъри, мрежови и устройства и други системи е създаването на резервни копия на данните, операционните системи и на конфигурациите.

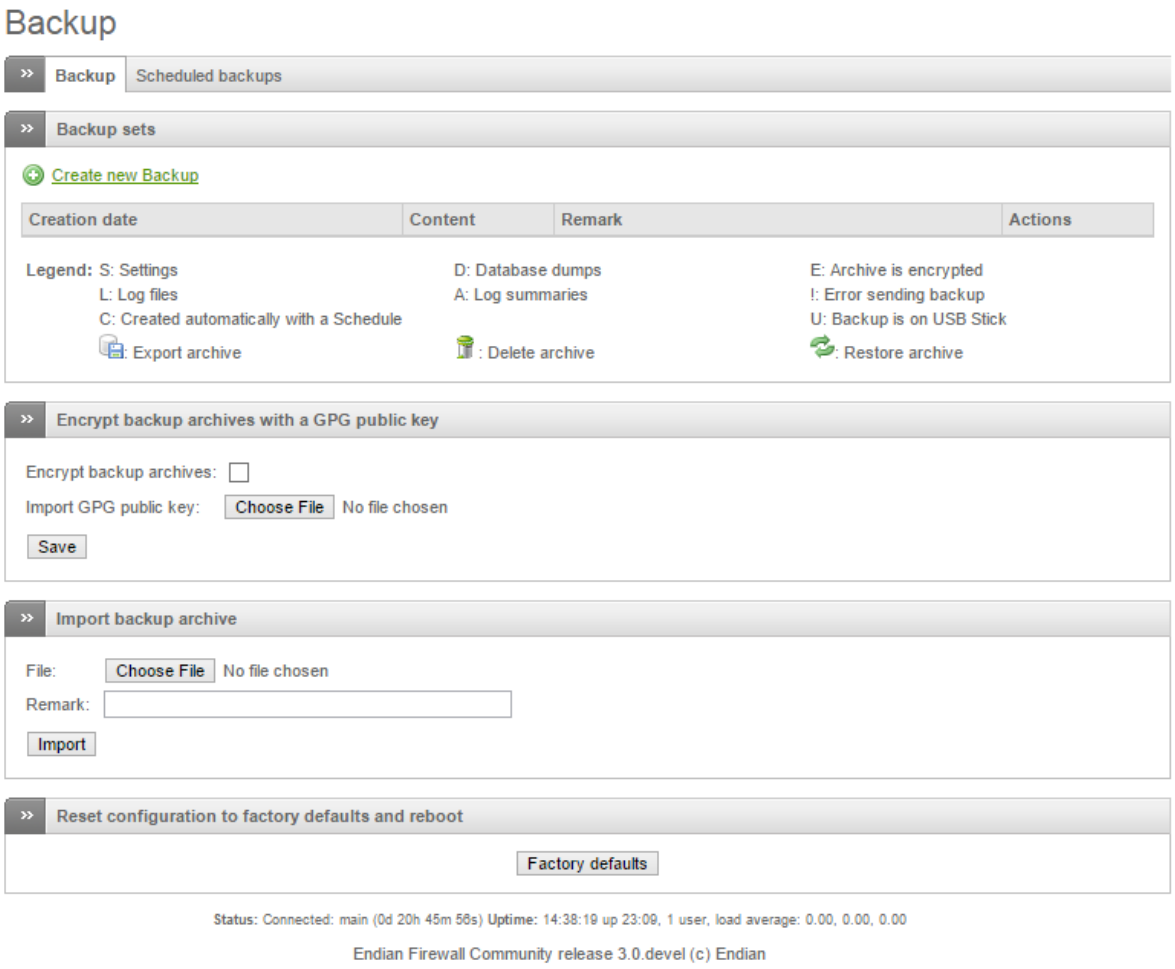
Подменюто “Backup” на страницата “Dashboard” съдържа всички необходими функции за създаване на резервни копия, периодично генериране на архиви и възстановяване на системата в случай на нужда.

Резервните копия се съхраняват в архивни файлове (tar.gz), като за тяхното създаване се използват стандартната Linux програма tar и gzip. Файловете от архива могат да бъдат разархивирани чрез tar и аргумент zxf и името на архивния файл. Ако се добави и параметъра v се извежда информация за всеки отделен файл от архива.

Името на архива е уникално и съдържа максимално количество информация, което го прави и сравнително дълго, например:

backup-20141023144326-efw-1413980873.localdomain-settings-db-logs-logarchive.tar.gz

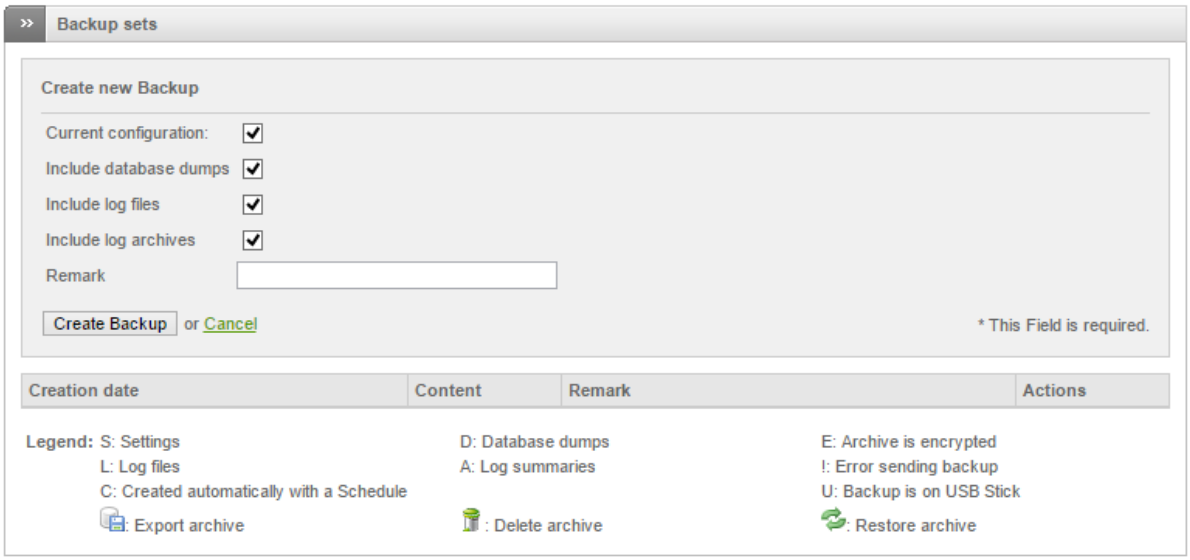
След текстът “backup” е датата и часа на генериране на архива (20141023144326), последвана от името на системата (efw-1413980873.localdomain), а на края на името са включените данни (settings-db-logs-logarchive).



Фиг. 5.32 Страница за създаване и възстановяване на резервни копие на EFW CE

Създаването на ново резервно копие може да се извърши при натискане на връзката “Create new Backup”, като е необходимо след това да се посочат, кои данни да бъдат архивирани:

- Current configuration – текущата конфигурация на EFW CE;
- Include database dumps – съдържанието на използваните бази данни;
- Include log files – съдържанието на файловете с журнали;
- Include log archives – архивираните журнални файлове.
- Remark – опционален текстов коментар за архива.



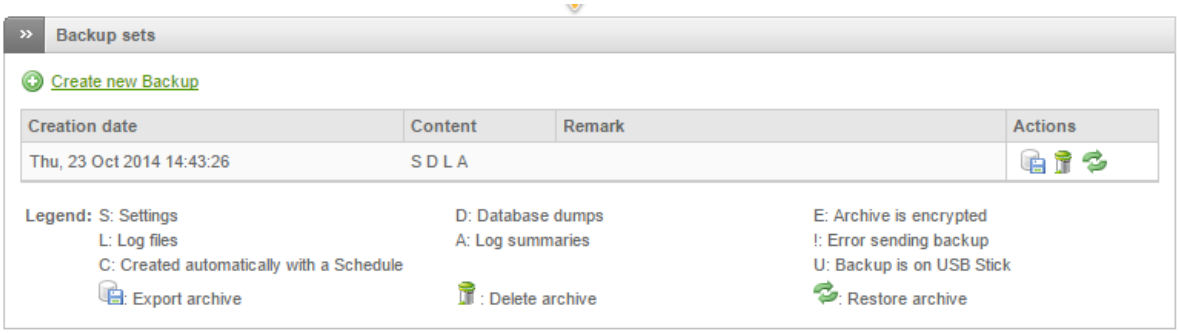
Фиг. 5.33 Избор на данни, които да бъдат включени в резервното копие

За да се стартира процедурата за създаване на архива се натиска бутона “Create Backup”.

Създаването на архивното копие отнема известен период от време, който варира в зависимост от обема на данните и системната производителност.

След приключване на архивирането на посочените данни в списъка с архиви се добавя и новосъздадения. Легендата под списъка показва съдържанието и типа на резервното копие:

- S – съдържа настройките на системата;
- L – включва журналните файлове;
- C – резервното копие е създадено автоматично чрез cron[57] действие;
- D – съдържа базите данни, използвани от EFW;
- A – съдържа обобщените журнали на EFW;
- E – резервното копие е шифрирано;
- ! – грешка при архивирането;
- U – резервното копие е налично на USB памет.



Фиг. 5.34 Списък с резервни копия

Съдържанието на резервното копие се записва в специална директория на EFW и ако е необходимо да се експортира на носител може да се избере първата икона на реда след името

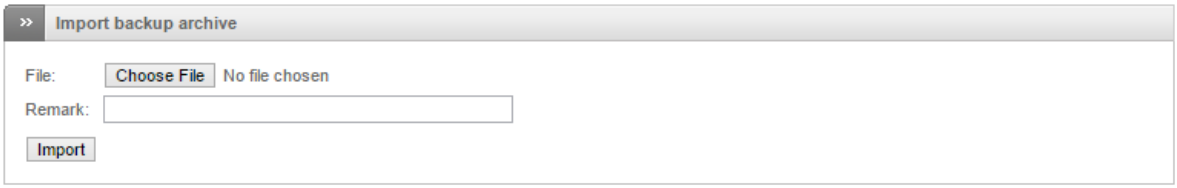
[57] Инструмент за автоматизиране на действия при Unix и Linux

на архива. Изтриването на резервното копие се извършва при натискане на втората икона, а третата икона възстановява системата от архива.

С цел повишаване на сигурността резервното копие може да бъде криптирано. За да се активира тази функционалност е необходимо да се избере опцията “Encrypt backup archives” и да се посочи GPG публичен ключ.

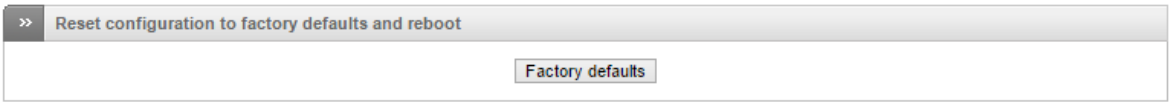
Фиг. 5.35 Шифриране на съдържанието на архивите с резервни копия

Импортирането на външен архив се извършва от секцията “Import backup archive”. Добавените данни се включват в списъка с резервни копия и могат да бъдат управлявани (възстановявани или изтривани) от там.



Фиг. 5.36 Импортиране на архиви, съдържащи резервни копия на данните на EFW

Възстановяването на системата в първоначално състояние (след инсталация) може да се извърши или от конзолата или от секцията “Reset configuration to factory defaults and reboot” при натискане на бутона “Factory defaults”.

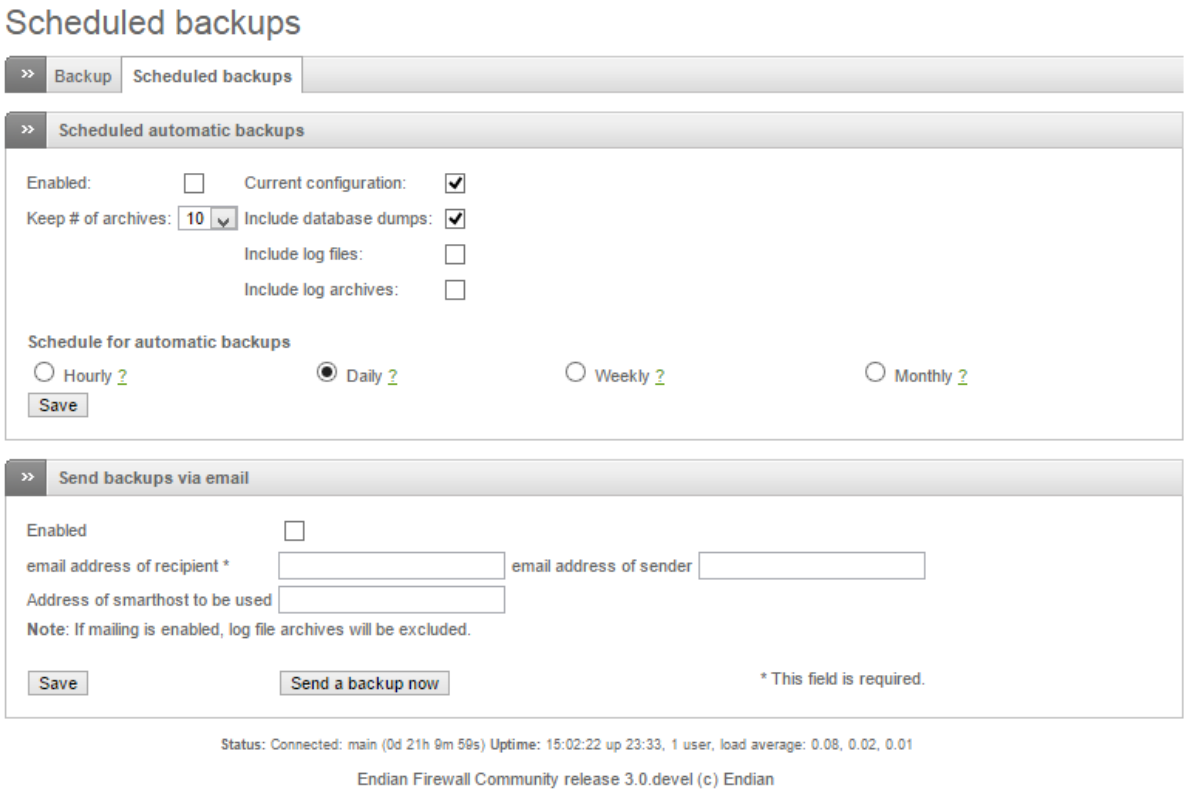


Фиг. 5.37 Възстановяване на системните настройки по подразбиране

Автоматичното генериране на архиви може да се зададе от втората секция на страницата в подменюто “Backup” – “Scheduled backups”, като конфигурационните параметри са:

- Enable – активира автоматичното генериране на резервни копия (cron task);
- Keep # of archives – брой на съхранявани резервни копия. Принципът на работа е цикличен и при надвишаване на посочената бройка (по подразбиране 10) най-стария файл се изтрива;
- Current configuration – архивира се текущата конфигурация на EFW CE;
- Include database dumps – архивира се съдържанието на използваните бази данни;
- Include log files – архивира се съдържанието на файловете с журнали;
- Include log archives – към резервното копие се включват и архивираните журнални файлове;
- Hourly – резервното копие се прави ежечасно (всяка първа минута на всеки нов час);
- Daily – всекидневно архивиране на системата (всеки ден в 1:25);
- Weekly – резервното копие се генерира всяка седмица (всяка неделя в 2:47);

- Monthly – резервното копие се прави ежемесечно (всяко първо число на месеца в 3:52). Активирането на направените настройки се извършва след натискането на бутона “Save”.



Фиг. 5.38 Конфигуриране на периодично създаване на резервни копия на настройките на EFW CE

Възможно е резервното копие да се изпраща по електронна поща, като тази функция се задава от втората секция на страницата. Необходимо е да се въведат адресите на получателя и изпращача, както и да се вземе под внимание, че при тази ситуация журналните файлове не се архивират с цел намаляване на размера на данните.

Подменю “Shutdown”

Последното подменю на страницата “Dashboard” е “Shutdown”, което предоставя следните две възможности:

1. Reboot – EFW CE се рестартира;
2. Shutdown – изключване на EFW CE.

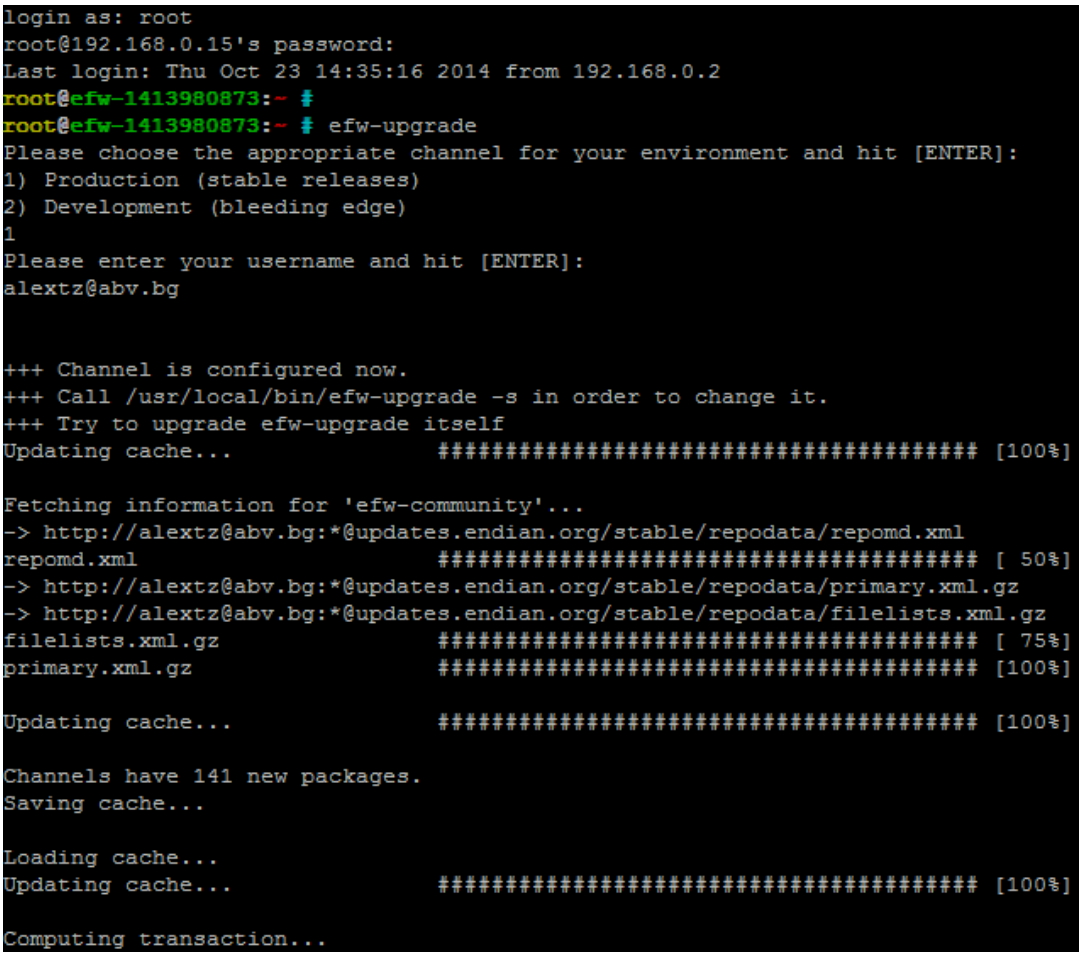
Фиг. 5.39 Меню за изключване или рестартиране на EFW

Обновяване на EFW CE

Обновяването на EFW CE се извършва чрез специален скрипт – efw-upgrade, който може да се стартира през конзолата или чрез SSH достъп до системата.

За да се обнови EFW CE през SSH е необходимо да се извършат следните действия:

1. Свързване към EFW CE през SSH клиент;
2. Включване към системата като потребител root;
3. Стартиране на скрипта efw-upgrade;
4. Избор на канала за обновявания (1 – стабилна версия, 2 – версия в етап на разработка);
5. Въвеждане на потребителско име – адрес за електронна поща (необходима е регистрация на сайта на Endian).
   Обновяването е автоматично и сравнително бързо.



Фиг. 5.40 Обновяване на EFW CE чрез скрипт, стартиран от конзолата

Системна конзола на EFW CE

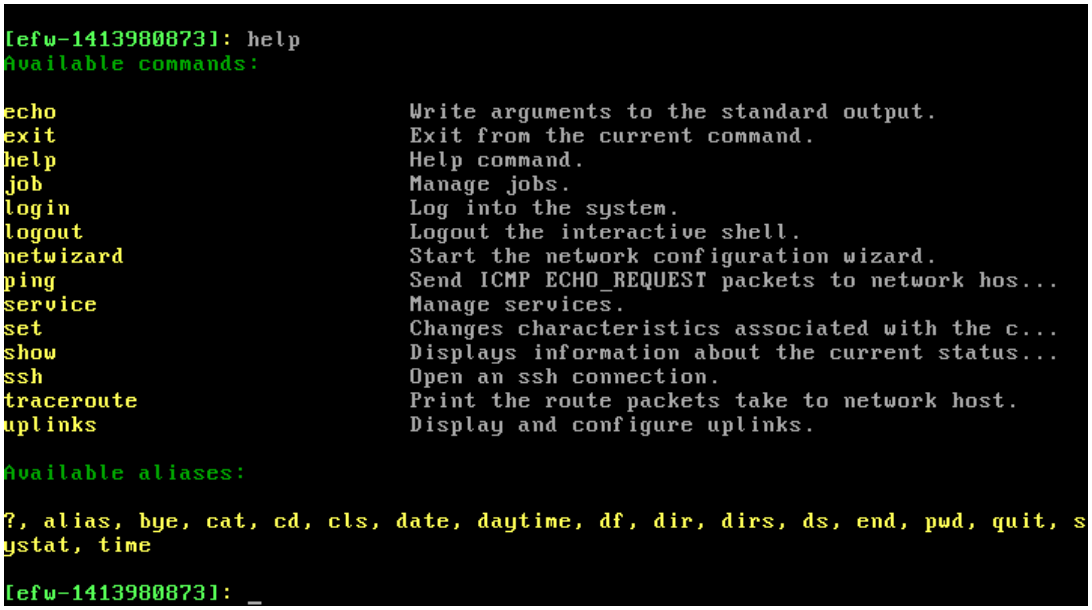
Системната конзола на EFW CE е достъпна от клавиатура и е необходим свързан към системата монитор, което поставя и изисквания за подсигуряване на физическия достъп до устройството и допълнителна конфигурация на използваната Linux дистрибуция.

Опциите, които се предлагат от конзолата са, свързани със shell достъп, рестартиране, промяна на пароли и възстановяване на настройките по подразбиране (factory defaults) на EFW.

За да се подсигури конзолата може да се редактира файла /usr/sbin/efw-console, който е скрипта, визуализиращ менютата. Най-лесният начин за тяхното изключване е да се направи промяна на масива ACTIONS.

Shell

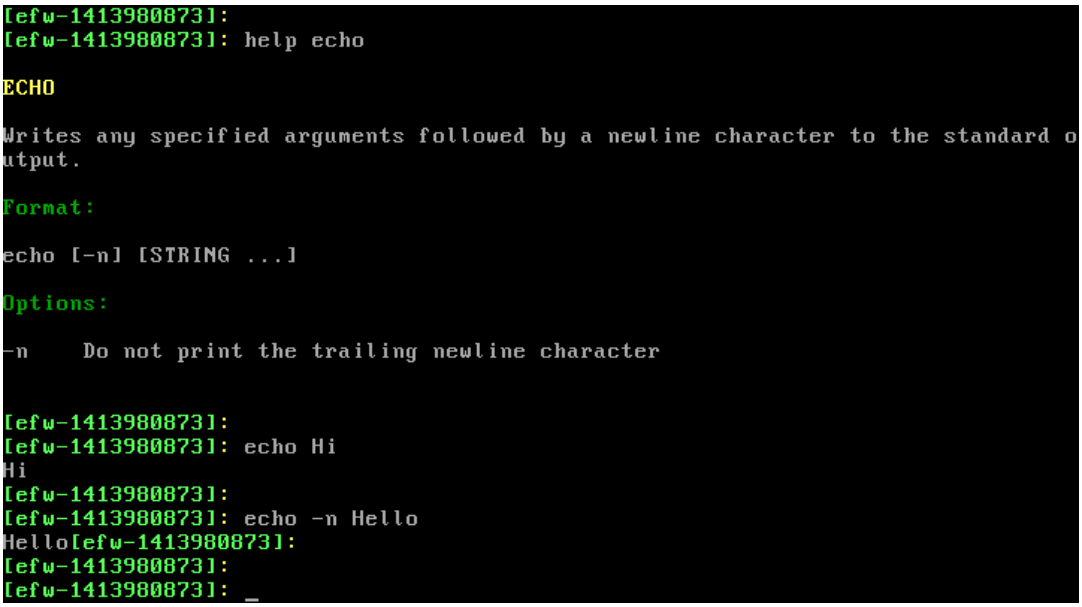
Достъп до специалния shell на EFW CE може да се получи след избор на опция 0 от основното конзолно меню. След стартиране на shell се извежда информация за системата и Prompt, който съдържа името на устройството (например [efw-1413980873]:). Shell интерфейсът е текстово базиран и необходимата функционалност се получава след въвеждане на команда и съответни аргументи.



Фиг. 5.41 Команди в конзолата на EFW CE

За да се види списък с основните команди може да се въведе “help” (фиг. 5.41):

- echo – изписва подадените аргументи към stdout;
- exit – изход от текущата команда;
- help – извеждане на информация за командата и нейните аргументи, например “help show”;
- job – управление на системните задачи;
- login – свързване към shell на операционната система (Linux);
- logout – изход от shell;
- netwizard – интерактивно текстово базирано конфигуриране на интерфейсите на EFW CE;
- ping – изпращане на ICMP Echo-request с цел проверка на свързаност;
- service – управление на услугите;
- set – промяна на настройките на текущата сесия;
- show – извеждане на информация за текущия статус и други параметри, свързани с конфигурацията и работата на EFW CE;
- ssh – стартиране на SSH сесия към отдалечено устройство;
- traceroute – проследяване на пътя на пакетите към отдалечена система;
- uplinks – информация за и конфигуриране на достъпа към външните мрежи и доставчиците.

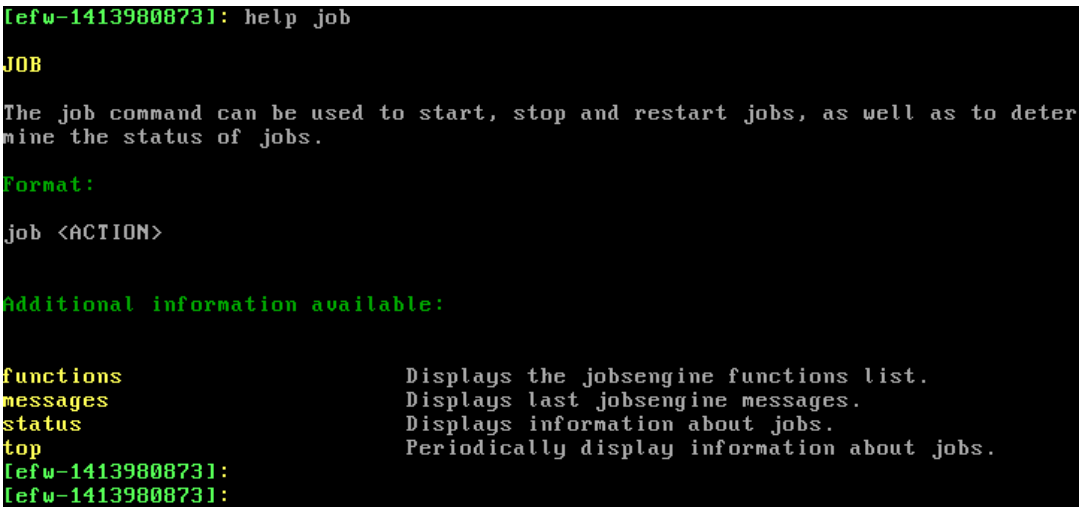


Фиг. 5.42 Извеждане на помощ за определена команда и употреба на echo

Командата echo препраща аргумента към стандартния изходен поток stdout. Ако се добави параметъра –n символа за нов ред не се препраща (фиг. 5.42).

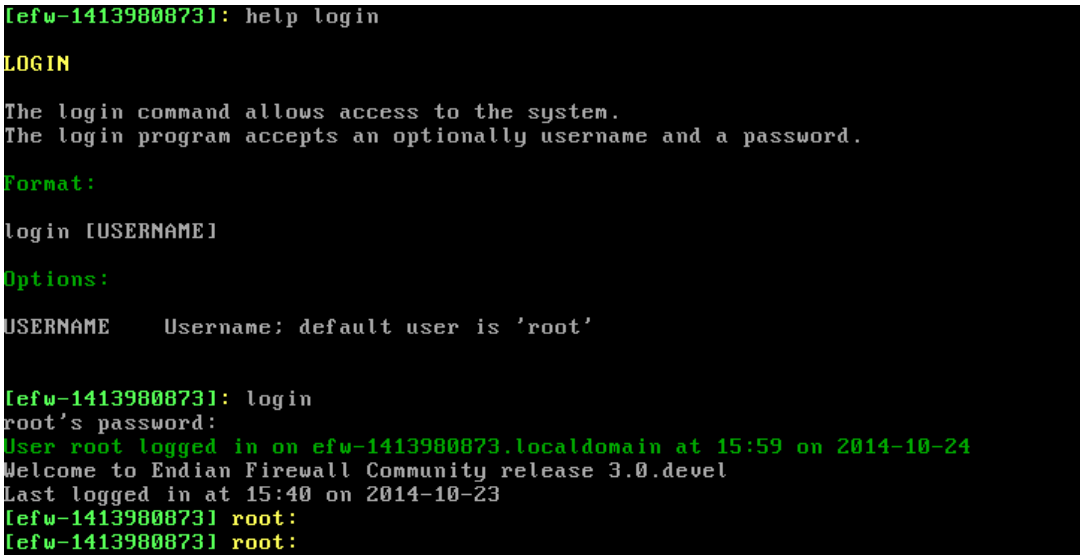
Job се използва за стартиране, спиране и рестартиране на задачи, като аргументите са:

- functions – извеждане на списъка със задачи;
- message – информация за съобщенията, свързани със задачите;
- status – информация за задачите;
- top – периодично наблюдение на задачите.



Фиг. 5.43 Аргументи за командата job

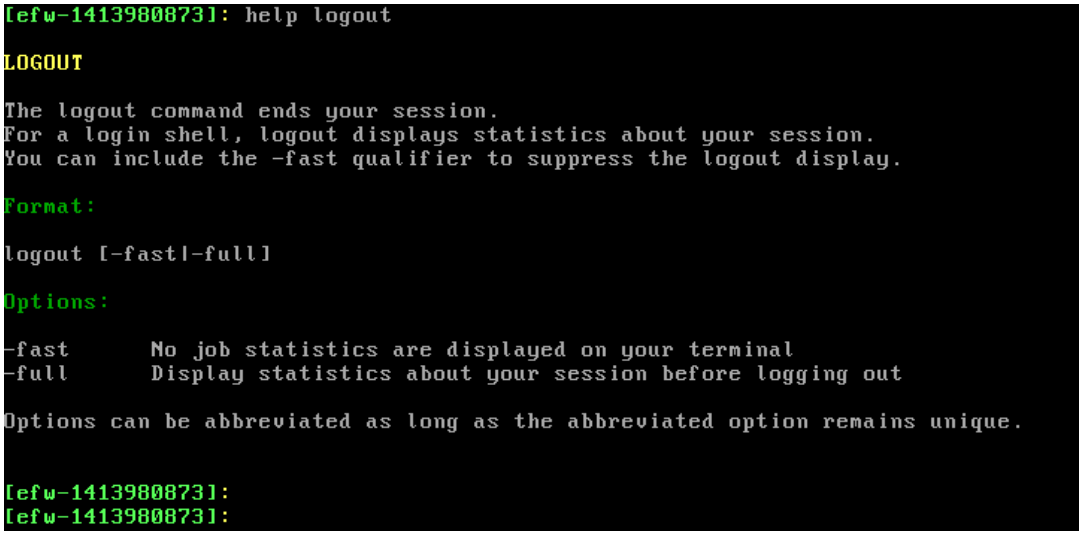
Командата login предоставя възможност за включване към Linux операционната система, която използва EFW CE. При стандартна конфигурация се изисква въвеждането на потребител (root) и парола с цел автентификация.



Фиг. 5.43 Аргументи за командата login

За да се прекрати работата на текущата сесия се използва командата logout, която има следните опционални аргументи:

- -fast – не се извежда статистика за задачите;
- -full – извеждане на статистика за сесията преди нейното прекратяване.



Фиг. 5.44 Аргументи за командата logout

Netwizard предоставя възможност за конфигуриране на мрежовите интерфейси на EFW CE чрез текстов диалог, като тази команда няма аргументи. Ако след нейното стартиране се натисне "Ctrl+C" въведените стойности не се използват за промяна на конфигурацията и netwizard се прекратява.

```
[efw-14139800873]: help netwizard

NETWIZARD

start the network configuration wizard

Format:

netwizard


[efw-14139800873]: netwizard
root's password:
Network Configuration Wizard
----------------------------

Hostname? efw-1413980873
Domain? localdomain
RED interface type <STATIC/DHCP/GATEWAY>? DHCP
RED device <eth0/eth1>? eth1
Green devices <eth0>? eth0
Green IPs (IP/CIDR)? 192.168.0.15/24
Orange devices <>?
Blue devices <>?
Enable SSH access <on/off>? on_
```

Фиг. 5.45 Команда netwizard

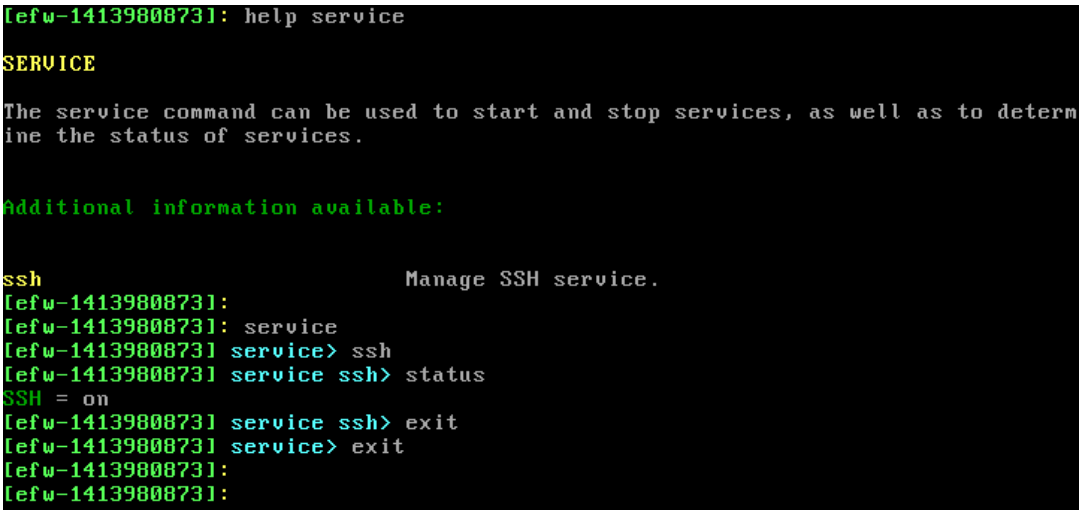
Командата ping се използва за изпращане на ICMP Echo-request с цел проверка на отдалечена свързаност, като аргумента е или IP адрес или DNS името на отдалечената система.

След стартирането на ping се започва генериране на ICMP пакети и визуализиране на резултата, като командата се прекратява чрез натискане на "Ctrl+C". След приключване на ping операциите се извежда обобщена статистика.

```
PING

send ICMP ECHO_REQUEST packets to network hosts.

Format:

ping DESTINATION

Options:

DESTINATION Host address


[efw-14139800873]: ping www.ict-academy.bg
PING www.ict-academy.bg (81.161.249.8) 56(84) bytes of data.
64 bytes from MTF4534-1.tu-sofia.bg (81.161.249.8): icmp_seq=0 ttl=64 time=0.588
 ms
64 bytes from MTF4534-1.tu-sofia.bg (81.161.249.8): icmp_seq=1 ttl=64 time=0.487
 ms

[efw-14139800873]:
--- www.ict-academy.bg ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.487/0.537/0.588/0.055 ms, pipe 2
_
```

Фиг. 5.46 Команда ping

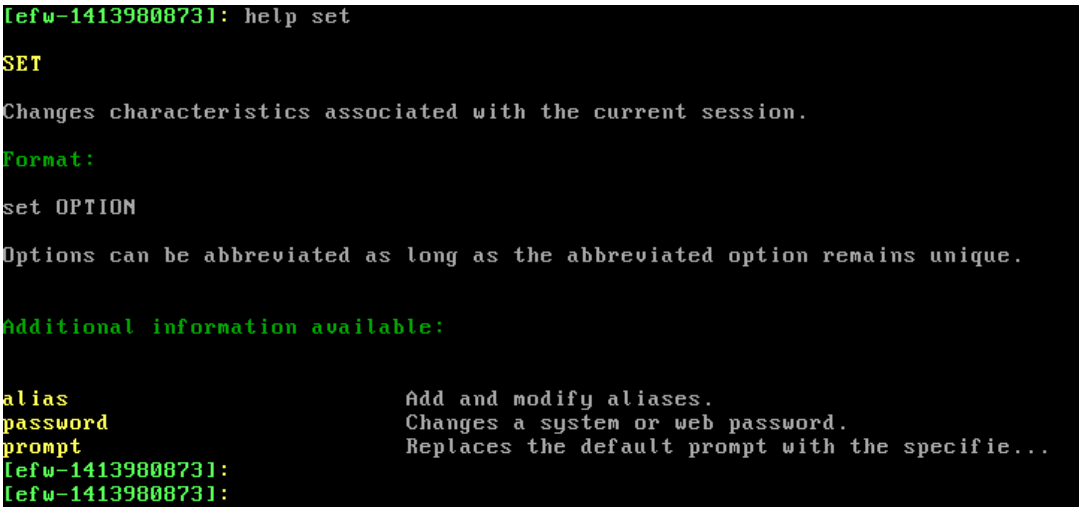
Командата service се използва са стартиране и спиране на услуги, като при EFW CE е възможно да се управлява единствено SSH демона. При стартиране на командата service се активират няколко подрежима, а командният ред се променя и описва даденото конфигурационно ниво (показано на фиг. 5.47).



Фиг. 5.47 Команда service и подрежими

Set се използва за промяна на конфигурацията на текущата сесия, като възможните аргументи са:

- alias – добавяне или промяна на конфигурираните съкращения на команди;
- password – промяна на парола за достъп до системата или до Web интерфейса на EFW CE;
- prompt – промяна на начина на изписване на Prompt.

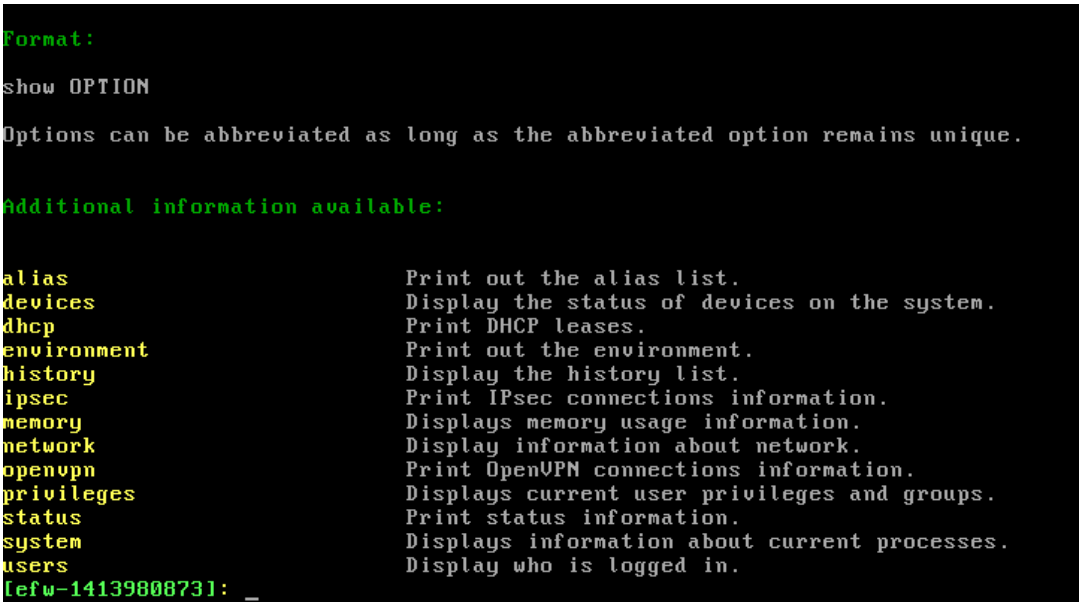


Фиг. 5.48 Команда set

Чрез командата show може да се визуализира информация за системата, настройките, услугите и др., като приложимите аргументи са:

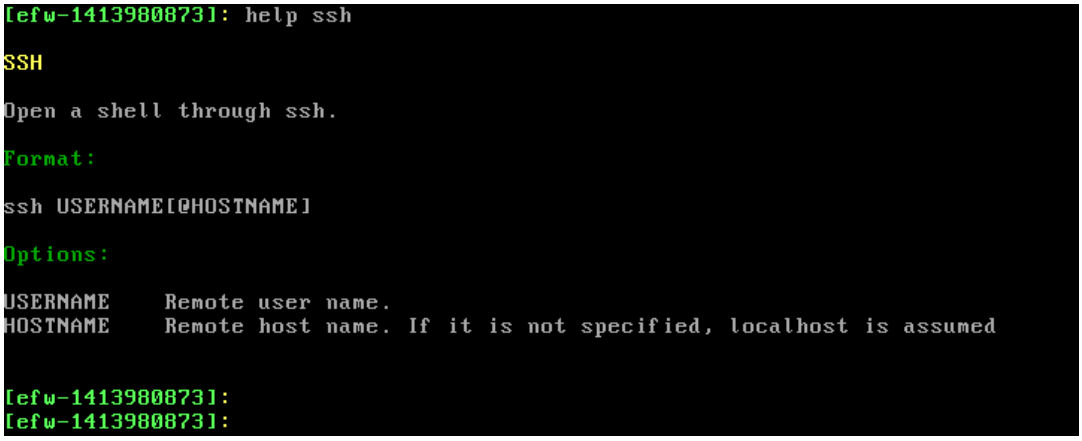
- alias – информация за съкратените команди;
- devices – статус на системните модули и файловите системи;
- dhcp – информация за използваните DHCP адреси (dhcp leases);
- environment – визуализиране на системните променливи (environment variables);
- history – списък с въведените до момента команди;
- IPsec – информация за IPsec връзките;
- memory – данни за използваната памет;

- network – информация за мрежовите интерфейси. Необходимо е да се добави и втори аргумент, посочващ какво да се визуализира (addresses, links, maddresses, neighbors, routes, rules, summary, tunnels);
- openvpn – данни за OpenVPN връзките;
- privileges – информация за нивото на достъп на текущия потребител;
- status – извежда се обобщена информация за статуса на EFW CE;
- system – данни за системните процеси;
- users – информация за потребителите, които в момента са включени към системата.



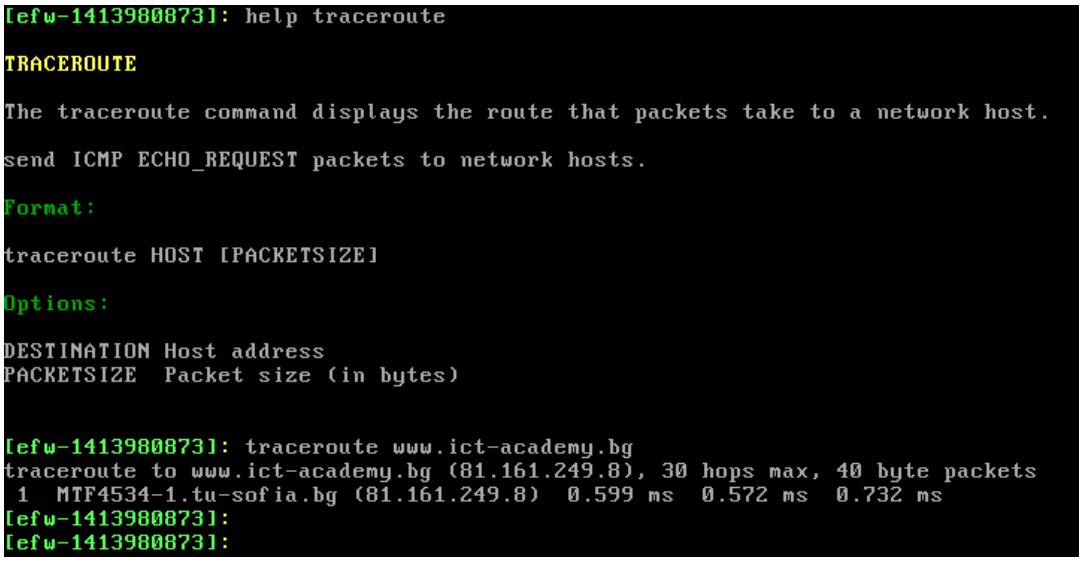
Фиг. 5.49 Аргументи на командата show при EFW CE

Ако е необходимо от конзолата на EFW CE да бъде стартирана SSH сесия към отдалечено устройство може да се използва командата SSH, последвана от адрес или DNS име и опционално потребителско име.



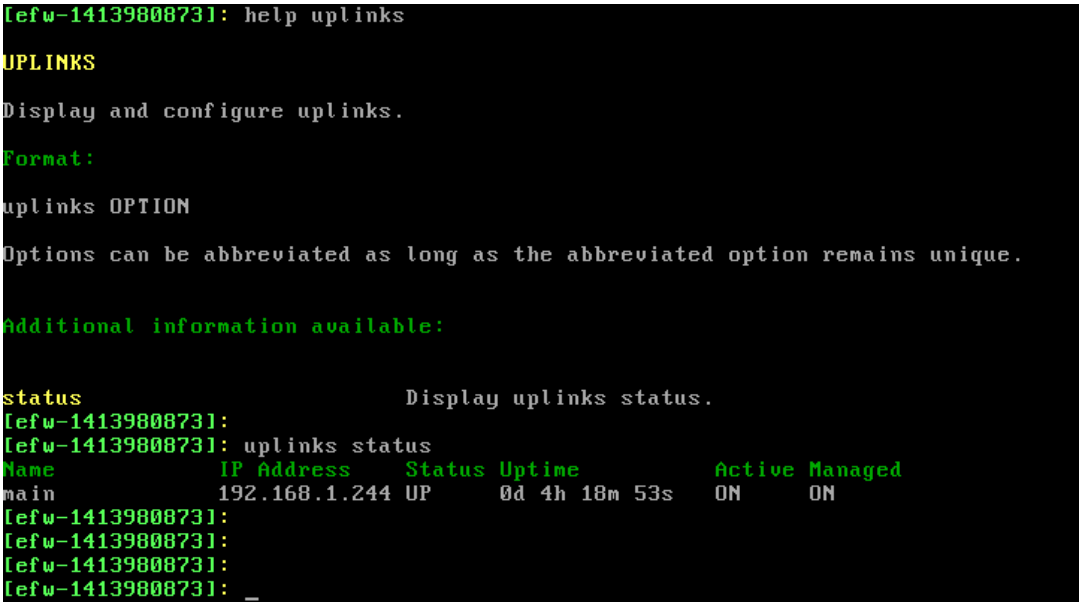
Фиг. 5.50 Аргументи на командата ssh

Проверката на текущия маршрут на IP трафика между EFW CE и отдалечена система се извършва чрез стандартната команда traceroute.



Фиг. 5.51 Аргументи на командата traceroute и визуализиране на пътя (маршрутизаторите) към устройство www.ict-academy.bg

За да се визуализира информация за интерфейсите, които имат връзка към Интернет доставчиците или WAN може да се използва командата uplinks, последвана от параметъра status.



Фиг. 5.52 Аргументи на командата uplinks и резултат от нейното стартиране

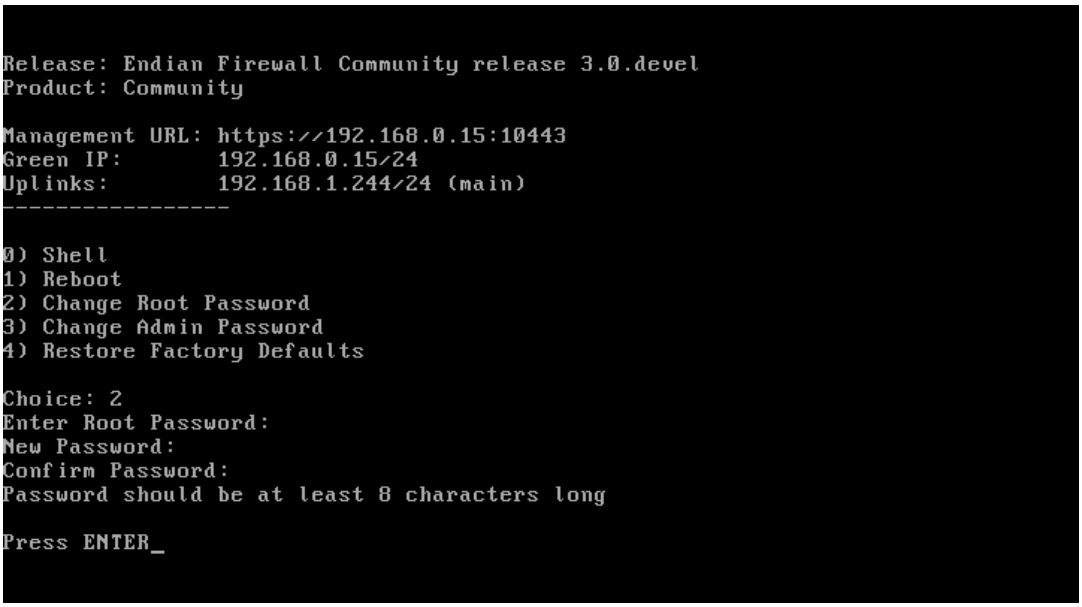
Рестартиране

Рестартирането на EFW CE от конзолата се извършва от опция 1 (Reboot). Необходимо е действието да се потвърди с “Y”.

Промяна на парола на потребител root

За да се промени текущата парола на потребителя root от конзолното меню се избира опция 2 (Change Root Password). Въвежда се текущата парола , новата парола, която се потвърждава.

Препоръчително е да се спазват правилата за надеждни пароли!

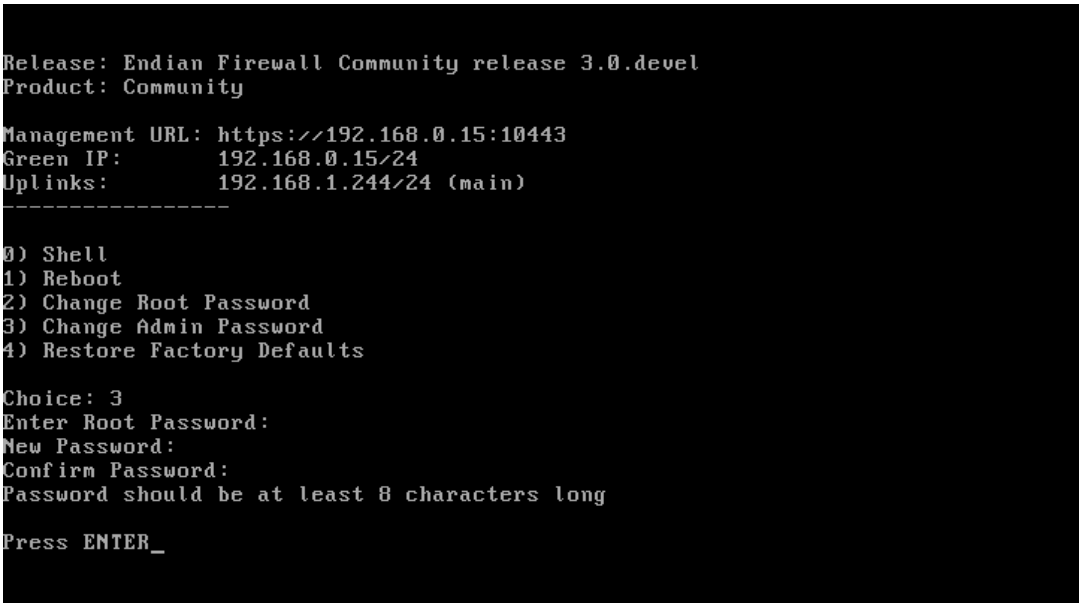


Фиг. 5.53 Промяна на паролата на потребителя root от конзолата на EFW EC

Промяна на парола на потребител admin

Аналогично на промяната на паролата на потребителя root от конзолата може да се зададе и нова парола за потребителя admin (достъп до графичния интерфейс). Първоначално се изисква да се въведе паролата на root потребителя, след което новата за admin, която е необходимо е да се потвърди.

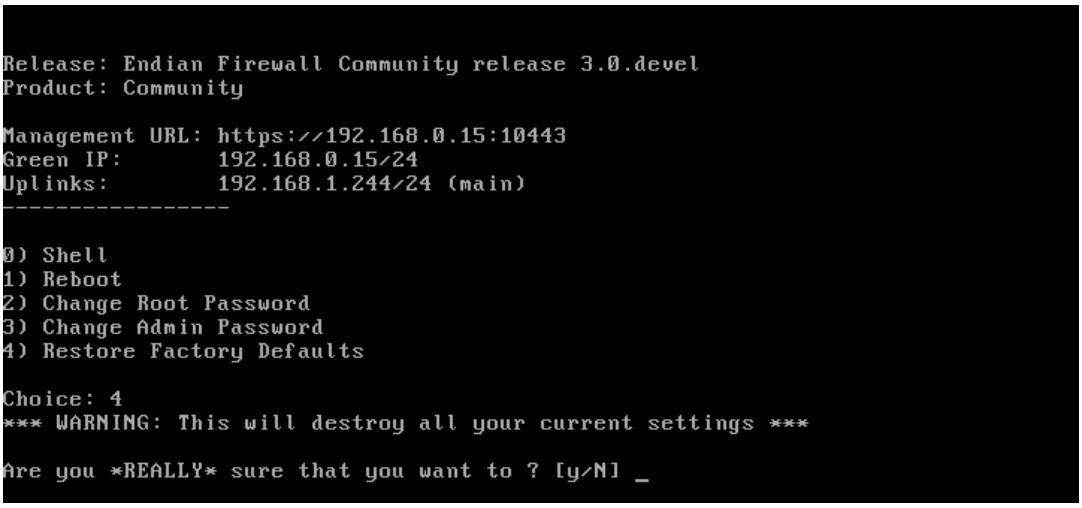
Отново се препоръчва да се спазват правилата и съветите за надеждни пароли!



Фиг. 5.54 Промяна на паролата на потребителя admin от конзолата на EFW EC

Възстановяване на настройките по подразбиране

От конзолното меню при избор на опция 4 (Restore Factory Defaults) се възстановяват настройките по подразбиране на EFW CE. Необходимо е действието да бъде потвърдено с “Y”.



Фиг. 5.55 възстановяване на настройките по подразбиране на EFW CE от конзолното меню

Заключение

Инсталирането на EFW CE е лесно и бързо, като хардуерните изисквания са ниски. Важно е да се отбележи, че в процесът на инсталиране цялото съдържание на твърдия диск се изтрива и се създават нови файлови системи.

Първоначалното конфигуриране на EFW CE е интуитивно и се извършва от специален web интерфейс.

Поддръжката, наблюдението и конфигурирането на EFW CE може да се извърши от конзолата, чрез клавиатура и монитор, като е възможно да се използва и пълен достъп до използваната операционна система (Linux).

Глава 6. Наблюдение на EFW CE и настройка на мрежовите интерфейси

Преди да се пристъпи към конфигурирането на защитната стена и правилата за филтриране на трафика е препоръчително да се провери правилната работа на мрежовите интерфейси, системните услуги и да се конфигурират някои допълнителни сървъри, като DHCP и Time. Също така е препоръчително да се анализира и маршрутизиращата таблица на EFW CE.

Меню “Status”

Менюто “Status” съдържа всички необходими функции (подменюта) за проверка на текущото състояние на EFW CE, като за разлика “Dashboard” предоставя подробна информация, която е изключително полезна при отстраняване на проблеми, свързани с работата на системата. Включените елементи в това подменю са:

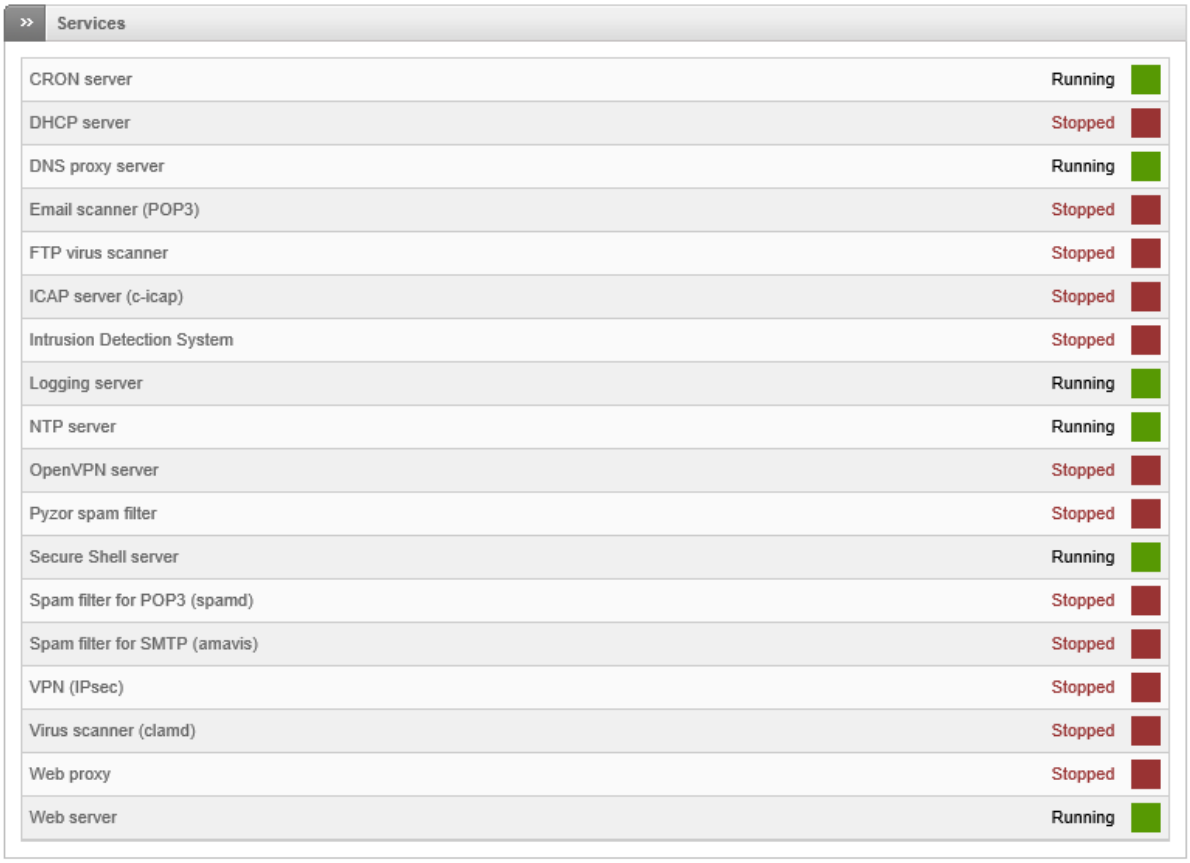
- System status – информация за състоянието на системата, списък със състоянието на отделните услуги, справка за използваната памет, данни за използваното дисково пространство, времето на работа на EFW, заредените модули и версията на ядрото;
- Network status – състояние и данни за конфигурацията на мрежовите интерфейси, маршрутизираща таблица и arp кеш;
- System graphs – графично представяне на натоварването на процесора(ите), използваната памет, използваната временна swap памет, заетото дисково пространство, заетото пространство на файловата система за конфигурация на EFW CE и за журнални файлове, както и на дяла за данни;
- Traffic graphs – графично представяне на натоварването на отделните интерфейси на EFW CE;
- Proxy graphs – графично представяне на състоянието на прокси услугите, включващо трафика за деня, заявките за деня, използвания кеш за деня и за 5 минутен интервал , както и данни за производителността на кеша;
- Connections – в табличен вид са представени текущите връзки (connections), използвания транспортен протокол и състоянието им;
- VPN connections – информация за VPN връзките;
- SMTP mail statistics – графично представяне на статистиката за SMTP протокола и сървъра;
- Mail queue – информация за опашката с електронна поща.

Подменю “System status”

Първата група с данни от подменюто “System status” е “Services” и при нея в табличен вид са показани стартираните системни услуги (демони) и тяхното състояние:

- Running – услугата е стартирана и работи;
- Stopped – дадената услуга е изключена (спряна);

Данните имат само информативен характер и от таблицата не могат да бъдат правени промени на услугите (стартиране или изключване).



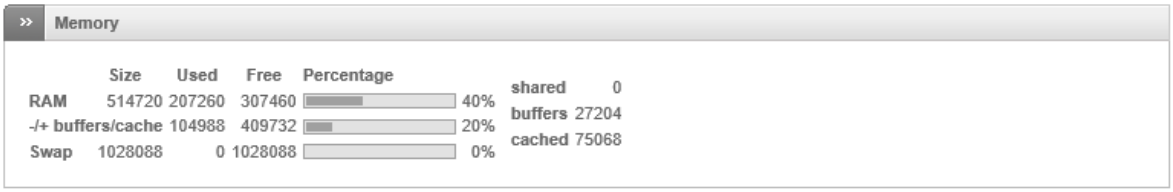
Фиг. 6.1 Списък с информация за състоянието на системните услуги

Втората група с данни от “System status” визуализира състоянието на паметта на системата, като се извеждат следните параметри:

- Size – обем на паметта в килобайта;
- Used – обем на използваната памет в килобайта;
- Free – свободна памет в килобайта;
- Percentage – процентно съотношение на заетата към свободната памет;
- Shared – споделена памет;
- Buffers – брой буфери;
- Cached – използвана кеш памет.

От Endian препоръчват да се наблюдава съотношението на свободната памет към използваната и ако за продължителен период от време се задържи над 80% е желателно към системата да се добавят нови RAM модули. Също така ако RAM паметта не е достатъчна Linux ще компенсира с чести операции към swap, което ще доведе до рязко намаляване на производителността – действие, критично за някои операции и водещо до драстично редуциране на броя на филтрираните пакети в секунда.

Подобен анализ може да се направи и за swap паметта, като там нормалната стойност за правилна работа на системата трябва да бъде до 20% натоварване.

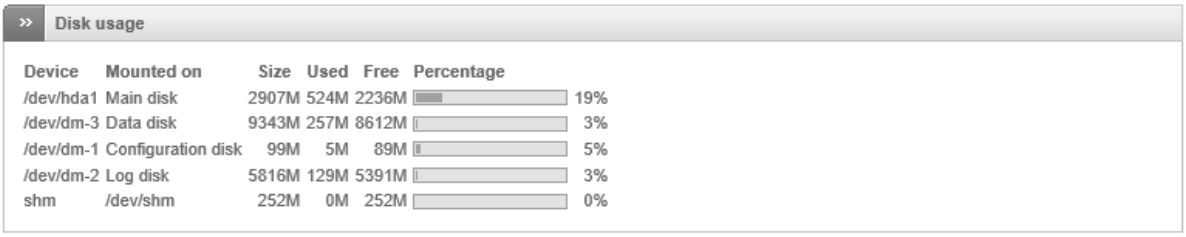


Фиг. 6.2 Данни за използваната памет от EFW CE

Под групата със състоянието на паметта е визуализирана информация, получена от Linux командата df, която включва обемът, използваното и свободно пространство, както и процентно съотношение (заето към свободно) за:

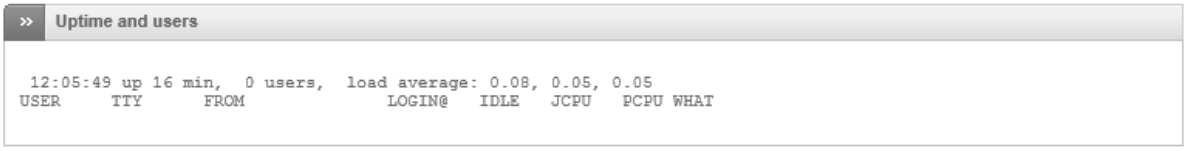
- Основната файлово система “Main disk” – /dev/hda1;
- Файловата система за съхранение на данни за EFW CE “Data disk” – /dev/dm-3;
- Конфигурационния диск “Configuration disk” – /dev/dm-1;
- Файловата система за журнални файлове “Log disk” – /dev/dm-2;
- Споделената памет “shm” – /dev/shm.

Съдържанието на файловите системи за данни и за журнали може да нарасне с времето за това се препоръчва първоначално за тях да бъде отделено повече пространство.



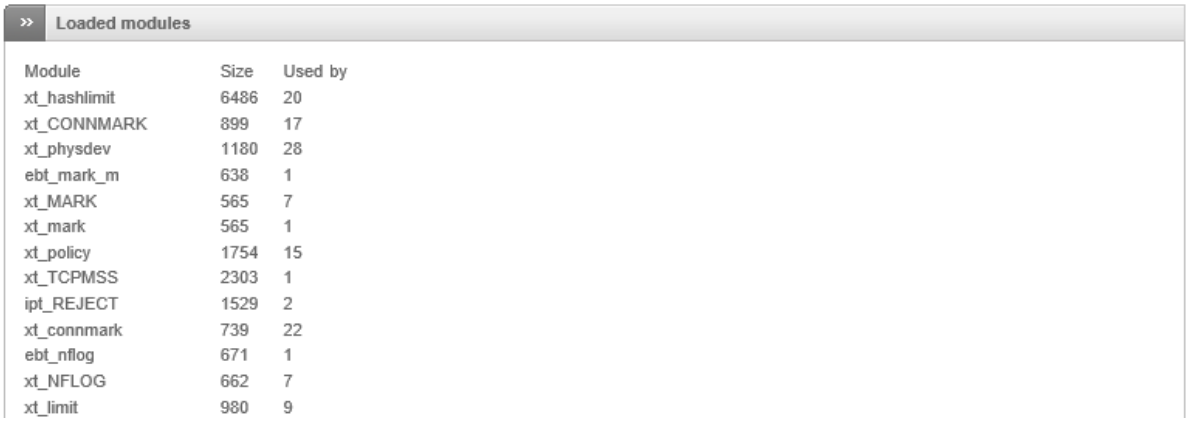
Фиг. 6.3 Данни за състоянието на файловите системи (отчетените стойности са в MB)

Към групата “Uptime and users” са включени данни за текущите потребители, които използват административен достъп до EFW CW, както и натоварването на системата. Времевата статистика показва времето на активност на EFW CE.



Фиг. 6.4 Данни за потребителите, свързани към момента и използващи административен достъп до EFW CE системата

Предпоследната група на страницата “System status” съдържа информация за заредените системни модули.



Loaded modules

Module	Size	Used by
xt_hashlimit	6486	20
xt_CONNMARK	899	17
xt_physdev	1180	28
ebt_mark_m	638	1
xt_MARK	565	7
xt_mark	565	1
xt_policy	1754	15
xt_TCPMSS	2303	1
ipt_REJECT	1529	2
xt_connmark	739	22
ebt_nflog	671	1
xt_NFLOG	662	7
xt_limit	980	9

Фиг. 6.5 Данни за заредените системни модули

На края на страницата е поместена информацията за ядрото на Linux операционната система.

Kernel version

2.6.32.43-57.e51.i586

Фиг. 6.65 Данни за версията на ядрото на Linux операционната система, която се използва от EFW CE

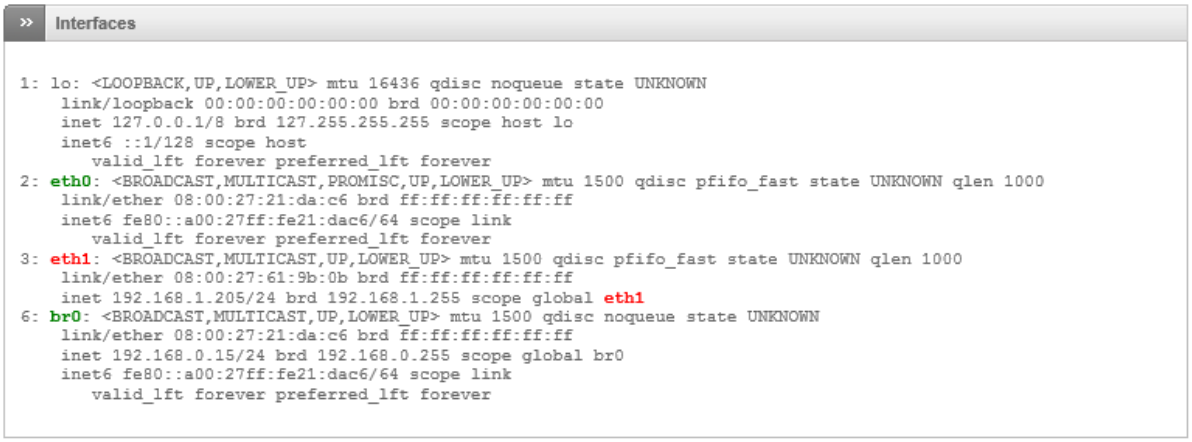
Подменю “Network status”

От подменюто “Network status” се получава достъп до обобщена информация за състоянието на мрежовите интерфейси.

Първата група с данни съдържа резултата от стартирането на командата ip addr show. За всеки отделен интерфейс е посочен цвета на зоната, в която се намира, IP адресите, MAC адресът (ако е наличен) и параметъра MTU.

При два физически интерфейса EFW CE използва:

- lo – loopback интерфейс;
- eth0 – първата мрежова карта, от примера (фиг. 6.6) се вижда, че е в зелената зона;
- eth1 – втората мрежова карта, от примера (фиг. 6.6) се вижда, че е част от червената зона;
- br0 – мостов интерфейс между eth0 и eth1.

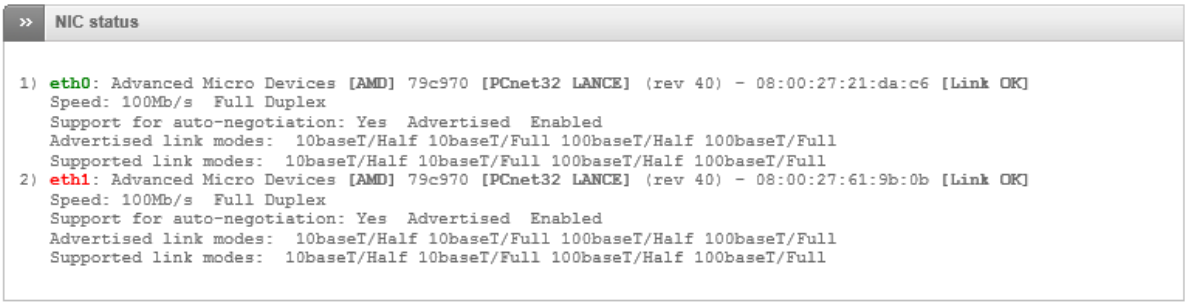


Interfaces

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 08:00:27:21:da:c6 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe21:dac6/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 08:00:27:61:9b:0b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.205/24 brd 192.168.1.255 scope global eth1
6: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 08:00:27:21:da:c6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.15/24 brd 192.168.0.255 scope global br0
    inet6 fe80::a00:27ff:fe21:dac6/64 scope link
       valid_lft forever preferred_lft forever
```

Фиг. 6.6 Данни за мрежовите интерфейси на EFW CE

Непосредствено след “Interfaces” е втората група “NIC status”, в която е поместено състоянието на отделните мрежови интерфейси (карти). Ако дадената карта е в работно състояние и не са открити проблеми, то тя е маркирана с [LINK OK]. Също така от тази група може да се провери скоростта на свързване, дуплекса и дали картата е с активирана auto-negotiation функционалност.

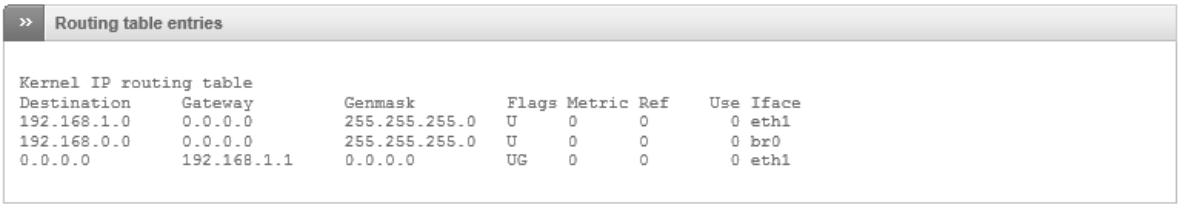


Фиг. 6.7 Данни за състоянието на мрежовите карти

На същата страница е поместено и съдържанието на маршрутизиращата таблица, получена в резултат на стартирането на командата route[58] –n. Използваните флагове при представянето на са:

- U (up) – пътят е в работно състояние;
- G (gateway) – пътят се използва като шлюз;
- H (host) – пътят сочи към хост;
- R (reinstate) – пътят е научен чрез протокол за динамично маршрутизиране;
- D (dynamical) – пътят е включен в маршрутизиращата таблица на динамичен принцип от демон или от ICMP пренасочване;
- M (modified) – пътят е модифициран от демон или от ICMP пренасочване;
- A (addrconf) – пътят е добавен от addrconf;
- C (cache) – кеширана стойност;
- ! (reject) – пътят е отхвърлен.

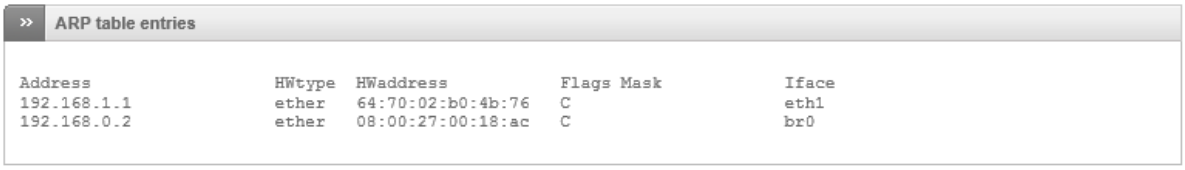
Стойността на метриката най-често е в “hop count” (брой маршрутизатори по пътя към отдалечената мрежа, спрямо текущото устройство.). Ref описва броя референции и не се използва от ядрото на Linux, а параметъра Use показва броя на извършените търсения на дадения път. Допълнително описание на командата route можете да намерите в съответната man страница.



Фиг. 6.8 Визуализиране на съдържанието на маршрутизиращата таблица при EFW CE

[58] linux.die.net/man/8/route

Последната група с данни е текущото съдържание на ARP кеша (командна arp[59] –n).

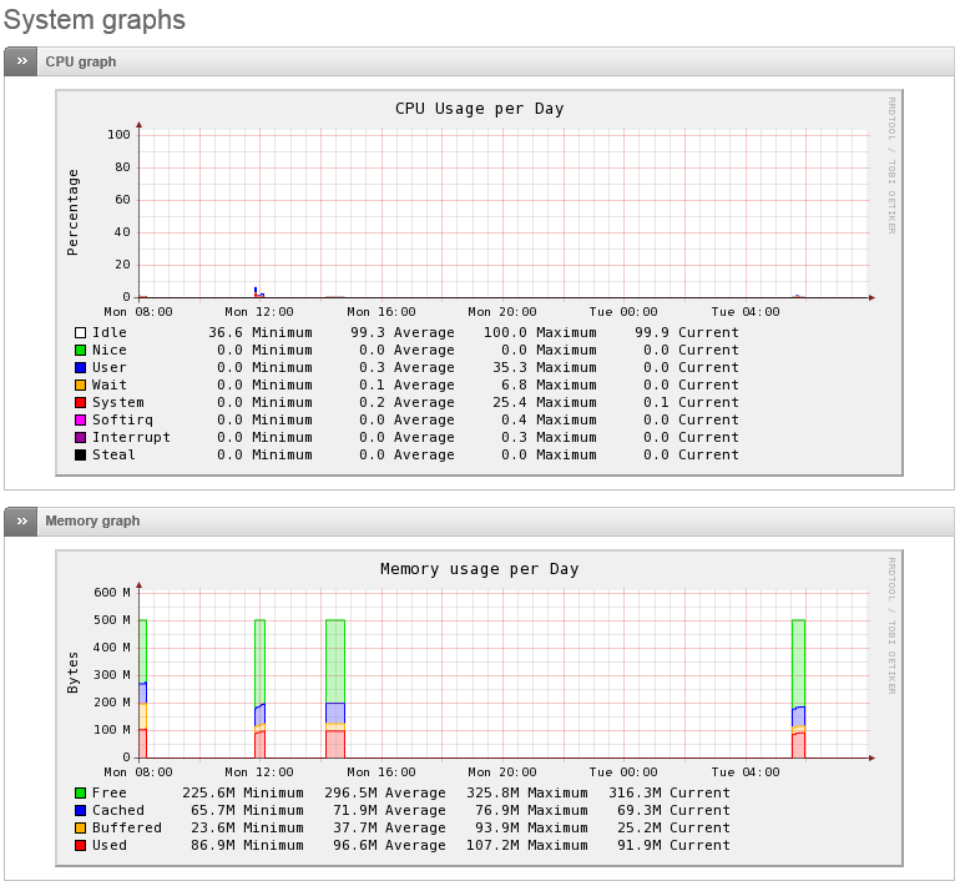


Фиг. 6.9 Съдържание на кеша на ARP протокола

Подменю “System graphs”

Подменюто “System graphs” визуализира в графичен вид:

- CPU graph – натоварване на процесорите на системата;
- Memory graph – данни за използваната от системата памет;
- Swap graph – информация за състоянието на swap паметта;
- Disk usage graph: Main disk – съотношение на свободно към заето място на “main disk”;
- Disk usage graph: Configuration disk – съотношение на свободно към заето място на “configuration disk”;
- Disk usage graph: Log disk – съотношение на свободно към заето място на “log disk”;
- Disk usage graph: Data disk – съотношение на свободно към заето място на “data disk”.



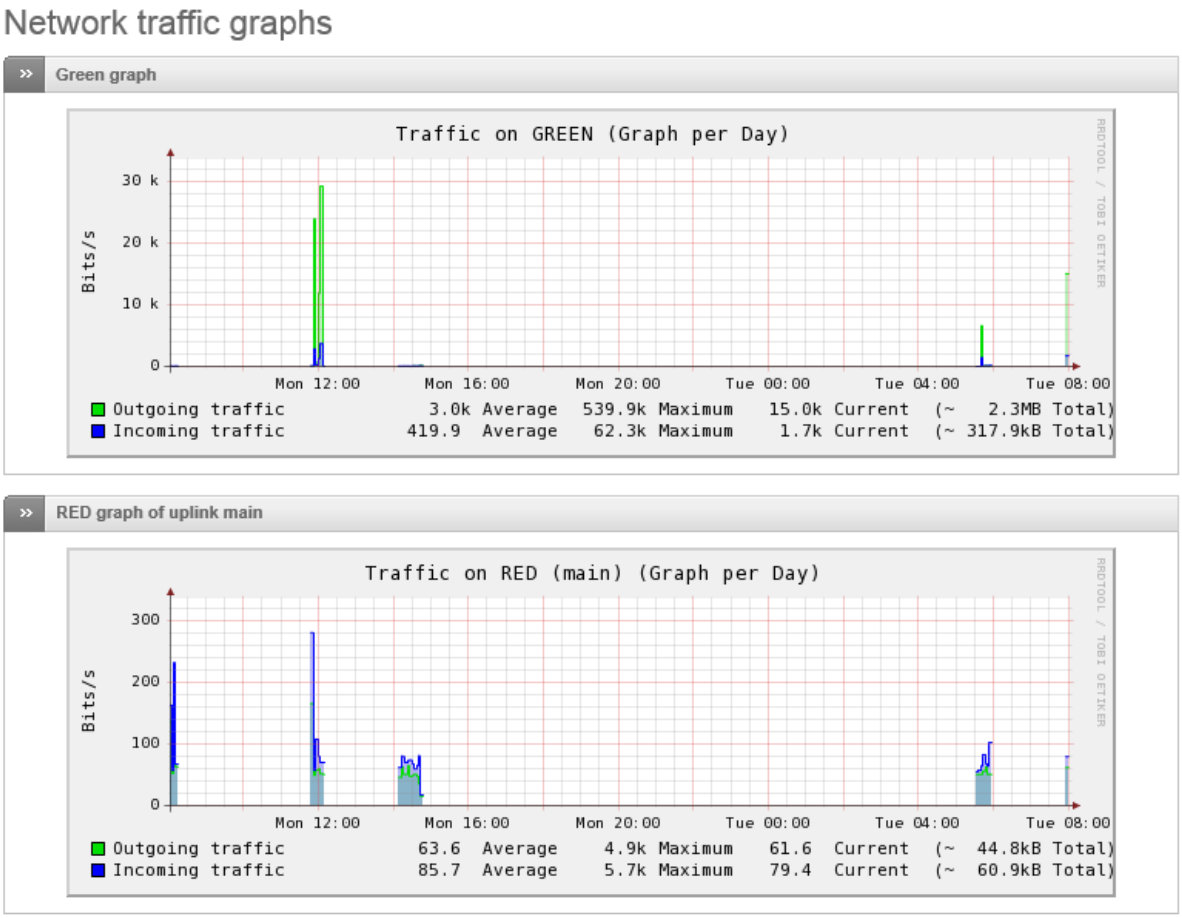
Фиг. 6.10 Графично представяне на натоварването на процесора и на използваната памет от EFW CE

[59]linux.die.net/man/8/arp

Подменю “Traffic graphs”

Аналогично на “System graphs” в подменюто “Traffic graphs” се визуализират данни в графичен вид, но отделните параметри са свързани с натоварването на отделните интерфейси и съдържат:

- Outgoing traffic – изходящ трафик за интерфейса;
- Incoming traffic – входящ трафик за интерфейса.



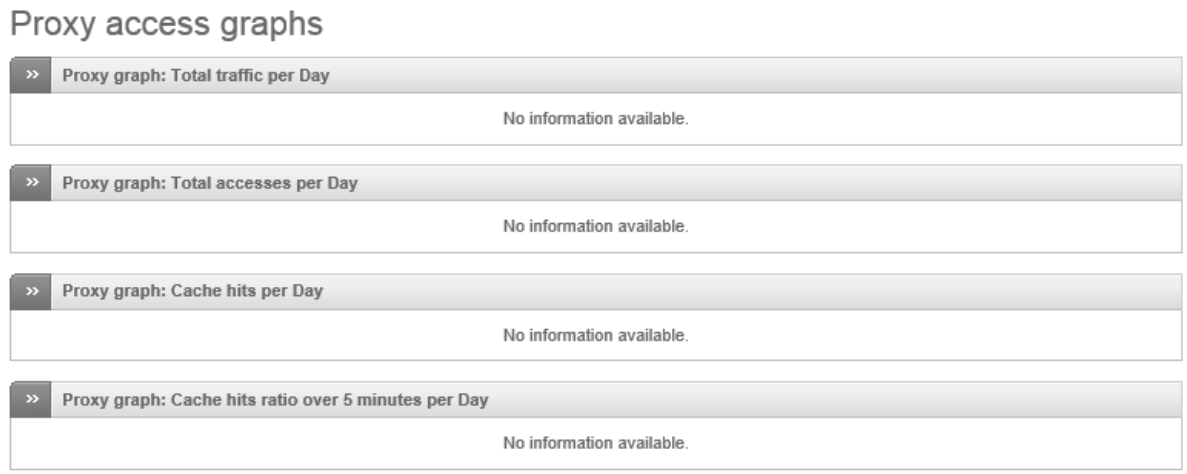
Фиг. 6.11 Графично представяне на натоварването на EFW CE

Подменю “Proxy graphs”

В подменюто “proxy graphs” са включени графики, които визуализират статистическите данни за прокси функциите за последните 24 часа:

- Total traffic per day – обемът на данните, обработени от прокси функциите. В син цвят са посочени входящите, а в зелено – изходящите;
- Total Accesses per Day – брой HTTP заявки, като в син цвят са посочени изходящите, а в зелен – входящите;
- Cache hits per day – направени заявки за обекти от кеша;
- Cache hits ratio over 5 minutes per day – заявки към обекти в кеша за последните 5 минути.

Графиките са налични след активиране и конфигуриране на прокси функционалността на EFW CE.

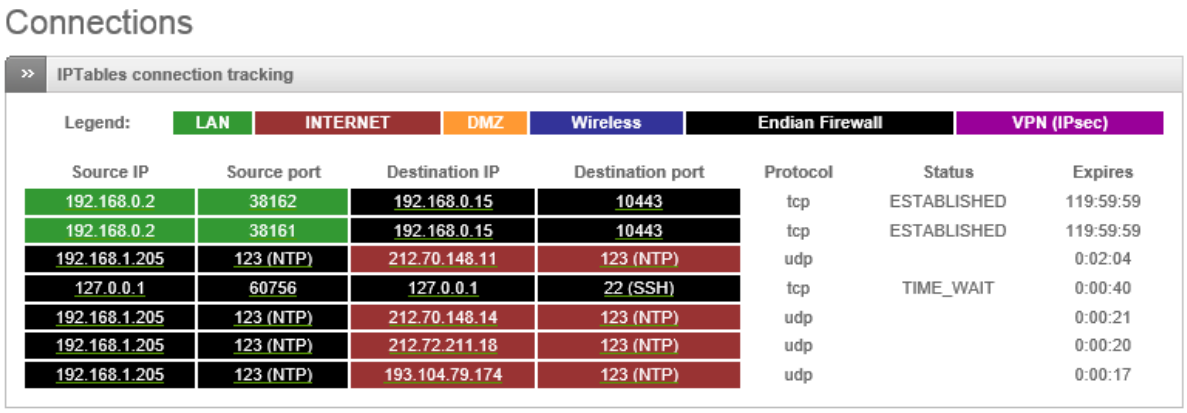


Фиг. 6.12 Данни за работата на прокси сървърите – поради неактивните услуги липсват и данни за визуализиране

Подменю “Connections”

Подменюто “Connections” съдържа таблица, в която са описани връзките от, към и през EFW CE. Данните са получени от таблицата conntrack на Linux ядрото. Връзките са маркирани в определени цветове, както следва:

- Червено, зелено, синьо и оранжево показват зоната;
- Черно посочва, че връзката е от демон или услуга (например SSHd);
- Лилаво маркира връзката като част от VPN.



Фиг. 6.13 Данни за conntrack на ядрото на използваната Linux операционна система

Отделните колони на таблицата съдържат:

- Source IP – IP адрес на източника на трафика;
- Source port – порт на приложените, източник на трафика;
- Destination IP – IP адрес на получателя на трафика;
- Destination port – порт на приложените, получател на трафика;
- Protocol – транспортен протокол;
- Status – състояние на връзката при TCP, съгласно RFC 793[60];
- Expires – оставащото време на връзката в текущото състояние.

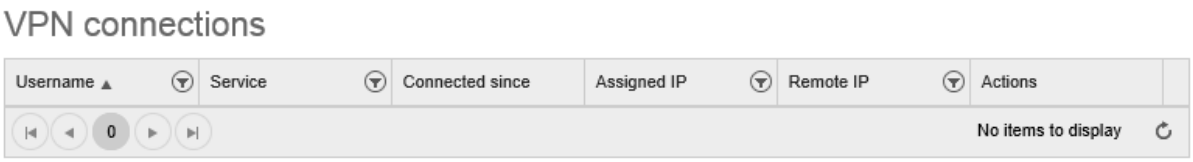
[60] tools.ietf.org/html/rfc793.html

Съдържанието на тази страница се обновява автоматично на всеки 5 секунди.

При натискане на връзката от даден адрес се извършва пренасочване към www.whois.net с цел извличане на допълнителни данни за адреса. Тази функционалност е удобна при извършване на справка за трафик от зловреден софтуер или при подозрение за атаки.

Подменю "VPN connections"

Подменюто "VPN connections" е налично от версия 3.0 на EFW CE и съдържа информация за VPN потребителите и IPsec тунелите.

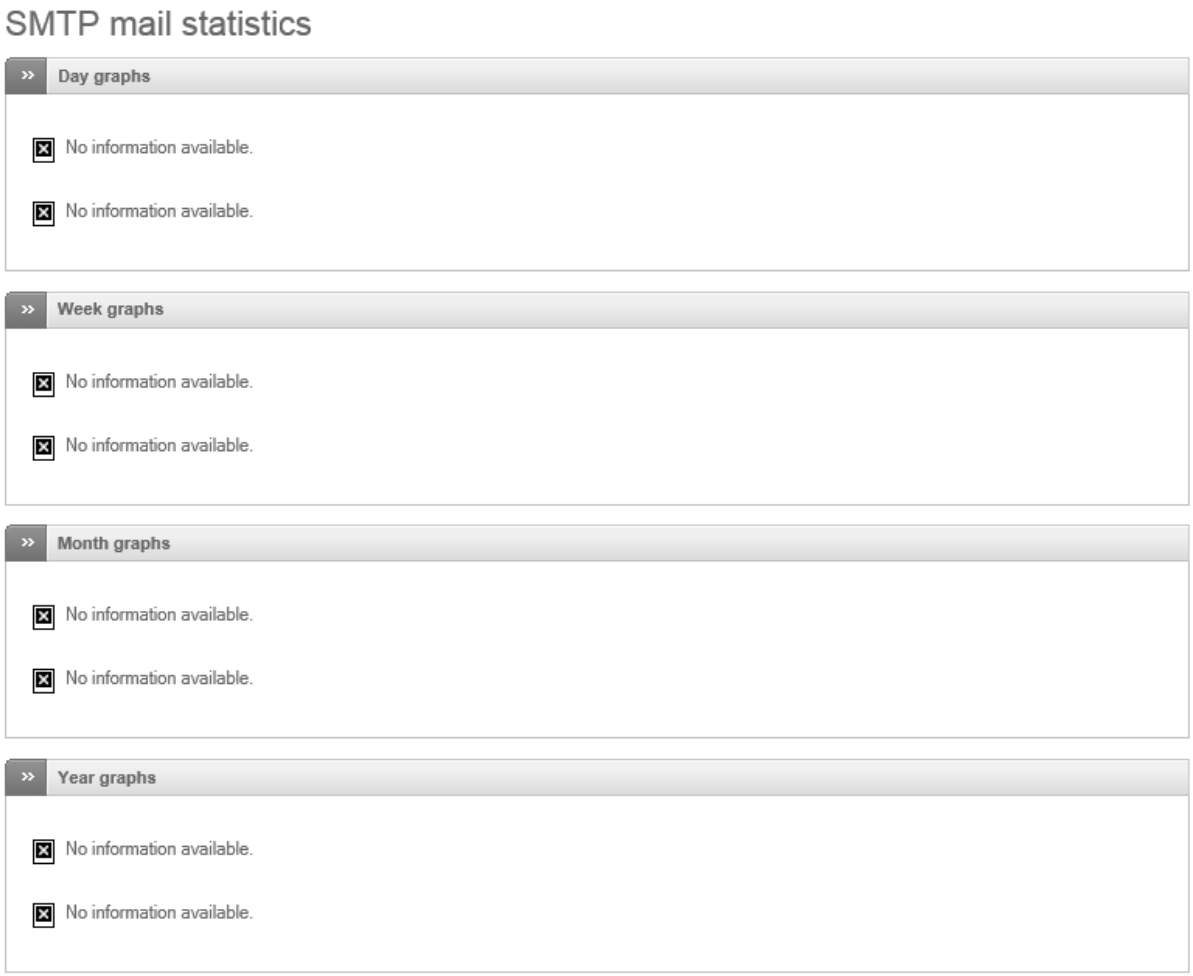


Фиг. 6.14 Информация за изградени VPN тунели (поради липсата на такива таблицата е празна)

Подменю "SMTP mail statistics"

"SMTP mail statistics" съдържа статистически данни, представени в удобен графичен вид за работата на SMTP сървъра, използван от EFW CE. Ако SMTP сървърът не е активен графиките не се визуализират и поради технически пропуск в текущата версия[61] се получава пренасочване към несъществуващ графичен обект (на фигурата).

[61] Стабилна версия към датата на писане на книгата



Фиг. 6.15 Статистически данни за работата на SMTP сървъра. Поради техническа неточност при липса на данни се извършва пренасочване към несъществуващ обект

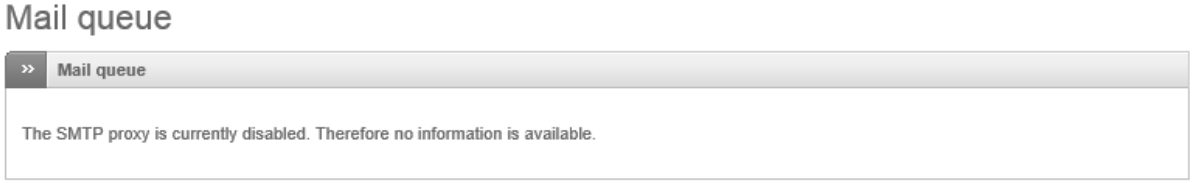
Ако има налични данни всяка една графика описва:

- По оста X – промяна в отчетената стойност;
- По оста Y – брой електронни писма за минута.

Информацията е обобщена по дни, седмици месеци и на годишна база.

Подменю “Mail queue”

Последното подменю в “Status” е “Mail queue”, което съдържа текущата опашка от електронна поща, ако SMTP прокси функциите са активирани.

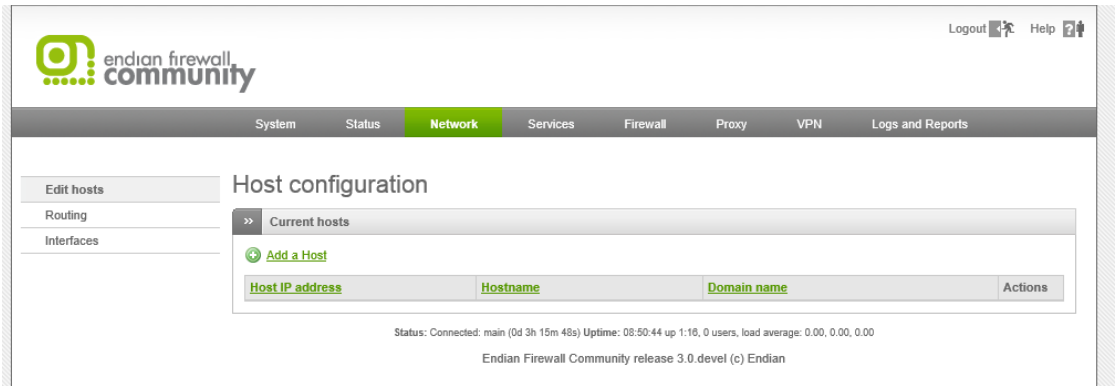


Фиг. 6.16 Данни за SMTP прокси функциите (липсата на данни се дължи на не активираното SMTP прокси)

Меню “Network”

Менюто “Network” съдържа всички необходими менюта за допълнително конфигуриране на базовата мрежова функционалност на EFW CE:

- Добавени хостове – конфигурация, включваща име на хоста, IP адрес и име на домейн;
- Редактирани пътища – добавяне или премахване на статични пътища, както и policy маршрутизиране – технология, която позволява на администраторите по определени критерии да пренасочат дадения пакет, например по адреса на източника (не бива да се бърка със source routing);
- Редактирани интерфейси – промяна на конфигурацията на uplink и добавяни и премахване на VLAN.



Фиг. 6.17 Меню “Network” при EFW CE

Хостове

Подменюто “Edit host” се използва за добавяне, редактиране или изтриване на хостове. Под хост се разбира запис от вида име/IP адрес, използван от локалния DNS сървър – dnsmasq[62].

Добавянето на нов запис се извършва при натискане на “Add a Host” и въвеждане на:

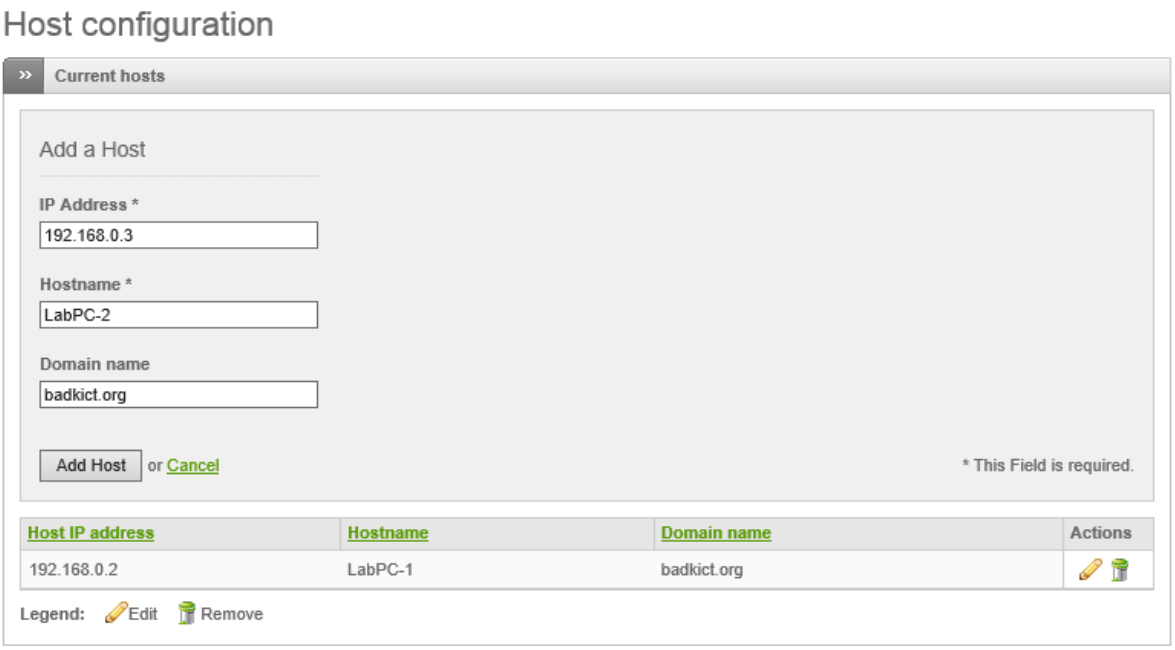
- IP address – IP адрес на хоста (полето е задължително);
- Hostname – име на хоста (полето е задължително);
- Domain name – опционално поле за името на домейна.

След като се попълнят полетата се натиска бутона “Add Host”.

В таблицата са включени всички конфигурирани хостове. В последната колона (Actions) са включени два бутона – първият е за редактиране на записа, а втория за изтриване. При изтриване на запис от таблицата не се изисква потвърждаване.

Важно е да се отбележи, че всеки ред съдържа една единствена комбинация IP адрес и име на хоста и ако е необходимо няколко адреса да сочат като едно и също име е необходимо да се добавят няколко отделни реда.

[62] www.thekelleys.org.uk/dnsmasq/doc.html



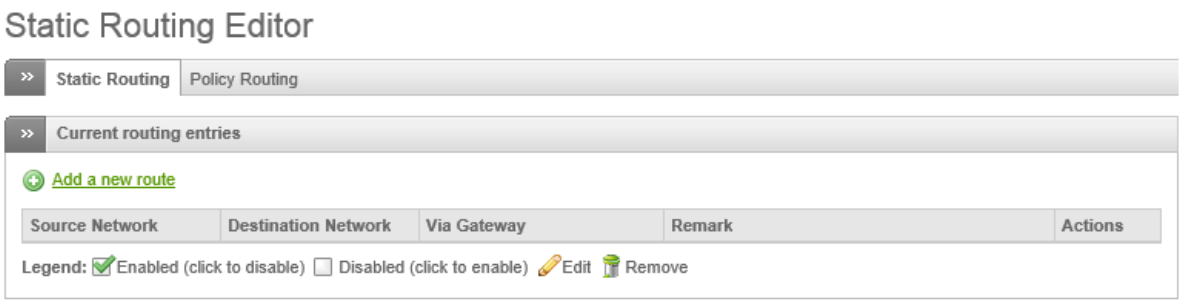
Фиг. 6.18 Подменю за добавяне и редактиране на хостове

Маршрутизиране

Към маршрутизиращата таблица, която е автоматично генерирана от EFW CE е възможно да бъдат добавени статични пътища и такива, описани чрез “policy routing”. Всяка направена промяна в допълнителните пътища трябва да бъде записана и в следствие на това да бъдат рестартирани необходимите системни услуги.

Статични пътища

Статичните пътища дефинират пренасочването на трафика от дадена мрежа към отдалечена през точно дефиниран шлюз (следващ маршрутизатор по пътя).



Фиг. 6.19 Подменю за добавяне и редактиране на статични пътища

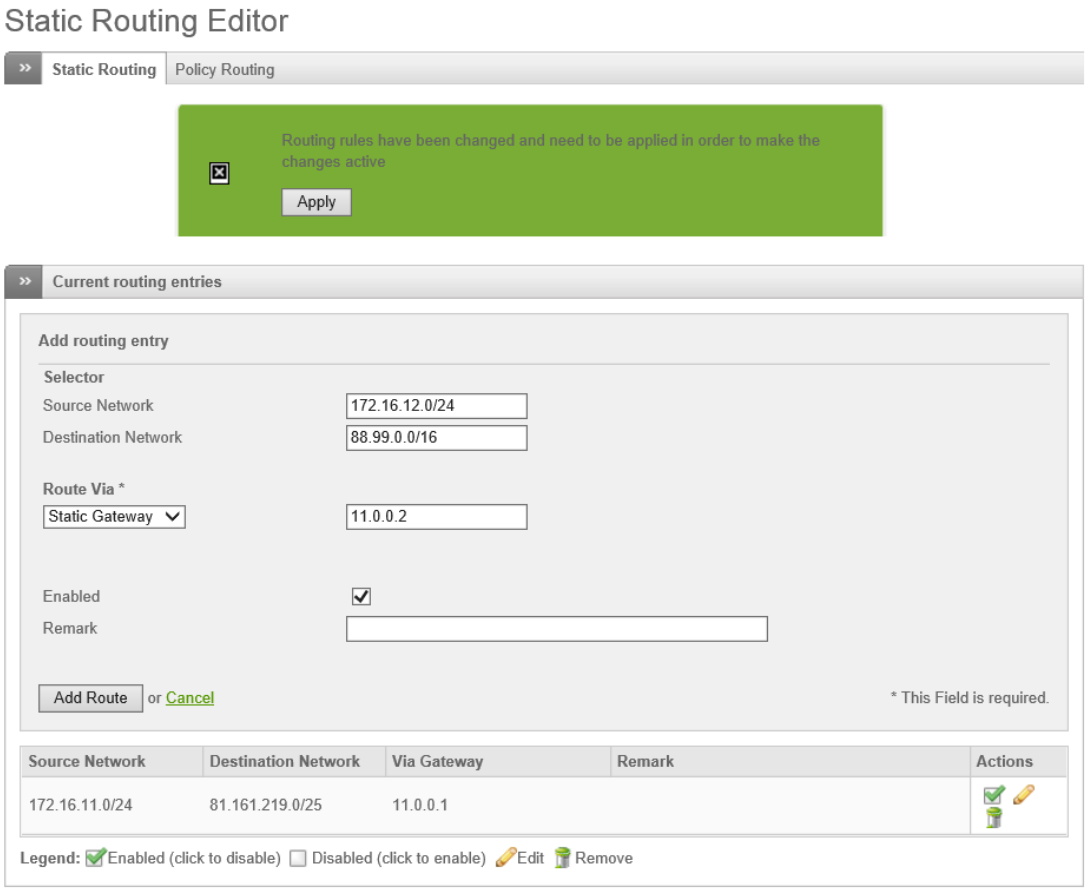
За да се дефинира нов статичен маршрут е необходимо да се въведат:

- Source network – мрежови адрес на мрежата, в която се намират източниците на трафика, например 11.0.0.0/8;
- Destination network – отдалечена мрежа, в която са свързани получателите на пакетите, например 99.88.0.0/24;
- Route via – възможни са три варианта за пренасочване на пакетите – Static gateway, Uplink, OpenVPN user. Ако е избрана опцията Static gateway е необходимо да бъде въведен адрес на шлюз. При избор на uplink на страницата се добавя падащо меню, от което се посочва изходящия мрежови интерфейс. Ако към отдалечената мрежа се

използва OpenVPN се избора последната възможна опция и отново от падащото меню се посочва избрания тунел;

- Enabled – ако опцията е активна статичния път също е активен;
- Remark – опционален коментар.

След въвеждане на необходимите параметри за да се добави статичния път се натиска бутона “Add Route” и поради промяната на конфигурацията се налага потвърждаване с бутона “Apply” (в зеленото поле в началото на страницата). Ако конфигурацията е потвърдена се рестартират определени услуги, свързани с мрежовите функции на EFW CE.



Фиг. 6.20 Пример за въвеждане на статичен път

Всички статични пътища са включени в таблицата, като последната колона “Actions” съдържа икони, чрез които пътя се активира или деактивира, редактира и изтрива.

Policy routing

Технологията policy routing позволява да се асоциират определени мрежови адреси, зони или типове трафик (протокол и порт) към дадена връзка към WAN или Интернет (uplink).

В таблицата са включени всички статични и policy routing пътища. Поради спецификата на този вид маршрутизиране от съществено значение е позицията на даденото правило. За EFW CE колкото едно правило е в по-предна позиция в таблицата, толкова неговия приоритет е по-висок и обратното.

При EFW CE технологиите policy routing, HTTP прокси и uplink взаимодействат една с друга и в зависимост от конфигурационните параметри, начина на филтриране на трафика може да е различен, най-вече при заявка от зелената към червената зона.

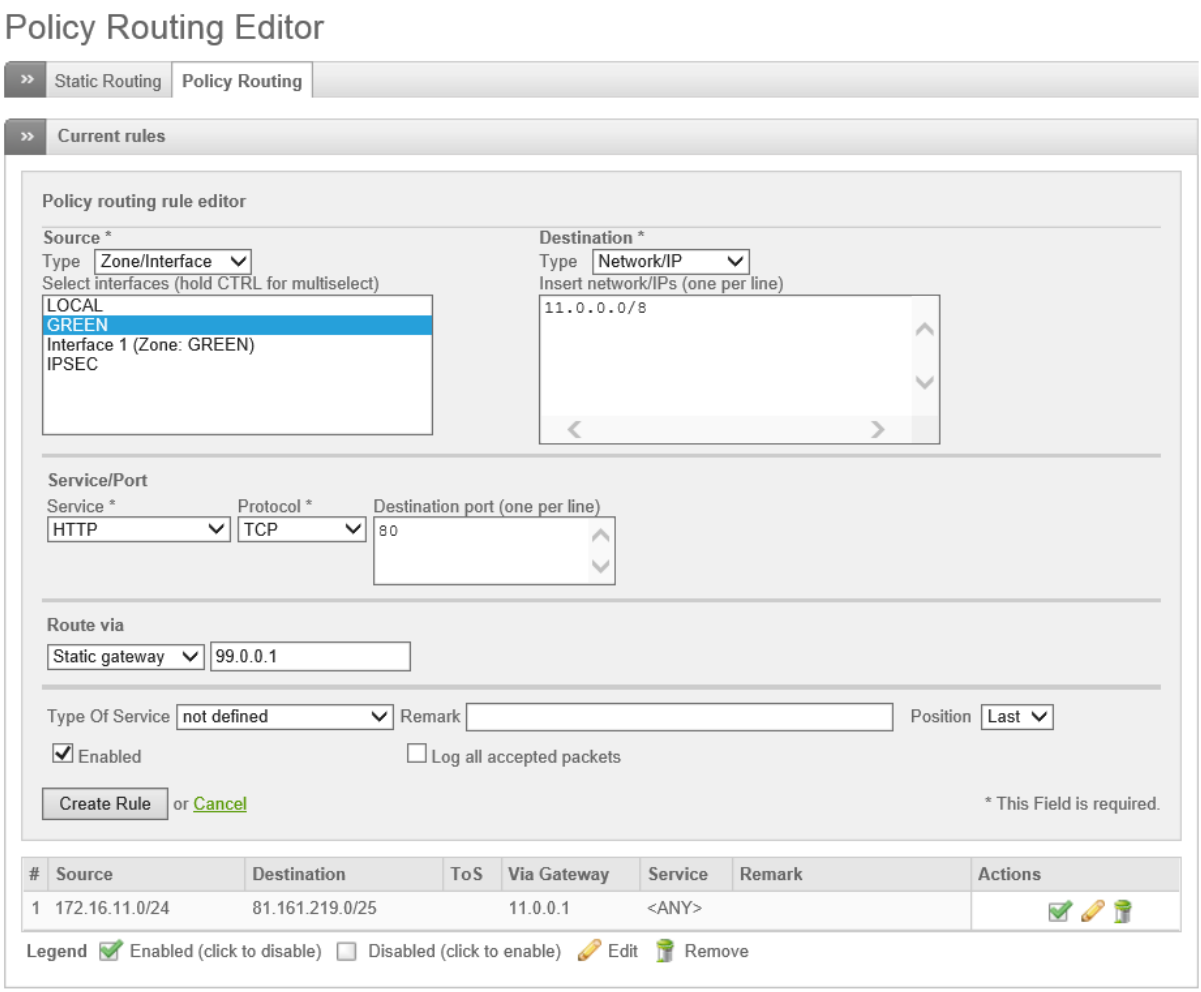
За да се разбере по-добре преноса на пакети през EFW CE е добре да се запомнят следните три основни правила, когато HTTP прокси е активно и има конфигуриран policy routing:

1. HTTP проксито използва основния uplink интерфейс (интерфейса, който води към червената зона);
2. HTTP проксито разделя връзката от клиента към отдалечения сървър на две – една от клиента към EFW и втора от EFW към сървъра;
3. Policy routing правилата се изпълняват след като трафика премине през HTTP проксито.

За да се добави ново правило е необходимо да се избере опцията “Create a policy routing rule” и да се въведат:

- Source – източник на трафика. Възможно е да се избере една от следните опции <ANY>, Zone/Interface, OpenVPN User, Netwrok/IP или MAC. Първата възможност (<ANY>) обхваща всички потенциални източници на пакета, втората (Zone/Interface) предоставя възможност за избор на дадена зона или интерфейс, свързани към EFW CE. OpenVPN User се отнася за VPN тунелите, а комбинацията Network/IP позволява чрез CIDR нотация да се опишат мрежи или точно определени хостове. Последната опция MAC дефинира списък от MAC адреси, които се използват за еднозначно определяне на източника на рамката;
- Destination – получател на трафика. Възможно е да се избере <ANY>, OpenVPN User или Network/IP, като значението на тези параметри е аналогично на това при източника (source), но се отнасят за получателите на пакетите;
- Service/Port – избор на услуга или на транспортен протокол и порт за трафика към отдалечената система. Налични са предварително зададени комбинации, като например HTTP, която съответства на TCP и порт 80. Възможните протоколи са TCP, UDP, TCP и UDP, ESP, GRE, ICMP или <ANY>. Портовете могат да се запишат поединично, като списък със запетая или като група от последователни стойности чрез посочване на първия порт, символа “:” и последния порт (включително) – например 1000:1100;
- Route via – аналогично на статичните пътища са възможни три варианта за пренасочване на пакетите – Static gateway, Uplink, OpenVPN user. Ако е избрана опцията Static gateway е необходимо да бъде въведен адрес на шлюз. При избор на uplink на страницата се добавя падащо меню, от което се посочва изходящия мрежови интерфейс. Ако към отдалечената мрежа се използва OpenVPN се избора последната възможна опция и отново от падащото меню се посочва избрания тунел;
- Type Of Service (TOS[63]) – тип на услугата, която се използва за приоритизиране на трафика. Възможните опции са not defined, default (0x00), lowdelay (0x10), throughput (0x08) и reliability (0x04);
- Remark – опционално текстово описание на правилото;
- Position – позиция на правилото;
- Enabled – ако опцията е активна и правилото е активирано;
- Log all accepted packets – информация за всички пакети, които се обработват от правилото се записва в журнала.

[63] За справка - en.wikipedia.org/wiki/Type_of_service



Фиг. 6.21 Конфигуриране на policy routing при EFW CE

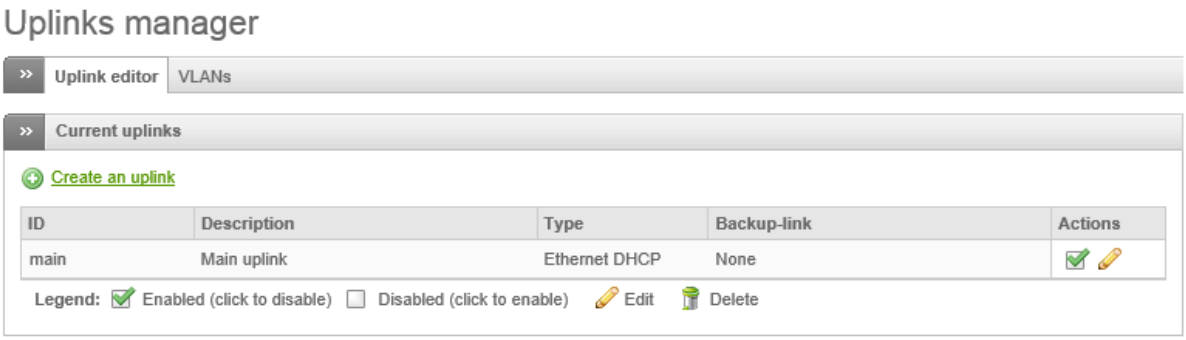
Въпреки, че добавянето на правилата изглежда по-сложно от статичните пътища, функционалността не е по-различна от гледна точка на конфигуриране – налични са повече опции.

Добър пример за конфигуриране на EFW CE с policy routing с цел балансиране на трафика е публикуван от Endian на адрес: **http://help.endian.com/entries/20059278**

В таблицата, с правилата отново има колона “Actions”, която съдържа функции за редактиране, изтриване, преместване в по-горна или по-долна позиция и за активиране или деактивиране. Направените промени трябва да се потвърдят с бутна “Apply” (поместен в зеленото поле в началото на страница), след което услугите на EFW CE, които имат отношение към направените конфигурационни промени се рестартират.

Интерфейси

Подменюто “Interfaces” съдържа функции, свързани с конфигурирането на uplink и VLAN.



Фиг. 6.22 Подменю “Interfaces” при EFW CE

Uplink editor

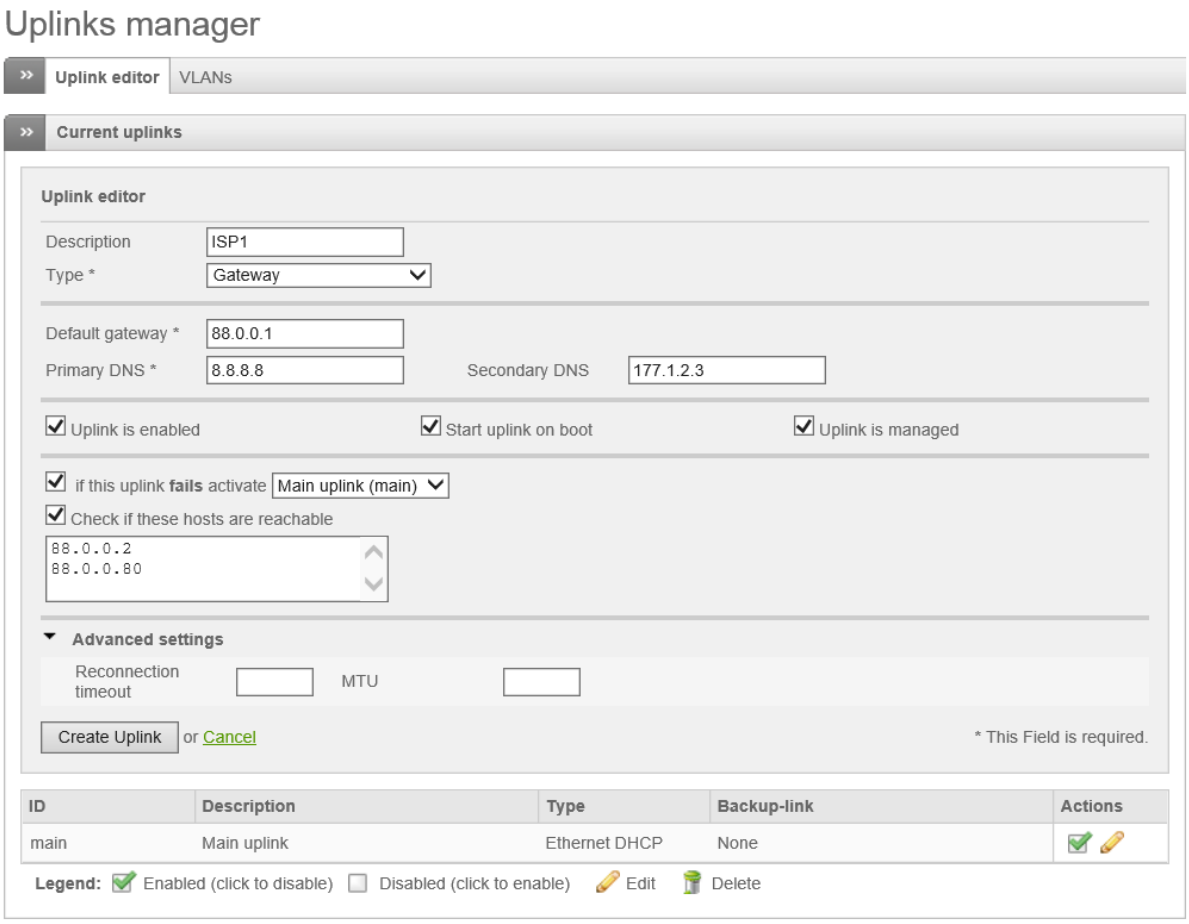
Подменюто “Uplink editor” предоставя достъп до функции, свързани с настройките на връзките към WAN или Интернет доставчици (uplink). Възможните действия са активиране или деактивиране на връзката, промяна на конфигурационните параметри, добавяне на нов uplink или изтриване. Едно съществено ограничение е, че main uplink не може да бъде изтрит.

Добавянето на нова връзка се осъществява от “Create an uplink”. Необходимо е да бъде конфигурирано:

- Description – опционално описание на връзката;
- Type – тип на връзката. Възможните стойности са ADSL modem, ANALOG/UMTS modem, ISDN modem, Ethernet DHCP, Ethernet static, Gateway, PPPoE или PPTP. В зависимост от направеният избор е необходимо да се въведат и допълнителните конфигурационни параметри;
- Uplink is enabled – ако опцията е активирана и uplink е активен;
- Start uplink on boot – стартиране на връзката при стартиране или рестартиране на EFW CE сситемата;
- Uplink is managed – позволява параметрите на връзката да бъдат управлявани;
- Reconnection timeout – интервал от време в секунди, след изтичането на който, ако не е изградена свързаност се опитва отначало;
- If this uplink fails activate – активиране на друга връзка при проблем със свързаността на текущото конфигурираната;
- Check if these hosts are reachable – проверка чрез ICMP на посочените в списъка хостове при повторено активиране на връзката. Ако хостовете не са достъпни връзката се счита за проблемна.
- MTU – maximum transmission unit.

След като параметрите са въведени добавянето на uplink е чрез натискане на бутона “Create uplink”.

Аналогично на пътищата в табличен вид са представени и наличните uplink. Чрез иконите в колоната “Actions” дадена връзка може да бъде редактирана, изтрита или активирана (деактивирана).



Фиг. 6.25 Конфигуриране на uplink при EFW CE

VLAN Manager

EFW позволява използването на технологията VLAN, като всяка VLAN може да бъде част от дадена зона. По този начин се получава допълнително сегментиране на мрежовата комуникация, което води до по-ефективно маршрутизиране и значително повишаване на сигурността.

Добавянето на VLAN се извършва от подменюто “VLAN Manager” чрез “Add new VLAN” и въвеждане на:

- Interface – интерфейсът, към който е свързана дадената VLAN;
- VLAN ID – номер на VLAN;
- Zone – зона.

След попълването на трите задължителни полета се натиска бутона “Add VLAN” и “Apply” и в зависимост от направената конфигурация EW CE рестартира необходимите системни услуги и процеси.

В таблицата са включени отделни редове за всеки VLAN като от иконата в колоната “Actions” е възможно да се изтрие определен ред.

Фиг. 6.26 Конфигуриране на VLAN при EFW CE

DHCP сървър

Вграденият в EFW DHCP сървър позволява динамично изпращане на IPv4 конфигурацията към хостовете в отделните зони. Възможно е тези параметри да бъдат конфигурирани по два различни начина:

1. **Dynamic** – конфигурацията се предоставя на хоста за определен период от време, след което се извършва нова DHCP заявка. Адресите се задават на база на заявки от мрежови сегмент;
2. **Fixed** – конфигурацията е предварително зададена за всеки хост и не се използва временно предоставяне (lease time).

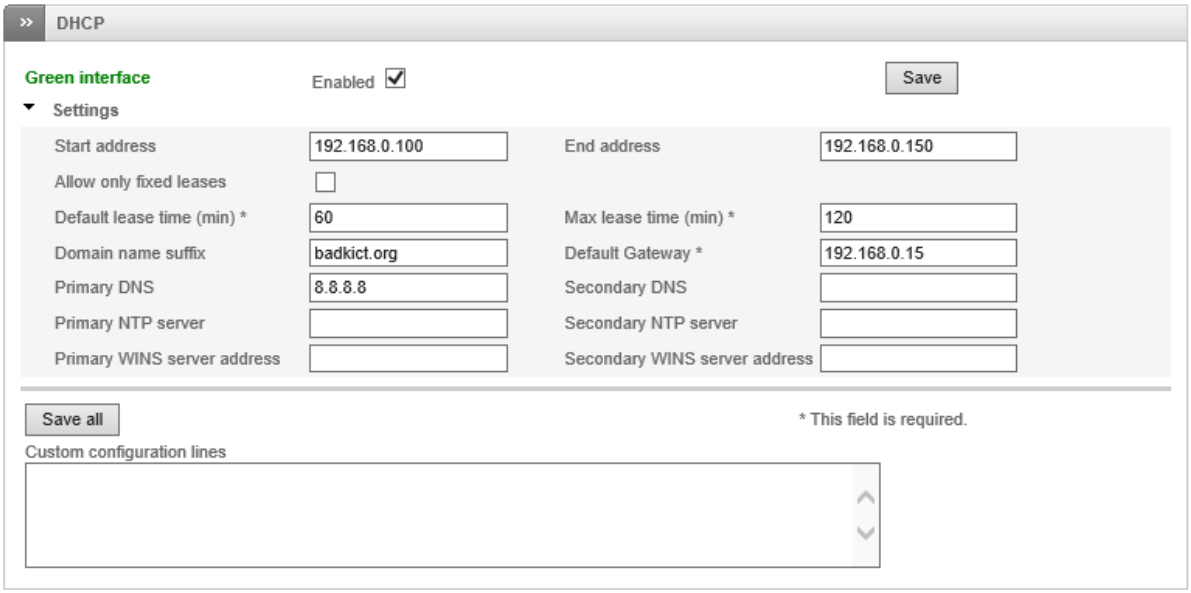
Конфигурирането на DHCP се извършва от менюто “Services” и подменюто “DHCP server”.

Фиг. 6.27 Съдържанието на страницата за конфигуриране на DHCP при EFW CE

Когато даден хост или мрежово устройство се свърже към мрежовия сегмент на определена зона (към която има конфигуриран DHCP сървър), и ако неговата IPv4 конфигурация не е зададена статично се преминава към процедурата за изпращане на заявка за адрес, маска, шлюз DNS и други параметри чрез протокола DHCP.

EFW CE предоставя възможност за стартиране на DHCP сървър за зеления, оранжевия и синия интерфейс, а конфигурационните параметри са:

- Enabled – активира или деактивира DHCP сървъра в дадената зона;
- Start address – начален адрес от списъка с динамично раздаваните към DHCP клиентите;
- End address – последен възможен адрес от списъка с динамично раздаваните към DHCP клиентите;
- Allow only fixed leases – изключва възможността за временно предоставяне на IPv4 конфигурацията на клиента;
- Default lease time – интервал от време в минути, който определя времето след което клиента трябва да заяви отново DHCP конфигурация на своите IPv4 параметри;
- Max lease time – аналогичен на “Default lease time” интервал от време, но определящ максималното изчакване;
- Domain name suffix – име на DNS домейна;
- Default Gateway – шлюз за мрежовия сегмент;
- Primary DNS – първи DNS сървър, използван от устройствата в дадената зона;
- Secondary DNS – втори DNS сървър, използван от устройствата в дадената зона;
- Primary NTP server – първи NTP сървър, използван от устройствата в дадената зона;
- Secondary NTP server – втори NTP сървър, използван от устройствата в дадената зона;
- Primary WINS server address – адрес на първи WINS сървър за дадената зона;
- Secondary WINS server address – адрес на втори WINS сървър за дадената зона;



Фиг. 6.28 Конфигурационни параметри на DHCP сървъра

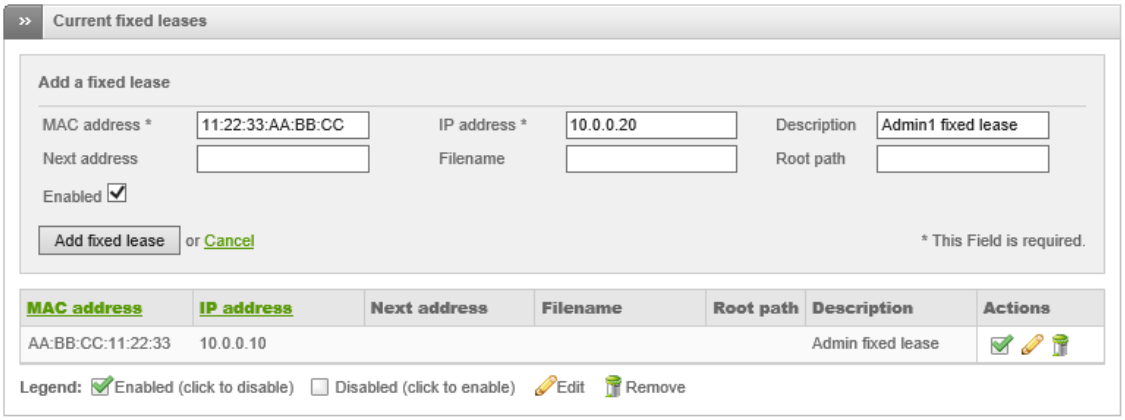
След въвеждане на необходимите параметри за да се потвърди направената конфигурация на DHCP сървъра за зоната се натиска бутана “Save all”. EFW CE стартира или рестартира необходимите системни услуги.

За да се добавят допълнителни команди към файла dhcpd.conf може да се използва полето “Custom configuration lines”. Важно е да се отбележи, че не се извършва проверка на въведените допълнителни опции, което може да доведе до проблеми при стартирането или рестартирането на демона dhcpd.

За разлика от DHCP при дефиниране на “fixed leases”, конфигурацията не се предоставя на хоста за определен интервал от време, а за постоянно. Добавянето на нова фиксирана конфигурация се извършва след натискане на връзката “Add a fixed lease” и въвеждане на:

- MAC address – MAC адресът на устройството клиент;
- IP address – IPv4 адресът, който винаги ще бъде изпращан на посочения по MAC адрес клиент;
- Description – кратко описание;
- Next address – адрес на TFTP сървър. Този параметър рядко се използва, но е полезен при VoIP телефони, които трябва да използват TFTP за изтегляне на конфигурационни файлове;
- Filename – отново рядко използвана опция, която съдържа името на файла за първоначално зареждане (boot file) от устройството клиент;
- Root path – пътя към boot file;
- Enabled – дадената фиксирана конфигурация е активна или неактивна;

Въведените параметри се записват към списъка след натискане на бутона “Add fixed lease”.



Фиг. 6.29 Параметри за фиксирана DHCP конфигурация

В таблицата под полетата за конфигуриране е поместен списък с всички въведени фиксирани комбинации от адреси. С иконите от колоната “Actions” даден ред може да бъде изтрит, редактиран или активиран (деактивиран).

Последната таблица от страницата за конфигуриране на DHCP съдържа списък с активните към момента динамично раздадени адреси.

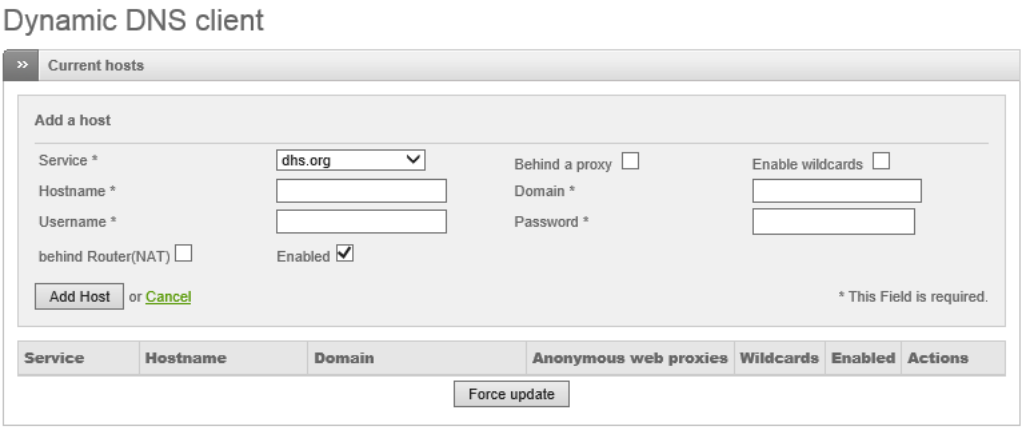
Фиг. 6.30 Поле с активни DHCP клиенти (в примера няма раздадени адреси)

Dynamic DNS

Dynamic DNS (DDNS) е услуга за автоматично обновяване в реално време на записите на DNS сървърите, което предоставя възможност и за автоматична промяна на DNS параметрите при смяна на IP адрес на дадено устройство. DDNS описва две технологии – първата е свързана с промяната на DNS информацията при работа с Интернет, а втората е оптимизиран софтуер за бързо опресняване на локалните DNS записи, използван от някои DNS сървъри. По често DDNS

се свързва с първия подход. За да се използва DDNS е необходимо потребителя да разполага с профил, към някои от доставчиците на тази услуга.

EFW CE съдържа възможност за DDNS работа в режим на клиент, като функциите са достъпни от менюто “Services” и подменюто “Dynamic DNS”.



Фиг. 6.31 DDNS конфигурация при EFW CE

Добавянето на нов хост се извършва чрез натискане на “Add a host” и въвеждане на:

- Service – доставчик на DDNS услуги;
- Behind a proxy – тази опция се отнася единствено за доставчика no-ip.com и трябва да се активира, ако EFW се намира зад мрежово прокси при достъпа към доставчика на Интернет;
- Enable wildcards – разрешава символите за обобщение (wildcard) при описание на домейните. Не всички DDNS доставчици имат подобна функционалност и е необходимо да се извърши проверка дали може да се приложат или не;
- Hostname – регистрираното име на хоста при DDNS сървъра;
- Domain – регистрираният домейн при DDNS доставчика;
- Username – потребителско име за достъп до DDNS сървъра;
- Password – парола за достъп до DDNS сървъра;
- behind Router (NAT) – опцията трябва да се активира, ако EFW е свързан към Интернет през NAT;
- Enabled – Активиране или деактивиране на конфигурацията на EFW CE за DDNS.

След приключване на въвеждането на параметрите и натискането на бутона “Add host” напавената конфигурация се активира и в таблицата в най-долната част на страницата се извежда списък с DDNS клиентите. Аналогично на повечето списъци с параметри при EFW CE от колоната “Actions” може да се избере действие (изтриване, редактиране, активиране и деактивиране).

Бутонът “Force update” стартира процедура за обновяване на информацията за DDNS.

За да работи правилно DDNS при определени топологии е необходимо да се пренасочат портове DNS към червената зона.

Time сървър

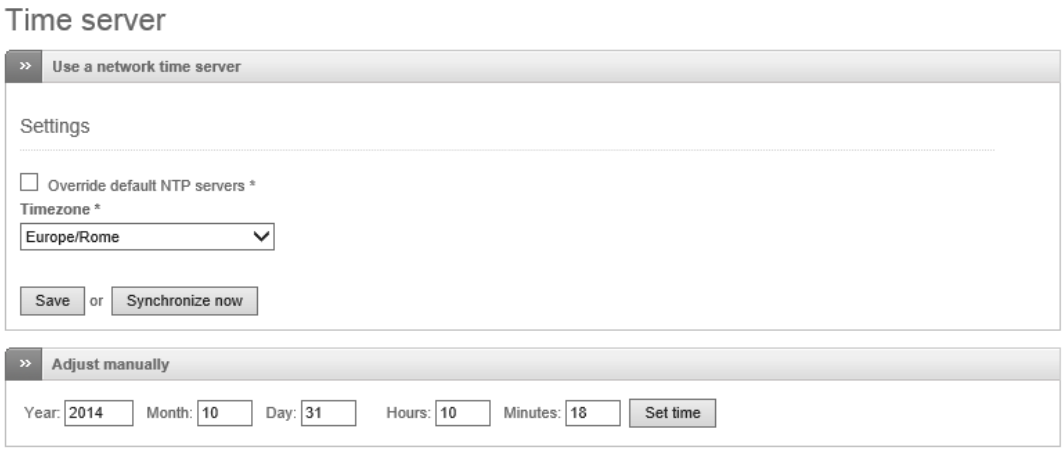
EFW CE използва протокола NTP за синхронизиране на текущата дата и час със специални сървъри в Интернет.

Настройките на тази функционалност са разделени в две групи:

1. Свързване към Интернет NTP сървър;
2. Ръчна настройка на датата и часа.

EFW CE автоматично определя NTP сървъра, който да използва на база на посочената в първоначалната конфигурация часова зона, която може да бъде променена и от тази страница.

Направените промени се записват при натискане на бутона “Save”, а чрез “Synchronize now” се извлича точното астрономическо време от NTP сървъра. Ако е необходимо датата и часа могат да бъдат въведени на ръка в полетата в най-долната част на страницата и новите стойности да се активират чрез “Set time”.



Фиг. 6.32 Конфигуриране на NTP при EFW CE

Ако от съображения за сигурност е необходимо да се изключи NTP сървъра на EFW CE е необходимо тази конфигурация да се извърши от Linux операционната система.

Заключение

Менюто “Status” съдържа няколко подменюта, чрез които работата на EFW CE може да бъде детайлно проследена и ако е необходимо да бъдат взети необходимите мерки за избягване на проблеми. Важно е да се извършва периодично наблюдение на натоварването на системата и ако то надхвърля определени прагови стойности да се разшири обема на наличната памет или да се мигрира към по-мощен хардуер.

EFW CE поддържа маршрутизиране със статични пътища, както и опцията “policy routing”, която дава допълнителен контрол върху трафика.

Ако се използват VLAN в мрежовата топология те могат да бъдат свързани към EFW CE, като необходимата конфигурация се извършва от менюто за интерфейси.

EFW CE може да се използва като DHCP сървър за отделните зони, както и да бъдат конфигурирани фиксирани конфигурации на IPv4 параметрите, които да се изпращат при заявка от определени хостове.

Ако е необходимо EFW CE може да бъде конфигуриран като DDNS клиент.

Актуалната дата и час по подразбиране се синхронизира през NTP със сървъри в Интернет.

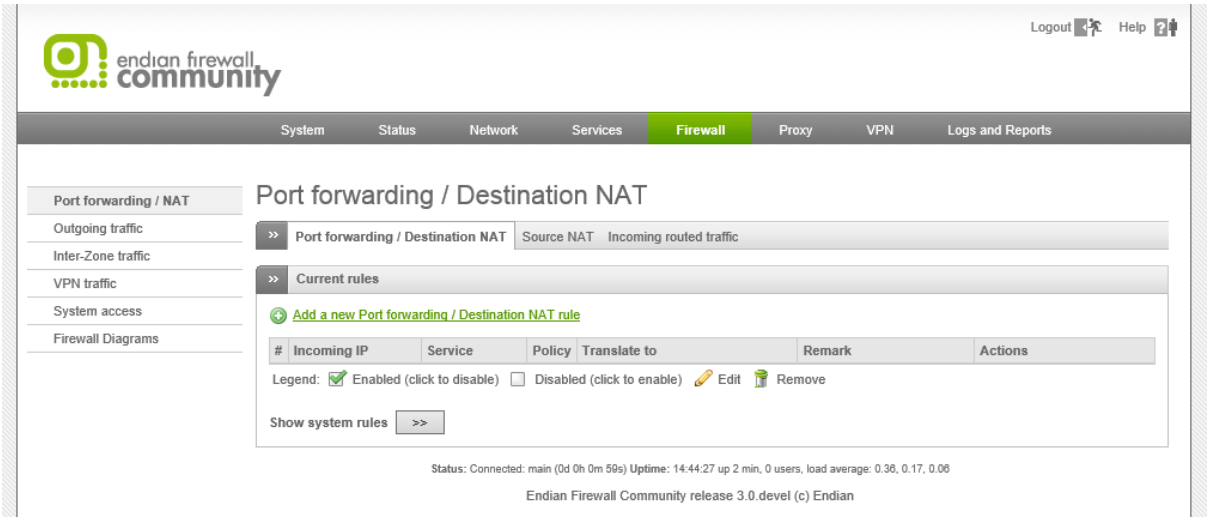
Глава 7. Проектиране и конфигуриране на динамична защитна стена (Stateful Firewall) и NAT

EFW CE поддържа базирана на зони защитна стена, която инспектира трафика до транспортното ниво спрямо референтния модел на OSI (т.нар. Stateful Firewall). Пренасочването на данните може да се извърши чрез статични пътища или по по-често използвания подход чрез технологията Network Address Translation (NAT), която позволява дадени "вътрешни" IPv4 адреси да бъдат транслирани към "външни" публични адреси[64]. Комбинацията от защитната стена и NAT позволява значително да бъде повишена сигурността на устройствата, поставени зад EFW (в зелената зона). Като недостатък на NAT могат да се посочат необходимостта от прекъсване на TCP сесията и започването на нова (от NAT системата към отдалеченото устройство), липсата на пълно проследяване на пътя и на сесиите между двете крайни системи, които си комуникират, както и забавянето и редуцирането на скоростта на трансфер на пакетите.

Конфигурирането на NAT и защитната стена на EFW се извършва от менюто "Firewall", а наличните подменюта са:

- Port forwarding/NAT – конфигуриране на пренасочването на портове между отделните зони и NAT;
- Outgoing traffic – филтриране на изходящия трафик;
- Inter-Zone traffic – политики за определяне на разрешения и забранения трафик между дефинираните логически зони на EFW;
- VPN traffic – филтриране на трафика през VPN;
- System access – административен достъп до EFW;
- Firewall Diagrams – примерни диаграми на топологии със защитна стена.

Отделянето на конфигурирането на защитната стена и мрежовите интерфейси е направено с цел по-лесно управление и наблюдение на сложни топологии, както и за по ясно разграничаване на различните потоци от пакети през EFW.



Фиг. 7.1 Меню Firewall при EFW CE

[64] Най-често вътрешните зони при NAT използват адреси, дефинирани в RFC 1918, а външните – публични адреси.

В отделните подменюта има редове, които са задължителни и не могат да бъдат изтрити. Също така позицията на реда е от съществено значение при филтрирането на информацията. Редовете с по-малък номер (по-предна позиция) се проверяват първи и т.н.

В общия случай въведените правила се трансформират в команди за iptables[65]. Максимално детайлно конфигуриране на параметрите на защитната стена може да се извърши от shell на използваната от EFW CE Linux операционна система.

Основни конфигурационни параметри

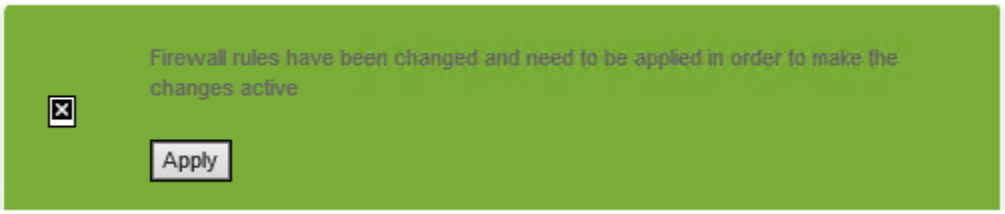
Съгласно документацията на EFW CE добавянето на нови правила в различните подменюта в голяма степен съдържа едни и същи параметри за отделните атрибути (например една и съща зона като източник на трафик и изходящ мрежови интерфейс), тъй като технологията която се използва в общия случай е iptables. Някои от по-важните термини, които се използват на няколко места в подменютата са:

- **Source** или **Incoming IP** – най-често този параметър се представя във вид на падащо меню, чрез което се задава типа на източника на трафика, който трябва да бъде анализиран. В зависимост от направения избор е възможно да бъде зададен параметър Network/IP/Range, който дефинира мрежови адреси, отделен IP адрес или последователност от IP адреси (address range). Друга възможност е да се посочи VPN клиент или uplink. Също така е възможно даден хост да бъде еднозначно посочен чрез MAC адрес (при Ethernet интерфейс).
- **Destination** или **Target** – тази настройка отново е във вид на падащо меню и посочва получателя на трафика. Аналогично е възможно да бъде зададена зона, IP адрес, последователност от IP адреси, VPN потребител и др.;
- **Service**, **Port** и **Protocol** – под service се разбира комбинацията от протокол и порт (например HTTP се описва като TCP:80). Чрез тези параметри могат да се филтрират транспортния протокол и използвания порт. Най-често в страницата са включени две падащи менюта, от които първото позволява директен избор на service, а второто дефинира отделно протокола, но е необходимо да се зададе и порт или група от портове. Включените транспортни протоколи са TCP, UDP, TCP+UDP, ESP, GRE и ICMP;
- **Policy** – задава действието, което трябва да се извърши спрямо пакета, ако условията на проверката са изпълнени. Налични са четири възможни опции. Първата е Allow with IPS, при която трафика се пропуска, но се анализира чрез Intrusion Prevention System (IPS). Allow – пакетът се пропуска без да се инспектира допълнително. Deny – пакетът се отхвърля и не се изпраща съобщение за грешка (ICMP) към източника. Drop – аналогично на Deny, но източника на трафика се уведомява за отхвърлените данни;
- **Enabled** – определя дали даденото правило е активно или не. По подразбиране при създаване на ново правило тази опция е включена. Деактивирането на правила се извършва най-често при анализ на работата на защитната стена и отстраняване на проблеми;
- **Log all accepted packets** – данните за всеки пропуснат пакет се включват към журнала. Това може да доведе до драстично нарастване на размера на журналните файлове, най-вече при по-голям трафик. Винаги се препоръчва периодично да се проверява състоянието на журналите и дали има достатъчно дисково пространство на съответната файлова система;
- **Remark** – кратък текст, който се използва като коментар за правилото;

[65] linux.die.net/man/8/iptables

- **Position** – позиция на новото правило. По подразбиране новите правила се добавят като последни в списъка;
- **Actions** – чрез иконите от тази колона в таблицата може да се промени позицията на правило или то да бъде редактирано, активирано (деактивирано) или изтрито.

След като бъдат въведени необходимите параметри се изисква потвърждаване на конфигурацията чрез бутона “Apply”, който е поставен в горната част на страницата в зелено поле. При неговото натискане се рестартират необходимите системни услуги, като необходимото време зависи от производителността на използвания хардуер.



Фиг. 7.2 Активиране на направените промени

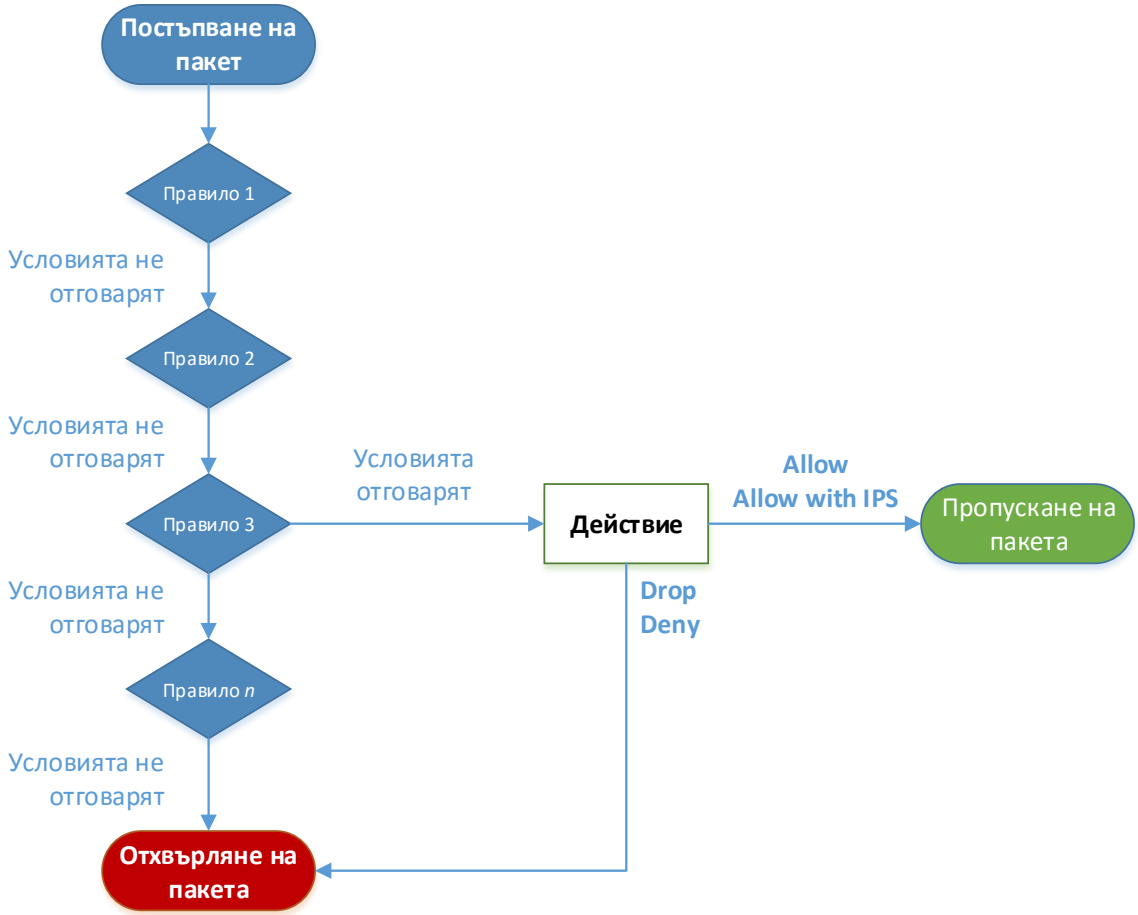
Правила и действия

Филтрирането на трафика при EFW се осъществява чрез два параметъра:

1. Условие за проверка;
2. Действие.

Проверката анализира пакетите на база на информацията до транспортното ниво на референтния OSI модел, което включва, физически и логически адреси на източника и получателя, портове за приложенията, които комуникират, транспортен протокол и при възможност състояние на сесията. При проверката се спазва последователността на дефиниране на правилата и първото, което отговаря на критериите активира посоченото действие.

Ако след стартиране на проверките данните за пакета не отговарят на нито една условие, то пакета се отхвърля. Това е рестриктивният модел на работа на защитните стени и ако е необходимо действието да се обърне (пакета да се пропусне, ако не отговаря на всички критерии) като последно правило трябва да се добави безусловно преминаване на всичкия трафик.



Фиг. 7.3 Последователност от проверка на условията на защитната стена

Необходимо е в зависимост от условията на проверката да бъде подбрано най-точното действие, свързано с дадения поток от пакети. Allow безусловно пропуска пакета, което води до най-бързото преминаване на трафика. Allow with IPS отново пропуска пакета, но преди да бъде изпратен на изходящия интерфейс той се сканира от IPS системата на EFW. Ако този анализ определи трафика като атака пакета може да бъде блокиран (действието на защитната стена е пропускане, но IPS не разрешава преминаването на данните). Това допълнително сканиране значително повишава сигурността, но намалява производителността. Действието Deny отхвърля пакета и за разлика от Drop не връща към източника ICMP съобщение, с което да го информира за филтрираните данни.

Allow е подходящо за случаите, когато е необходимо минимално негативно влияние върху скоростта на трафика и ако източниците са в надеждна зона (например зелената). Въпреки това винаги е препоръчително да се активира опцията Allow with IPS, дори ако източник е система, свързана в зелената логическа зона. Ако разгледаме пример, при който устройство, намиращо се в зелената зона е заразено със зловреден код, то може да има изтичане на конфиденциална информация, ако действието е Allow. Този зловреден код може да бъде засечен от IPS функциите на EFW CE и да се предотврати кражбата на информация.

Deny е подходящо действие за блокиране на адреси, които са участвали в доказани атаки или сканиране. Връщането на ICMP би дало възможност на злонамерените лица да разберат, че система по пътя блокира трафика, докато загубата на пакета без информация отново може да означава филтриране, но много по-трудно би било определено устройството, което е извършило това действие.

NAT и пренасочване на портове

По подразбиране EFW CE стартира NAT процес и транслира зелената зона към червената. Конфигурирането на NAT се извършва от менюто “Firewall” и подменюто „Port forwarding/NAT” на EFW CE (виж фиг. 7.1). Страницата съдържа три секции:

1. Port forwarding/Destination NAT – използва се за ограничаване на трафика от несигурни зони към сигурни (например от червената зона към зелената), както и за пренасочване на трафика по определен транспортен протокол и номер на порт към устройство, в надеждна зона;
2. Source NAT – SNAT е полезна технология, ако сървър е поставен в надеждна зона на EFW CE, и този сървър винаги е достъпен по предварително определен IP адрес от несигурна зона;
3. Incoming routed traffic – тази част на страницата предоставя възможност за управление на маршрутизирания през EFW трафик.

Destination NAT (simple mode)

Destination NAT (DNAT) най-често се използва с цел ограничаване на трафика от източници, намиращи се в несигурна логическа зона към такава с по-висока степен на надеждност. Дефинират се пренасочвания на портове, като за целта се описват параметри, свързани с източника, транспортния протокол и устройството, към което се препращат пакетите.

Добавянето на ново пренасочване на порт става след натискане на “Add a new Port forwarding/destination NAT rule” и въвеждане на три полета с данни:

1. Incoming IP – източник на трафика;
2. Incoming Service/Port – входящ транспортен протокол и порт;
3. Translate to – пренасочване към устройство.

В частта “Incoming IP” може да се посочи логическа зона, мрежа, IP адрес или последователност от адреси или OpenVPN потребител. Например, ако трафика винаги започва от несигурната червена зона се избира Type → Zone и интерфейс uplink main.

Транслираната услуга може да се посочи от списъка с предварително дефинираните комбинации от транспортен протокол и порт или да се въведат стойностите по отделно. Например за пренасочване на Microsoft RDP се въвежда протокол TCP и порт 3389.

Устройството, към което се пренасочва трафика се дефинира със следните параметри:

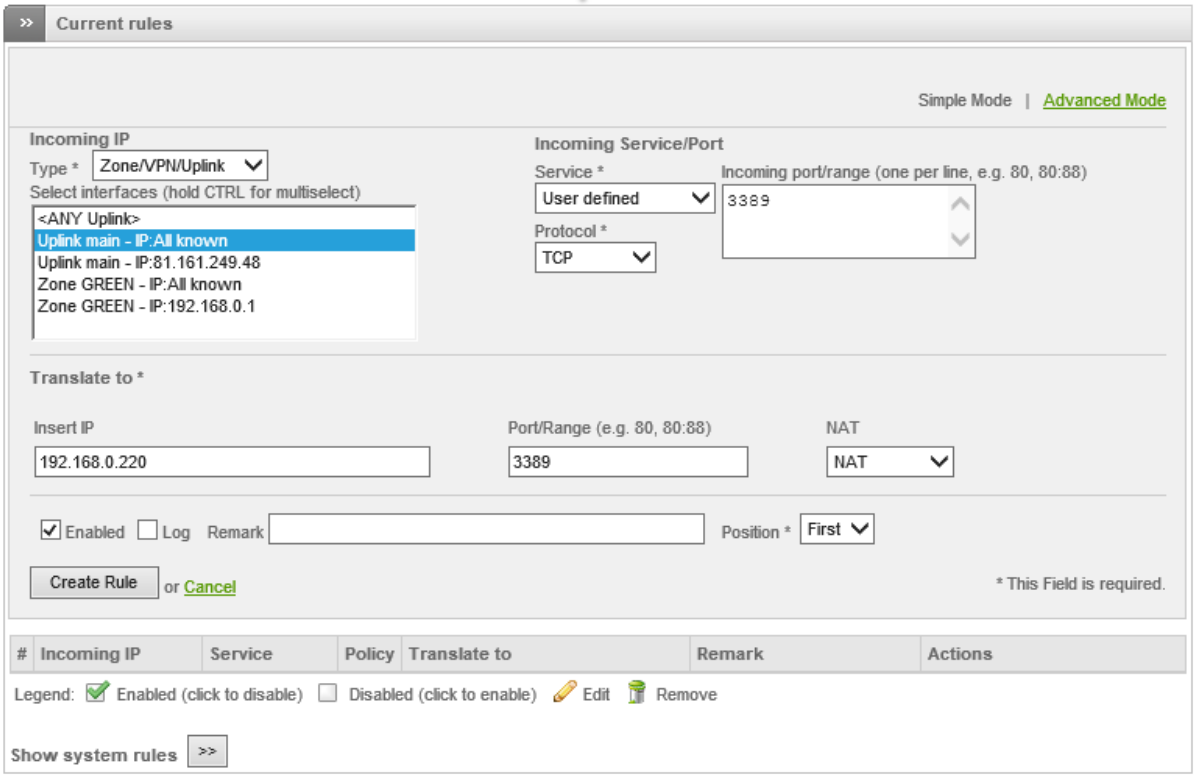
- Internal IP – IP адрес от вътрешна зона;
- PortRange – последователност от портове. Ако се въведат повече от 1 стойност може да се използва изброяване с “,” или последователност дефинирана от начален порт, символа “:” и последен порт (например 1000:1200).
- NAT – дали пакетите да преминават през NAT процеса или не.

Последната част от настройките е дали транслирането да бъде активно (Enable), дали да се записва в журналите информация за пакетите (Log) и опционален коментар (Remark).

Отново е възможно да се посочи и позицията на правилото, което рядко се налага, защото при пренасочването на портове най-често няма повтарящи се комбинации от параметри.

След попълване на данните и натискане на бутона “Create rule” се натиска бутона “Apply”, което води до рестартиране на необходимите системни услуги и активиране на правилата.

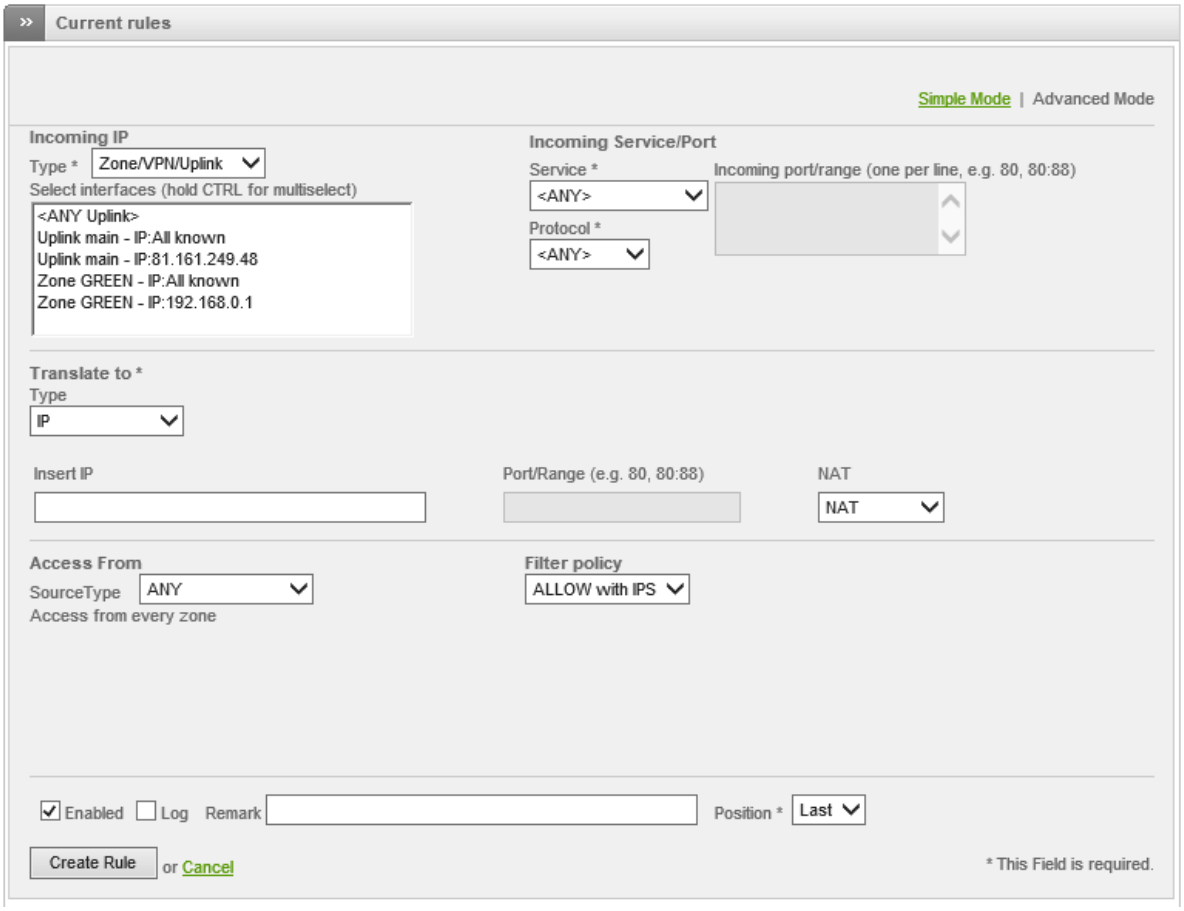
Всички въведени правила са показани в табличен вид, а чрез иконите в колоната “Actions” даден запис може да бъде активиран (деактивиран), редактиран или изтрит.



Фиг. 7.4 Пренасочване на портове при EFW CE (simple mode)

Добавянето или редактирането на пренасочването на портове може да се извърши в два режима:

1. Simple mode – опростен начин на конфигуриране, който предоставя само най-важните параметри;
2. Advanced mode – включва всички опции от simple mode, но допълнително може да се конфигурират “Access From” – параметър, показващ от коя зона, интерфейс, IP или кой VPN потребител може да използва правилото. Също така този режим включва и опцията “Filter policy”, която дефинира и действието на пренасочването - ALLOW with IPS, ALLOW, DROP и REJECT.



Фиг. 7.5 Пренасочване на портове при EFW CE (advanced mode)

Source NAT

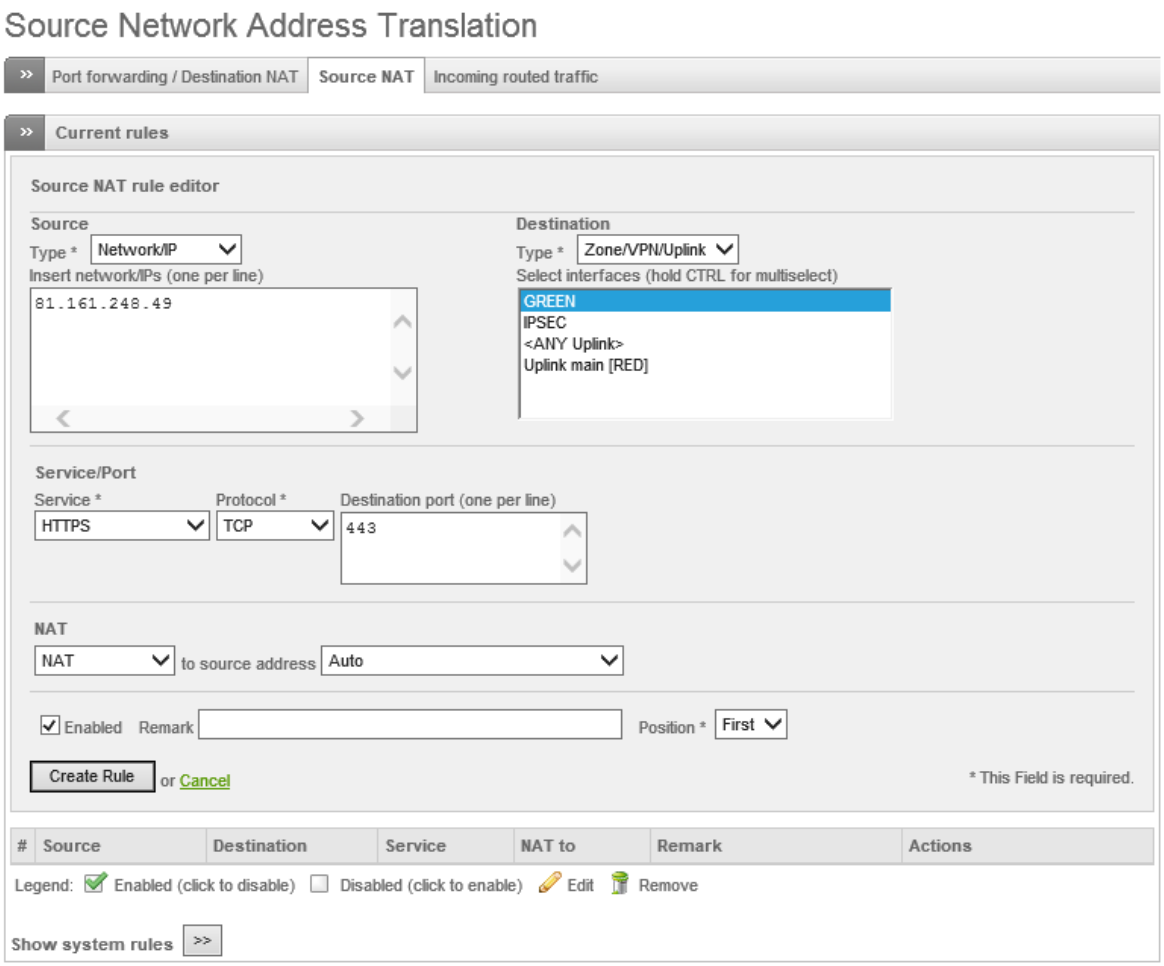
Source NAT (SNAT) се използва, ако в мрежата има свързан специализиран сървър или устройство, за което е наличен отделен IP адрес, различен от тези, използвани на червения интерфейс за NAT на вътрешните системи, но в същата IP мрежа. SNAT позволява EFW CE да бъде конфигуриран така, че трафика към сървъра да се пренасочва не по REDIP, а по допълнителния адрес. Добавянето на SNAT конфигурация се извършва от менюто “Firewall”, подменюто “Port forwarding/NAT” от секцията “Source NAT”.

След натискане на “Add a new source NAT rule” е необходимо да се въведат конфигурационни параметри:

- Source – източник на трафика, като възможните опции са Network/IP или VPN потребител. Поради спецификата на SNAT тук не е наличен избор на зона;
- Destination – пренасочване към зона, мрежа, IP адрес или VPN потребител;
- Service/Port – описание на услугата или конфигуриране на транспортен протокол и портове;
- NAT – наред със стандартните параметри това поле дефинира дали да се използва NAT, дали трафика да преминава без NAT. Налична е опцията Auto, която автоматично определя типа на пренасочването, както и “Map network”, която извършва статично транслиране на адресите на цяла мрежа към друга такава.

Когато отделните параметри се въведат записването и активирането на транслирането се извършва чрез бутона “Create rule” и след това чрез “Apply”.

Аналогично на по-голямата част от менютата на EFW CE всички въведени правила се показват в табличен вид и от колоната Actions могат да бъдат редактирани, активирани или деактивирани, както и изтривани.

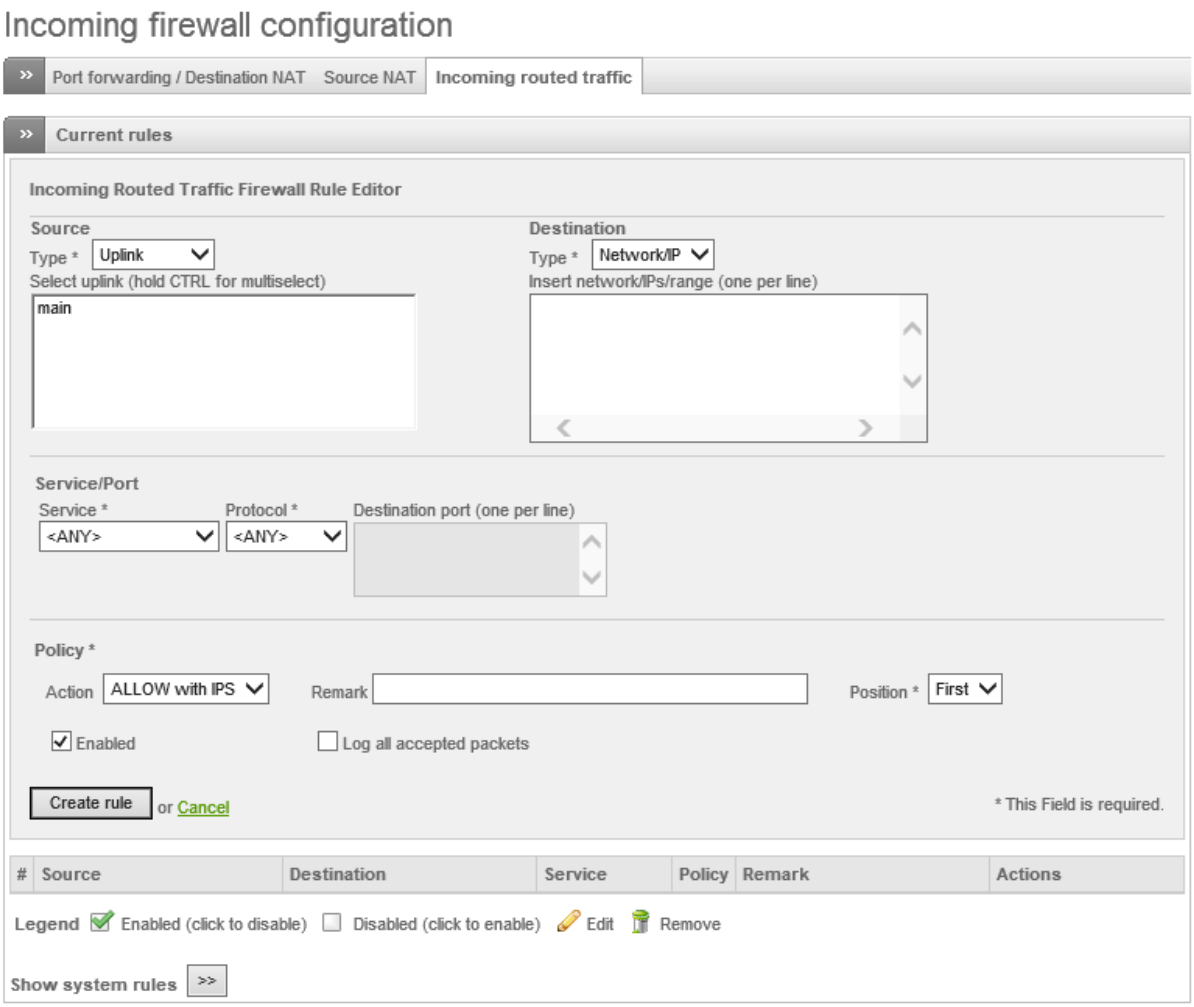


Фиг. 7.6 конфигуриране на SNAT при EFW CE

Филтриране на входящ маршрутизиран трафик

Последната секция на страницата от подменюто “Port forwarding/NAT” се използва за конфигуриране на пренасочването на входящите пакети, които са маршрутизирани през EFW CE. Тази опция е много полезна, ако част от публичните адреси се използва в зона (например DMZ) и трафика към тях не трябва да преминава през NAT.

Конфигурирането на отделните параметри е аналогично на това при описанието на защитната стена и отново всички правила са представени в табличен вид.



Фиг. 7.7 Филтриране на входящия маршрутизиран трафик през EFW

Филтриране на изходящ трафик

Едно важно правило, което вече няколкократно беше посочено, е че при защитните стени реда на правилата е от съществено значение, както и факта, че в общия случай ако нито една проверка не се изпълни трафика се отхвърля. Също така при инсталиране или първоначално конфигуриране на защитните стени или няма никакви предварително заложени правила или производителя дефинира подразбиращи се филтри на трафика, както е и случая при EFW.

След инсталиране на EFW CE изходящия трафик (зелен, син и оранжев интерфейс към червена зона) се анализира и филтрира на база на предварително заложени правила по подразбиране, които включват:

1. Трафикът от зелената и синята зони към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP и отдалечения порт е 80;
2. Трафикът от зелената и синята зони към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP и отдалечения порт е 443;
3. Трафикът от зелената зона към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP и отдалечения порт е 21;
4. Трафикът от зелената зона към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP и отдалечения порт е 25;

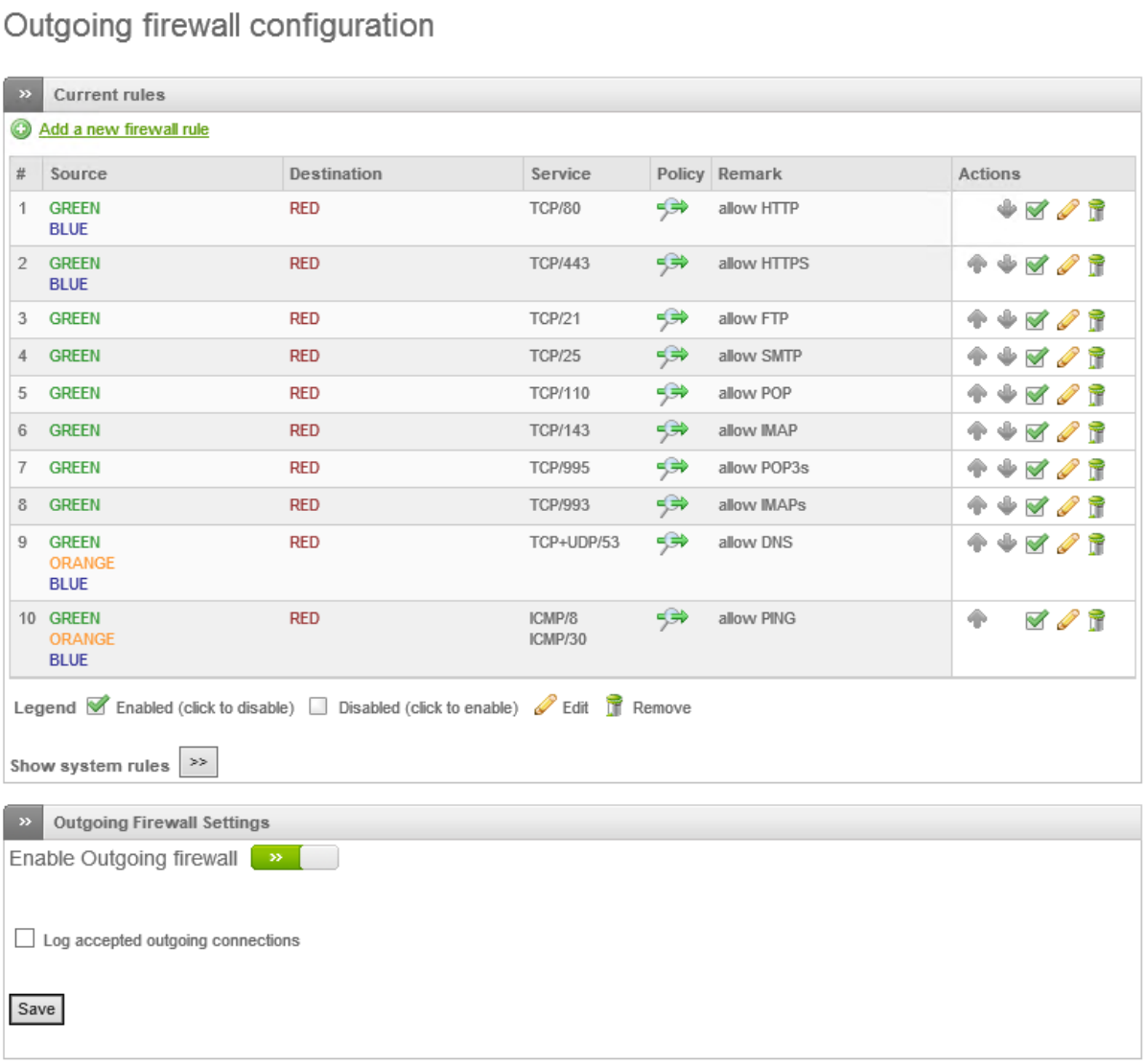
5. Трафикът от зелената зона към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP и отдалечения порт е 110;
6. Трафикът от зелената зона към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP и отдалечения порт е 143;
7. Трафикът от зелената зона към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP и отдалечения порт е 995;
8. Трафикът от зелената зона към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP и отдалечения порт е 993;
9. Трафикът от зелената, оранжевата и синята зони към червената е разрешен, но се инспектира с IPS, ако транспортния протокол е TCP или UDP, а отдалечения порт е 53;
10. Трафикът от зелената зона към червената е разрешен, но се инспектира с IPS, ако използваният протокол е ICMP, а услугите са 8 (Echo) и 30 (Traceroute).

Тези предварително дефинирани проверки са необходими за да са гарантира, че системните услуги ще работят безпроблемно след приключване на първоначалната инсталация и няма да се налага сложно допълнително конфигуриране.

По подразбиране защитната стена филтрира изходящия трафик, но ако е необходимо тази функционалност да бъде изключена от менюто “Firewall”, подменю “Outgoing traffic”, чрез бутона Enable Outgoing Firewall, който стартира или изключва услугата. Допълнителна опция е да се активира съхранение в журнала на данни за изходящите сесии, чрез активиране на “Log accepted outgoing connections”.

Направените промени се отразяват в конфигурацията след натискане на бутона “Save”, като не е необходимо да се потвърди допълнително (липса на “Apply” бутон).

В табличен вид са представени дефинираните правила към момента, като от колоната Actions те могат да бъдат премествани (реда е от съществено значение и се анализират от позиция 1 към последната), активирани или деактивирани, редактирани и изтривани.



Фиг. 7.8 Филтриране на изходящия трафик при EFW CE

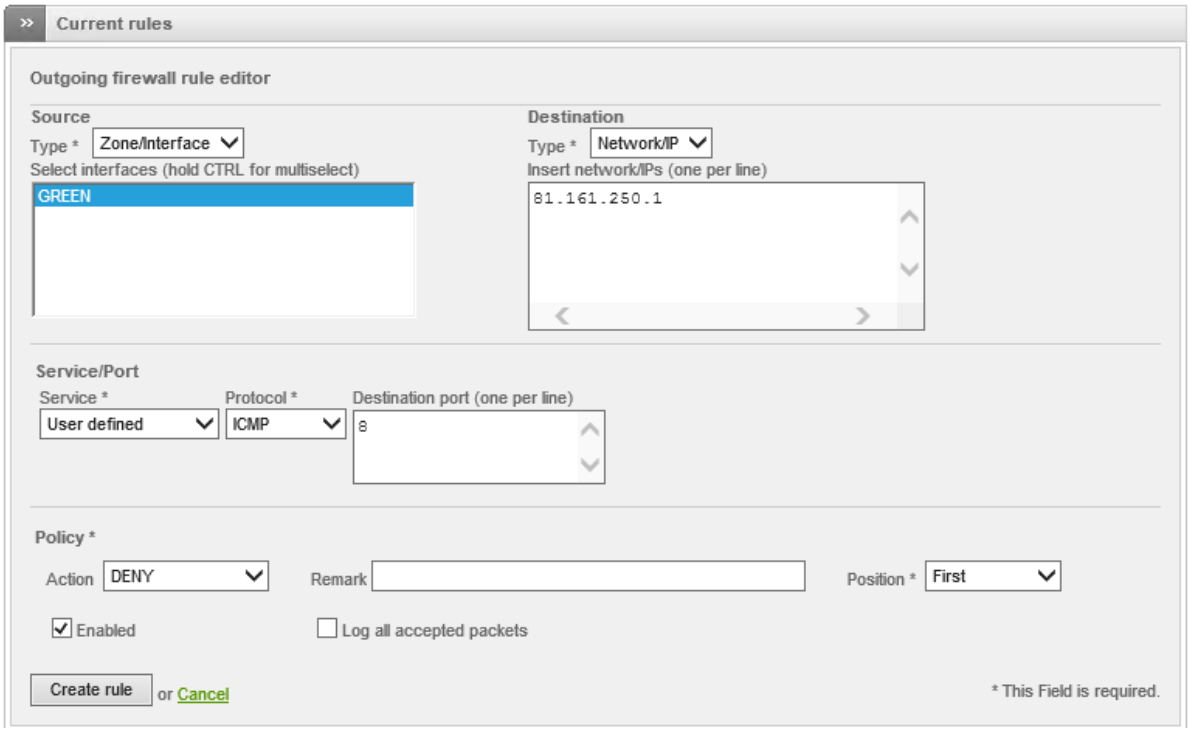
Добавянето на ново правило за проверка и филтриране на изходящия трафик се извършва чрез връзката “Add a new firewall rule” и въвеждане на:

- Source – източник на трафика (зона, адрес, мрежа, MAC адреси или всички);
- Destination – получател на трафика (мрежа или IP адреси, червен интерфейс или uplink);
- Service/Protocol – услуга или транспортен протокол и портове ва получателя;
- Action – действие, което може да бъде ALLOW with IPS, ALLOW, DENY или REJECT;
- Remark – коментар;
- Position – позиция на правилото, като по подразбиране новите правила се добавят след последното съществуващо;
- Enabled – правилото се разрешава или забранява;
- Log all accepted packets – в журналите се добавя информация за всеки пакет, който отговаря на посочените критерии.

След описание на правилото, за да се запише направената конфигурация се използва бутона “Create rule” и конфигурацията се потвърждава с “Apply”.

При дефиниране на ICMP проверка се използва типа на съобщението (ICMP Type[66]), като някои от по-важните стойности са:

- 0 – Echo Reply;
- 3 – Destination Unreachable;
- 4 – Source Quench (остаряло);
- 5 – Redirect;
- 8 – Echo;
- 9 – Router Advertisement;
- 10 – Router Solicitation;
- 11 – Time Exceeded;
- 12 – Parameter Problem;
- 13 – Timestamp;
- 14 – Timestamp Reply;
- 30 – Traceroute (остаряло).



Фиг. 7.9 Редактиране на правилата за проверка и действие при филтриране на изходящия трафик

Трафик между зоните на защитната стена

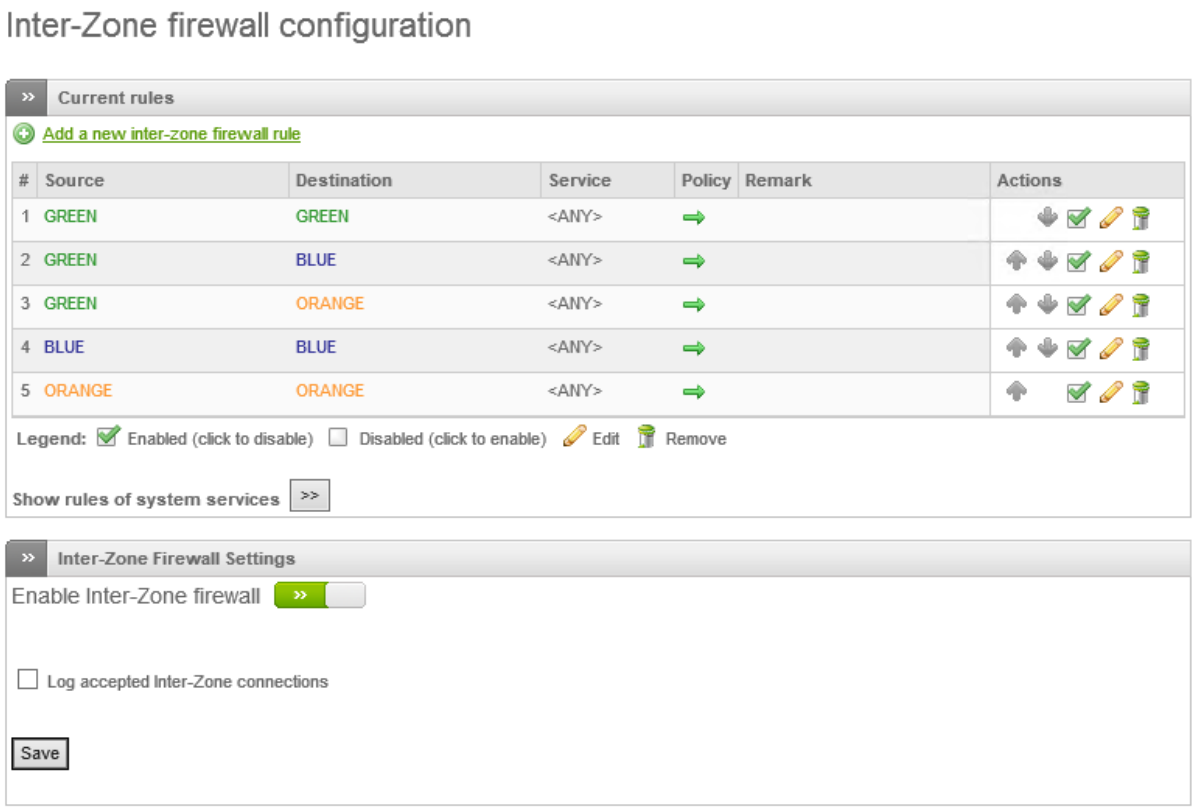
Филтрирането на трафикът между зоните на EFW CE е важна конфигурационна задача, която може да окаже съществено влияние върху работата на системата като цяло, както и на сигурността на информацията и на потребителите. Аналогично на изходящия трафик след приключване на процедурата по първоначалната инсталация се дефинират и активират правила по подразбиране, които пропускат пакети между наличните зоните, без червената, по следните критерии:

- Ако източникът и получателя се намират в зелената зона трафика е разрешен;
- Ако източника е в зелената зона, а получателя е в синята зона трафика преминава;

[66] http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml#icmp-parameters-types

- Ако източника е в зелената зона, а получателя е в оранжевата зона трафика се пропуска;
- Ако източникът и получателя се намират в синята зона трафика е разрешен;
- Ако източникът и получателя се намират в оранжевата зона трафика е разрешен.

Трафикът към червената зона се описва в секцията “Outgoing traffic”, както и в “Port forwarding/NAT”.



Фиг. 7.10 Правила за филтриране на трафика между зоните

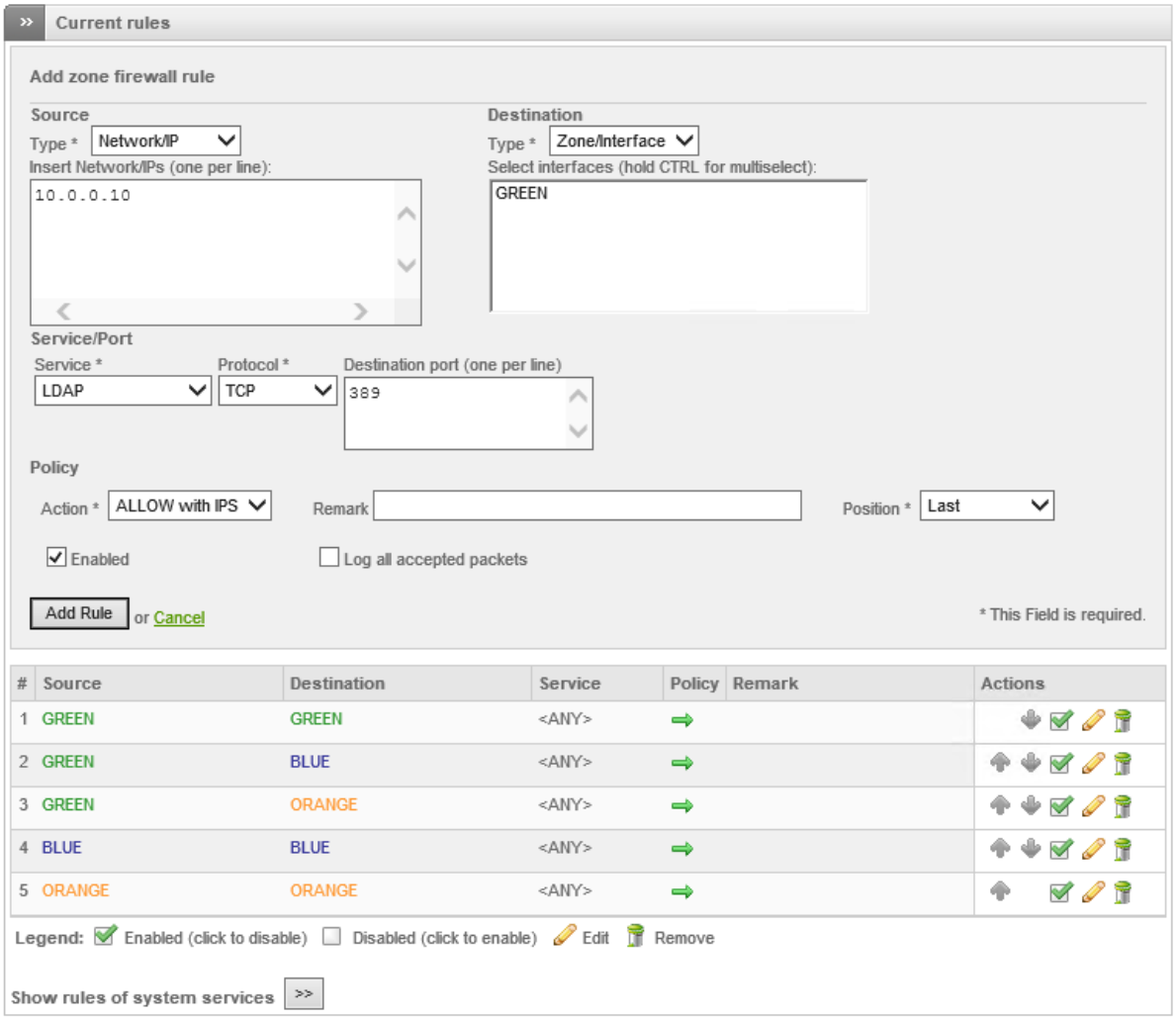
Дефинираните правила по подразбиране се достатъчни за правилната работа на EFW CE, но при необходимост те могат да бъдат деактивирани или изтрити от съответната икона от колоната Actions.

Добавянето на ново правило се извършва от “Add a new inter-zone firewall rule”, като е необходимо е да се дефинират:

- Source – източник на трафика, като възможните опции са ANY, зона или интерфейс, мрежа или IP адреси, както и MAC адрес;
- Destination – получател на пакетите. Възможно е да бъде посочен ANY, зона, мрежа или IP адрес(и);
- Service – услуга или комбинация от транспортен протокол и порт;
- Policy – действие, отнасящо се за трафика. Възможностите са аналогични на тези от настройките на изходящия трафик и включват ALLOW, ALLOW with IPS, DENY и REJECT;
- Remark – кратък коментар, описващ правилото за трафик между зоните;
- Position – позиция на правилото. По подразбиране и аналогично на разгледаните до тук менюта всяко ново правило се добавя като последно;
- Enabled – ако тази опция е активирана и правилото е активно;

- Log all incoming packets – В журналите се добавя информация за всички входящи пакети, които отговарят на даденото правило.

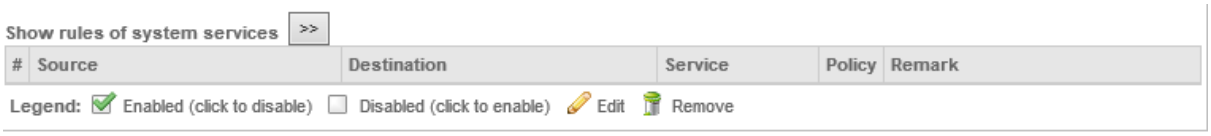
След като се въведат необходимите параметри чрез бутона “Add Rule” направената конфигурация се добавя и е необходимо да бъде потвърдена с “Apply”.



Фиг. 7.11 Редактиране на правилата за анализ при филтриране на трафика между зоните

Правилата, за трафик между зоните, дефинирани от системните услуги на Linux са описани в отделна таблица, която по подразбиране не е визуализирана. За да се види списъка с тези правила е необходимо да се използва бутона “Show rules of system services”.

Системните правила не могат да бъдат конфигурирани, активирани или изтривани от интерфейса за конфигуриране и наблюдение на EFW CE.



Фиг. 7.12 Правила за системни (Linux) услуги при EFW CE

Отново аналогично на изходящия трафик функцията на защитната стена за филтриране на пакетите между зоните може да бъде активирана и деактивирана чрез бутона “Enable Inter-Zone Firewall”. Опцията “Log accepted Inter-Zone connections” записва информация за всяка връзка между отделните зони в журналните файлове.

Фиг. 7.13 Активиране или деактивиране на анализа на трафика между зоните на EFW

VPN трафик

Подменюто “VPN traffic” позволява да се активира и конфигурира защитна стена за пакетите, пренасяни през VPN тунелите, конфигурирани на EFW CE. По подразбиране тази функционалност не е активирана и ще бъде разгледана подробно в темата за конфигуриране на VPN.

Фиг. 7.14 Активиране или деактивиране на анализа на VPN трафика при EFW CE

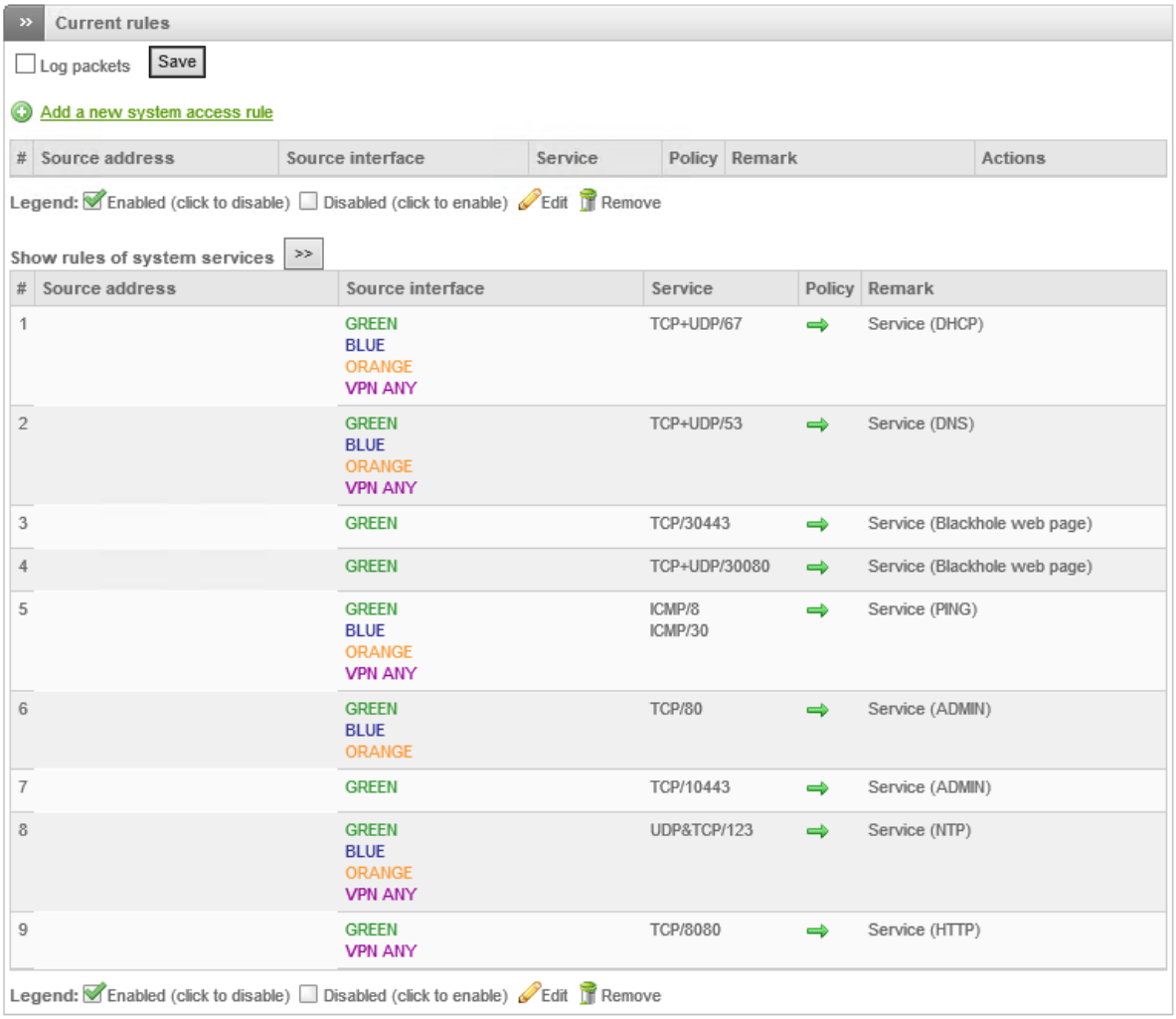
Отдалечен достъп до EFW CE

Административният достъп до самата EFW CE, както и трафика към някои специфични системни услуги може да се контролира от подменюто “System access”. Аналогично на разгледаните до момента функции за филтриране и тук са налични правила по подразбиране, като те са дефинирани от системни приложения и демони. Тези правила разрешават:

- Достъп от зеления, синия, оранжевия и лилавия интерфейс до EFW, ако транспортния протокол е TCP или UDP и порта на получаващото приложение (destination port) е 67 (DHCP);
- Достъп от зеления, синия, оранжевия и лилавия интерфейс до EFW, ако транспортния протокол е TCP или UDP и порта на получаващото приложение е 53 (DNS);
- Достъп от зеления интерфейс до EFW, ако транспортния протокол е TCP порта на получаващото приложение е 30443 (Blackhole web page);
- Достъп от зеления интерфейс до EFW, ако транспортния протокол е TCP порта на получаващото приложение е 30080 (Blackhole web page);
- Достъп от зеления, синия, оранжевия и лилавия интерфейс до EFW, ако протокола е ICMP, а услугата е 8 (echo) или 30 (traceroute);

- Достъп от зеления, синия и оранжевия интерфейс до EFW, ако транспортния протокол е TCP, а порта на получаващото приложение е 80 (страница за пренасочване към Web интерфейс на EFW);
- Достъп от зеления интерфейс до EFW, ако транспортния протокол е TCP, а порта на получаващото приложение е 10443 (Web интерфейс на EFW);
- Достъп от зеления, синия, оранжевия и лилавия интерфейс до EFW, ако транспортния протокол е TCP или UDP и порта на получаващото приложение е 123 (NTP);
- Достъп от зеления и лилавия интерфейс до EFW, ако транспортния протокол е TCP, а порта на получаващото приложение е 8080 (HTTP).

Добавянето на ново правило за достъп до EFW CE се извършва от “Add a new system access rule” и въвеждане на необходимите конфигурационни параметри, които са аналогични на тези при дефиниране на филтрирането на трафика между зоните или на изходящите пакети. Единствено трябва да се обърне внимание, че получателя (Destination) е точно определен интерфейс на EFW, uplink или VPN тунел.



Фиг. 7.15 Правила за административен достъп и специфични системни услуги при EFW CE

Диаграми на защитната стена

Последното подменю „Firewall diagrams” съдържа изображения, които в графичен вид показват начина на преминаване на пакетите между зоните, както и кой тип трафик е забранен и кой се анализира от EFW CE.

Фиг. 7.16 Диаграми на защитната стена

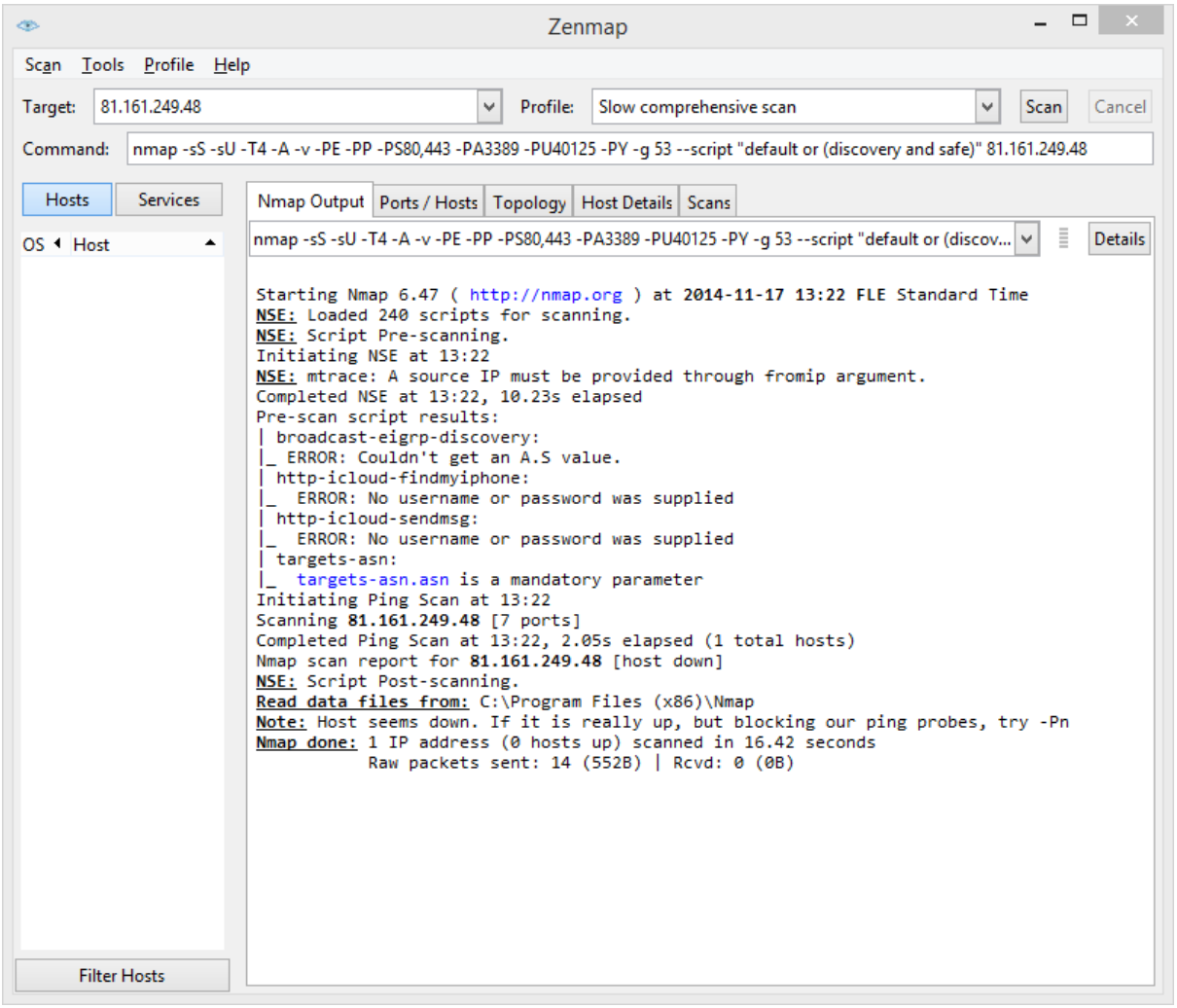
Методи за проверка на конфигурацията

Препоръчително е след извършването на първоначалната конфигурация на защитната стена и на NAT, както и след промяна на някои от правилата да се извършва проверка на получените резултати, която да анализира:

- Дали трафика между отделните зони преминава спрямо въведените правила и дали отговаря на предварително заложените цели?
- Дали трафика към червената зона преминава единствено за разрешените протоколи и услуги?
- При сканиране на EFW CE от червената зона дали има открити отворени портове, които не са пренасочени?

- Дали може да се определи типа на защитната стена (операционна система, сървърни приложения и др.) при сканиране от червената зона?

На фигура 7.17 е показано сканиране на EFW CE от червената зона. Вижда се, че резултатът от сканирането е отрицателен – хостът изглежда изключен или успешно блокира опитите за анализ.



Фиг. 7.17 Резултат от сканиране на EFW CE през червения интерфейс с nmap

Препоръки

1. Конфигурирането на правилата за филтриране на трафика между зоните, както и на NAT функциите трябва да се извърши внимателно и на база на корпоративната политика за защита и топологията на мрежата.
2. Винаги е препоръчително зададените критерии за филтриране на трафика да са максимално рестриктивни и да се пропускат само и единствено необходимите протоколи и услуги.
3. Задължително е журналите на защитната стена да се анализират внимателно и периодично.
4. Препоръчително е след всяка промяна на настройките на NAT и/или на защитната стена да се провежда тест на постигнатите резултати.

Заключение

Конфигурирането на NAT с EFW CE е лесно, а след приключване на процеса за първоначално конфигуриране се въвеждат и необходимите правила за безпроблемно пропускане на най-основния трафик от вътрешните зони към червената зона.

Конфигурирането на трафика към червената зона е обособено в отделно меню с цел по-лесно описание на правилата и по-голяма яснота на филтрирането на пакетите, преминаващи през EFW.

Повечето от елементите, които са свързани с конфигурирането на NAT и на защитната стена са еднотипни и описват източника на трафика, получателя на пакетите, услугата или комбинацията от транспортен протокол и порти, както и действието, което ще се извърши.

Необходимо е да се извърши предварителен анализ на политиката за защита, както и на топологията за да се конфигурират най-точните критерии за анализ.

Глава 8. Въведение в технологиите IDS/IPS и тяхното приложение с EFW CE

Сложността на злонамерените мрежови атаки, както и на използваните за тези цели инструменти непрекъснато нараства. Все повечето различни подходи и начини за опити за преодоляване на системите за мрежова защита води до трудно ограничаване на атакуващите устройства в даден мрежови сегмент. Следенето за наличие на атаки се препоръчва да се извършва във всички части на мрежата, като е необходимо трафика да се следи както във входяща, така и в изходяща посока. Защитните стени (работещи на 3, 4 и 7 ниво на OSI референтния модел) не предоставят необходимите механизми за предпазване при сложни атаки, което води до промяна на парадигмата за защитата на мрежовата архитектура. Необходимо е да се използват устройства, които предоставят висока производителност при анализ на сложните мрежови атаки, но и които да бъдат ценово изгодни. Такъв тип системи са Intrusion Detection System (IDS) и Intrusion Prevention System (IPS).

Една изключително опасна ситуация, която в общия случай отново не може да бъде анализирана с класическа защитна стена е т.нар. “zero-day” атака. Този тип заплахи се базират на най-новите открити технологични пропуски в софтуерни пакети или протоколи, които все още не са коригирани от съответните производители. Още един термин, свързан с “zero-day” е “zero-hour”, който дефинира момента, в който пропускът е открит и вече е известен. От “zero-hour” до пускането на коригирания софтуер от производителя или разработчиците мрежите могат да бъдат уязвими на атаки. Технологичните пропуски се обявяват на редица официални и “неофициални” сайтове, като един от най-популярните е www.exploit-db.com.

За да се защити мрежови сегмент от “zero-day” атаки е необходимо де се използва комплексен метод, както и сложни технологии, които най-често са налични и обединени в IDS/IPS системите.

Основи на IDS/IPS технологията

Въпреки, че технологията на защитните стени предоставя изключително висока степен на надеждност на защитата на мрежовите сегменти чрез филтриране на трафика, тя не може да предпази устройствата от “zero-day” атаки, особено в случаите, когато атаката се извършва с легитимни пакети (разрешени от правилата на защитната стена). Нуждата от подобряване на защитата води до разработването на технологията IDS, чието действие при компютърните мрежи може да се обобщи като извършване на активни опити за разпознаване и откриване на злонамерени атаки в мрежовия трафик.

Терминът IDS има по-общо значение и може да се отнася не само за мрежовите технологии, но и за анализ на други софтуерни процеси или опити за използване на системни ресурси без необходимите права.

Реализирането на IDS е софтуерно и изисква специално разработени алгоритми, които най-често използват сигнатури за откриване на атаките. Сигнатурите съдържат необходимите описания на атаката и могат да бъдат активирани или деактивирани от конфигурацията на IDS системата. Аналог на тези модули има и при антивирусните продукти – дефинициите на зловредния код.

История на IDS системите

Историята на IDS технологията е описана в редица източници, като SANS Institute в своята публикация “The History and Evolution of Intrusion Detection” систематизират основните етапи

през които преминава развитието. През 1972 година USAF[67] публикуват доклад, написан от Джеймс Андерсън, в който се дефинира, че USAF ясно осъзнават новите проблеми, свързани със сигурността при използваните от тях компютърни системи, които се отразяват на почти всички техни дейности. По това време USAF предоставят споделено използване на техните компютърни системи, което се базира на класифициране на достъпа, необходимост от познаване на използваните технологии от страна на потребителите и на ясно и стриктно дефинирани нива на достъп. Това води до един съществен проблем, който е налице и към момента – как да се дефинират отделни зони с различни нива на достъп без да се редуцира сигурността?

През 1980 година отново Джеймс Андерсън публикува доклад, в който описва методи за подобрявана на сигурността на компютърните системи, чрез проверка (audit) и наблюдение (surveillance) на клиентите. Този доклад се явява основополагащ за технологията Intrusion Detection (ID), която се използва първоначално за откриване на злоупотреба при работата на потребителите с мейнфрейм системи. При ID първата цел е да се дефинират заплахите, които съществуват за определената компютърна система. Това важи и за мрежовите IDS, където отново е необходимо да са известни заплахите, атаките и начина на генериране и пренасяна не пакетите. Андерсън описва необходимостта от анализ на риска, който да се използва като база за оценка на потенциалните заплахи и който да доведе до разработване на политика за защита.

Фиг. 8.1 Мейнфрейм система (източник Интернет)

В периода между 1984 и 1986 година Дороти Денинг и Питър Нюман разработват първия модел за IDS система, която работи в реално време, наречена Intrusion Detection Expert System (IDES). Първоначално IDES използва алгоритми, които се базират на предварително обучение от експерти с цел откриване на злонамерени действия. В последствие моделът е разширен до т.нар. Next-Generation Intrusion Detection Expert System (NIDES). Проучванията на Андерсън и моделите IDES и NIDES се считат като основополагащи за развитието на IDS технологията, във вида в който я използваме към момента.

IDS технологията непрекъснато се развива и в много литературни източници могат да се срещнат класификации, като:

- **Host-Based** – мрежовия трафик към дадено устройство (хост), както и отделните софтуерни процеси се анализират с цел откриване на следи за атаки;
- **Network-Based** – мрежовите IDS системи анализират трафика в дадени мрежови сегменти и отново се търсят признаци за наличие на атаки;

[67] United States Air Force

- **Откриване на аномалии** – IDS системата дефинира модел на нормална работа и търси отклонения от него. Един от недостатъците на този вариант на работа е изключително високия брой на грешно определени като атаки нормални действия (или трафик);
- **Анализ за злоупотреби (misuse model)** – IDS системата анализира и търси признаци за злоупотреба на база на предварително заложени правила.

Прилики и разлики между IDS и IPS

За да се изгради максимално ефективна защита е необходимо администраторите и мрежовите инженери ясно да разграничават IDS и IPS технологиите една от друга.

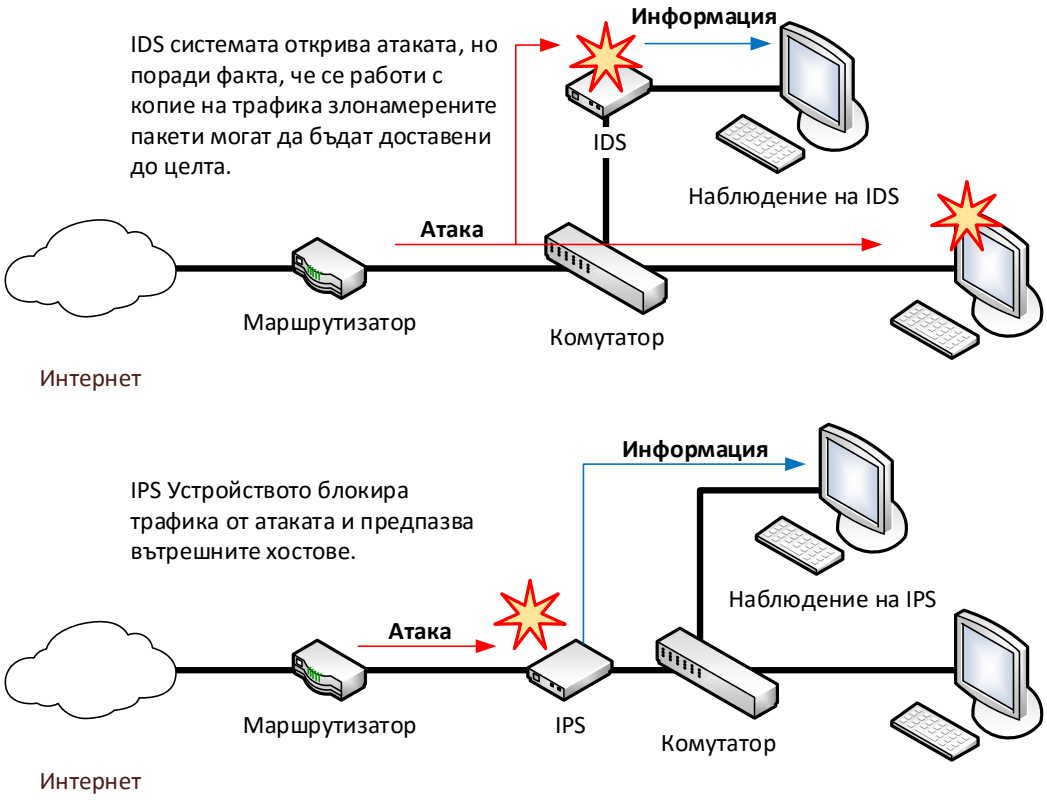
Основните **разлики** между технологиите IDS и IPS са:

1. IDS системите извършват сканиране на трафика в реално време, но работят с копие на пакетите;
2. IPS системата сканира трафика в реално време, като пакетите преминават през устройството;
3. IDS системата оказва минимално негативно влияние върху скоростта на пренос на данни и производителността на мрежата;
4. IPS технологията може значително да редуцира скоростта на обмяна на информация.

IDS и IPS си **приличат** по:

1. И двете технологии спомагат за повишаване на защитата на мрежите от комплексни атаки;
2. Използват сигнатури;
3. И двете технологии се инсталират като сензори;
4. Могат да имат вградени евристични алгоритми за анализ;
5. При грешна или неточна конфигурация могат да окажат негативно влияние върху сигурността;
6. Изискват периодично наблюдение и внимателно конфигуриране на използваните сигнатури и начини на сканиране;
7. Могат да докладват грешна информация (отчитат нормален трафик при атака и обратното) и др.

IPS технологията се базира на IDS, но работи в т.нар. “inline” режим, при който трафика преминава през устройството и при открита заплаха пакетите се блокират и не се препращат към останалите мрежови сегменти (виж фиг. 8.2). Това поставя изискване за анализ на нива 3 и 4 от OSI модела, но при някои по-сложни атаки е необходимо и проверка на ниво 7. Използването на сигнатури и вградените алгоритми могат да доведат до натоварване на устройството, което от своя страна да редуцира и пропускателната способност, забавяйки обмяната на информация от и към вътрешните мрежови сегменти.



Фиг. 8.2 Принцип на работа на IDS и IPS

Дали да се използва IDS или IPS? Отговорът е, че изборът зависи от няколко фактора, като по-важните са:

- Какво е желаното или необходимото ниво на сигурност?
- Какъв е толерансът при редуциране на скоростта на обмяна на данни?
- Какъв е бюджетът?
- Имат ли администраторите достатъчно знания и опит за конфигуриране на даденото устройство?

IDS е подходящ при необходимост от минимално забавяне на трафика, но при ясно условие, че засечените атаки могат да бъдат успешни, докато IPS е по-подходящ при повишаване на сигурността но за сметка на забавянето на трафика.

Отговорът на горният въпрос е, че в по-сложни мрежи се използват и двете технологии, в зависимост от изискванията към сигурността и скоростта на обмяна на данните за дадените сегменти.

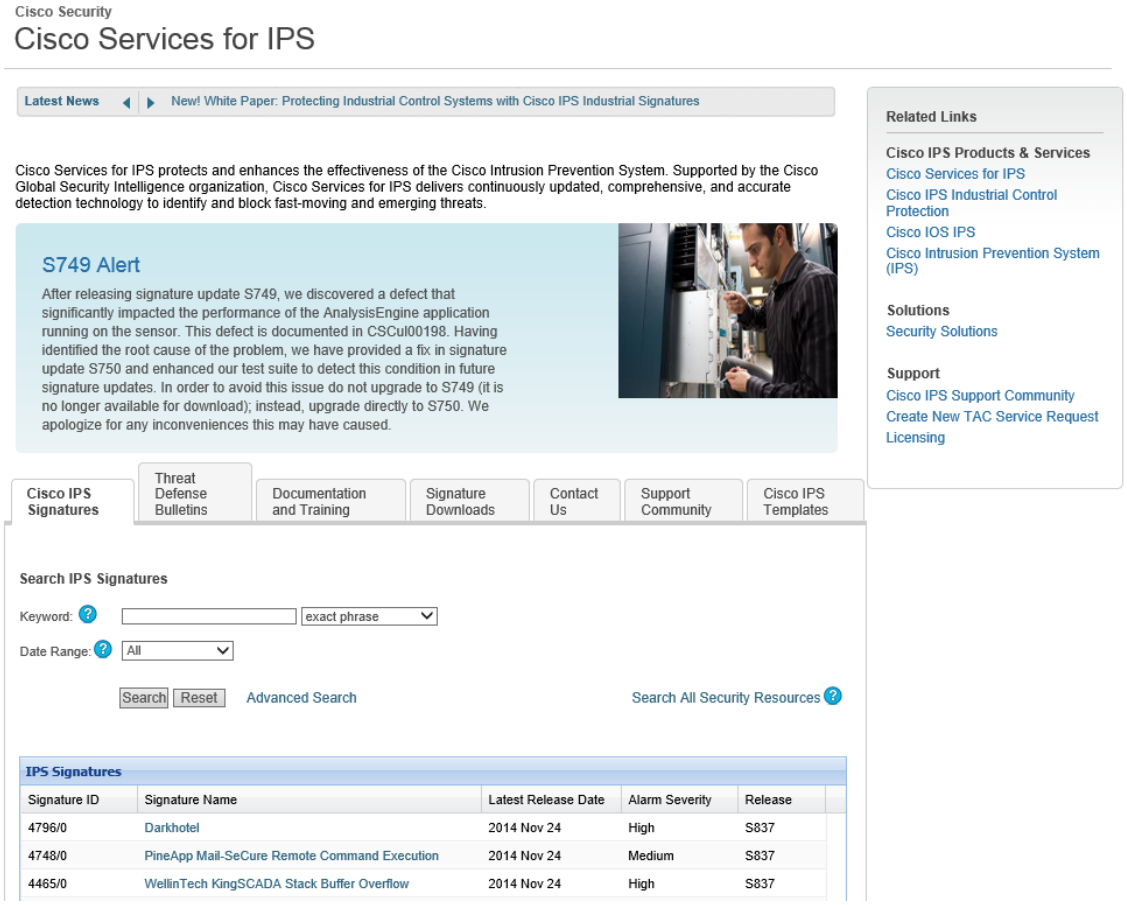
Сигнатури

И двете технологии - IDS и IPS използват специални сигнатури при анализ на трафика и откриването на потенциалните атаки. Сигнатурите могат да се опишат като набор от правила и данни, чрез който дадено действие се описва като безопасно или като злонамерено. Ако се разгледа анализът на мрежовия трафик и ако дадено злонамерено действие се разпознае като такова само от един единствен пакет сигнатурата се определя като “atomic”. При необходимост от сканиране на повече от един пакет се използват т.нар. “composite” сигнатури.

Веднага след откриването на нови заплахи повечето компании и разработчици на IDS/IPS системи пускат необходимите сигнатури, които да бъдат инсталиране на техните продукти. В интервалът от време от откриването на пропуска до наличието на сигнатура системите са уязвими, но след обновяването и активирането на сигнатурите могат да бъдат блокирани дори опасните “zero-day” атаки. Някои компании създават и осигуряват достъп до сигнатурите за критични атаки в рамките на няколко часа от откриването на заплахата, което е изключително важно особено за големи корпоративни клиенти.

В общия случаи сигнатурите са във вид на файлове (подписани с електронен подпис и/или надеждно шифрирани), които се изтеглят чрез подсигурена връзка от Интернет и със специална процедура (варира за различните производители) се инсталират на IDS/IPS устройствата. Повечето IDS/IPS системи поддържат опция за автоматично обновяване на сигнатурите, но е възможно това действие да се извършва от администраторите само при необходимост.

След като сигнатурите са инсталирани на устройството администраторите конфигурират кои от тях да бъдат активни и кои – не. Колкото повече сигнатури се използват за анализ на потока от информация, толкова повече се редуцира производителността на системата, поради необходимите системни ресурси. За да се получи максимално бързодействие, повечето IDS/IPS системи използват паралелни алгоритми и операции.



Фиг. 8.3 Сигнатури за Cisco IPS

Популярни IPS решения

Някои от най-популярните мрежови IDS/IPS системи са:

- FortiGate на FortiNet[68];
- IBM Security Network Intrusion Prevention System[69];
- Cisco Intrusion Prevention System[70];
- Snort[71] и др.

Фиг. 8.4 Фамилия от мрежови IPS системи на FortiNet (източник Интернет)

SNORT

Snort е мултиплатформена мрежова IDS/IPS система, която е създадена през 1998 г. от Мартин Роеш. Изходният код е написан на езика C, което позволява той да бъде компилиран и изпълняван под различни операционни системи. Най-често Snort се използва при Linux или BSD и по-рядко под Microsoft Windows.От 2001 година Snort се развива и разработва от Sourcefire, в която основател отново е Роеш. През месец октомври 2013 година Cisco Systems® закупуват Sourcefire и към момента те развиват и поддържат Snort.

През 2009 година Snort е отличен като един от най-значимите проекти с отворен код от InfoWorld и заема почетно място в т.нар. “Open Source Hall of Fame”.

Както вече беше дефинирано Snort е мрежова IDS система, която анализира трафика и отделните пакети чрез:

- Анализ в реално време на мрежовите и транспортните протоколи;
- Анализ на съдържанието на пакетите;
- Сравнение с правила.

[68] www.fortinet.com/products/fortigate
[69] www-03.ibm.com/software/products/en/network-ips
[70] www.cisco.com/c/en/us/products/security/intrusion-prevention-system-ips/index.html
[71] www.snort.org

Чрез тези техники Snort може да защити мрежовата комуникация до най-високото (приложното) ниво на OSI референтния модел.

Работният режими на Snort може да бъде конфигуриран като:

1. **Sniffer** – информацията за всеки отделен пакет се извлича и се визуализира в конзолата;
2. **Packet logger** – всички данни за пакета (хедър и информация) се записват в база данни на диска на системата;
3. **Intrusion detection** – трафика се анализира и на база на предварително конфигурирани правила се класифицира като безопасен или като заплаха.

Към Snort могат да бъдат добавени и допълнителни разширения включително такива, разработени от други производители на софтуер.

Предимства и недостатъци

Някои от по-важните **предимства** на системата Snort са:

- Поддръжката на различни платформи, сред които Linux, UNIX, BSD, Microsoft Windows и др.;
- Активна общност от потребители, разработчици и анализатори, която допринася за по-бързото генериране но нови сигнатури и правила. Също така големия брой инсталирани системи и честото докладване на новооткрити греши спомага за по-бързото отстраняване на технологичните проблеми. Като цяло Snort е изключително надежден и рядко може да доведе до проблеми в работата на системата за анализ на мрежовия трафик;
- Отвореният код не изисква лицензиране, а също така Snort може да бъде изтеглен напълно безплатно, като са налични и безплатни правила за анализ на трафика;
- Ако е необходимо да се използва комерсиална поддръжка на Snort редица компании имат разработени платени продукти на негова база и с тях може да бъде сключен договор за лицензиране на необходимия софтуер и за неговата поддръжка;
- Snort не изисква допълнителни софтуерни инструменти за анализ на пакетите и може лесно да бъде интегриран към вече изградена мрежова комуникационна инфраструктура или към защитна стена.

Някои от **недостатъците** на Snort са:

- Snort не е един единствен продукт, а комбинация от няколко софтуерни пакета;
- Грешно или неточно конфигуриране на правилата може да доведе до проблеми с комуникацията (това е валидно за повечето IDS/IPS системи);
  Необходимо е запознаване с документацията на продукта при изграждане на комплексни системи за анализ на трафика. Пълната документация на Snort е свободно достъпна на manual.snort.org.

SNORT правила

Snort използва максимално олекотени правила за анализ на трафика, които се дефинират чрез специализирано описание, което е изключително гъвкаво и в същото време предоставя богати възможности. До версия 1.8 всяко правило се дефинира на един ред, а след това е възможно де се използват няколко реда за описание на правилото, като новия ред се маркира със символа “\”.

Всяко правило е разделено на две логически части:

1. **Хедър** – описва действието (action), протокола, адресите на източника и получателя, както и транспортните портове;
2. **Опции** – съдържа допълнителни конфигурационни параметри, които дефинират, коя част на пакета да се инспектира и типа на генерираната аларма. Опциите се ограждат в скоби за да бъде правилото по-лесно разчетено.

Пример за Snort правило е:

```
alert tcp any any -> 10.0.0.0/8 (content: "|00 AA BB CC|"; msg:"tftpd64")
```

Опциите не са задължителни, но чрез тях може да се извърши много по-точно описание на анализирания трафик и на потенциалните заплахи.

Действието може да бъде:

- **Alert** – генериране на аларма в зависимост от избрания за целта метод и записване на данните за пакета в журнал;
- **Log** – записване на данните за пакетът в журнал;
- **Pass** – пакета се игнорира и пропуска през системата;
- **Activate** – генерира се аларма и се стартира ново динамично правила;
- **Dynamic** – правилото не е активно докато не се извърши специална заявка, след което трафика се инспектира от него и данните за пакета се добавят към журнал;
- **Drop** – пакетът се блокира и данните за него се записват в журнал;
- **Reject** – пакетът се блокира и се описва в журнал. След това се изпраща TCP reset (ако транспортния протокол е TCP) или ICMP Unreachable (при UDP);
- **Sdrop** – пакетът се блокира, но в журнала не се записва никаква информация.

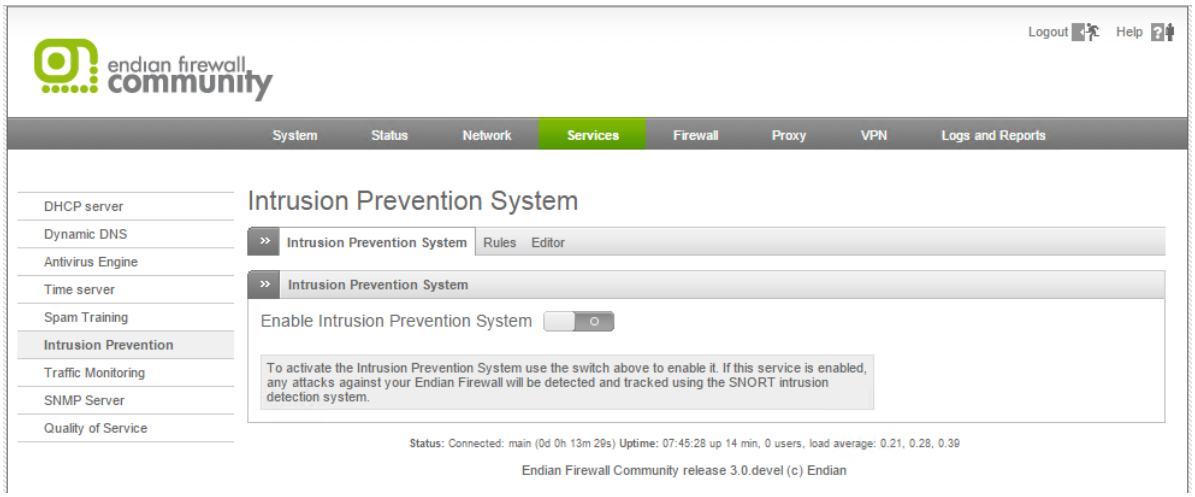
Правила за Snort могат да бъдат изтеглени от Интернет, като е важно техният източник да бъде внимателно проверен. Като основно правило може да се обобщи, че не се препоръчва да се използват сигнатури или правила при IDS/IPS от неясни източници.

IDS система при EFW CE

В Endian Firewall е вградена системата за превенция от атаки Snort. Тя е тясно интегрирана в системните iptables и позволява допълнително анализиране на трафика на база на конфигурираните правила.

По подразбиране IPS функциите на EFW са изключени и те могат да бъдат активирани и конфигуриране от менюто “Services” и подменюто “Intrusion Prevention”.

За да се активира IPS е необходимо плъзгащия бутон “Enable Intrusion Prevention System” да се постави във включено състояние. Важно е да се отбележи, че Snort се активира за всички зони на EFW.

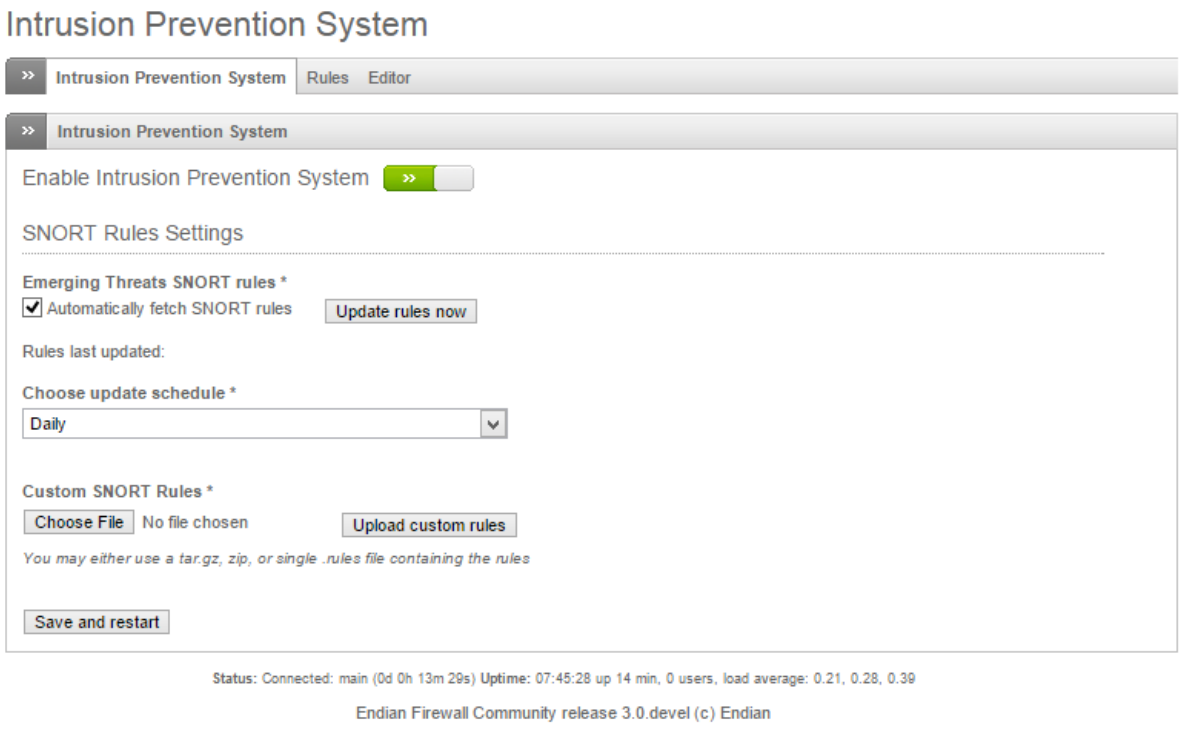


Фиг. 8.5 IPS система при EFW

Автоматично обновяване на правилата

След активирането на IPS се рестартират част от системните услуги и процеси. Препоръчително е непосредствено след това да се обновят правилата на Snort, както и да се дефинира периода за тяхното автоматично опресняване от Интернет.

Ако опцията “Automatically fetch SNORT rules” е активна правилата ще се обновяват автоматично спрямо посочения под полето интервал от време, който по подразбиране е ежедневно, а другите възможни стойности за времевия интервал са всеки час, всяка седмица или всеки месец. Ежедневната проверка е оптимална, но в случай на новооткрита сериозна заплаха може да се извърши и проверка за наличие на нови правила чрез бутона “Update rules now”, чиято функционалност е да стартира обновяване на правилата без да се изчаква да изтече посочения интервал от време.



Фиг. 8.6 Настройки на автоматичното обновяване на IPS правилата при EFW CE

След изтегляне на нови правила IPS системата се рестартира, което отнема известен интервал от време, който зависи от производителността на хардуера и по време на който може да се наблюдава по-сериозно използване на системните ресурси.

Датата и точния час на последното обновяване се посочват в “Rules last update”.

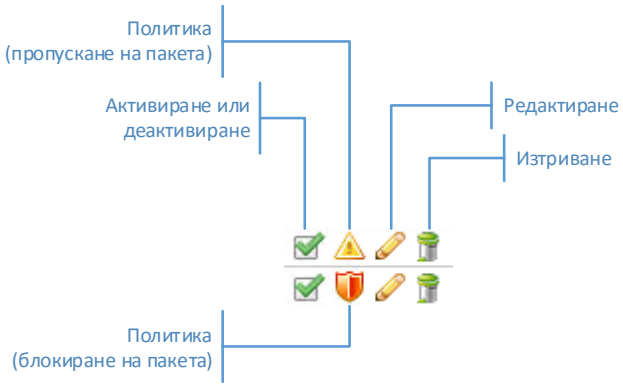
Потребителски SNORT правила

Ако е необходимо да се използват потребителски Snort правила (rules), те трябва да бъдат описани във файл, който да се изпрати към EFW чрез натискане на бутоните “Choose file” и “Upload custom rules” (виж фиг. 8.6).

IDS правила

Правилата са една от най-важните части на IPS системата и за да могат по-лесно да бъдат конфигурирани те са групирани, като всяка отделна група може да съдържа множество отделни правила, които имат определена функционалност или се отнасят до определен тип атаки или трафик.

Групите са представени в табличен вид, като от колоната “Actions” може да се избере определено действие, свързано с тях.



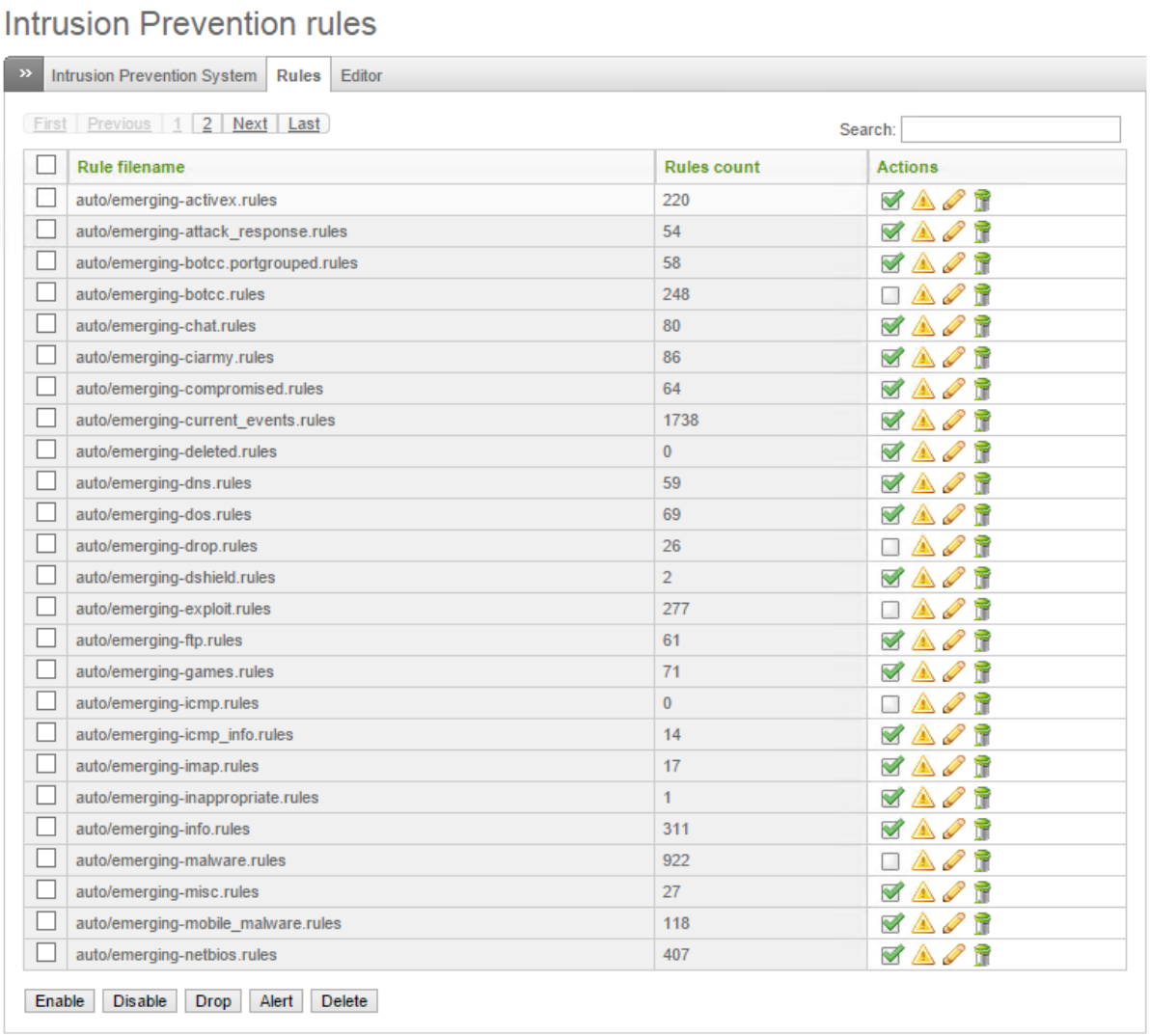
Фиг. 8.7 Икони за работа с правилата от колана “Actions”

Групи с IDS правила

Списъкът с правила може да се види от групата “Rules” в подменюто “Intrusion prevention”. Поради големият брой редове са налични няколко страници, като преминаването между тях става от препратките над таблицата.

От колоната “Actions” могат да бъдат активирани или деактивирани цели групи с правила, както и да се конфигурира действието (политиката) за цялата група. След промяна на параметрите се налага рестартиране на IPS функциите на EFW.

Отново над таблицата има текстово поле за търсене в списъка с правила.



Фиг. 8.8 Групи с правила при IPS на EFW CE

IPS Rule “Editor”

Всяка група с правила може да съдържа едно или повече, като освен активиране на цялата група EFW предоставя възможност и за избор на отделни елементи от нея. Тази функционалност е достъпна през страницата “Editor” в “Intrusion prevention”.

Аналогично на групите всяко отделно правило може да бъде активирано или деактивирано, както и да се посочи желаното действие от колоната “Actions”.

Изборът на група с правила става от списъка в най-горната част на страницата, като отново ако са налични голям брой редове таблицата се разделя на части, които са номерирани и са достъпни от препратките под списъка.

Под таблицата са поместени бутони, които могат да определят действието на правилото като:

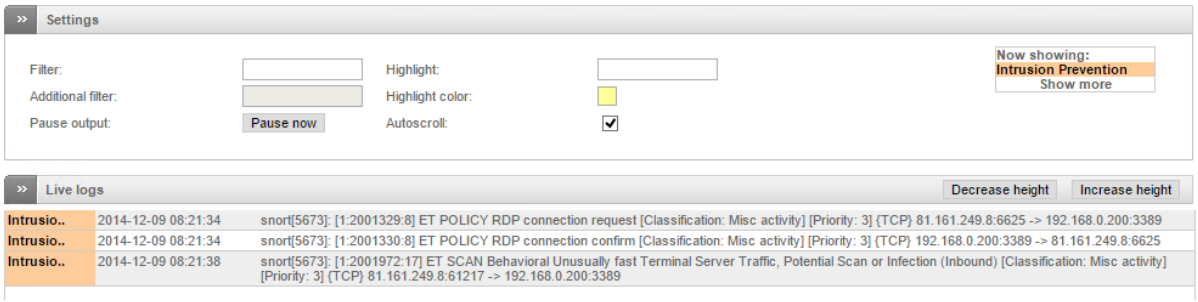
- Enable – активиране на правило;
- Disable – деактивиране на правило;
- Drop – пакетът се отхвърля;
- Alert – пакета се пропуска, но се генерира съобщение в журнала;

- Delete exception – изтриване на конфигурирано изключение.

За по-бързо откриване на дадено правило може да се използва полето за търсене в дясната част на страницата, непосредствено над таблицата.

Проверка на IDS/IPS системата

След конфигуриране на IPS системата е препоръчително да се извърши проверка на конфигурацията, тъй като грешно зададени правила могат да блокират легитимен трафик или да пропуснат атаки. Най-добрият вариант е IPS системата а се конфигурира на тестово устройство, което да се анализира чрез инструментите, вградени в Kali Linux, като се наблюдават журналите в реално време (live logs) и натоварването на EFW.



Фиг. 8.9 Информация в реално време за работата на IPS при EFW CE

Препоръки

Някои от по-важните препоръки, свързани с IPS при EFW CE са:

- IPS системата по подразбиране не е активна и е препоръчително тя да се включи с цел повишаване на сигурността;
- IPS повишава използването на системните ресурси и е желателно да се провери дали използваната памет няма да достигне до горната препоръчителна граница;
- Обновяването на правилата може да бъде зададено да е автоматично, като интервала за проверка от един ден в общия случай е оптимален;
- Правилата могат да бъдат обновявани и по заявка на администратор;
- Ако се активират всички групи от правила може да се блокира легитимен трафик;
- Препоръчително е изборът на правилата да се направи внимателно след задълбочен анализ на нормалния мрежови трафик;
- След всяка промяна на конфигурацията на IPS е препоръчително да се извършва и проверка на направената конфигурация и дали легитимния трафик не е блокиран.

Заключение

IPS системите предоставят възможност за предпазване на мрежовите сегменти от сложни атаки, които не могат да бъдат блокирани от технологията на защитната стена.

EFW използва Snort системата, която е тясно интегрирана с iptables и чрез графичния интерфейс могат лесно и бързо да бъдат конфигурирани необходимите правила и действия.

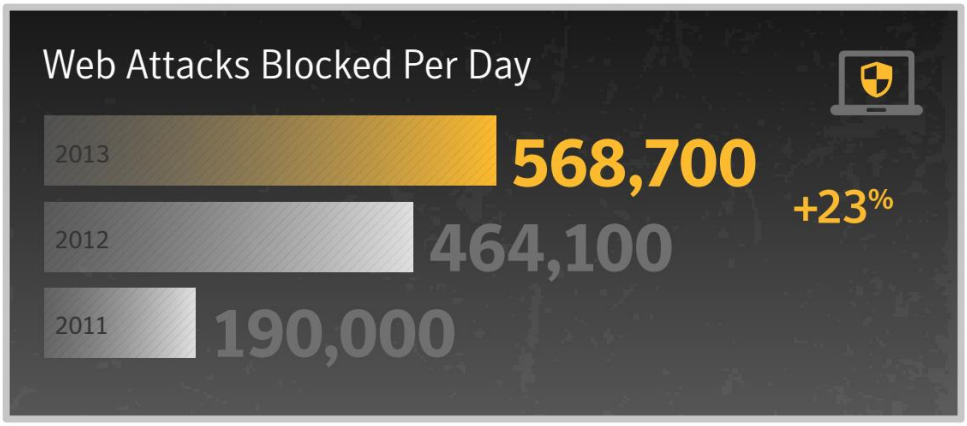
Препоръчително е при активиране на IPS правилата де се подберат само необходимите, тъй като грешна конфигурация може да блокира легитимен трафик и да доведе до използване на повече системни ресурси.

Източници

1. http://www.sans.org/reading-room/whitepapers/detection/history-evolution-intrusion-detection-344
2. http://www.infoworld.com/article/2631146/open-source-software/the-greatest-open-source-software-of-all-time.html

Глава 9. Защита на WEB трафик - http proxy, антивирусно сканиране, филтриране на съдържанието и управление на потребители

В докладът на Symantec, обобщаващ заплахите, към информационните и комуникационни технологии през 2013 година (докладът е публикуван през 2014 година) са обобщени редица параметри, като ясно се вижда, увеличения с 23% брой на блокираните ежедневно WEB атаки. Общият брой от над 568000 ежедневни атаки е силно притеснителен и поставя WEB трафика и приложенията сред най-популярните цели. Изключително важно е да се вземат необходимите мерки за подсигуряване на HTTP трафика, електронната търговия, достъпът до Web сайтове, както и всички останали услуги, свързани с Интернет.



Фиг. 9.1 Обобщена статистика за Web атаките от доклада на Symantec, обобщаващ заплахите за комуникационните технологии за 20134 година

WEB атаки

Компанията Imperva[72] предлага специализирани решения за защита на Web приложения, системи за управление на бази данни, файлови сървъри, както и за облачни приложения и услуги. В свой доклад, публикуван на техния сайт са описани и обобщени някои от най-честите Web заплахи за 2014 година:

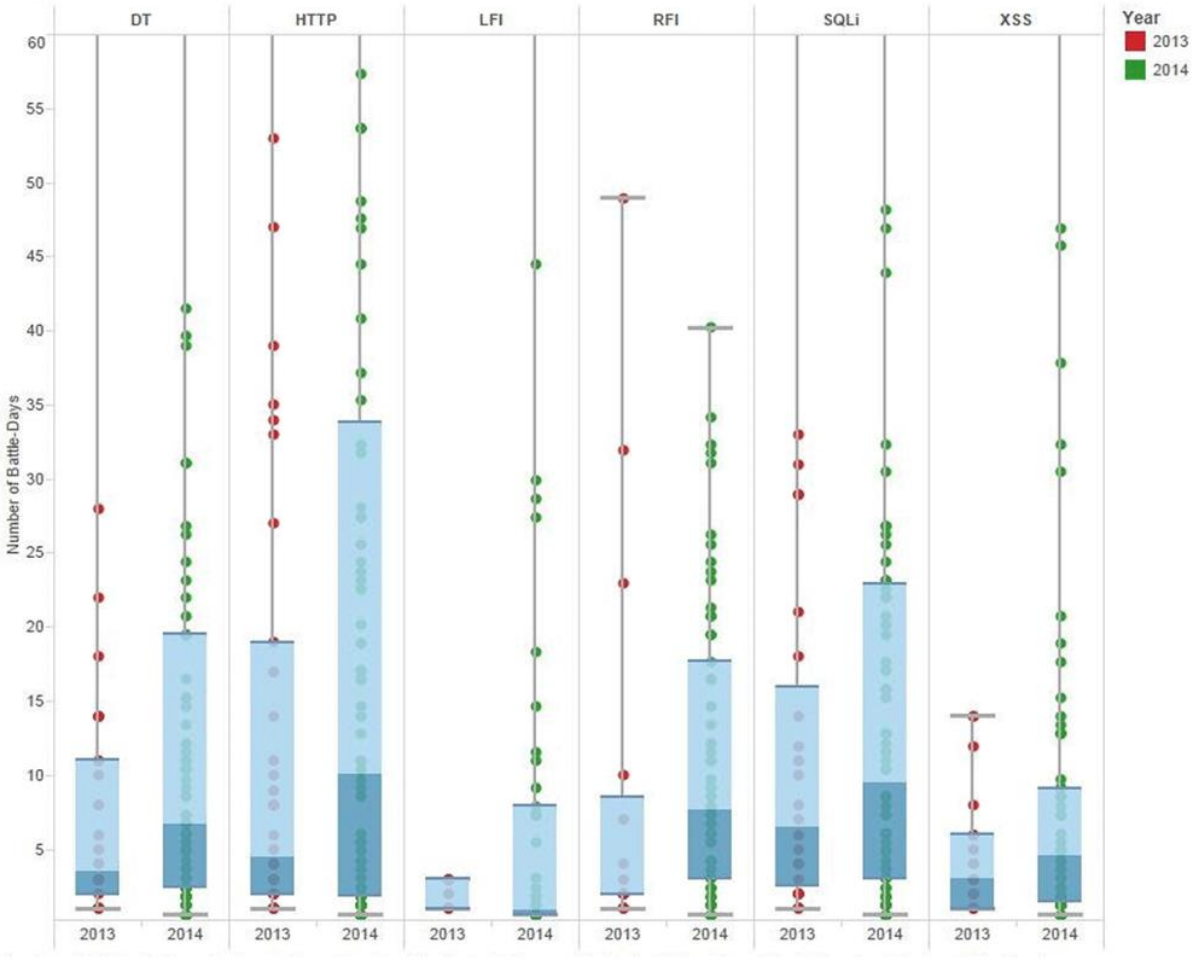
- **Directory Traversal (DT)** – специална атака, която цели дадено приложение да достъпва файлове, намиращи се в директория, до която то няма права за достъп. Този тип атаки най-често използват грешно конфигурирани права на достъп или открити потребителски имена и пароли;
- **Local File Inclusion (LFI)** – при тази атака към Web сървъра се изпращат файлове, които се добавят към директории на определени приложения. Най-често тяхното съдържание е със зловреден код. Типичен пример за такъв подход е добавянето на ASCII символа NULL (DEC код 0) към пътя и заобикалянето на механизмите за защита, ако не са конфигурирани някой от най-основните мерки;
- **Remote File Inclusion (RFI)** – добавяне на нови файлове към Web сървър чрез скриптове, които могат да модифицират съдържанието или да добавят зловреден код;
- **SQL Injection (SQLi)** – този тип атака се базира на проблеми в сигурността или грешна конфигурация на системите за управление на бази данни, използващи SQL. Чрез

[72] www.imperva.com

специално създадени SQL заявки атакуващите могат да модифицират съдържанието или структурата на бази данни, намиращи се на отдалечени сървъри;

- **Cross-Site Scripting (XSS)** – атаката позволява да се стартират скриптове на браузъра на потребителите на Web сайт или приложение, като най-честата цел е да се пренасочи сесията или да се получи достъп до данни за потребители и пароли.

За повечето атаки има разработени редица специализирани инструменти и скриптове, които са налични в Интернет и които могат да бъдат използвани както с цел анализ на сигурността, така и злонамерено.



Фиг. 9.2 Обобщена статистика от доклада на Imerva за Web атаките, през 2014 година

Необходимост от защита и филтриране на HTTP

Освен атаките към Web сървърите и приложенията, които се извършват от злонамерени лица и започващи от отдалечени системи за HTTP трафика е изключително важно да се подсигури и HTTP трафика, който се инициира от вътрешните устройства или който е отговор на външна заявка. По този начин могат да се избегнат редица опасни атаки като:

- Пренасочване на сесии;
- Фишинг;
- XSS;
- Кражба на конфиденциални данни;
- Инсталиране на зловреден код;

- SQLi;
- Сканиране на системи и много други.

Към момента се предлагат разнородни системи за подсигуряване на Web трафика, които най-често са:

- Специализирани хардуерни устройства;
- Специализиран софтуер;
- Виртуални системи;
- Облачно-базирани системи.

Аналогично на мрежовите защитни стени всяка технология има своите предимства и недостатъци, а изборът е от съществено значение за постигане на максимална степен на сигурност.

Прокси (proxy)

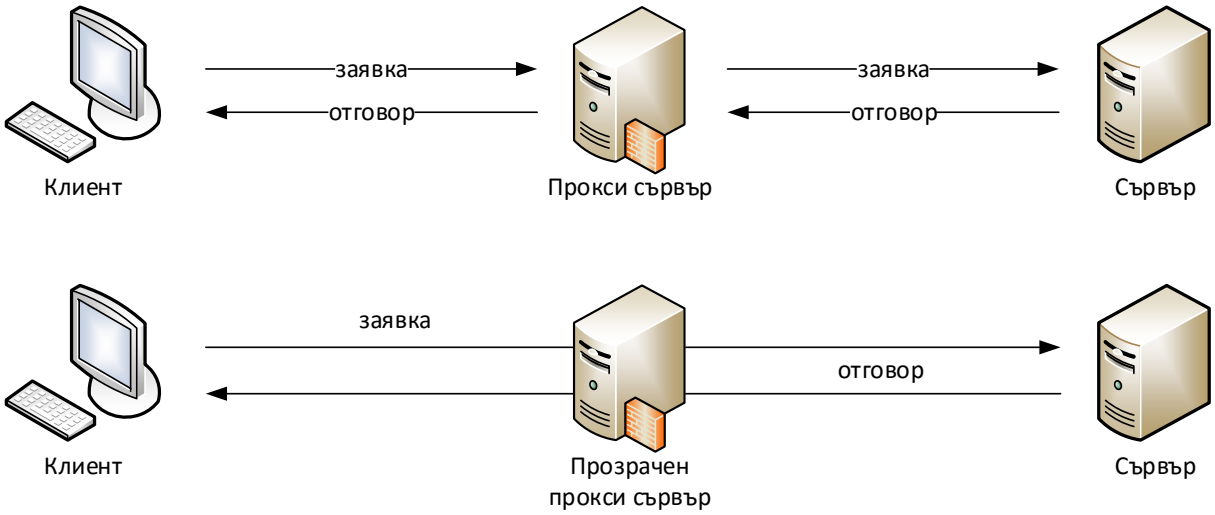
Прокси е софтуерна технология, която позволява даден сървър (хардуерно устройство или софтуер) да се явява междинно звено при комуникация между клиент и отдалечен (в определени случаи и локален за рамките на мрежовия сегмент) сървър. Клиентът се свързва с прокси сървъра, като изпраща необходимата заявка. В последствие прокси устройството извършва нова заявка към търсения от клиента сървър и препраща данните към него. Аналогично отговора на заявката преминава първо през прокси системата, която препраща данните към клиента.

Този тип комуникация има предимства от гледна точка на възможността за анализиране на трафика и сканиране за зловреден код, както и повишаването на сигурността на комуникацията.

Някои от недостатъците са, че има забавяне в скоростта на обмяна на данните, както и, че отпадането на прокси устройството може да доведе до срив в комуникацията.

Технологично спрямо потребителите прокси сървъра може да бъде:

1. **Видим** – необходимо е софтуера, който използва прокси сървъра да има конфигуриран неговия адрес и друга необходима информация (порт, потребител, парола и др.);
2. **Прозрачен (transparent)** – прокси сървъра функционира и анализира трафика, но не е необходимо потребителите да знаят за него, както и да се извършва допълнителна конфигурация на техните системи този подход се използва най-често при филтриране на Web трафик или при кеширане на HTTP данни с цел повишаване на производителността.



Фиг. 9.3 Видим и прозрачен прокси сървър

Защита на HTTP при EFW CE

За да се повиши сигурността, която EFW предлага в системата са интегрирани редица услуги, които могат да се използват чрез вградения прокси сървър:

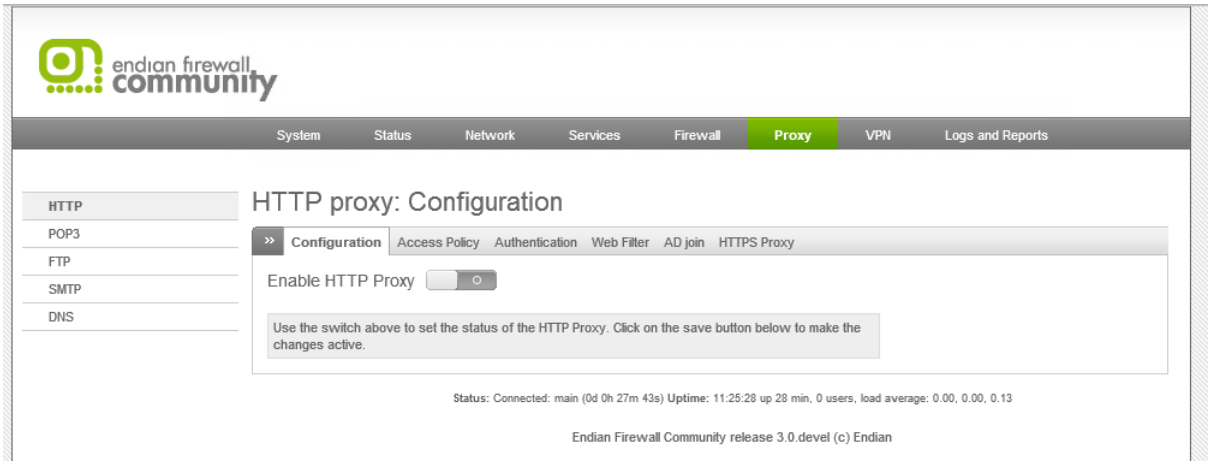
- **HTTP/HTTPS прокси** – предоставя възможност за кеширане, автентификация, URL филтриране, филтриране на съдържанието, антивирусно сканиране и др.;
- **FTP прокси** – кеширане и антивирусно сканиране на FTP трафик;
- **POP3 прокси** – антивирусно сканиране на електронната поща и филтриране на спам;
- **SMTP прокси** – антивирусно сканиране на електронната поща и филтриране на спам, маршрутизиране на домейни, DSN базиран на зони;
- **DNS прокси** – кеширане и anti-spyware проверка.

Всяка прокси услуга може да се конфигурира и активира (деактивира) независимо от останалите, например ако се стартира POP3 прокси при необходимост се инициализира и функционалността на POP3 протокола.

В сравнение с по-старата версия 2.x на EFW, актуалната 3.x има изцяло преработена прокси система, която значително подобрява производителността. В предходната версия се използва метода за кеширане на данните, базиран на прокси сървъра squid[73], което при определени ситуации и па-сериозно натоварване изисква значителни системни ресурси. За да се коригира този недостатък squid е заменен от ICAP (Internet Content Adaptation Protocol), който въпреки по-сложната системна архитектура има редица предимства. За ICAP може да се обобщи, че това е протокол, дефиниран в RFC 3507, който позволява съдържанието на Web страница да се анализира и при необходимост модифицира, което позволява да се извърши задълбочено филтриране и антивирусно сканиране.

Прокси сървърите при EFW се конфигурират от основното меню “Proxy” и съответните подменюта.

[73] www.squid-cache.org



Фиг. 9.4 Меню “Proxy” при EFW

HTTP прокси

Основната цел на интегрирането на HTTP прокси е да се увеличи бързодействието при зареждане на Web страници и обекти чрез кеширане. Тази технология позволява да се получи и разширена функционалност, обхващаща филтриране по URL или по съдържание, антивирусно сканиране и др. При конфигурирането на HTTP прокси е необходимо да се зададе дали то да бъде прозрачно или не, от гледна точка на мрежовите потребители.

Активирането или деактивирането на HTTP прокси услугите се извършва от менюто “Proxy”, подменюто “HTTP”, чрез преместване на бутона “Enable HTTP Proxy” в активно или изключено състояние.

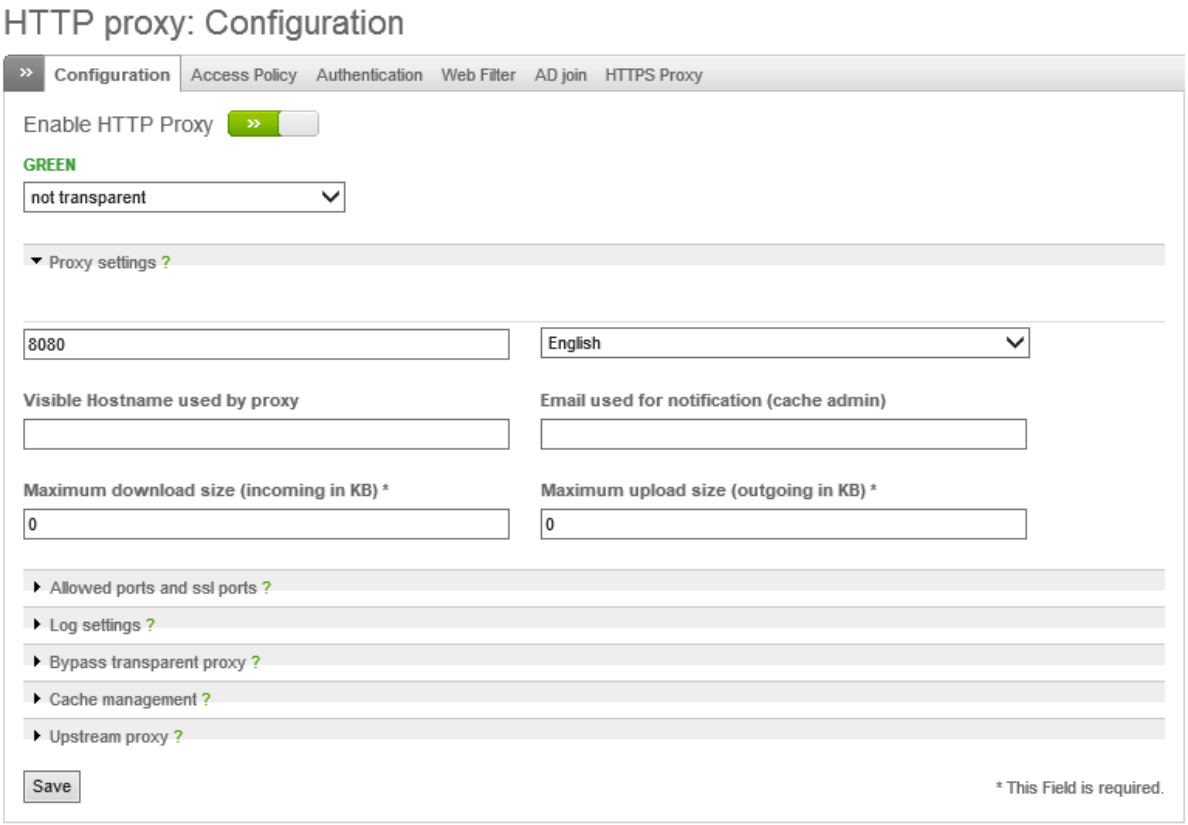
След активирането е необходимо да бъдат конфигурирани няколко секции с параметри:

1. Proxy settings;
2. Allowed ports and SSL ports;
3. Log settings;
4. Bypass transparent proxy;
5. Cache management;
6. Upstream proxy.

Фиг. 9.5 Групи с конфигурационни параметри при HTTP прокси

Основните настройки на HTTP прокси услугите включват:

- Интерфейс и тип на проксито – прозрачно (transparent) или непрозрачно (not transparent);
- Порт – TCP порта, на който е активна прокси услугата, като по подразбиране се използва порт 8080;
- Език – избор на език за визуализиране на съобщенията за грешки;
- Visible Hostname used by proxy – името на устройството, което се извежда и в съобщенията за грешки;
- Email used for notification (cache admin) – адрес за електронна поща, който се включва в съобщенията за грешки;
- Maximum download size (incoming in KB) – максимален размер на обектите в KB, които могат да бъдат изтеглени през HTTP - стойност 0 дефинира неограничен размер;
- Maximum upload size (outgoing in KB) - максимален размер на обектите в KB, които могат да бъдат изпратени към HTTP сървър - стойност 0 дефинира неограничен размер.

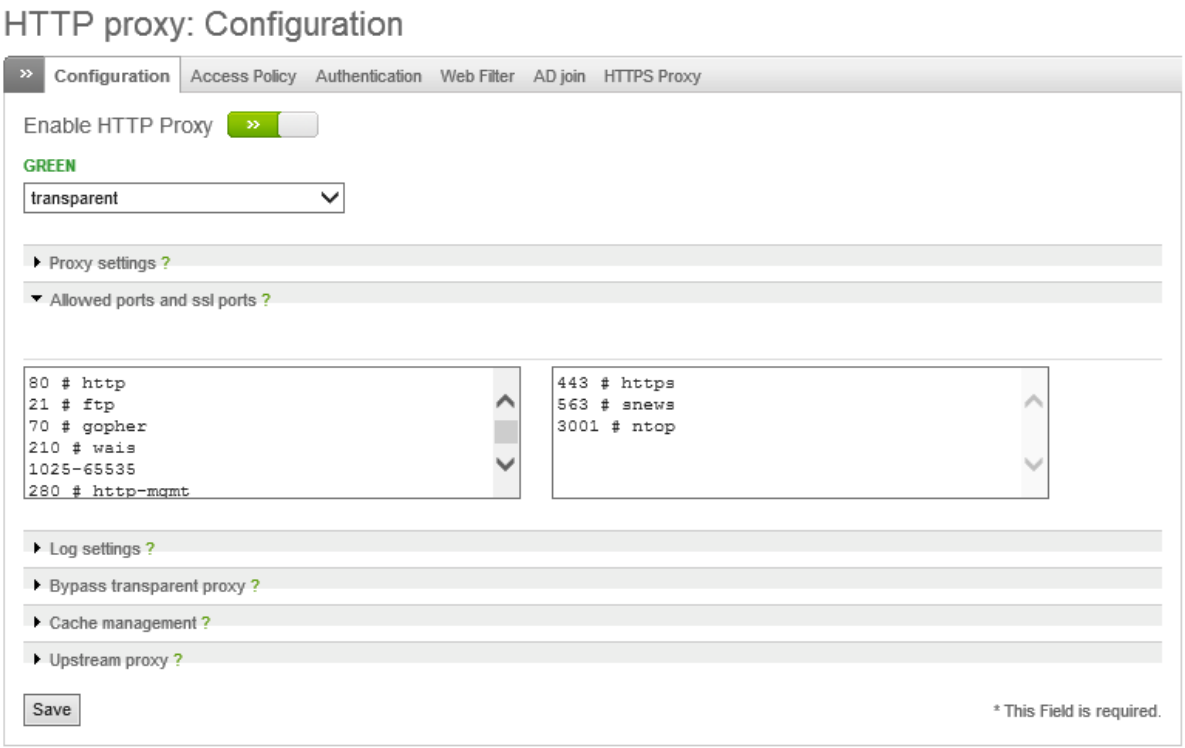


Фиг. 9.6 Основни настройки на HTTP прокси при EFW CE

След конфигурирането на основните параметри на HTTP прокси сървъра е необходимо да бъдат зададени портовете, които клиентите могат да използват с HTTP и HTTPS протоколите.

Списъка е разделен на две части, като в лявата са HTTP портовете, а в дясно – HTTPS.

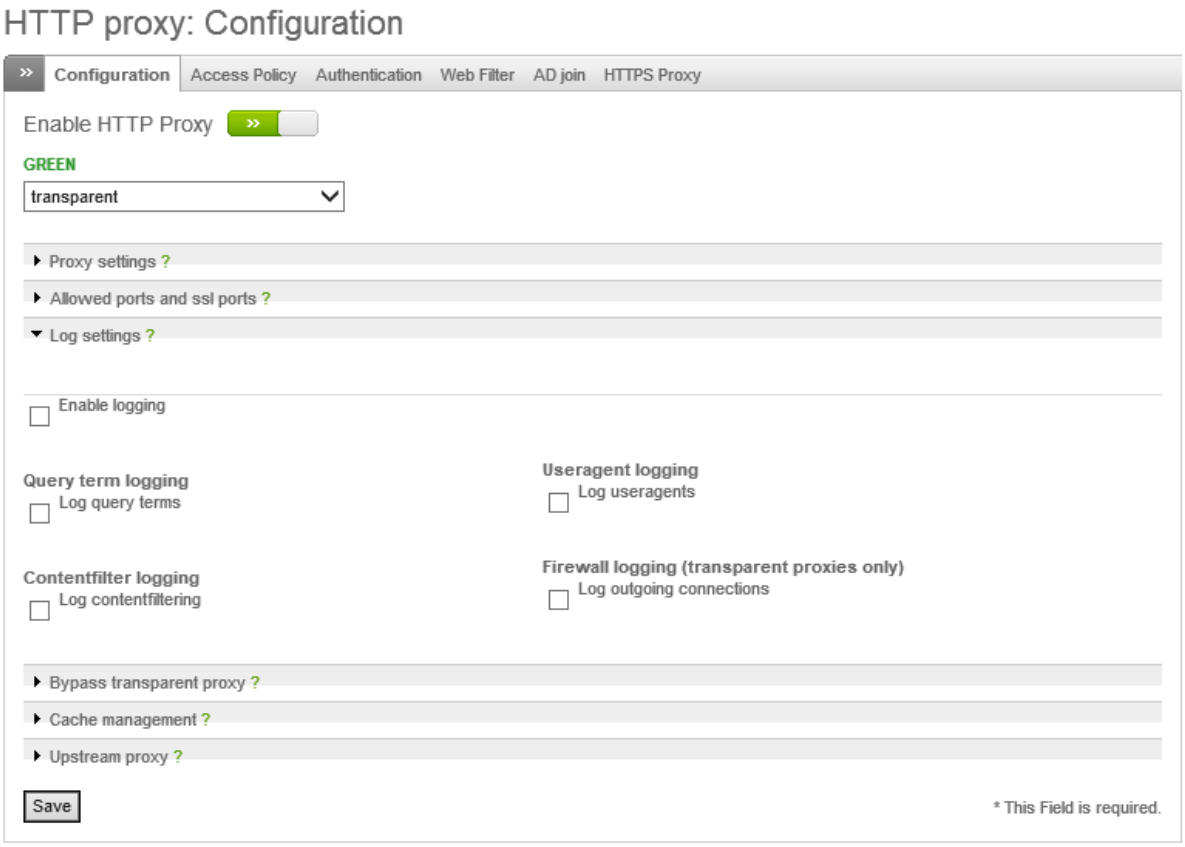
След номера на всеки порт следва символа #, който определя че последващият текст е коментар.



Фиг. 9.7 Настройки на разрешените портове при HTTP прокси сървъра на EFW CE

Прокси услугата, свързана с HTTP и HTTPS може да генерира записи в журналите на EFW CE, като за целта е необходимо да бъдат зададени параметрите от групата “Log settings”:

- Enable logging – активира или деактивира генерирането на журналните записи;
- Query term logging – записване и на параметрите от заявките, например (?id=1024);
- Useragent logging – добавяне на информация за “User agent” на клиента;
- Contentfilter logging – включване в журнала на информация за филтрирането на съдържанието;
- Firewall logging (transparent proxies only) – информация за филтрирането на изходящия трафик от защитната стена - тази функция може да се използва единствено с прозрачно прокси.

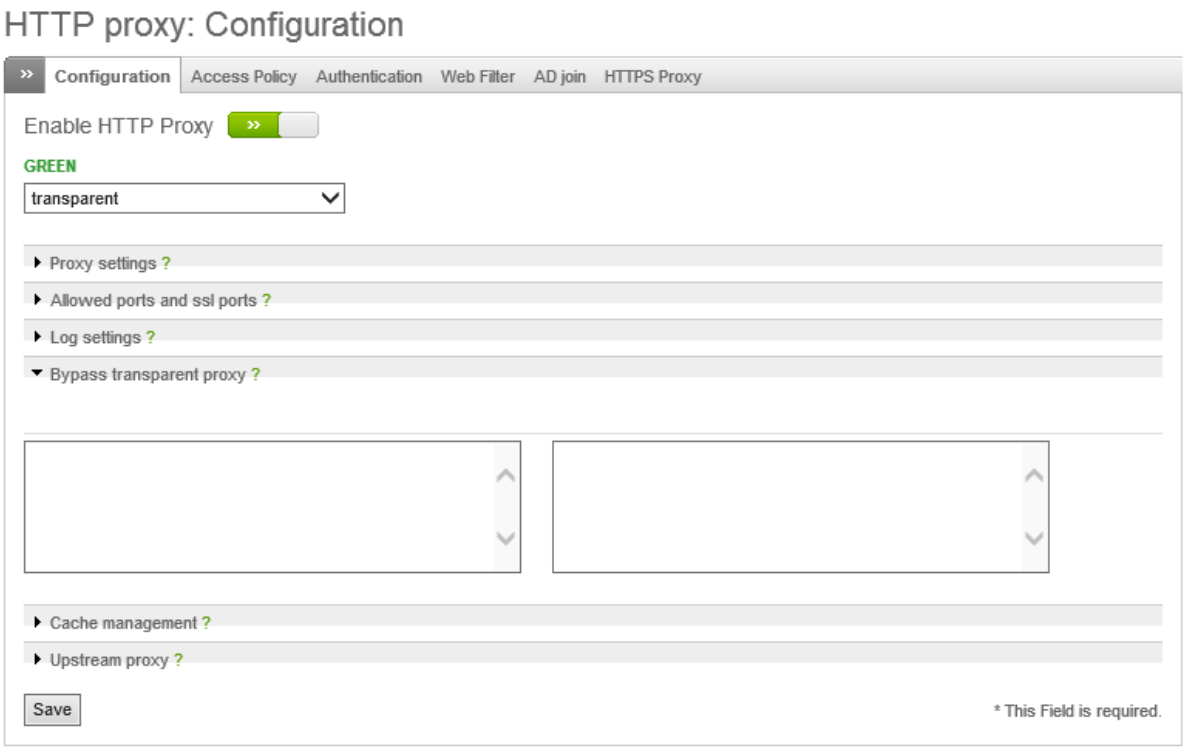


Фиг. 9.8 Конфигуриране на параметрите за запис на данни в журналите за HTTP прокси

Ако HTTP прокси сървъра работи в прозрачен режим могат да бъдат конфигурирани изключения, които да обхващат:

- Bypass transparent proxy from SUBNET/IP/MAC – прокси функцията не се използва, ако изпращача е в посочената мрежа, или използва зададения IP/MAC адрес;
- Bypass transparent proxy to SUBNET/IP – прокси функцията не се използва, ако получателя е в посочената мрежа, или използва зададения IP/MAC адрес.

По подразбиране тези две групи с параметри са празни и всичкия трафик се сканира от прозрачното прокси.

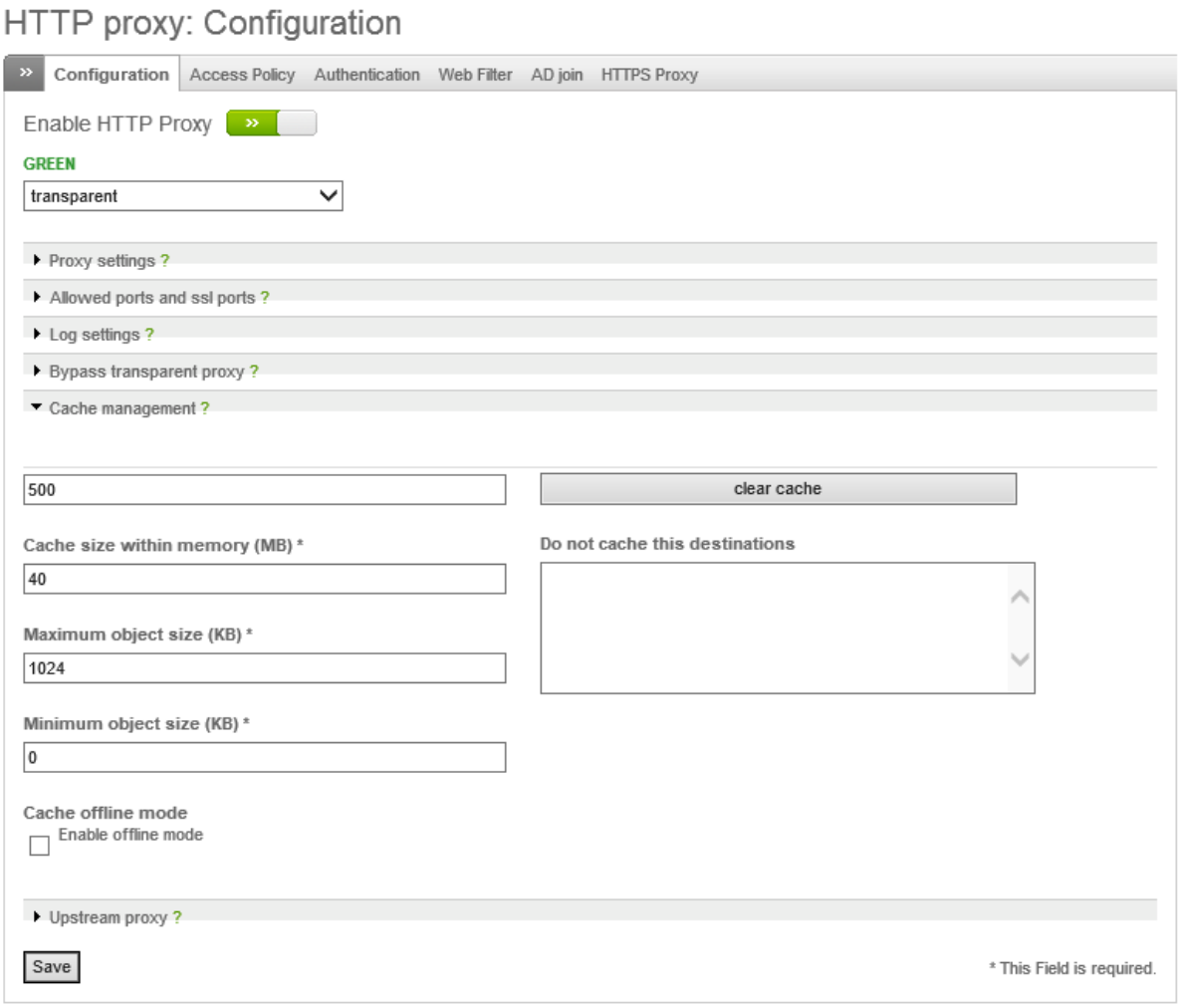


Фиг. 9.9 Изключения за HTTP прокси функциите при EW CE

Една от по-важните настройки на HTTP прокси е свързана с управлението на кеша. Тези параметри могат да бъдат зададени от частта “Cache management” на страницата “Configuration”.

Възможно е да бъдат конфигурирани:

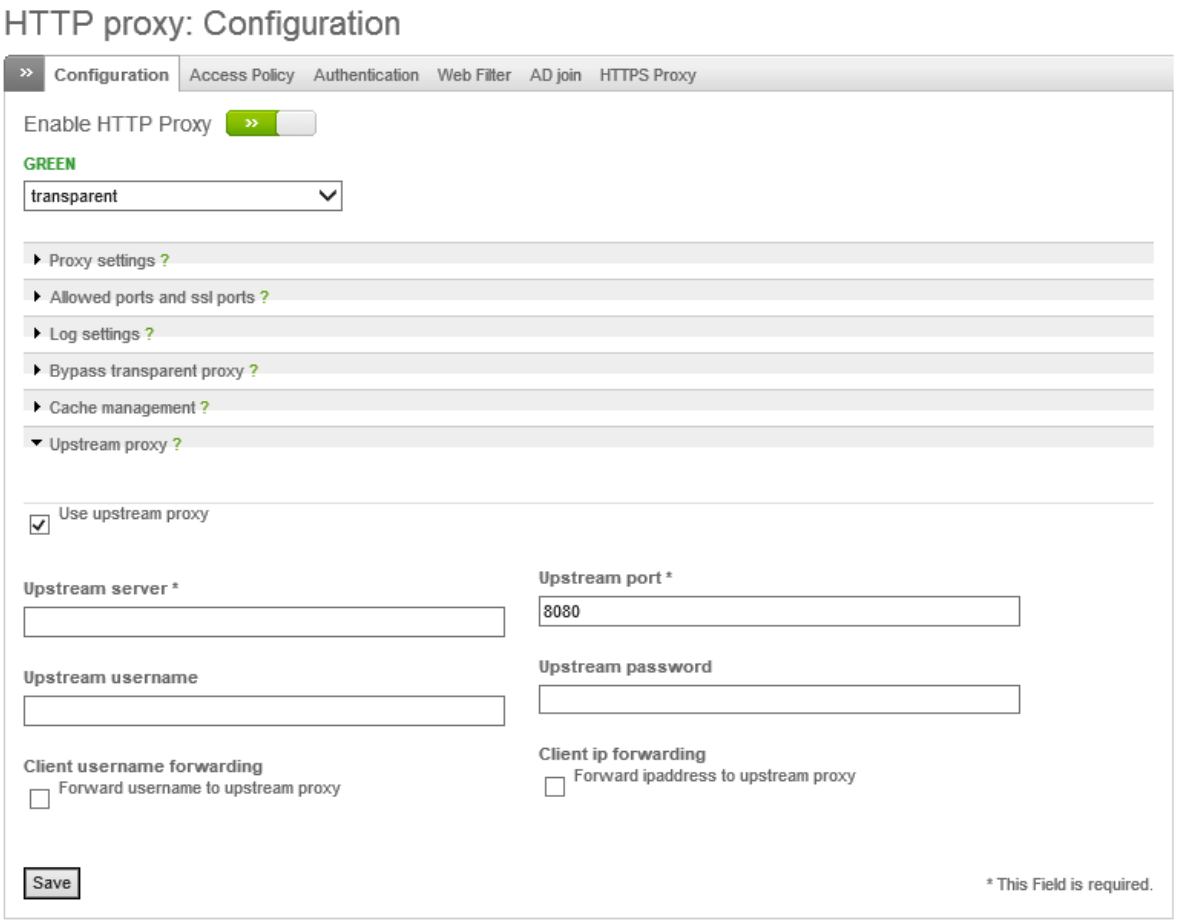
- Размер на кеша – стойност в MB, която дефинира максималния размер, който да се съхранява на файловата система на EFW CE. По подразбиране големината е 500 MB;
- Cache size within memory (MB) – максимален размер на кеша в MB, който се съхранява в RAM на EFW системата. Стойността по подразбиране е 40 MB;
- Maximum object size (KB) – максимален размер на кешираните обекти, зададен в KB;
- Minimum object size (KB) – минимален размер на кешираните обекти, зададен в KB;
- Do not cache these destinations – списък с домейни, чиито предоставени обекти няма да се кешират;
- Enable offline mode –ако тази опция е активирана HTTP прокси сървъра на EFW CE няма да кешира обектите от предходното прокси (upstream proxy).
  Съдържанието на кеша може да се изчисти при натискане на бутона “Clear cache”.



Фиг. 9.10 Управление на кеша при HTTP прокси на EFW CE

Последната група с конфигурационни параметри е свързана с т.нар. “Upstream proxy” – наличие на друг прокси сървър в рамките на мрежата, който се явява междинно звено при заявката за отдалечен обект спрямо локалния прокси. За тази функционалност може да се конфигурира:

- Upstream proxy – активиране или деактивиране на достъпа до “upstream proxy” система;
- Upstream server – адрес или DNS име на междинния прокси сървър;
- Upstream port – порт, на който работи междинния сървър;
- Upstream username / password – потребител и парола за “upstream proxy” системата;
- Client username forwarding – пренасочване на въведените от клиента потребител и парола към “upstream proxy” система;
- Client IP forwarding – пренасочване на IP адреса на клиента към “upstream proxy” устройство.



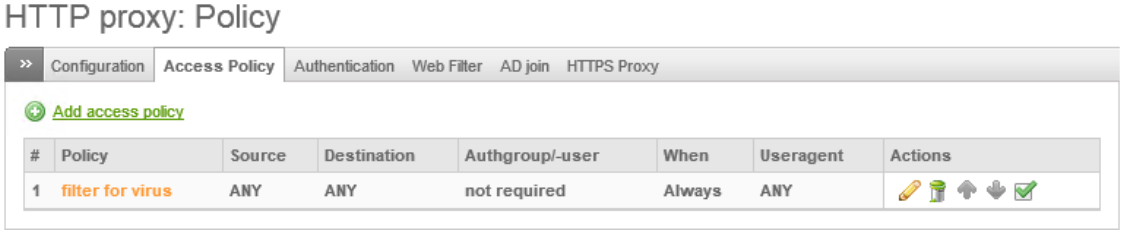
Фиг. 9.11 Конфигуриране на “Upstream proxy”

Политики за достъп (access policy)

Политиката за достъп се отнася за всеки отделен клиент, чиито трафик преминава през HTTP прокси функциите без значение, дали сървъра работи в прозрачен режим или не. Всяка политика е изградена от едно или няколко правила, които определят дали достъпа е разрешен или забранен, включително в даден интервал от време и на база на зададени критерии (потребител, “user agent” и др.).

Страницата за конфигуриране на политиките за достъп съдържа в табличен вид списък с дефинираните правила, както и стандартната за EFW колона “Actions”, в която има бутони за редактиране, изтриване, промяна на позицията на правилото и за неговото активиране или деактивиране.

Позицията на правилото е от значение и логиката е аналогична на обработката на правилата за филтриране на пакетите при защитната стена.

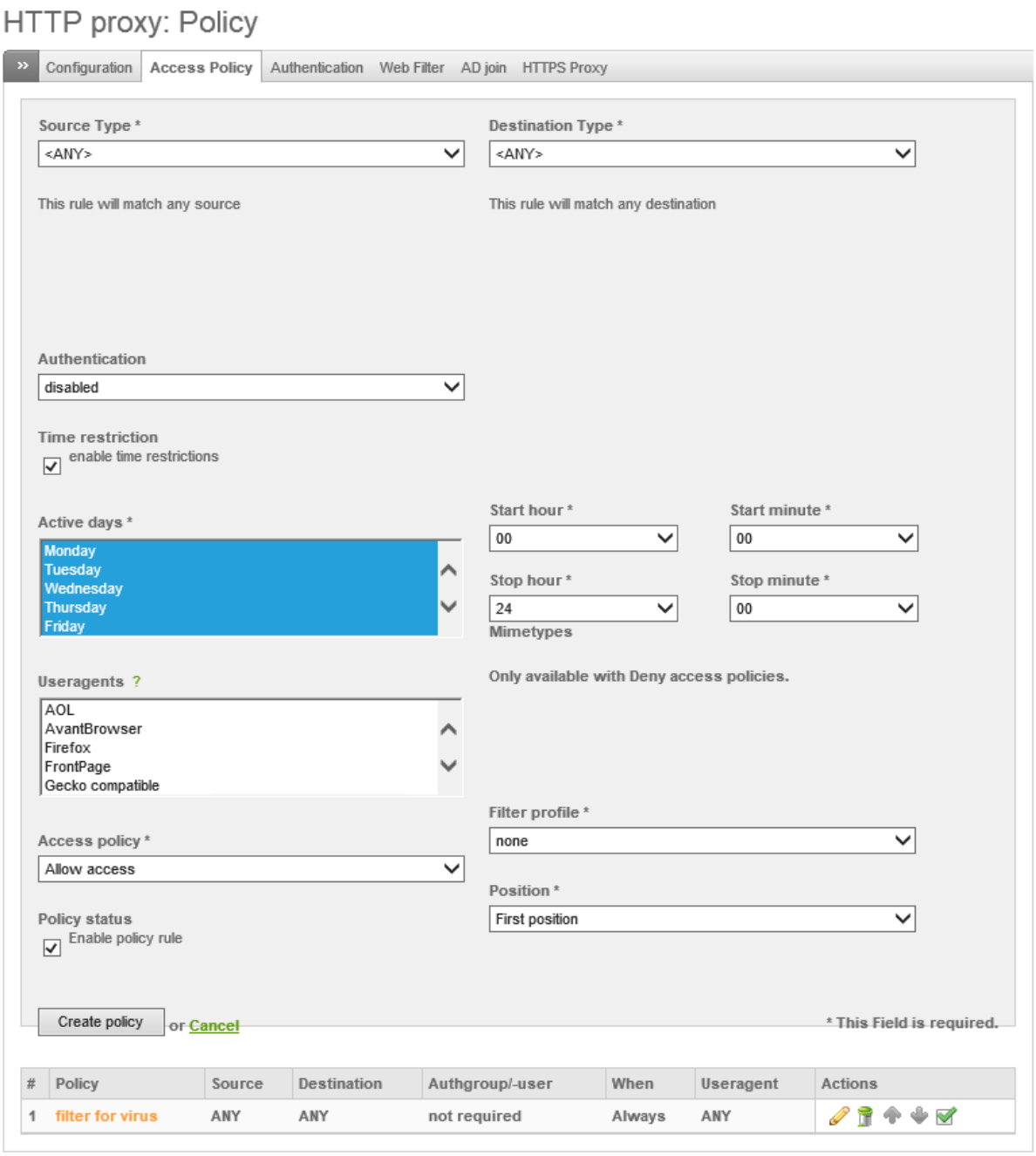


Фиг. 9.12 HTTP политики за достъп

Добавянето на ново правило за политиката за достъп при HTTP прокси става след натискане на връзката “Add access policy”, като е необходимо да се дефинират следните параметри:

- Source type – източник на трафика, с опции ANY, определена зона, интерфейс или IP/MAC адрес;
- Destination type – получател на трафика, с опции ANY, определена зона, интерфейс или IP адрес;
- Authentication – дефинира метода за автентификация, която се изисква от клиентите. Възможните варианти са “disabled” – без автентификация, “user based” – изисква се въвеждане на потребител и парола, или “group based”, при който се изисква потребителя да е член на отделната група, която е дефинирана в конфигурацията на EFW. Добавянето на групи и потребители става от страницата “Authentication” при HTTP прокси;
- Enable time restrictions – активира правилото в зададения период от време. Необходимо е след стартиране на тази функция да се посочат дните от седмицата, както и началния и крайния част на активно състояние;
- Useragents – типа на клиента, който се използва от потребителя;
- Mimetypes – типа на MIME (Multipurpose Internet Mail Extensions), които да се блокират, като тази опция е валидна единствено ако действието на политиката е “deny access”;
- Access policy – действие на политиката. Възможните варианти са разрешаване (allow access) или забраняване (deny access);
- Filter profile – ако действието на политиката е “allow access” от тук може да се посочи правилото за филтриране. По подразбиране, ако няма допълнително дефинирани филтри се предоставят две възможности - първата опция е “none”, при която не се извършва филтриране, а втората - “virus detection only”, при която се извършва единствено сканиране на вируси и ако е открит зловреден код обекта не се пропуска;
- “Enable policy rule” – активира или деактивира правилото от политиката;
- Position – посочва позицията на правилото

След въвеждане на параметрите за да се създаде правилото е необходимо да се натисне бутона “Create rule”. Аналогично на защитната стена след добавянето на правилото е необходимо да се натисне бутона “Apply”, намиращ се в зеленото поле в най-горната част на страницата.



Фиг. 9.13 Правила при HTTP политиките за достъп на EFW CE

Прокси функциите при EFW поддържат 4 вида автентификация:

1. Local Authentication (NCSA);
2. LDAP[74];
3. Windows Active Directory;
4. RADUIS[75].

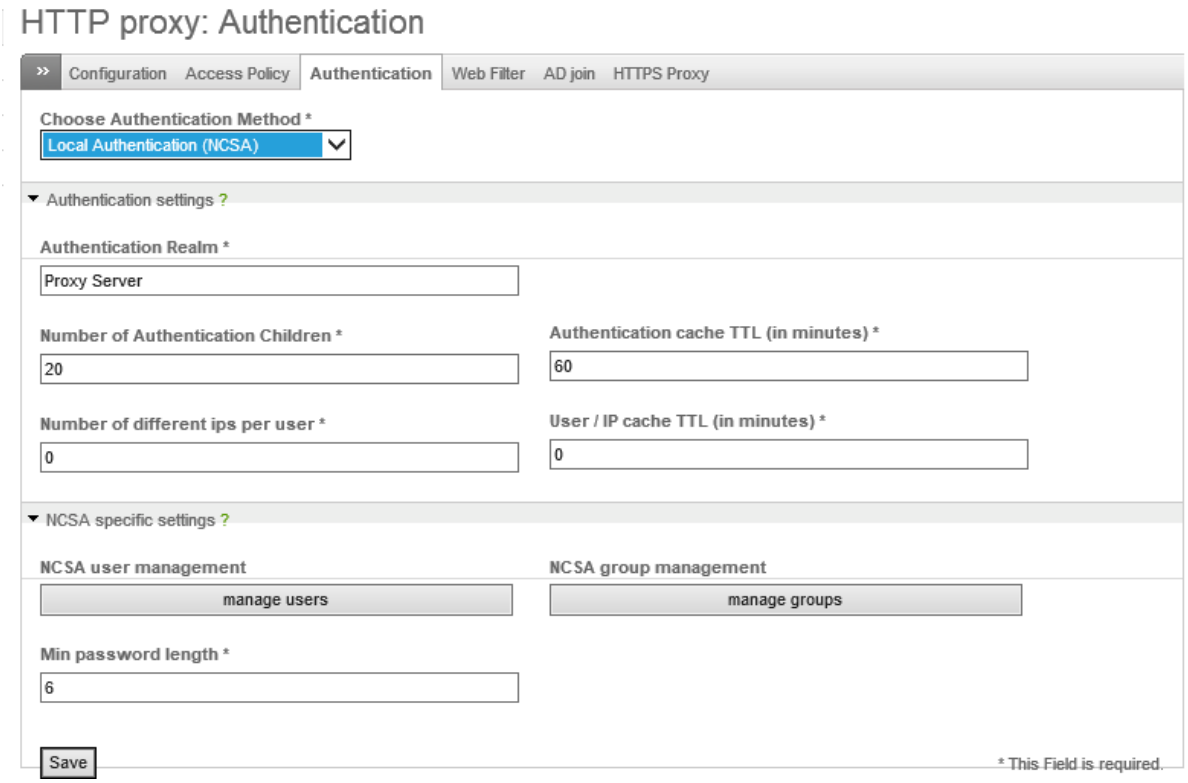
Изборът на метода се извършва от страницата “Authentication” в подменюто HTTP прокси.

[74] Lightweight Directory Access Protocol
[75] Remote Authentication Dial-In User Service

Конфигурационните параметри са разделени на две групи. Първата включва общите настройки на автентификацията:

- Authentication realm – текст, който се добавя в диалога за автентификация, или като параметър към някои от методите (например Kerberos). Ако се използва Microsoft Active Directory е необходимо в това поле да се въведе FQDN[76] на PDC[77];
- Number of Authentication Children – максималния брой на паралелните процеси за автентификация;
- Authentication cache TTL – интервала в минути, по времето на който данните за автентификацията са кеширани;
- Number of different IPs per user – максималният брой на различните IP адреси, от които едновременно може да се автентифицира един конкретен потребител;
- User / IP cache TTL – времето в минути, през което даденият потребител е асоцииран с IP адрес.

Втората част с параметри варира в зависимост от типа (протокола) на автентификацията, която HTTP проксито на EFW използва.



Фиг. 9.14 Конфигуриране на HTTP прокси автентификация

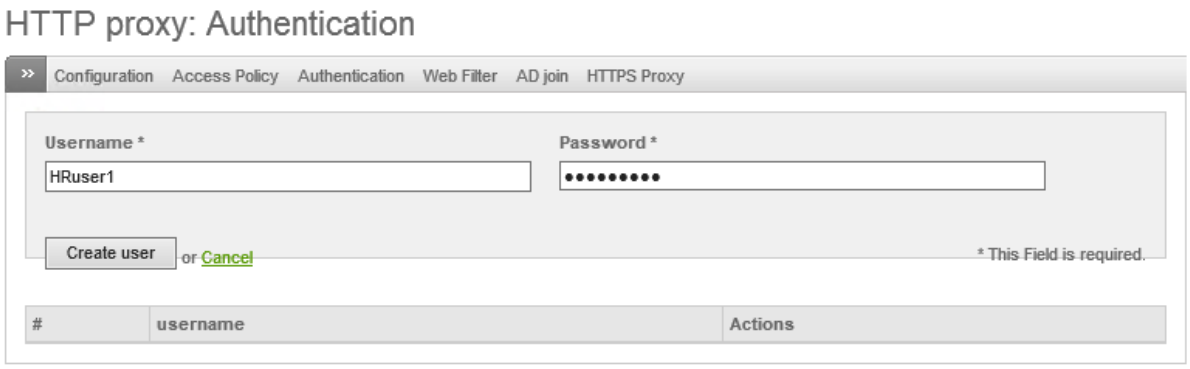
Добавянето, изтриването или редактирането на локален потребител (NCSA) се извършва след натискане на бутона “manage users”. За да се добави нов потребител се използва връзката “Add NCSA user” и се въвеждат:

- Username – потребителско име;
- Password – парола;

[76] Fully Qualified Domain Name
[77] Primary Domain Controller

Чрез бутона “Create user” се създава новия потребител. Препоръчително е винаги въведените пароли да спазват критериите за надеждност.

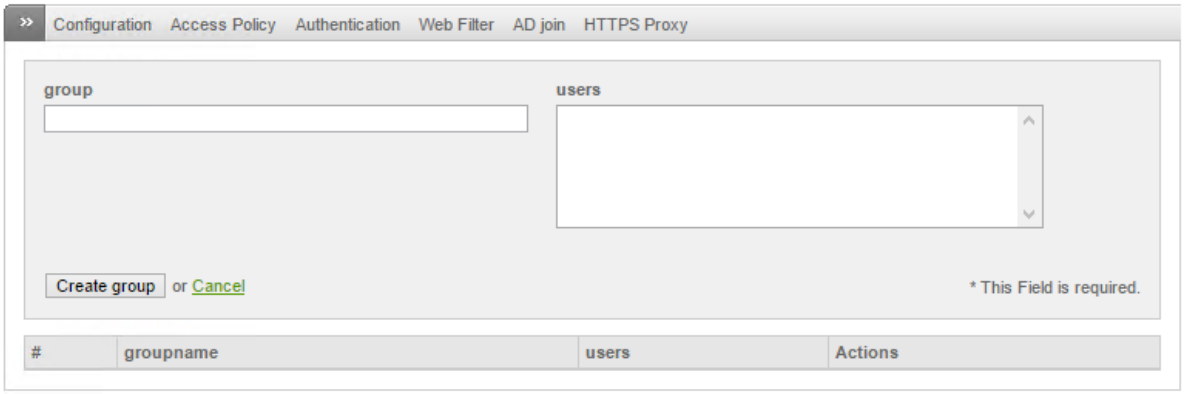
Аналогично на останалите таблици от колоната “Actions” може да се избере действието за определения ред.



Фиг. 9.15 NCSA потребители при EFW CE

Управлението на групите е аналогично на потребителите и се извършва след натискане на бутона “manage groups”. За да се добави нова група трябва да се използва връзката “Add NCSA group” и да се въведат име на групата и един или няколко потребителя.

В последствие при редактиране на групи може да се променя списъка на включените потребители.



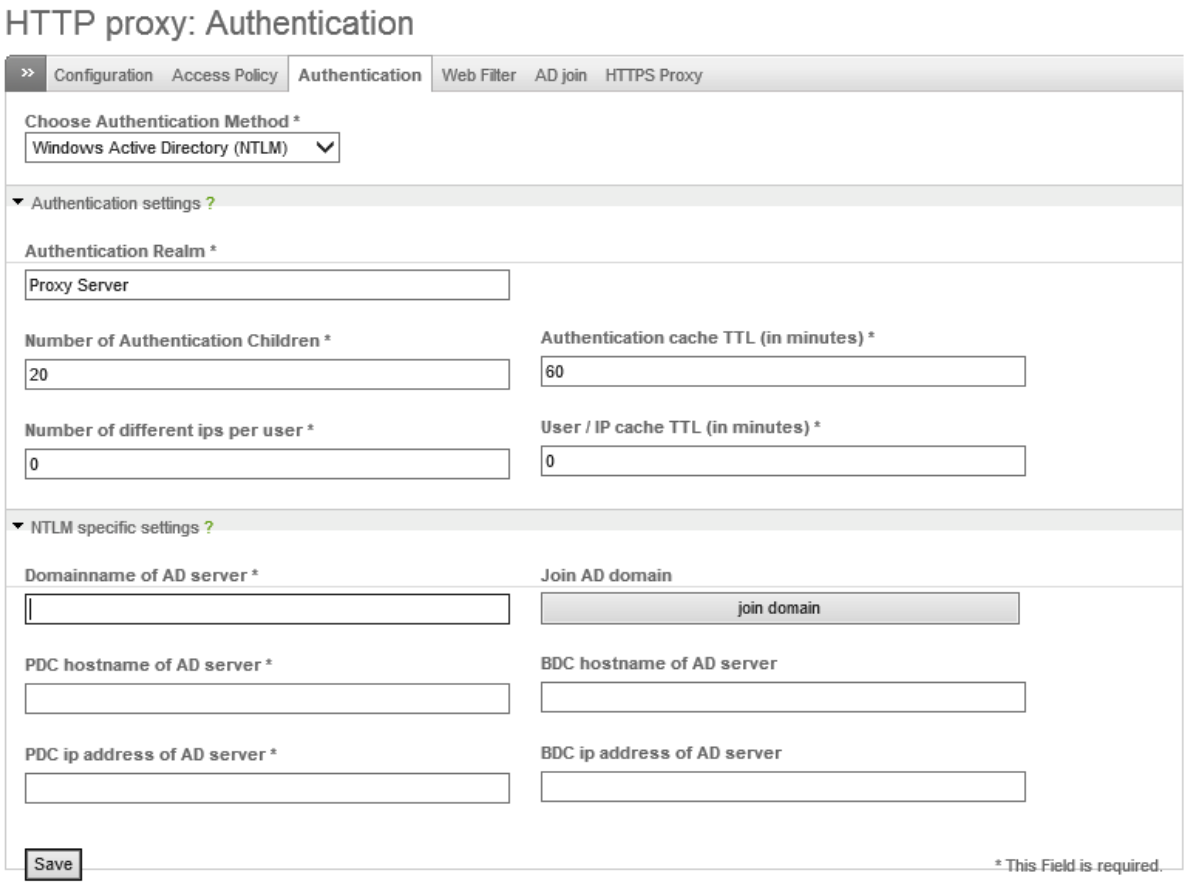
Фиг. 9.16 NCSA групи при EFW CE

Ако методът за автентификация е Microsoft Active Directory е необходимо да бъдат конфигурирани:

- Domainname of AD server – името на домейна;
- PDC hostname of AD server – името на PDC;
- PDC ip address of AD server – IP адресът на PDC;
- BDC hostname of AD server – името на BDC[78];
- BDC ip address of AD server – IP адресът на BDC.

Бутонът “Join domain” добавя EFW системата към посочения домейн.

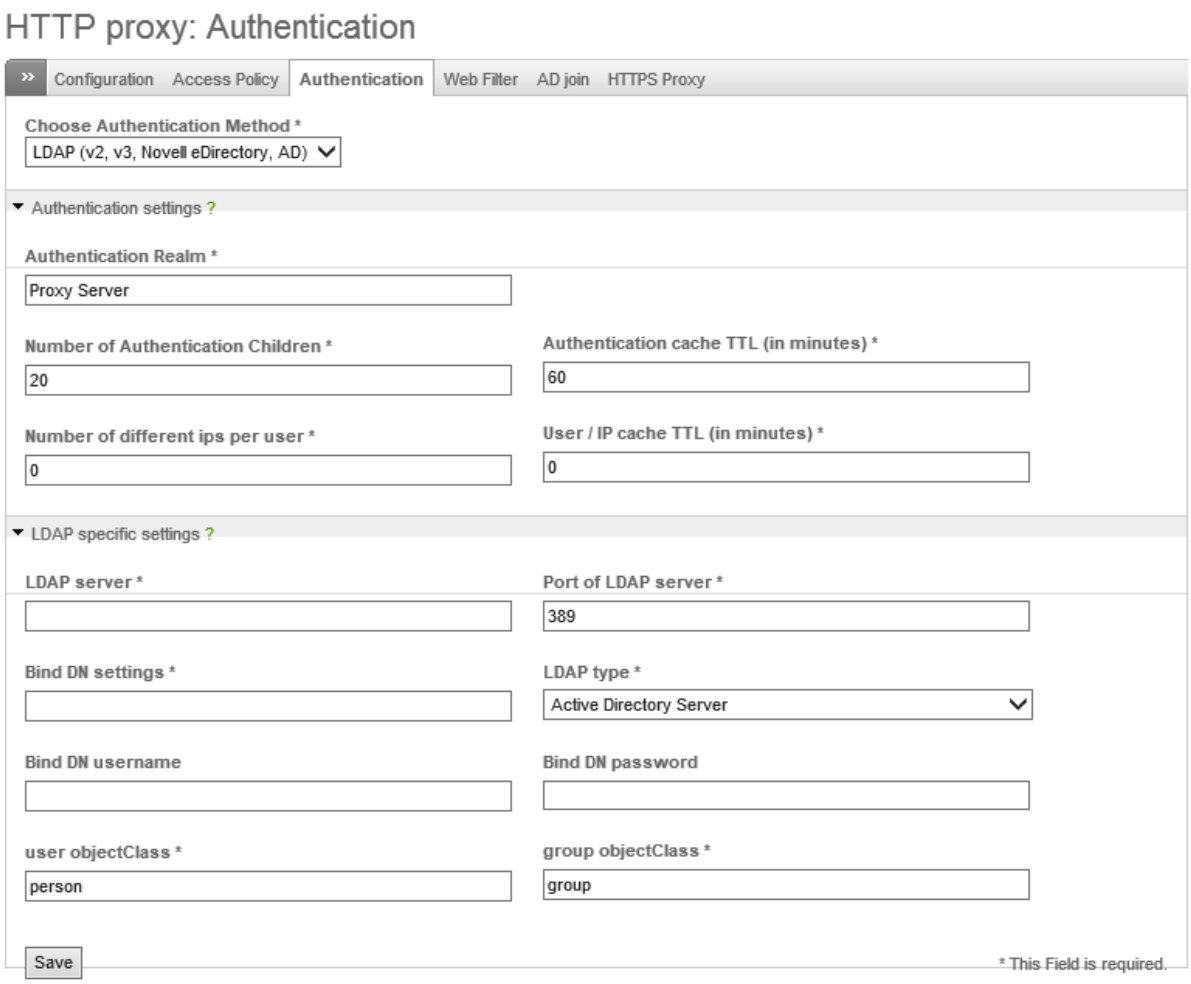
[78] Backup Domain Controller



Фиг. 9.17 Конфигуриране на AD автентификация при EFW CE

При автентификация с LDAP трябва да се конфигурират:

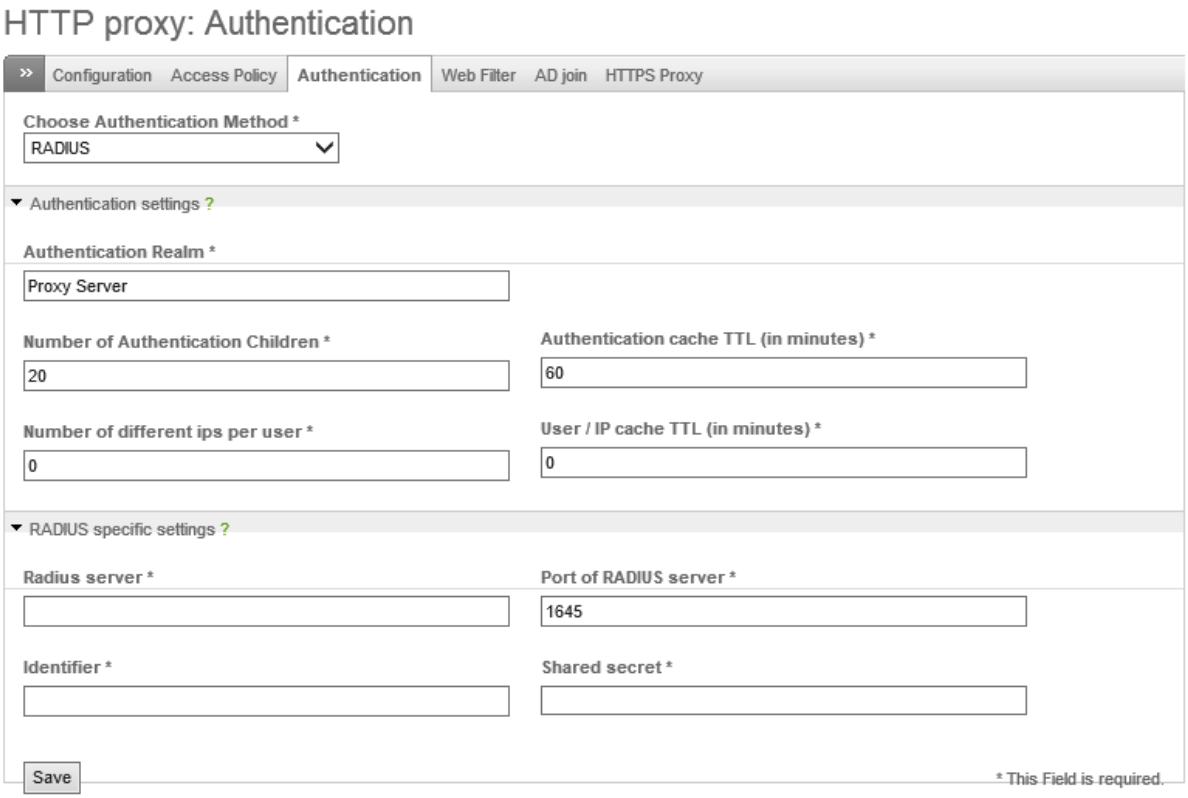
- LDAP server – IP адресът или FQDN на използвания в LDAP сървър;
- Port of LDAP server – портът, на който работи LDAP сървъра, като по подразбиране стойността е 389;
- Bind DN settings – началната точка (Base Distinguish Name);
- LDAP type – типа на сървъра;
- Bind DN username – пълното DN (Distinguish Name), което е обвързано с потребителя. (Забележка: необходимо е да има конфигуриран достъп за четене);
- Bind DN password – паролата на потребителя, чрез който EFW CE се свързва LDAP сървъра;
- user objectClass и group objectClass – обекти в LDAP.



Фиг. 9.18 Конфигуриране на LDAP автентификация при EFW CE

RADIUS автентификацията изисква въвеждане на:

- RADIUS server – адрес или FQDN на RADIUS сървъра;
- Port of RADIUS server - портът, на който работи RADIUS сървъра. По подразбиране стойността е 1645;
- Identifier – допълнителен идентификатор;
- Shared secret – споделената парола за достъп до RADIUS сървъра.



Фиг. 9.19 Конфигуриране на RADIUS автентификация при EFW CE

В EFW е използвана технологията на Cyren[79], която предоставя мощни функции за URL филтриране, базирани на два подхода:

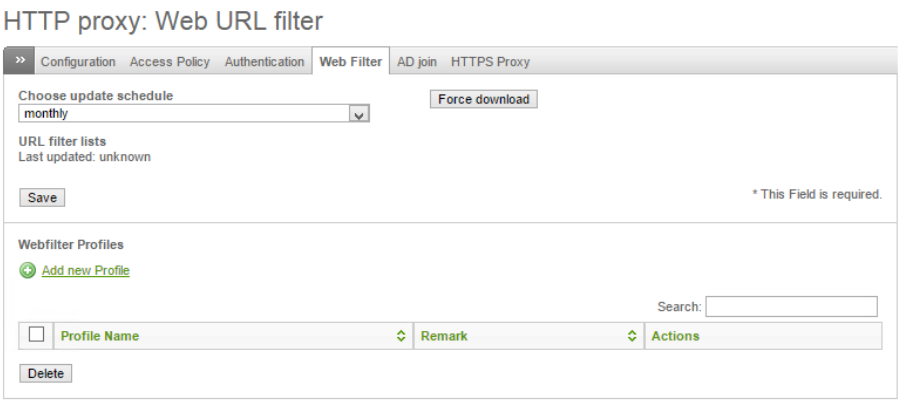
1. Гъвкав метод за категоризиране на съдържанието на Web страници;
2. Включване на URL в т.нар. “whitelist” и “blacklist” списъци. Всяка заявка от страна на потребителите се проверява в тези два списъка и ако URL е открит в “whitelist” може да се изпълни.

Ако EFW системата не е регистрирана на сайта на Endian не е възможно да бъдат изтеглени списъците, необходими на URL филтъра и се извежда предупредително съобщение. Необходимо е да се конфигурира интервал за обновяване на тези списъци, като възможните настройки са на всеки час, всеки ден, всяка седмица или всеки месец. При натискане на бутонът “Force download” данните се изтеглят от Интернет и при необходимост списъците се обновяват.

За да се използва URL филтрирането от HTTP прокси услугата при EFW CE е необходимо да се създаде нов профил, свързан с URL. Това се извършва от връзката “Add new Profile”. Първо се въвежда името на профила в полето “Profile Name”. Ако е необходимо да се активира и антивирусно сканиране се посочва опцията “Activate antivirus scan”, която по подразбиране е активна.

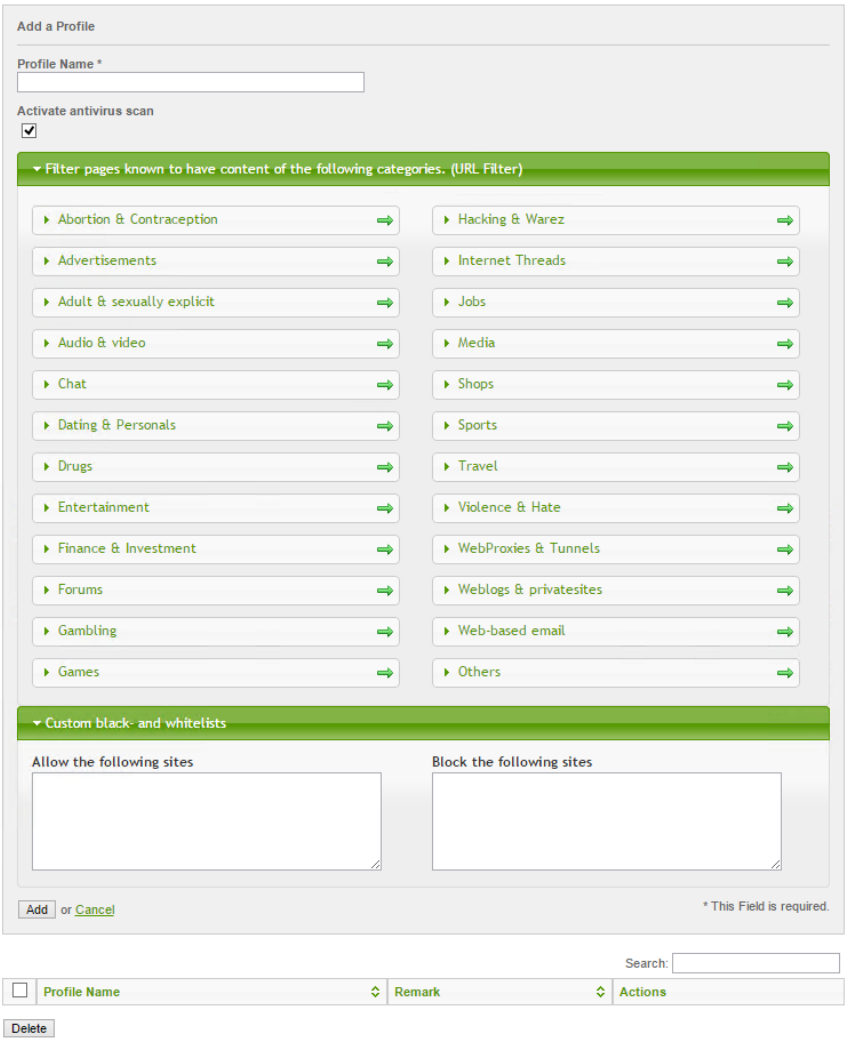
В секцията “Filter pages known to have content of the following categories.” може да се посочи при коя категория съдържание страницата да се блокира. За разлика от функциите на защитната стена, при която по подразбиране трафика се забранява и се разрешава на базата на политиката за сигурност, при URL филтрите всички категории са разрешени и трябва да бъдат блокирани необходимите.

[79] www.cyren.com



Фиг. 9.20 Конфигуриране на интервал за обновяване на списъците на Cyren

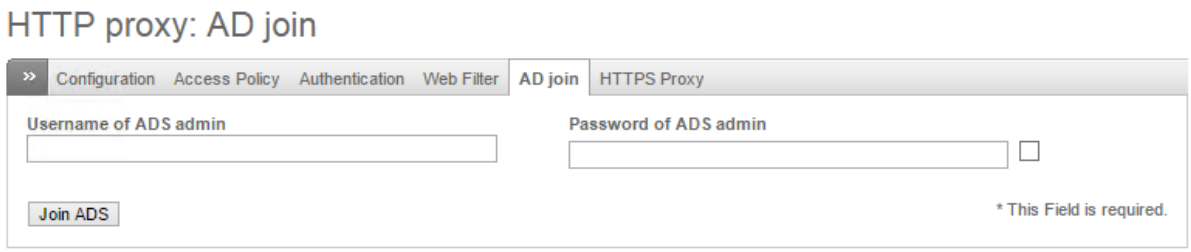
Блокирането или разрешаването на дадена категория (подкатегория) се извършва от иконата със стрелка в дясно от категорията. Добавянето на URL в “blacklist” или “whitelist” може да се направи в двете текстови полета под частта с категориите за филтриране. Когато политиката е готова е необходимо да се добави от бутона “Add” и след това да се активира в “Access policy”.



Фиг. 9.21 Конфигуриране на URL филтър с Cyren при EFW CE

В страницата “AD join” е възможно да се въведат необходимите параметри за включване на EFW към Microsoft Active Directory:

- Username of ADS admin – потребител с необходимите права за добавяне на устройства към AD;
- Password of ADS admin – парола за потребителя, която по подразбиране е скрита, но може да бъде визуализиране при активиране на “checkbox” полето до текстовото.

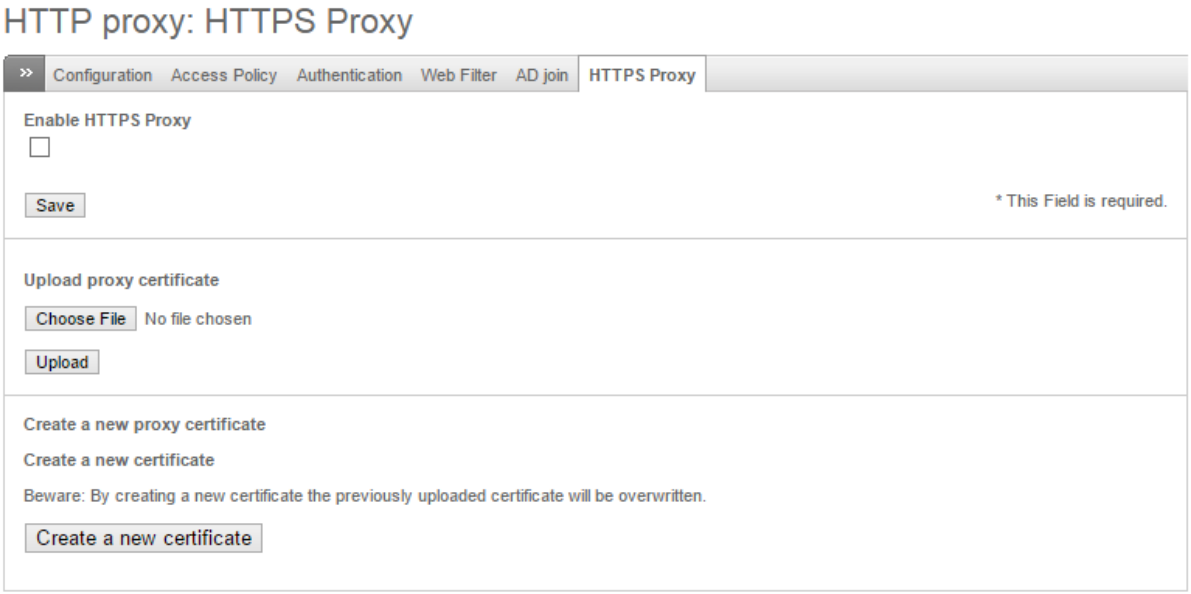


Фиг. 9.22 Включване на EFW към Microsoft AD

HTTPS прокси

HTTPS прокси функциите на EFW позволяват да се филтрира SSL криптиран трафик. При тяхното активиране, всяка заявка от страна на клиентите към отдалечен сървър се анализира и филтрира аналогично на тези при HTTP. Съществена разлика е, че е необходимо EFW да може да използва цифровия сертификат, свързан с трафика (за да може да извърши прокси заявката и съответния анализ). HTTPS прокси настройките са в три групи:

1. Активиране или деактивиране на HTTPS прокси – в тази група е включена опцията “Enable HTTPS Proxy”;
2. Управление на сертификатите;
3. Генериране на нов прокси цифров сертификат.



Фиг. 9.23 Конфигуриране на HTTPS прокси при EFW CE

POP3 и SMTP прокси

Функциите на EFW CE, свързани с POP3 и SMTP прокси са разгледани в главата, разглеждаща подробно защита на електронната поща с EFW.

FTP прокси

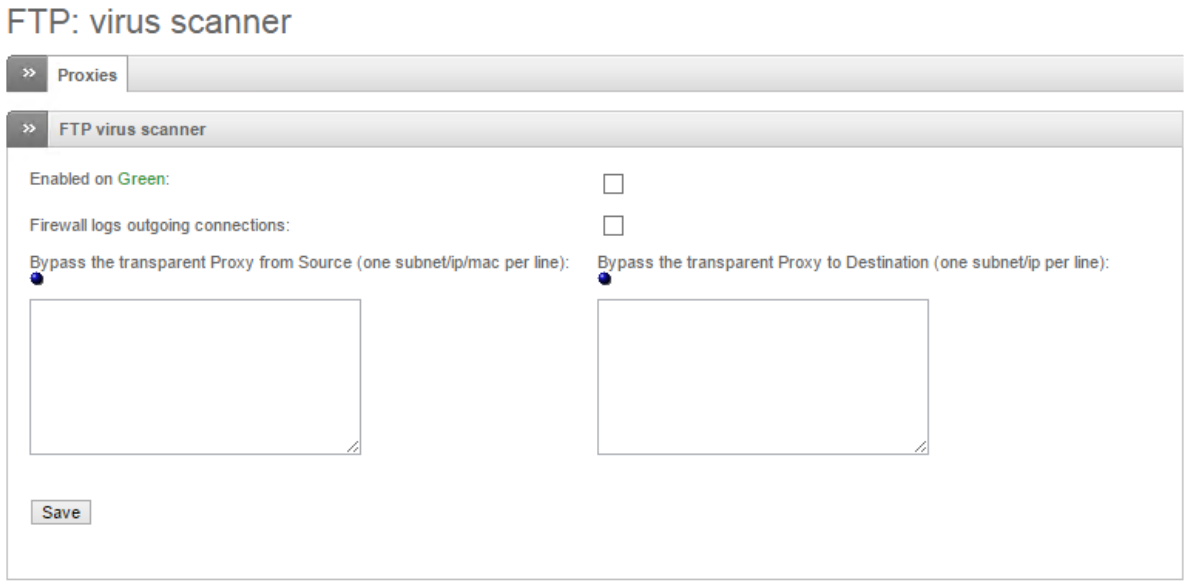
EFW CE може да работи като прозрачно FTP прокси, между посочените от администраторите зони, като изтегляните файлове през FTP протокола се сканират за наличие на зловреден софтуер. Използвания софтуерен пакет е frox[80]. Важно е да се отбележи, че единствено връзките, осъществени на TCP порт 21 се пренасочват към този прокси процес.

Параметрите, които могат да бъдат конфигурирани са:

- Enabled on GREEN, Enabled on BLUE, Enabled on ORANGE – активиране на FTP прокси за съответната зона;
- Firewall logs outgoing connections – записване на данни за съответната изходяща връзка в журнал;
- Bypass the transparent Proxy from Source – "заобикаляне" на FTP прокси функцията за посочените IP адреси, MAC адреси или мрежи, ако източника е включен в списъка;
- Bypass the transparent Proxy to Destination – "заобикаляне" на FTP прокси функцията за посочените IP адреси, MAC адреси или мрежи, ако получателя е включен в списъка.

След натискане на бутонът "Save" направената конфигурация не е нужно да бъде потвърждавана (не се налага натискане на бутона "Apply").

Препоръчително е да се обърне внимание на възможностите на frox при анализ на FTP в активен и пасивен режим, които кратко са описани на страницата с официалната документация на Endian - http://docs.endian.com/3.0/utm/proxy/ftp.html.



Фиг. 9.24 Конфигуриране на FTP прокси при EFW CE

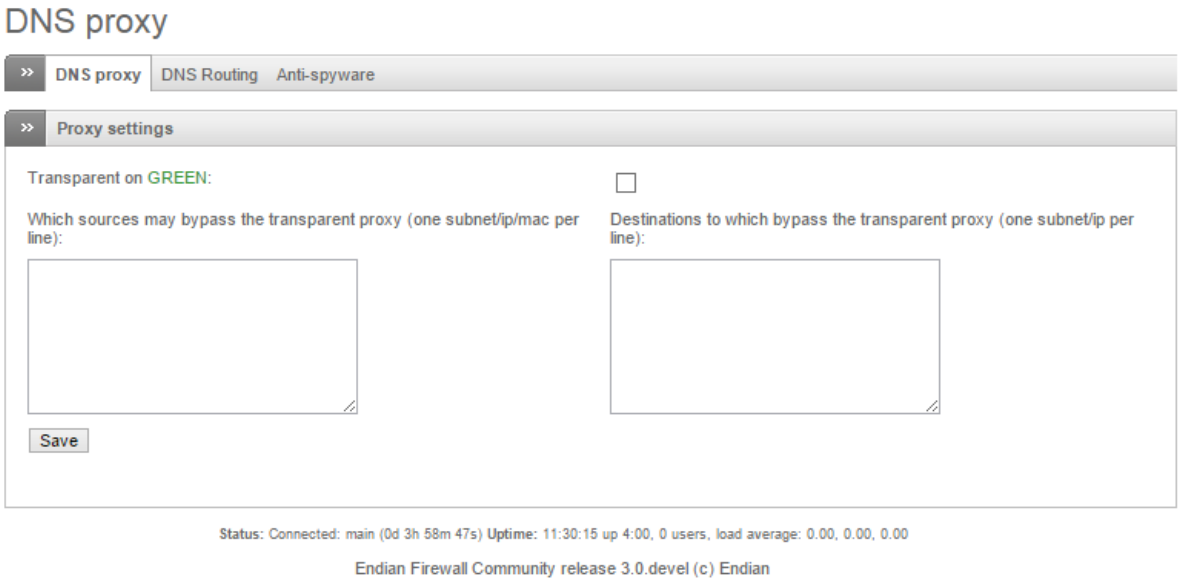
[80] frox.sourceforge.net

DNS прокси

DNS прокси функциите получават DNS запитванията от клиентите, анализират ги и препращат отговорите към тях. Предимство е, че по този начин не винаги е необходимо да се инициира DNS заявка към отдалечен сървър, което може да доведе до значително повишаване на бързодействието от гледна точка на потребителите (най-вече при често повтаряни еднакви търсения).

Основните конфигурационни параметри на този прокси сървър са:

- Transparent on Green, Transparent on Blue, Transparent on Orange – активиране на прозрачно DNS прокси за посочените зони;
- Which sources may bypass the transparent proxy – източниците на DNS запитване, посочени като IP или MAC адрес или в дадената мрежа не се обработват от прокси функцията;
- Destinations to which bypass the transparent proxy – получателите на DNS запитване, посочени като IP адрес или в дадената мрежа не са обект на анализ на прокси функцията.



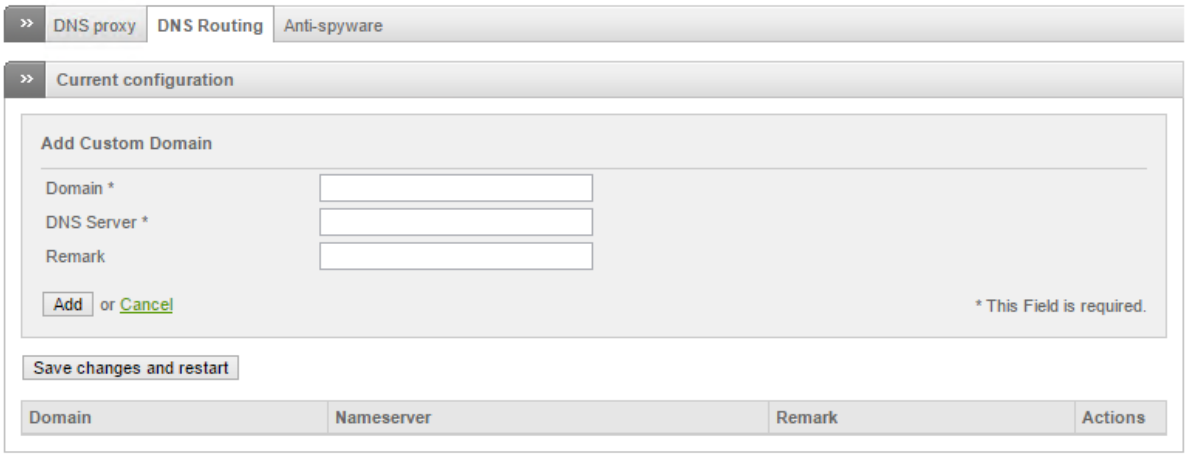
Фиг. 9.25 Конфигуриране на DNS прокси с EFW CE

DNS прокси поддържа и конфигурационни параметри, свързани с пренасочването на DNS трафика, които са:

- Domain – домейна, за който ще се използва посочения DNS сървър;
- DNS Server – IP адресът на DNS сървъра;
- Remark – опционален коментар.

След въвеждане на параметрите и натискане на бутона “Add” направената конфигурация се добавя и аналогично на повечето подобни страници в EFW данните се визуализират в таблица, като от колоната “Actions” може да се редактират или изтрият.

Пълната конфигурация се записва и активира чрез “Save changes and restart”.

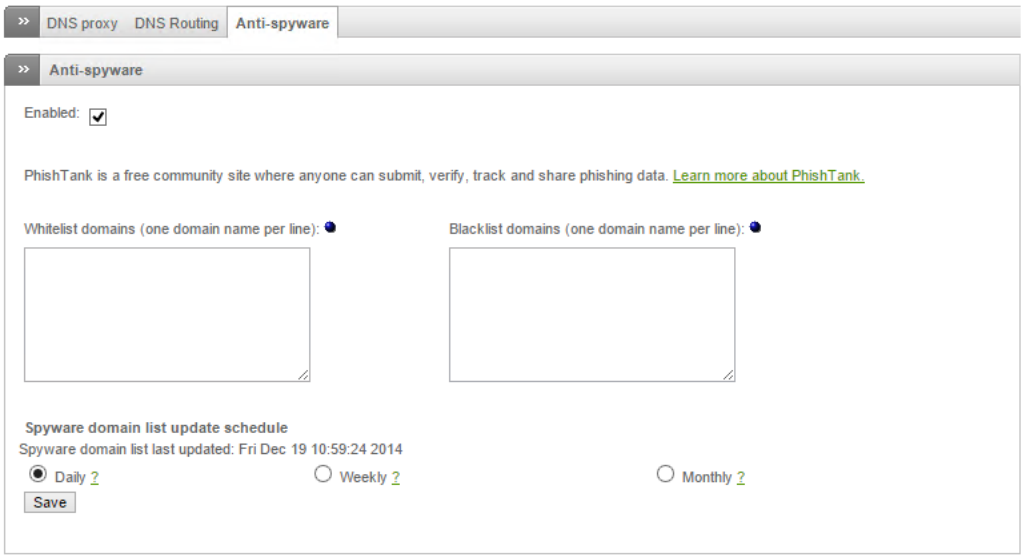


Фиг. 9.26 Конфигуриране на DNS маршрутизирането с EFW CE

Последната група от конфигурационни параметри е свързана с Anti-spyware проверките при DNS. Използва се публичната услуга Phish Tank[81], която предоставя възможност всеки да добави потенциални phishing сайтове, които в последствие да бъдат проверени и оценени като такива.

Конфигурационните параметри в тази страница са:

- Enabled – активиране или деактивиране на Anti-spyware проверката;
- Whitelist domains – включените домейни не се приемат за spyware, дори да са описани като такива от Phish Tank;
- Blacklist domains - включените домейни се приемат за spyware, дори да не са описани като такива от Phish Tank;
- Spyware domain list update schedule – период за опресняване на данните за spyware домейните.



Фиг. 9.27 Конфигуриране на Anti-spyware при DNS прокси с EFW CE

[81] www.phishtank.com

Препоръки

1. Конфигурирането на HTTP и HTTPS прокси е необходимо за да може съответния трафик да се анализира задълбочено от EFW и потребителите да могат да разчитат на антивирусно сканиране, както и на филтриране на потенциално опасни URL;
2. Конфигурационните параметри на HTTP и HTTPS прокси сървърите трябва да са направени на база на корпоративната политика за сигурност;
3. Препоръчително е конфигурацията, свързана с филтрирането на трафика да се направи на тестова EFW система, след което да бъде активирана и на системите, които анализират и филтрират реалния мрежови трафик;
4. Силно препоръчително е журналите, свързани с прокси функциите да се преглеждат внимателно и периодично.

Заключение

Включените прокси функции при EFW позволяват да се извърши задълбочено анализиране на HTTP и HTTPS трафика, като по този начин може да се активират услуги като URL филтриране, автентификация и антивирусно сканиране. Чрез тази функционалност EFW може да се използва не само като мощна защитна стена, но и като пълноценно UTM решение.

Използваните програмни пакети са изключително надеждни, като необходимите антивирусни дефиниции и списъци с правила могат да бъдат обновяване на всеки час (при актуализиране от страна на разработчиците).

URL филтрирането води до повишаване на защита на потребителите от достъп до сайтове с потенциално наличен зловреден код, както и до увеличаване на производителността на отделните екипи.

Източници

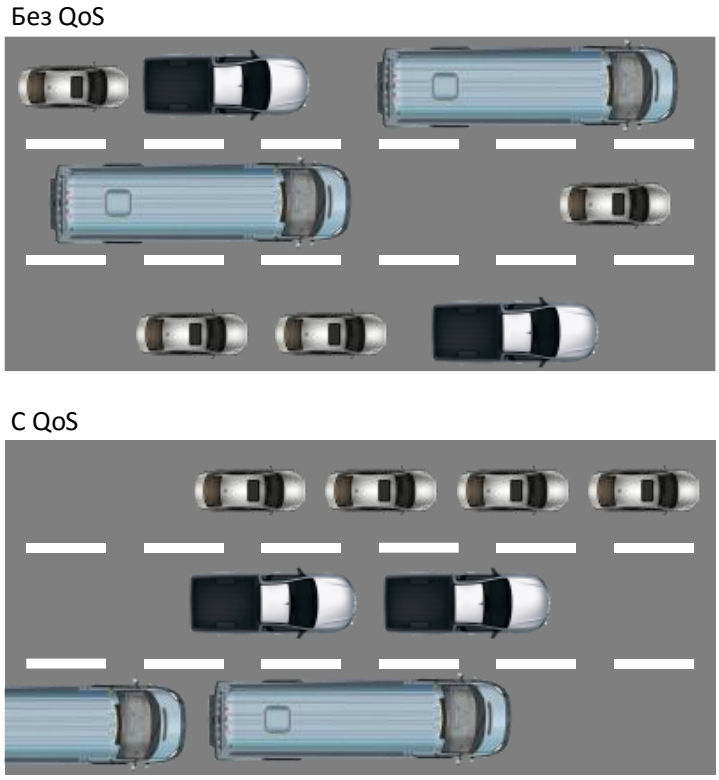
1. http://www.symantec.com/security_response/publications/threatreport.jsp

Глава 10. Конфигуриране на QoS при Endian CE

Пренасянето на различен вид трафик в една комуникационна система е често срещано, а смесването на потоците от данни, свързани с файлове, Web съдържание, VoIP и видео е нормално за почти всички типове потребители – от домашните до големите фирмени мрежи. Много често даден комуникационен поток може да използва почти целия наличен канал, което да доведе до значително забавяне на други типове трафик, а това от своя страна и до проблеми за приложенията - например обмяната на големи обеми от данни, свързани с пренос на файлове може да забави VoIP пакетите. Именно при VoIP трафика има изискване към забавянето – то не трябва да надхвърля 150 ms, тъй като над тази стойност вече може да се получи осезаем за потребителите проблем в качеството на връзката.

За да се избегне проблема от претоварване на канала от определено приложение може да се използва технологията Quality of Service (QoS). Много често концепцията за QoS се счита за сложна и трудна за конфигуриране, но при внимателно разглеждане и разбиране тя е лесна както за конфигуриране, така и за поддръжка.

Една проста аналогия на QoS е движение на превозни средства по път с няколко ленти. Ако по-бавните превозни средства използват всяка лента произволно те биха забавили по-бързите от тях. Ако обаче по-бавно движещите се използват най-дясната лента, а най-бързите – съответно най-лявата трафика ще се придвижва по-лесно и за единица време ще премине по-голям брой от превозни средства.



Фиг. 10.1 Аналогия на QoS

Технология QoS

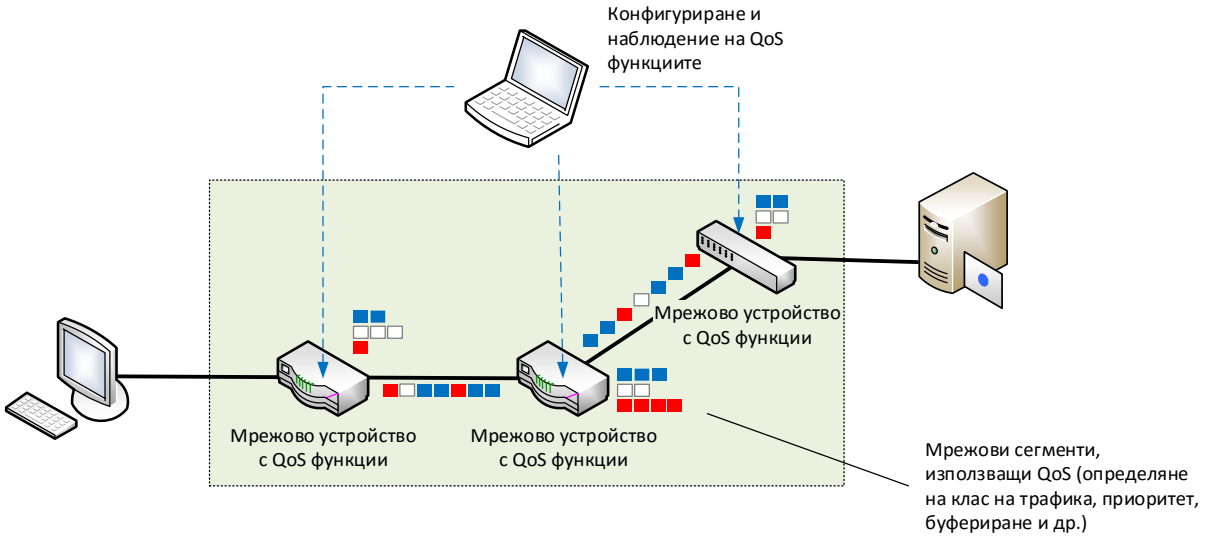
Най-общо за технологията QoS може да се каже, че тя подобрява работата на отделните услуги, чрез определяне на клас и приоритизиране на трафика в отделни потоци. Всеки един поток има определен приоритет, на базата на който пакетите от него се поставят в определен

буфер. Аналогично на потоците и буферите имат приоритет, чрез който се посочва дали тяхното съдържание де се обработва по-бързо или по-бавно.

Съществуват различни модели и средства за поддържане на QoS при мрежовите устройства, както и са достъпни редица стандарти.

Основната архитектура на QoS зависи от това дали се използва едно мрежово устройство или няколко по пътя на пакетите от източника към получателя и обратно, но в общия случай тя съдържа:

1. Функции за определяна на класа на трафика и маркиране на пакетите;
2. Функции за буфериране на пакетите на база на QoS маркерите;
3. Софтуер за конфигуриране и наблюдение.



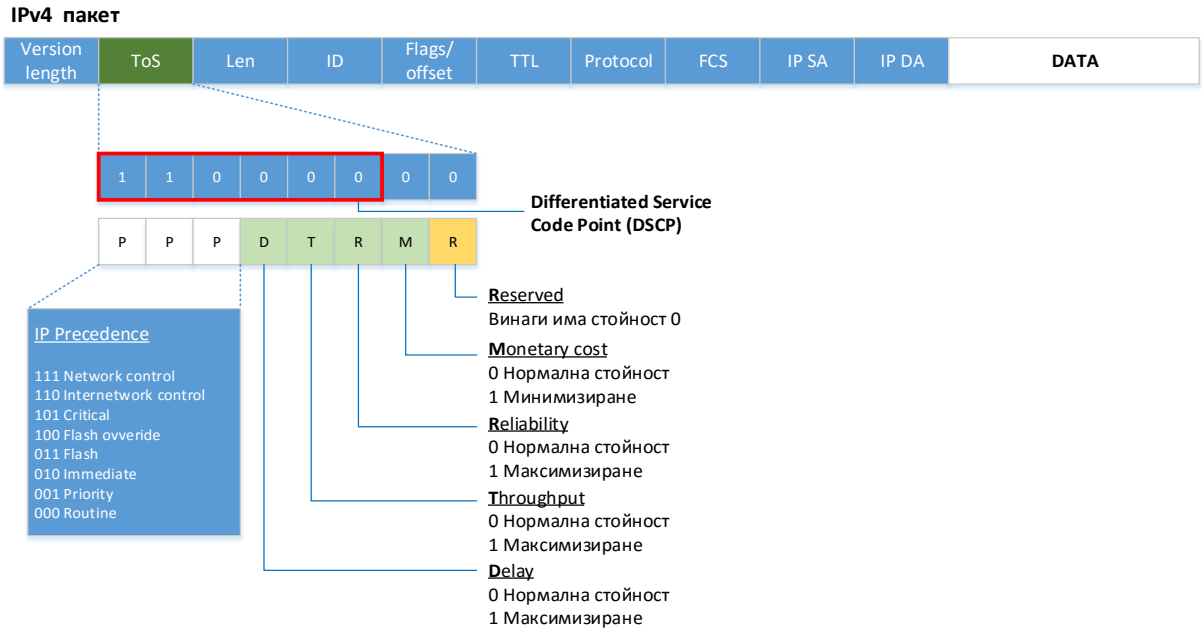
Фиг. 10. 2 Обобщен модел на технологията QoS

Класове трафик

За да може да се зададе приоритет (степен на важност) на трафика е необходимо предварително пакетите да бъдат дефинирани като конкретен тип и в последствие към тях да бъде добавен маркер (в дадени ситуации може да не се пристъпи към маркиране). Тези две действия дефинират процеса на задаване на класа при QoS.

Ако пакетът е определен като тип, но не е маркиран се дефинира т.нар. класификация на база устройство (per-hop classification). Тази ситуация най-често се получава, когато пакетите остават в рамките на системата и не се пренасочват към маршрутизатор. Също така това се отнася и за опашките с приоритет PQ (Priority Queuing) и за допълнителните опашки CQ (Custom Queuing).

Ако IP пакетите са маркирани и се пренасочват по отделните мрежови сегменти се използват съответните битове в IP хедъра (IP Precedence Bits) – фиг. 10.3.



Фиг. 10.3 Значение на IP Precedence Bits

Най-често определянето на типа на трафика се извършва на базата на ACL (Access Control Lists), маршрутизиране с политики (Policy Based Routing), CAR (Committed Access Rate) или NBAR (Network-Based Application Recognition).

QoS при една система

За всяка отделна мрежова система, която поддържа QoS се използват:

- Управление на натоварването (**congestion management**);
- Управление на опашките (**queue management**);
- Ефективност на връзките (**link efficiency**);
- Ограничаване на трафика и политики (**traffic shaping and policing**).

Управлението на натоварването е важно, най-вече при пренос на VoIP, видео потоци или големи обеми от данни, които могат да претоварят връзката. При тази ситуация мрежовите устройства могат да буферират постъпващите пакети и да ги обработват на принципа FIFO. Друг подход е да се извърши отново буфериране, но на база на тип на трафика. При този подход се използват опашки PQ, CQ, WFQ (Weighted Fair Queuing) и CBWFQ (Class-Based Weighted Fair Queuing).

Размерът на буферите е ограничен и това води до възможност за тяхното препълване – ситуация, при която ново постъпилите пакети се отхвърлят (tail drop). Ако отново се върнем към примера за пренасяне на гласови или видео потоци се вижда, че по този начин има опасност VoIP трафика да бъде забавен, както и е налична възможност за потенциална загуба на информация (VoIP и протоколите за поточно видео най-често използват не-надежден пренос).

За да се избегне тази проблемна ситуация за **управлението на опашките** се използват следните два подхода:

1. Анализ на буферите с цел избягване на тяхното препълване, което гарантира налично пространство за пакети с висок приоритет;
2. Дефиниране на критерии за отхвърляне на пакетите - ако е необходимо първо се отхвърлят пакетите с нисък приоритет.

Технологията Weighted Random Early Detect (WRED) използва горните два механизма.

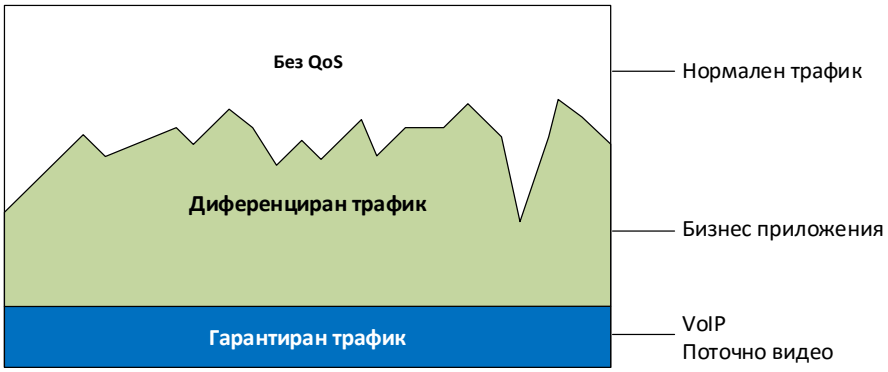
Много често бавните връзки се явяват проблем, най-вече при пренос на пакети с малка големина, например, ако трябва да се пренесе VoIP пакет, непосредствено след пакет с максимална големина при 56 kbps[82] линия забавянето ще бъде приблизително 214 ms, което надхвърля изискването на VoIP. **Ефективното използване на връзките** може да фрагментира големите пакети до размер приблизително еднакъв с този на VoIP пакетите, както и да редуцира някои от излишните служебни битове в хедърите (например компресия на RTP хедъра).

За да се оптимизират потоците от информация QoS технологията позволява да се зададат ограничения за използвания капацитет (traffic shaping). Разликата с ограничаването на трафика по политики, е че при политиките, ако обема на данните надвишава праговата стойност пакетите не се буферират.

QoS при няколко мрежови устройства

При пренос на пакети в хетерогенни мрежи, използващи множество мрежови устройства, от гледна точка на QoS може да има:

1. **Липса на QoS** (Best-effort) – използват се FIFO буфери и липсва диференциране на потоците;
2. **Диференциране на потоците** (Soft QoS) – някои типове трафик имат по-висок приоритет и се обработват с предимство. Използват се PQ, CQ, WFQ и WRED;
3. **Гарантиране на потоците** (Hard QoS) – пълно резервиране на мрежови ресурси за определен тип трафик. Използват се RSVP и CBWFQ.



Фиг. 10.4 QoS при няколко мрежови устройства

Както вече беше дефинирано за да се зададе приоритет на даден поток от данни пакетите предварително трябва да бъдат селектирани и при необходимост маркирани. Този процес се нарича “classification”. Първоначално мрежовите устройства използват технологията за филтриране на трафика (най-често ACL) за да определят пакетите, които трябва да се маркират. В последствие PQ и CQ и дефинирането на класа се използват за всяко устройство по отделно. При определени системи и конфигурации се използва и CBWFQ.

По-новите технологии за маршрутизиране на трафика се базират на политики и при тях QoS е свързан с CAR, като по този начин степента на гъвкавост се повишава значително, защото трафика може да се определи и на базата на параметри като потребител, източник и получател, приложение и др. Най-често се дефинират QoS настройки за IP мрежи или подмрежи, което прави целесъобразно тази функционалност да се използва максимално близко до условната

[82] Стара технология, но подходящ пример за дефиниране на нуждата от QoS

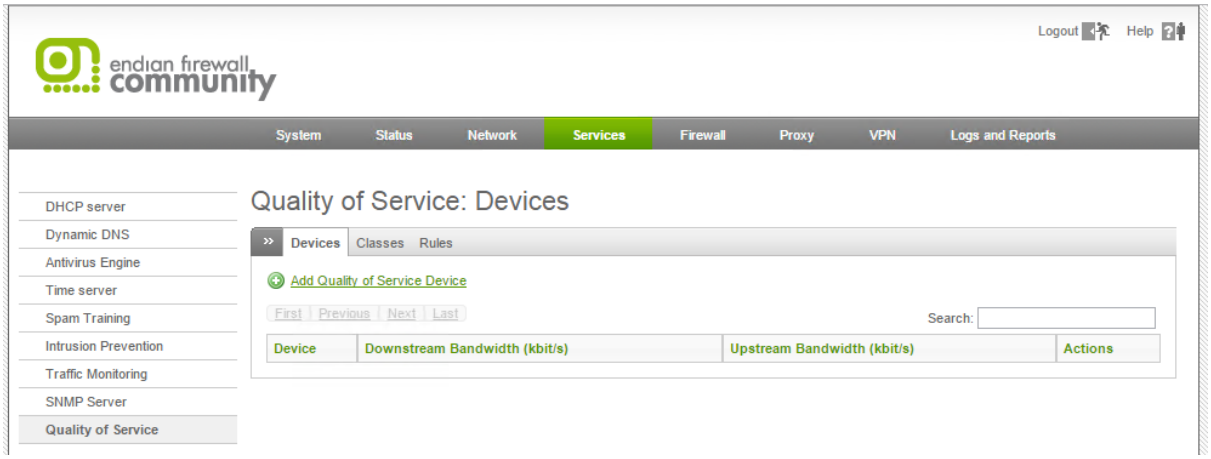
граница на мрежата (network edge).NBAR позволява трафика да се селектира още по-точно, като се включват и параметри като URL, анализ на HTTP пакети и др.

Конфигуриране на QoS с EFW CE

QoS функциите при EFW позволяват да се зададе приоритет на IP трафика, който преминава през системата, на база на използваните услуги. Конфигурирането се извършва от менюто “Services” и подменюто “Quality of Service”.

QoS настройките съдържат три части:

1. Устройства (Devices);
2. Класове (Classes);
3. Правила (Rules).



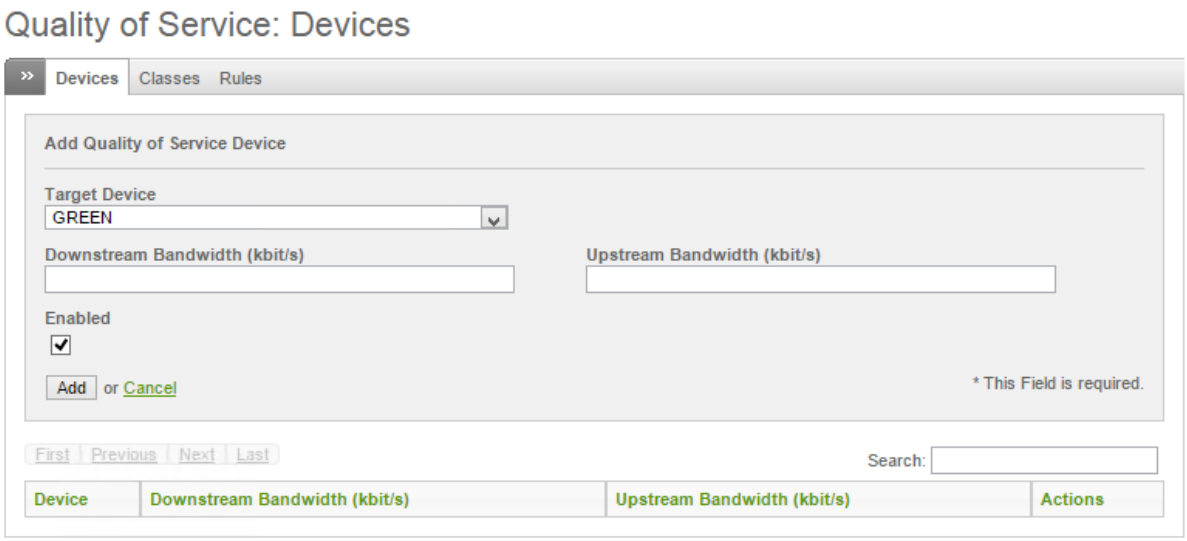
Фиг. 10.5 Страница за конфигуриране на QoS при EFW CE

След инсталиране на EFW CE в QoS конфигурацията няма дефинирани устройства и таблицата в частта “Devices” съответно е празна. Добавянето на нова система става от “Add Quality of Service Device” като е необходимо да се зададат следните параметри:

- Target Device – мрежови интерфейс, който се използва от системата. Възможно е да се посочи зона, физически интерфейс, uplink или VPN тунел;
- Downstream Bandwidth (kbit/s) – скоростта на трансфер на изходящия трафик от интерфейса към източника, зададен в Kbps;
- Upstream Bandwidth (kbit/s) – скоростта на трансфер на изходящия трафик от интерфейса към получателя, зададен в Kbps;
- Enabled – активиране или деактивиране на настройката.

След въвеждане на параметрите е необходимо да се натисне бутона “Add” и промените да се потвърдят чрез “Apply”.

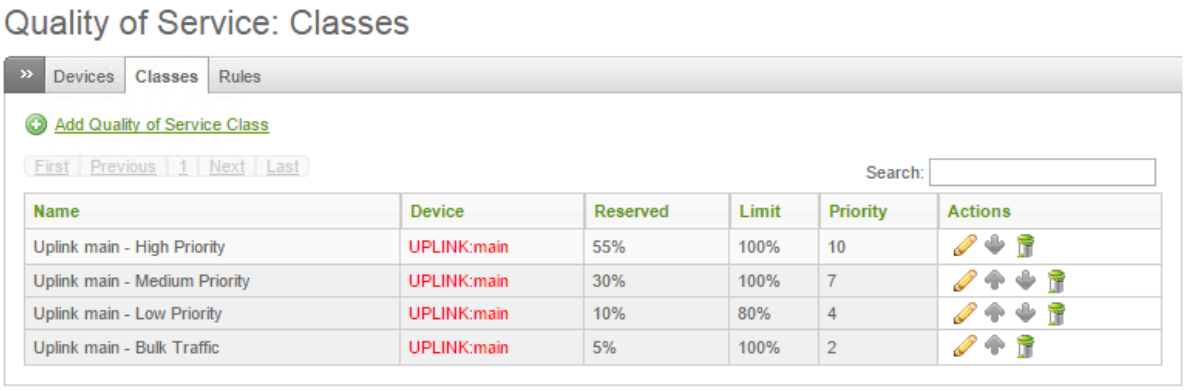
Аналогично на повечето страници при EFW от колоната “Actions” в таблицата данните за устройствата могат да бъдат редактирани, изтривани или активирани и деактивирани.



Фиг. 10.6 Добавяне на устройство за QoS при EFW CE

Страницата “Classes” съдържа информация и предоставя възможност за конфигуриране на класовете трафик за QoS услугите. По подразбиране са дефинирани 4 класа:

1. **Висок приоритет** (High Priority) – резервирани са 55% от пропускателната способност на main uplink, максималното натоварване може да достигне 100%, а стойността на приоритета е 10;
2. **Среден приоритет** (Medium Priority) – резервирани са 30% от пропускателната способност на main uplink, максималното натоварване може да достигне 100%, а стойността на приоритета е 7;
3. **Нисък приоритет** (Low Priority) – резервирани са 10% от пропускателната способност на main uplink, максималното натоварване може да достигне 80%, а стойността на приоритета е 4;
4. **Останал трафик** (Bulk Traffic) – резервирани са 5% от пропускателната способност на main uplink, максималното натоварване може да достигне 100%, а стойността на приоритета е 2.



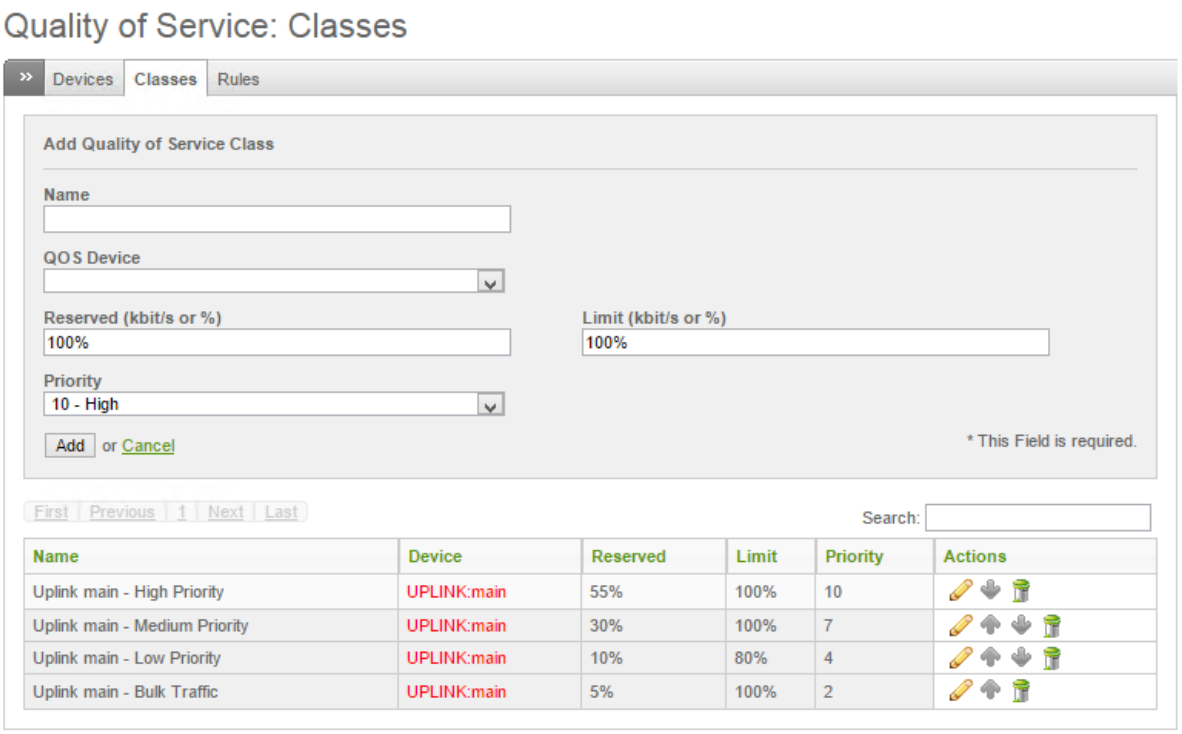
Фиг. 10.7 Класове трафик по подразбиране за QoS услугите на EFW CE

Добавянето на нов клас за трафика се извършва от връзката “Add Quality of Service Class”, като се конфигурират:

- Name – име на класа;

- QoS Device – списък с устройства, които са дефинирани чрез страницата Devices;
- Reserved – резервиран процент от пропускателната способност;
- Limit – максимален процент на натоварване, спрямо пропускателната способност;
- Priority – приоритет със стойност от 10 (максимален) до 0 (минимален).

Добавянето на класа се извършва от бутона “Add”.



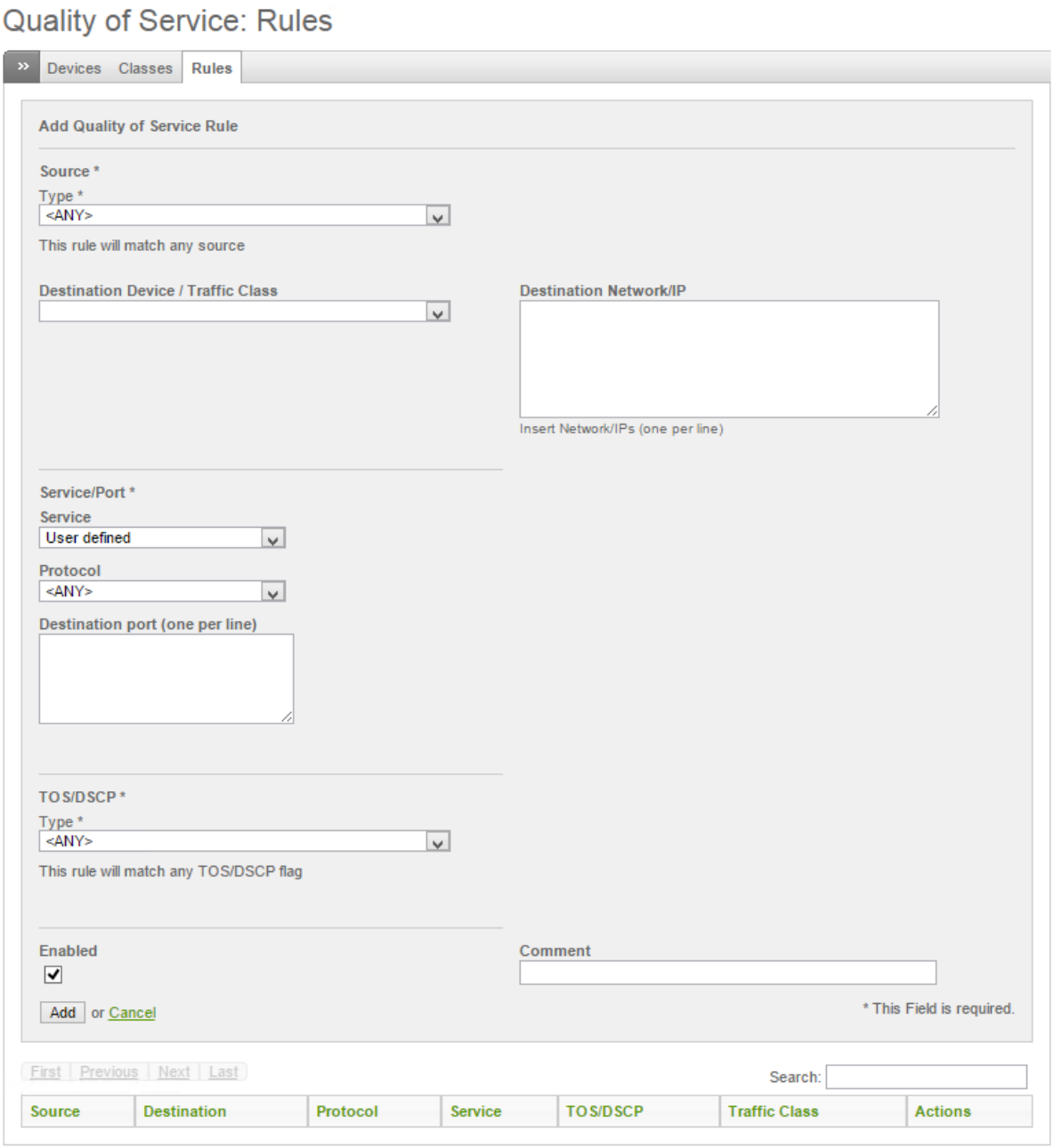
Фиг. 10.8 Дефиниране на нов клас трафик за QoS услугите на EFW CE

Третата страница от “Quality of Service” е “Rules” и съдържа необходимите конфигурационни параметри, свързани с описанието на връзката между класа на трафика и устройствата. Добавянето на ново правило се извършва от “Add Quality of Service Rule”.

Необходимо е да бъдат конфигурирани следните параметри:

- Source – източник на трафика. Може да бъде посочена зона, интерфейс, IP или MAC адрес. В зависимост от посочения тип се дефинират допълнителни параметри (например при избор на зона се посочва нейното име);
- Destination Device/Traffic Class – посочва устройството получател или класа на трафик, които се конфигурират от останалите страници в “Quality of Service”;
- Destination Network/IP – определя отдалечената мрежа или IP адрес, които трябва да се достъпват от устройството или от класа на трафика;
- Service/Port, Protocol – услуга или комбинация от транспортен протокол и порт;
- TOS/DSCP – параметри ToS и DSCP. В зависимост от избрания параметър е необходимо да се конфигурира и комбинацията трафик-клас;
- Enabled – активира или деактивира правилото;
- Remark – опционален коментар.

Добавянето на правилото се извършва от бутона “Add”.



Фиг. 10.9 Дефиниране на правила за QoS услугите на EFW CE

Заключение

Технологията QoS позволява да се оптимизират потоците от данни, което е важно при преноса на VoIP или поточно видео.

Конфигурирането на QoS при EFW CE е сравнително лесно, а графичният потребителски интерфейс предоставя всички необходими параметри.

Препоръчително е след промяна на QoS конфигурацията да се извършат и тестове на производителността спрямо класовете трафик и устройствата.

Източници

1. http://docwiki.cisco.com/wiki/Quality_of_Service_Networking

Глава 11. Въведение в криптографията

Много мрежови технологии (протоколи, услуги и др.) използват средства за защита на информацията, базирани на криптографски методи. Това е само една от многото причини мрежовите специалисти и администратори да мат познания, свързани със защитата на информацията и в частност с науките кроптография и криптографски анализ.

На 3.11.1990 година пред сградата на ЦРУ в град Лангли е открита скулптурата на Джеймз Санбърн, наречена Криптос (Kryptos). Този паметник е една от най-актуалните мистерии, свързани с криптографията, като в своя блог Весела Христова е дава кратко, но изключително интересно описание. Терминът “криптос” от гръцки се превежда като таен, а поставянето на паметник пред обществени сгради не е нещо ново или уникално само по себе си. Интересното в този случай е, че Криптос се намира на най-подходящото за него място, тъй като творбата представлява шифрован текст, което е в пълен унисон с някои от дейностите на сградата до него. Инсталацията е изградена от червено дърво, бял кварцов камък, мед и бетон. Металната плоча е оформена във вида на буквата “S” и върху нея е разположен шифрирания (криптирания или кодирания) текст. В продължение на над 20 години криптоаналитиците се опитват да разшифроват съдържанието на текста и към момента имат успех в три от четирите части. Последната секция, наречена K4 все още не е декодирана и този текст се е превърнал в една от най-интересните неразгадани тайни днес.

Фиг. 11.1 Криптос (източник Интернет)

За създаването на паметника и включения в него код Санбърн използвал опита на известния криптограф Шайд. Броят на поставените символите е 865, а стойността на скулптурата – 250 000 долара.

Първите седем години след откриването на паметника не бил постигнат никакъв напредък по разшифроването на текста. Пробивът е дошъл през 1998 година, когато агентът на ЦРУ Дейвид Стейн представил своята версия. Дейвид прекарал над 400 часа в опити да свали булото на мистерията. След половин година криптоаналитикът Джим Джилогли, използвайки

собствена програма и домашния си компютър, успешно разшифрова трите секции. При кодирането на първата секция е прилаган модифицираният шифър на Виженер (Vigenère), който използва ключови думи, а втората и третата секции използват и други криптографски методи. Четвъртата секция се оказва доста по-загадъчна и сложна за декодиране.

Текстът на К1 (първият фрагмент/секция) е дешифриран като “Between subtle shading and the absence of light lies the nuance of iqlusion”. В думата “iqlusion” има преднамерена правописна грешка. В превод изречението гласи: "Между затъмнението и липсата на светлина има нюанс на илюзия”

К2 е телеграфен текст, дешифриран като "...Възможно ли е това. Те използват магнитните полета на Земята. Информацията е събрана и предадена на неизвестно място под Земята. Знаят ли в Лангли за това? Би трябвало – там някъде нещо е заровено. Кой знае точното място? Само WW…”и следват географски координати в близост до щаб квартирата на ЦРУ. Естествено, инициалите WW приковават вниманието на любителите на загадки и раждат различни трактовки. Писателят Дан Браун изказал предположението, че това са инициалите на Мария Магдалена, само че обърнати, но през 2005 година Санбърн го опровергава – WW означава Уилям Уебстър, директор на ЦРУ по времето, когато Криптос е поставен.

Третият фрагмент цитира част от дневника от 26.11.1922 на египтолога Картър при отварянето на гробницата на Тутанкамон: “Бавно, много бавно скалните останки, с които бе затрупана долната част на прохода, бяха отстранени. С трепереща ръце направих малка дупка в левия горен ъгъл. След като разширих отвора, мушнах вътре свещ и погледнах. Горещият въздух, който излизаше от камерата, караше пламъка да трепти, но вътрешността тънеше в мрак. Виждате ли нещо?”.

По повод 20-годишния юбилей на скулптурата Санбърн разкрива пред „Ню Йорк Таймс” част от загадката: съчетанието от буквите NYPVTT, които са от 64 до 69 позиция в К4, означава BERLIN.

Някои наричат Санбърн “агент на дявола” заради думите му, че никога не ще разкрие тайната на произведението си. Това, разбира се, е доста преувеличено и незаслужено, защото според него идеята на инсталацията е да се отдаде почит на бойците на "тихия фронт", както и да се отправи предизвикателство към техните умения за разкриване на тайни.

Важни термини

Някои от по-важните термини, свързани с криптографията и криптографския анализ са:

- **Открит (изходен) текст** — данни (не задължително текстови), предавани без използване на криптография;
- **Шифрован (закрит) текст** — данни, получени след използване на криптосистема с указан ключ;
- **Криптосистема** — семейство обратими преобразувания на откритият текст в шифрован;
- **Ключ** — параметър на шифъра, определящ избора на конкретно преобразуване на даденият текст. В съвременните шифри алгоритъма на шифриране е известен и криптографската устойчивост на шифъра изцяло се определя от секретността на ключа;
- **Криптоанализ** — наука, изучаваща математическите методи за нарушаване на конфиденциалността на информацията. Криптографията и криптоанализът съставят криптологията, като единна наука за създаване и разбиване на шифри;
- **Криптоаналитик** — лице, създаващо и прилагащо методите на криптоанализа;

- **Криптографска атака** — опит на криптоаналитик да предизвика отклонения в атакуемата защитена система за обмен на информация. Успешната криптографска атака се нарича разбиване или отваряне;
- **Шифриране (криптиране)** — процес на нормално прилагане на криптографско преобразуване на открит текст на основата на алгоритъм и ключ, в резултат на което възниква шифрован текст;
- **Дешифриране (декриптиране)** — процес на нормално прилагане на криптографско преобразуване на шифриран текст в открит.
- **Криптографска устойчивост** — способността на криптографския алгоритъм да противостои на криптоанализ.

Криптография

Криптографията е науката, свързана с разработването и изследването на методите и алгоритмите за създаване на секретни кодове, както и за съответното шифриране и дешифриране на информацията. По-точно е определението, че “криптография” се използва при разработката и прилагането на системите за шифроване, а “криптоанализ” при използването и разработването на методи за “разбиване” на кодове.

Исторически сведения

Криптографията се прилага от векове, като една от най-важните и цели е да се пренесе информация между две страни, която да е надеждно защитена и при попадане в трети лица тя да не може да бъде разчетена.

В своята книга "Живота на 12-те Цезари" римският историк Сетон описва методите, които Юлий Цезар е използва за да шифрова своята кореспонденция. Този начин на кодиране е известен като “шифър на Цезар” и използва заместване (субституция) на букви от азбуката с други при предварително известни критерии. Цезар е използвал два диска, с надписани по тях букви. Дисковете са симетрично поставени и позволяват да се завъртят един спрямо друг (спрямо общ център). По този начин с определен брой премествания може да се определи на коя буква от откритият текст, коя трябва да се изпише при криптирането и обратно.

Фиг. 11.2 Дискове, прилагани за шифриране и дешифриране на текстове с шифъра на Цезар

Кодирания текст се изписва слято, например:

Оригинален текст:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ако стъпката е 23 се получава кодиран текст:	XYZABCDEFGHIJKLMNOPQRSTUVW

Този метод е бил прост и ефективен, но от гледна точка на сигурност не е надежден. Много лесно може да се направи последователно заместване на даден символ с друг и след максимум 23 опита текста ще се декодира успешно. В момента най-ефективният начин за декодиране на този код е да се използват статистически алгоритми, които на база на най-често използваните символи в азбуката могат лесно и бързо да предскажат стъпката на преместване.

Друг пример за използването на крипотография през вековете е скитала (scytale). Това е инструмент за кодиране на съобщения чрез транспозиция (промяна на мястото на символа), използван от древните гърци и спартанци (приблизително през 7ми век преди Христа). Скитала представлява дървено трупче, оформено в многостен, върху който се навива кожена лента, на която се изписва текста по редове. След разплитане на лентата от трупчето се получава вертикална последователност от объркани букви, която може успешно да се разчете ако се увие около многостен със същия профил, като на използвания при изписването на текста.

Аналогично на шифъра на Цезар при последователно увиване на лентата около многостени с повишаване на броя на страните текста ще се разчете успешно.

Фиг. 11.3 Скитала

През 16ти век френският математик Виженер (Blaise de Vigenère) създава алгоритъм за шифриране на текст, който използва текстов ключ и специално дефинирана таблица за заместване. Броят на символите на ключа трябва да съвпада с този на текста и ако се използва по-къса фраза, то тя се дописва многократно (т.нар. цикличен ключ). След като дължината на ключа и на текста се уеднакви от таблицата се извършва търсене по редове и колони спрямо двата символа и се записва откритата стойност. Таблицата се нарича “tabula recta” и е описана от Йоханес Тритемиус в своя труд “polygraphia” – първата книга, свързана с криптографията (фиг. 11.4).

Кодираният текст може да се декодира ако е известен ключа.

Пример за шифроване с метода на Виженер е:

Ключ:	VIGENERE
Текст:	DECEMBER
Кодиран текст:	YMIIZFVV

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Фиг. 11.4 Таблица за заместване при шифъра на Виженер

Английският математик Чарлз Бабидж разработва тест, който може да открие дължината на ключа на Шифърът на Виженер, и тогава разглежда шифъра като сбор на обикновени Цезарови шифри. Отново ако се използва метода на криптоанализа, базиран на статистика за употребата на буквите кода сравнително лесно може да бъде разбит.

Дискът на Джеферсън или още познатия като “шифър на Джеферсън” е метод за криптиране на данни, разработен от Томас Джеферсън и представляващ подреждане на дискове с по 26 символа по средната им ос. Редът на буквите е различен на всеки един от дисковете и тяхната позиция е случайна. Всеки диск е номериран и след като се изпише текста е необходимо да се запомни подредбата на дисковете спрямо техните номера.

Фиг. 11.5 Диск на Джеферсън

Криптографските методи намират изключително широко приложение при военна комуникация. Това води до развитие както на алгоритмите, така и до създаване на все по-сложни криптиращи и декриптиращи устройства. Едно от най-известните е машината Енигма. Енигма (на немски: Enigma) е тип преносима шифровъчна машина, използвана за шифриране и дешифриране на секретни съобщения през 20-те години на XX век. Тя е използвана от началото на 20-те години на XX век за търговски цели, както и във военните и държавни служби и учреждения на много страни, но най-широко разпространение получава в Нацистка Германия по време на Втората световна война. Затова под „Енигма“ най-често се разбира немският военен модел Енигма на Вермахта (Wehrmacht Enigma). За изобретател на Енигма се счита немският електроинженер д-р Артур Шербиус, който на 23 февруари 1918 година получава първи патент за тази своя разработка.

Фиг. 11.6 Шифровъчна машина Енигма

Машината Енигма-I тежи около 10 kg и е с размери 310 mm x 255 mm x 130 mm и на пръв поглед прилича на пишеща машина. Както и другите роторни машини, Енигма се състои от комбинация от механически и електрически системи. Механичната част включва клавиатура, набор от въртящи се дискове (ротори), които са разположени около вала и са долепени до него и степенен механизъм, задвижващ един или повече ротора при всяко натискане на клавиш. Конкретният начин на работа може да е различен, но общият принцип е един и същ: при всяко натискане на клавиш най-десният ротор се измества с една позиция, а при определени условия се изместват и другите ротори. Движението на роторите води до различни криптографски преобразувания (самото шифриране) при всяко следващо натискане на клавиш на клавиатурата.

Механичните части се движат, образувайки променяща се електрическа верига, тоест фактически шифрирането на буквите се осъществява електрически. При натискането на клавиш веригата се затваря, токът преминава през различните компоненти и накрая включва една от множеството светлини индикатори (лампи), изобразяваща буквата, която ще излезе на изхода. Например, при шифроване на съобщение, започващо с ANX..., операторът първо натиска клавиш A и светвал индикатор Z, тоест Z ставала първата буква на криптограмата. Операторът продължавал шифрирането с буквата N по същия начин и т.н.

Видове криптографски алгоритми

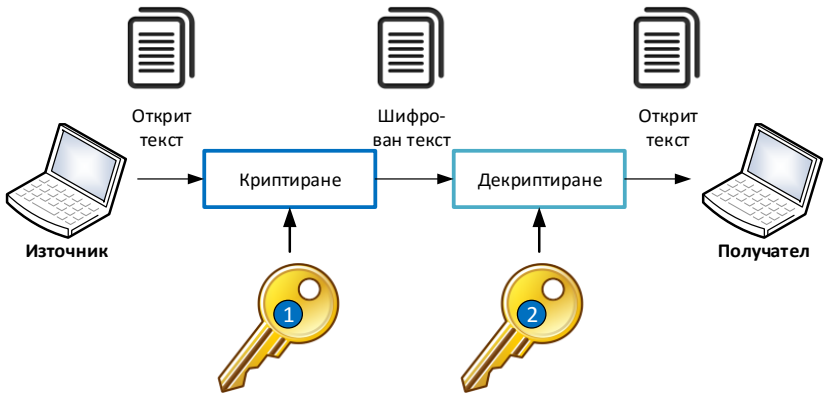
От направените по-горе примери се вижда, че алгоритмите за криптиране най-често се разделят на няколко основни типа:

- Симетрични;
- Асиметрични;
- С блоково шифроване;
- С поточно шифроване.

Симетричните алгоритми използват един и същи ключ за шифриране и дешифриране на данните, докато асиметричните се базират на двойка ключове – един за криптиране и втори за декриптиране.

Симетричен алгоритъм

Асиметричен алгоритъм



Фиг. 11.7 Симетричен и асиметричен криптографски алгоритъм

Блоковото шифроване използва алгоритъм, който последователно извлича блокове с данни от входящия открит текст, които в последствие се трансформират и стават част от изходния поток от данни. Поточното шифроване се извършва последователно, най-често при изпращането на данните, като всяка отделна единица (най-често байт) се кодира последователно с конкретен ключ от предварително генериран набор ключове.

Много често се поставя въпроса, кой алгоритъм е по-надежден или кой е по-бърз. Изборът на стандарт за шифриране най-често зависи от характера на откритата информация. Може да се обобщи, че асиметричните алгоритми изискват по-дълги ключове и са по-бавни, особено при криптиране на голям обем от открит текст. Симетричните алгоритми са по-бързи, но за сметка на това, ако ключа е известен на трето лице може данните лесно да бъдат дешифрирани.

Тъй-като повечето криптографски алгоритми са публично достъпни надеждността на защитата се определя от дължината и сложността на ключовете, както и на тяхното подсигуряване при споделяне.

Основната цел на криптоанализът е да се разработят методи за дешифриране на шифрирани текстове.

Колкото дължината на ключа е по-голяма, толкова е по-голям и размера на адресното пространство на ключовете (възможните комбинации), например:

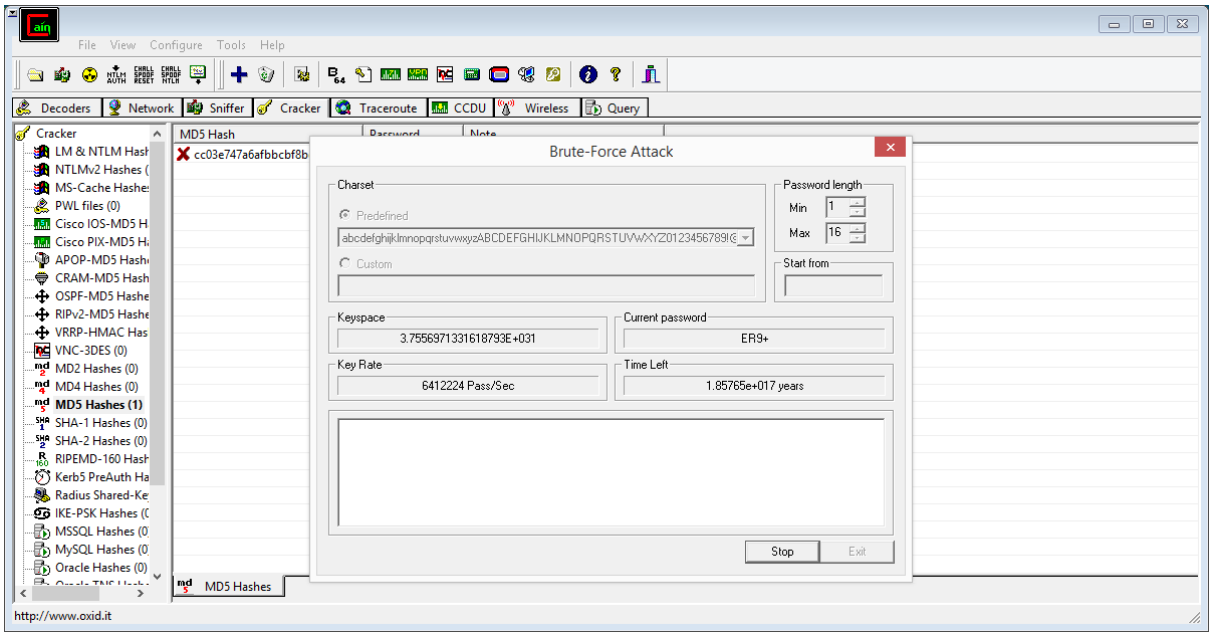
- 1 битов ключ – 2 комбинации (0, 1);
- 2 битов ключ – 4 комбинации (00, 01, 10, 11);
- 3 битов ключ – 8 комбинации (000, 001, 010, 011, 100, 101, 110, 111);
- 10 битов ключ – 1024 комбинации;
- 11 битов ключ – 2048 комбинации и т.н.

Вижда се, че добавяне само на 1 бит към дължината на ключа удвоява броя на възможните ключове.

Един от методите на криптоанализа, който винаги би дал успешен резултат е т.нар. “метод на грубата сила”, като при него се проверяват всички възможни комбинации от символи за ключ и се анализира получения дешифриран текст. Този метод е успешен при прости криптографски алгоритми, но при актуалните за момента, времето за неговото изпълнение може да надхвърли милиарди години. За да се ускори атаката с груба сила често се използват специални паралелни алгоритми и GPU ускорение, но въпреки това при надеждни ключове необходимото време за приключване на атаката е практически неизползваемо за атакуващия.

Пример за GPU оптимизиран алгоритъм за анализ на хешове е IGHASHGPU – който може да анализира на ATI HD5870 приблизително 3650 милиона MD5 хеша в секунда и около 1360 милиона SHA1 хеша в секунда.

На фигура 11.8 е показана атака по метода на групата сила към MD5 хеширана стойност, която на използвания хардуер ще приключи след $1.86x10^{17}$ години.



Фиг. 11.8 Софтуерен инструмент Cain and Abel

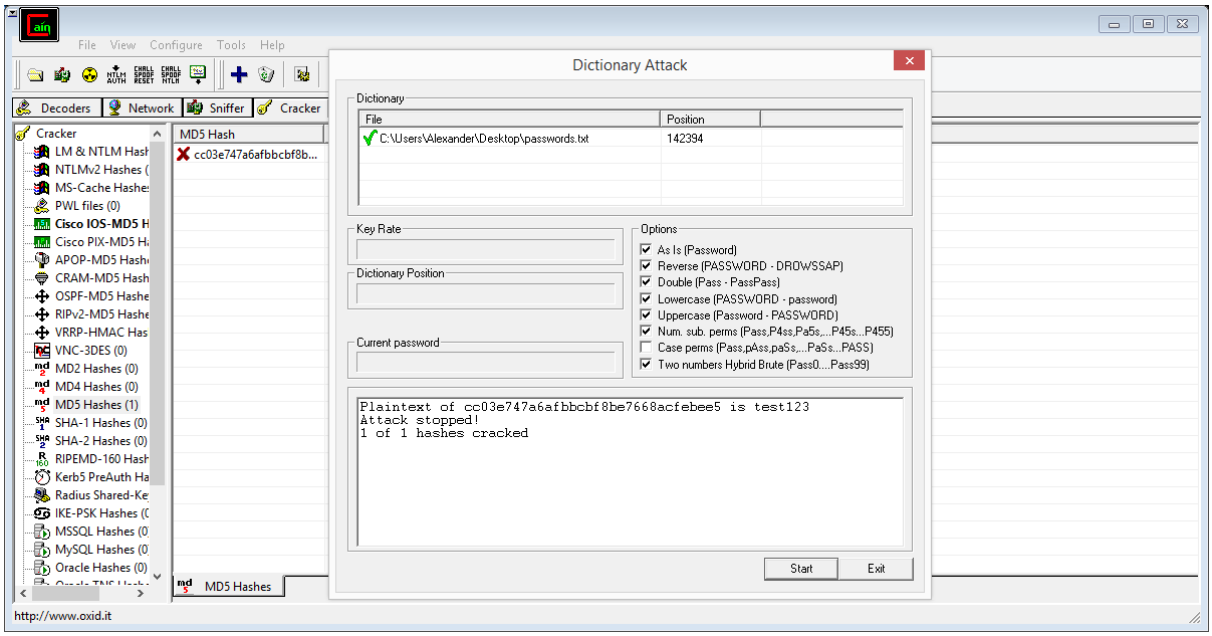
В повечето случаи методът на грубата сила е практически неизползваем (въпреки гарантирания успешен резултат), поради неговото времетраене. При атаки към шифриране текстове или хеш стойности, свързани с пароли по-често се използва т.нар. речникова атака, при която се проверяват списък с най-често използваните пароли, които се кодират или хешират със съответния алгоритъм и се сравняват получените стойности и известните такива. Примерен речник с най-често използваните пароли за определена система може да бъде с няколко милиона записа, а проверката би отнела от няколко часа до няколко дена. Въпреки, че има голяма вероятност търсената парола да се намира в речника, при спазване на правилата за надеждни пароли атакуващия ще трябва да прибегне до метода на грубата сила.

В Интернет има редица класации на най-често използваните пароли, като сайта Slashgear[83] ги описва в следната последователност:

1. 123456
2. password
3. 12345678
4. qwerty
5. abc123
6. 123456789
7. 111111
8. 1234567
9. iloveyou
10. adobe123

В примера на фигура 11.9 отново е показана атака, насочена към MD5 хеширана стойност, но този път с речников подход. Времето за откриване на текста е по-малко от секунда (в сравнение с максималното време от 1.86×10^{17} години при метода на грубата сила. Файлът с паролите, използвани при анализа съдържа 14344392 стойности.

[83] www.slashgear.com



Фиг. 11.9 Резултат от успешна речникова така с Cain and Abel

Един интересен подход, който се използва при анализ на хешове е генерирането и използването на т.нар. “rainbow table”. По същество това са бази данни, при които данните са съхранени в табличен види и които в най-основен вид съдържат колона с хеш и колона със съответната открита стойност (включително специализирани редуциращи функции). Много често се използват алгоритми, които генерират всички възможни комбинации от тези два параметъра. В последствие търсенето се извършва по хеширания текст, като благодарение на технологии като индексиране то е изключително бързо и ако стойността е налична в базата данни се връща като резултат съответния открит текст.

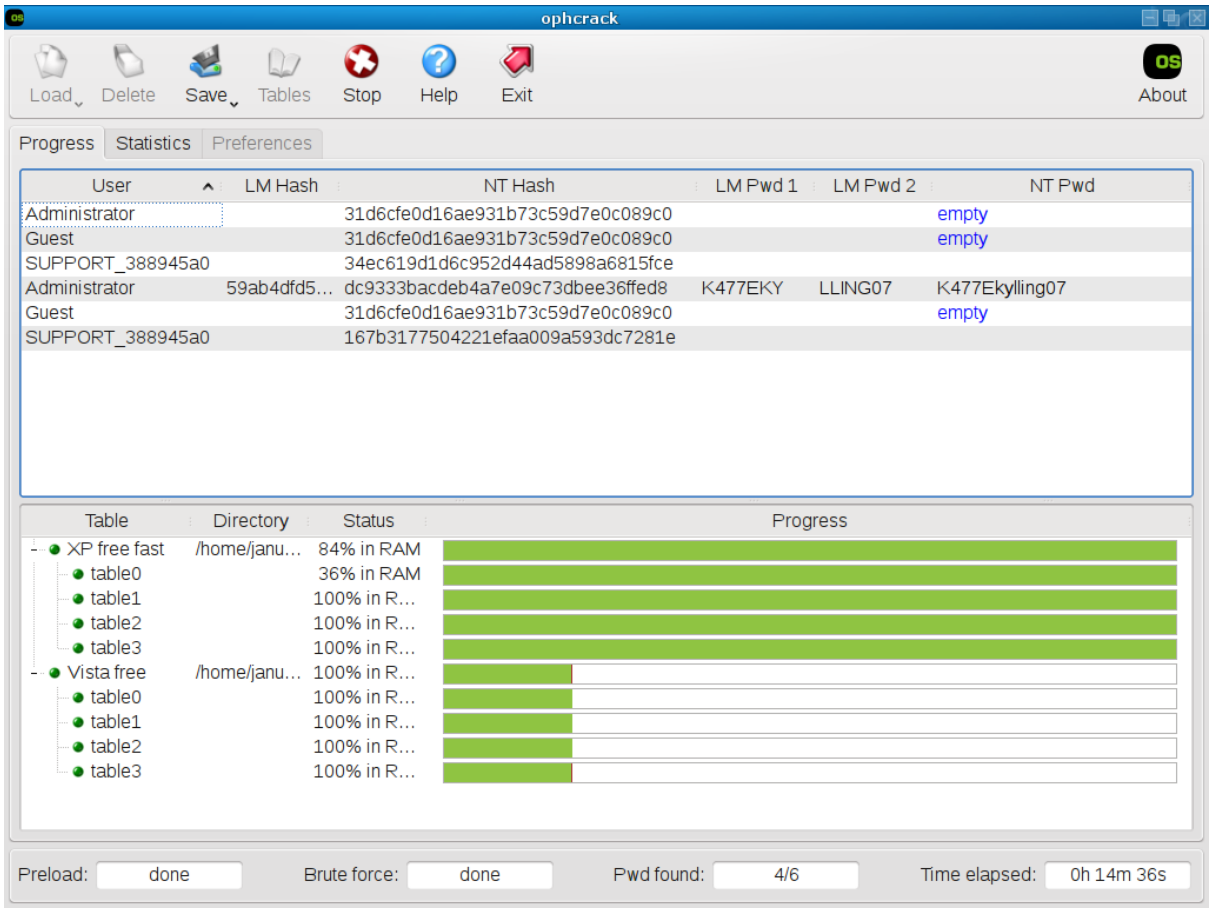
Този подход е изключително популярен и се използва от редица проекти, някои от които са свободно достъпни в Интернет, като:

- https://www.freerainbowtables.com/
- http://project-rainbowcrack.com
- Ophcrack[84] и др.

Ако в таблицата са включени всички възможни комбинации от хеш и открит текст може да се каже, че сигурността на паролите, които използват дадения алгоритъм е ниска и е препоръчително алгоритъма да бъде заменен с друг или да се добави т.нар. стойност “salt”. “Salt” е специална добавка, която е случайно генерирана и която се прибавя към входящия поток от данни, който се хешира.

“Salt” и хешът се съхраняват надеждно и при проверка се използва “Salt”, въведения текст и получената стойност се сравнява с хеша. По този начин въвеждайки псевдослучайни символи вероятността “rainbow table” да се използва успешно значително се редуцира.

[84] ophcrack.sourceforge.net



Фиг. 11.10 Криптографски анализ с “rainbow tables” чрез ophcrack

Подсигуряване на комуникацията

Както вече беше споменато криптографските алгоритми се използват при подсигуряването на комуникацията, като най-често те се прилагат в три насоки:

1. **Интегритет** (integrity) – добавяне на стойност с цел проверка дали изпратения поток от данни не е бил модифициран по време на преноса му от изпращача към получателя. Най-често се използват алгоритми за хеширане (MD5, SHA и др.) или цифрови сертификати;
2. **Автентификация** (authentication) – проверка дали двете страни в комуникацията са легитимни и имат конфигурирани необходимите пароли или други параметри за да могат да осъществят преноса на данни помежду си. Най-често този процес се базира на HMAC (Hash Message Authentication Code) или асиметрични алгоритми като RSA, DSA и др.;
3. **Конфиденциалност** (confidentiality) – шифриране на потока от данни с цел при прихващане на части от него или на цялото му съдържание информацията да не може да се дешифрира без точните ключове. Някои от най-често използваните алгоритми са 3DES, AES, RC и др.

Интегритет и автентификация

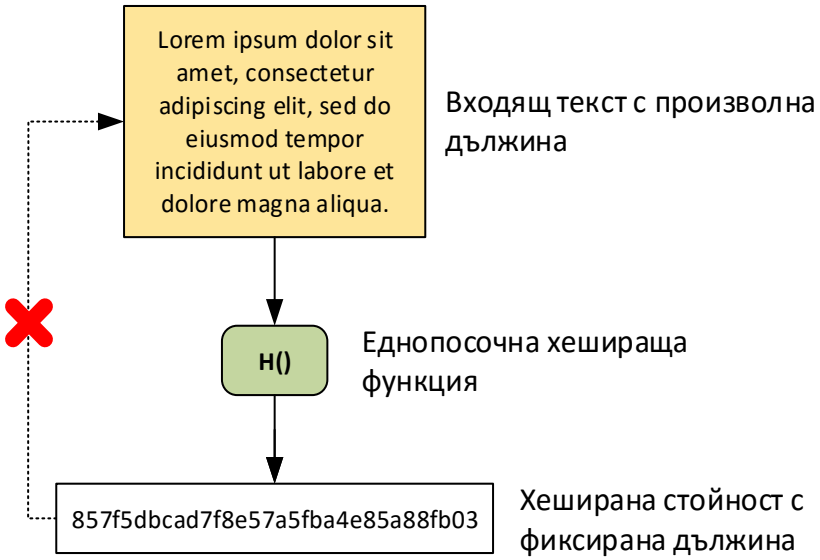
Под терминът “интегритет” се разбира добавяне на стойност или използване на метод, които гарантират, че изпратените съобщения не са били подправени и, че та са получени във вида на тяхното изпращане (в получаващата страна). Тази функционалност наподобява използването на CRC при преноса на рамки и пакети, но за разлика от CRC алгоритъма, при който е възможно да се генерират различни данни с една и съща контролна сума криптографските

методи предоставят по-висока степен на надеждност и свеждат до минимум възможността за подправяне на пренасяната информация.

Автентификацията при мрежовите технологии гарантира, че двете системи, които комуникират са наистина тези, за които се представят. Най-често за автентификация и подсигуряване на интегритета на пренасяните съобщения се използват криптографските функции за хеширане.

Хеширане

Хеширащите функции позволяват на база на входящи данни с произволна големина да се генерира стойност (наречена хеш – “hash”), която е с фиксирана дължина. Разлика от само един единствен бит във входния поток от информация гарантира, че се генерират напълно различни хешове. Хеширащите функции са еднопосочни математически модели, което означава, че от изчислената стойност не може да бъде възстановен открития (входящия) текст. Пресмятането на резултата е бързо, а надеждните алгоритми нямат стойности, които водят то колизии. Под колизия при хеширане се разбира една изключително неприятна ситуация, при която след обработката на два различни входни текста се генерира един и същи хеширан резултат.



Фиг. 11.11 Хеширане на данни

Хеширането може да се приложи за:

- Генериране на информация с цел доказателство на автентичност на информацията;
- Автентификация чрез т.нар. “one-time” и “one-way” отговори на криптографски заявки (challenges);
- Добавяне на данни за проверка на интегритета на съобщенията.

Към момента два от най-често използваните алгоритъма за хеширане са Message Digest 5 (MD5) и Secure Hash Algorithm (SHA).

MD5

Message Digest 5 (MD5) е алгоритъм за хеширане, който като резултат генерира 128 битова стойност, която най-често се представя с 32 шестнадесетични символа. Въпреки, че това е един от най-често използваните алгоритми към момента той постепенно се измества от значително по-надеждния SHA.

MD5 е разработен от Рон Ривест (Ron Rivest) през 1991 година и цел да замени вече остарелия MD4. MD5 е подробно описан в стандарта RFC 1321, който е свободно достъпен.

През 1996 година в MD5 е открита уязвимост, която води до препоръката да се използва SHA, който в последствие също се оказва, че съдържа уязвимости. През 2004 година е доказано, че са налични и колизии. Отново през 2004 година е създаден и метод за генериране на двойка файлове, които имат различно съдържание, но след обработка с MD5 се получават еднакви хеширани стойности (колизия). Това дефинира MD5 като неподходящ за SSL и цифрови сертификати.

SHA

Secure Hash Algorithm (SHA) е фамилия от криптографски функции, разработени и стандартизирани от National Institute of Standards and Technology (NIST)[85]. SHA включва следните разновидности:

- SHA-0 – Първоначалната версия на алгоритъма, която генерира 160 битова хеширана стойност. Почти веднага след нейното публикуване е заменена от SHA-1, поради открит значителен пропуск;
- SHA-1 – Резултатът е 160 битова стойност, като след анализ са открити няколко пропуска и към момента този алгоритъм се счита за не-надежден от криптографска гледна точка;
- SHA-2 – две хеширащи функции, използващи различна големина на блоковете, които са познати като SHA-256 (32 битов блок) и SHA-512 (64 битов блок).
- SHA-3 – алгоритъмът е познат още като “Keccak” и е избран през 2012 сред разработки на специалисти, нямащи отношение към NSA. Дължината на хешираната стойност е аналогична на SHA-2, но логиката на обработка е коренно различна.

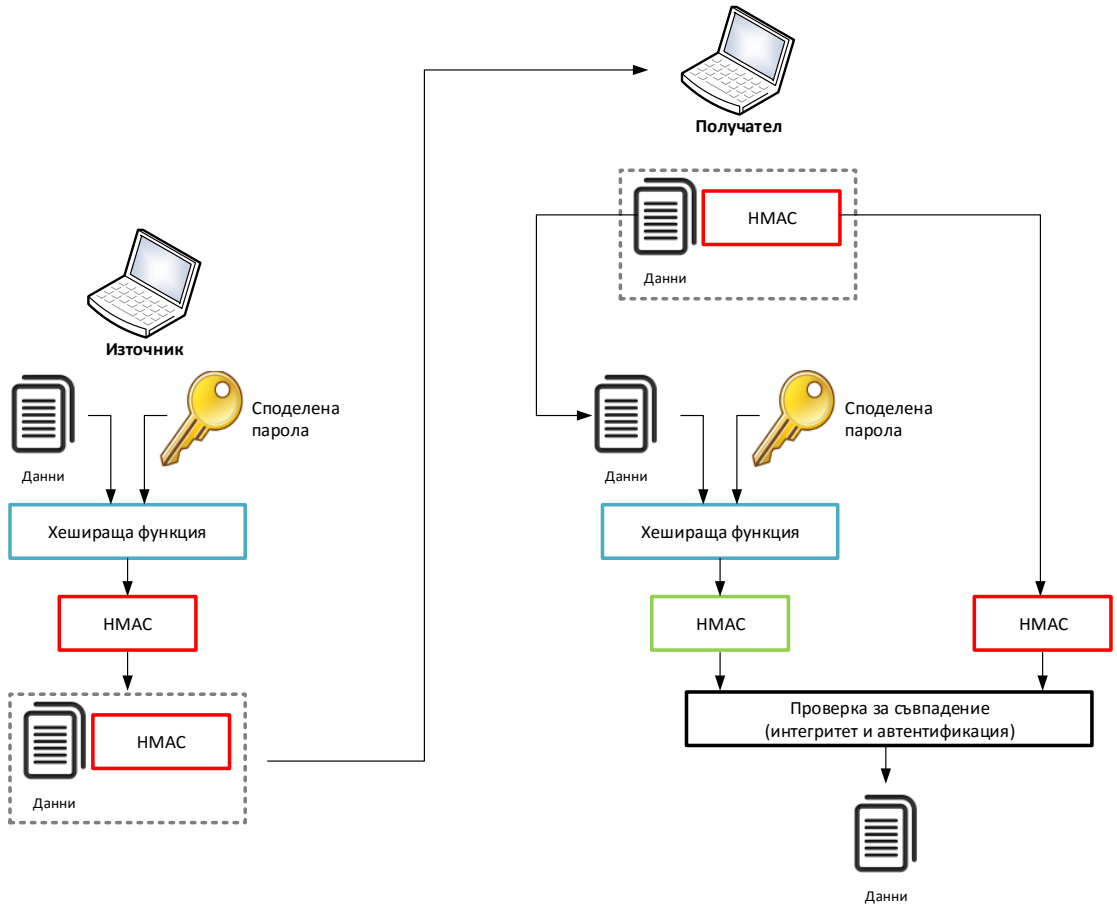
Алгоритъм	Резултат (битове)	Размер на блока (битове)	Максимален размер на входните данни (битове)	Етапи на обработка	Операции	Криптографска устойчивост
MD5	128	128	$2^{64} - 1$	64	And, Xor, Rot, Add (mod 2^{32}), Or	Открити са колизии
SHA-0	160	160	$2^{64} - 1$	80	And, Xor, Rot, Add (mod 2^{32}), Or	Открити са колизии
SHA-1	160	160	$2^{64} - 1$	80	And, Xor, Rot, Add (mod 2^{32}), Or	Надежден
SHA-2 **SHA-224** **SHA-256**	224 256	256	$2^{64} - 1$	64	And, Xor, Rot, Add (mod 2^{32} mod 2^{64}), Or, Shr	Много надежден
SHA-2 **SHA-384** **SHA-512** **SHA-512/224** **SHA-512/256**	384 512 224 256	512	$2^{64} - 1$	80	And, Xor, Rot, Add (mod 2^{64}), Or, Shr	Много надежден
SHA-3 **SHA3-224** **SHA3-256** **SHA3-384** **SHA3-512**	224 256 384 512	1600	∞	80	And, Xor, Rot, Not	Много надежден

[85] www.nist.gov

Автентификация с хеширащи алгоритми

В криптографията методът “Keyed-Hash Message Authentication Code” (HMAC или KHMAC) предоставя възможност за автентификация на база на съобщения (Message Authentication Code – MAC). HMAC изисква да бъдат конфигурирани споделени пароли, съвпадащи на двете системи, които ще комуникират. Тази споделена парола се прибавя към текста, който се хешира и по този начин може да се гарантират както интегритета на данните, така и да се определи, дали устройството от което са изпратени е легитимно (има въведена същата парола).

HMAC е важна функционалност, която е базисна при изграждане на IPsec VPN.



Фиг. 11.12 Описание на HMAC

Управление на ключовете

Една от най-важните криптографски функции е свързана с управлението на ключовете, което включва:

- Генериране на нови ключове;
- Проверка на ключовете;
- Съхранение;
- Деактивиране и унищожаване на ключове;
- Обмяна.

В момента повечето криптографски системи използват автоматизирано генериране на ключове, като се използват или псевдо генератори на случайни числа или в по-редки случай специални хардуерни модули за генериране на истинска случайната стойност.

Проверката на ключовете е важна задача, поради факта, че е възможно криптографския алгоритъм да има слаби ключове (weak key), при които шифрираните данни съвпадат с входящия открит текст.

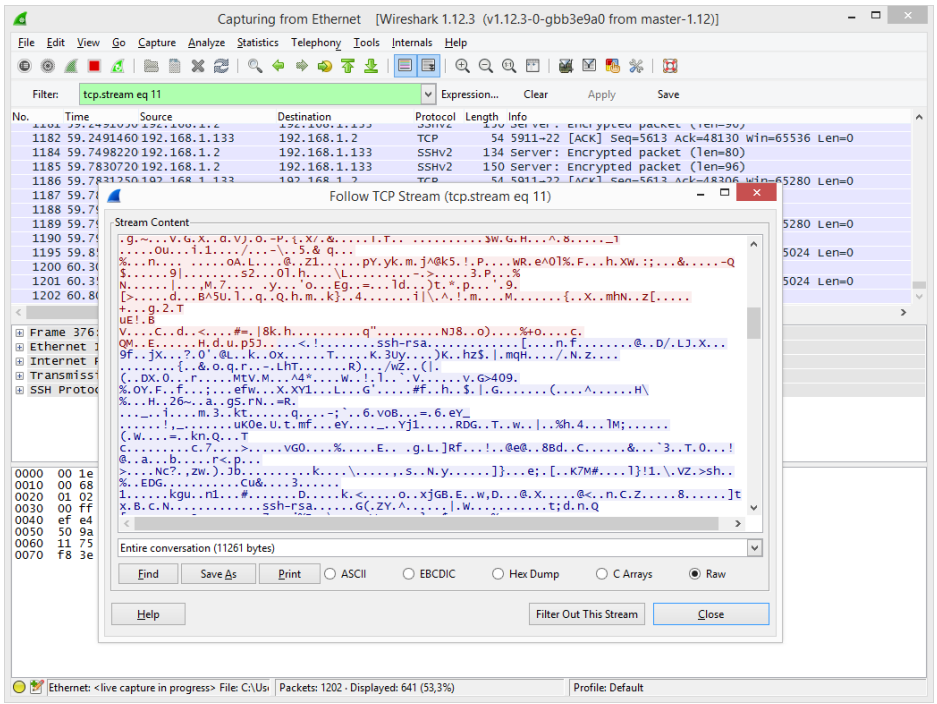
Ако ключовете се съхраняват в споделена памет е възможно дадено приложение да получи неоторизиран достъп до тях. Това е ситуация, която потенциално може да се случи при повечето модерни операционни системи, които поддържат паралелни процеси и едновременно изпълнение на няколко приложения. Също така при наличие на зловреден код е възможно трето неоторизирано лице да си осигури достъп до криптографските ключове. Това поставя специални изисквания за изолирането на ключа в отделен контейнер с ограничен достъп. Също така при съхраняването на секретните ключове при асиметричните алгоритми не трябва да се използват общодостъпни устройства (USB памети и други незащитени носители).

Обмяната на ключовете между две криптографски системи може да е сложна задача, която да изисква допълнително криптирани на потока от данни или да се използват алгоритми, подобни на Diffie-Hellman (DH).

Изтриването на ключ не е свързано само с посочването на обекта от програмния код, а с цел повишаване на сигурността се използва и нулиране на клетките в паметта, където е бил съхранен ключа.

Конфиденциалност

Конфиденциалността при преноса на данни през мрежови технологии е свързана най-вече с предпазване от опасността при прихващане на трафика от трети лица информацията да бъде лесно разчетена. За тази цел се използват криптографски алгоритми за шифроване, които могат да бъдат приложени на различни нива спрямо референтния модел OSI. Разработени от производителя методи и алгоритми за криптиране на трафика на каналното ниво. На мрежовото ниво може да се използват мрежови протоколи с вградени методи за шифроване като IPsec и др. Преносът на данни на транспортното ниво може да бъде защитен чрез технологии като Secure Sockets Layer (SSL) или Transport Layer Security (TLS). Повечето приложения, които използват мрежова комуникация също имат вградени криптографски функции.

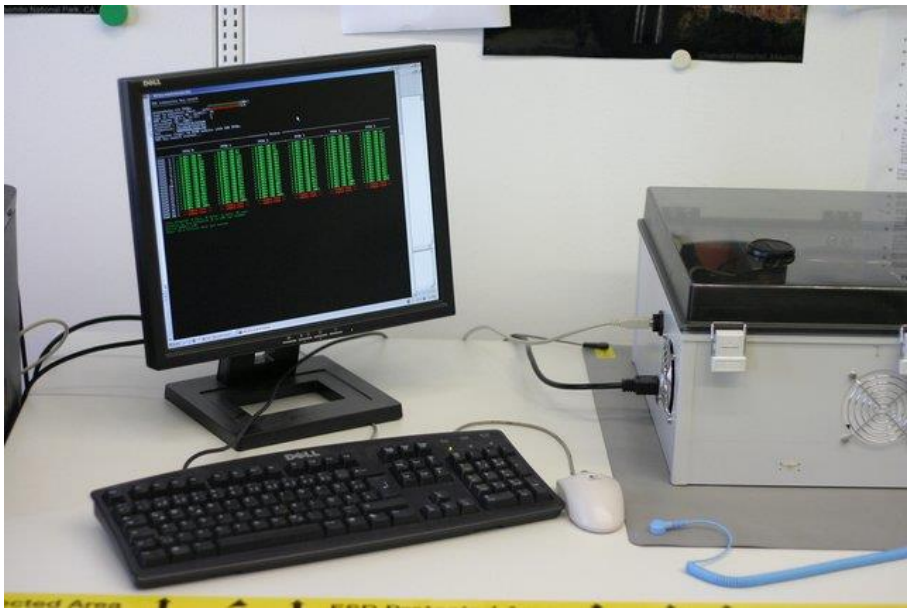


Фиг. 11.13 Визуализиране на прихванати данни от шифрована комуникация (SSH)

DES

Data Encryption Standard (DES) е бил един от най-често използваните алгоритми за шифриране на данни. Разработен в периода 1970 – 1977 година този стандарт се явява един от основополагащите за модерните криптографски системи.

Важно е да се отбележи, че в момента DES се счита като криптографски ненадежден поради малката дължина на ключовете (56 бита), както и разработените специализирани устройства, които успяват да направят напълно успешна атака, насочена към DES само за 22 часа и 15 минути.

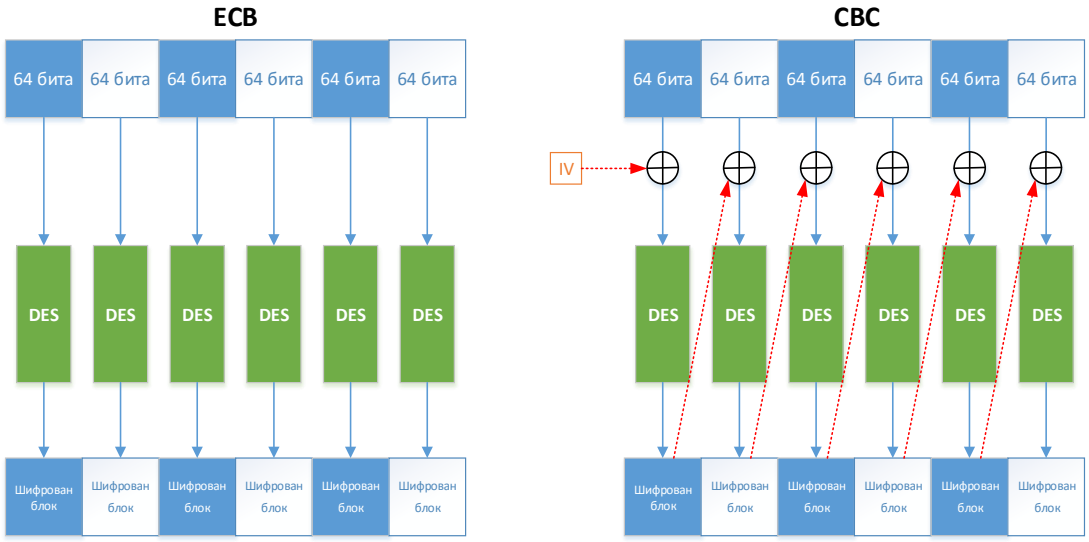


Фиг. 11.14 Специализирано устройство за атака на DES ключове (източник www.copacobana.org)

DES разделя данните на 64 битови блокове, които участват в процеса на шифроване, а използваният алгоритъм се базира на множество етапи на пермутация и заместване на битове, които комбинират данни и посочения ключ. DES е симетричен алгоритъм, което означава, че едни и същи функции се използват за шифриране и дешифриране, както и един и същи ключ. DES ключовете са с фиксирана дължина от 64 бита, като то тях 56 формират ключа, а останалите 8 се използват за проверка по четност (parity). Възможно е да се използват и по-слаби 40 битове ключове.

Въпреки, че DES най-често се използва в режим на блоково шифроване, алгоритъма позволява и поточна работа. При блокова обработка на данни с размер над 64 бита DES може да работи в един от следните два режима:

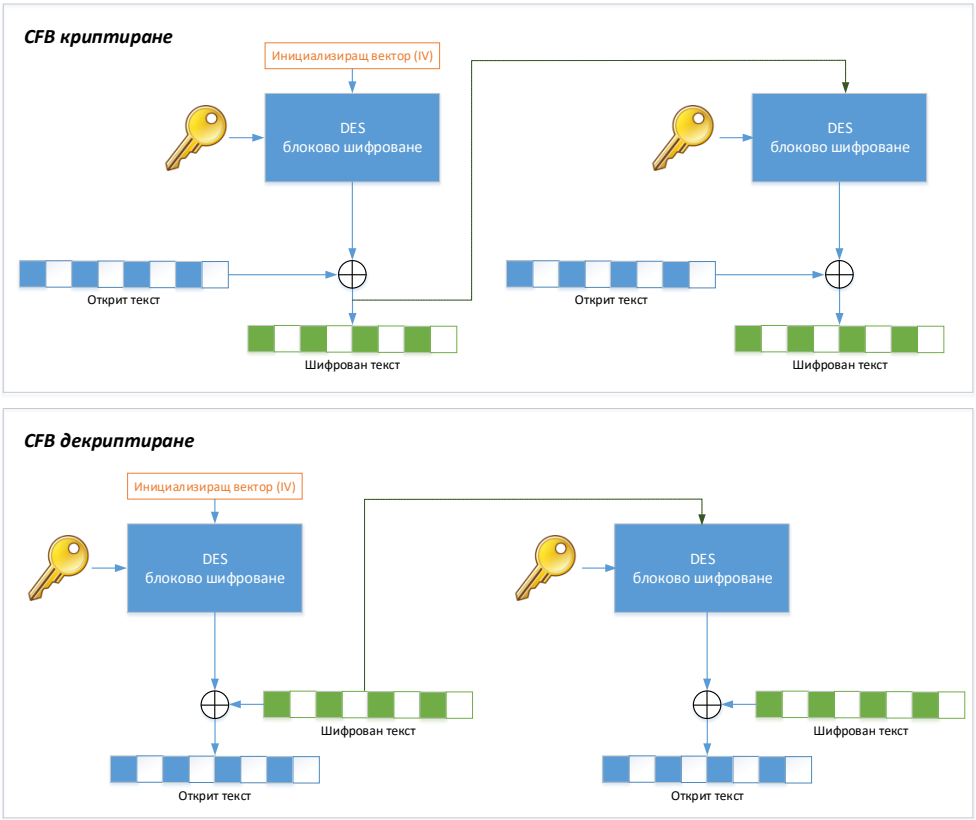
1. Electronic Code Book (ECB) – всички 64 битови блокове се шифрират с 56 битовия ключ. Ако съдържанието на двата блока е идентично се генерират и еднакви криптирани данни;
2. Cipher Block Chaining (CBC) – всеки блок се обработва с допълнителна операция XOR, като се използва предходния (или инициализиращ вектор - IV). По този начин шифроването на всеки отделен блок се базира на предходния, което повишава сигурността на алгоритъма.



Фиг. 11.15 ECB и CBC режим на работа на DES

Ако DES се използва в режим на поточно шифриране, на база на предходния шифрован текст и на ключа се генерира псевдо случайна последователност от битове. Дешифрирането използва същия ключ и само с него може да се възстанови точния открит текст. В този случай режимите на работа са:

1. Cipher feedback (CFB) – аналогичен на CBC, може да шифрова отделени битови или символи;
2. Output feedback (OFB) – генерират се блокове, които в последствие се предават към операция XOR (изключващо ИЛИ) и към блокове от открития текст.



Фиг. 11.16 CFB шифриране и дешифриране при DES

Някои от по-важните особености на DES са:

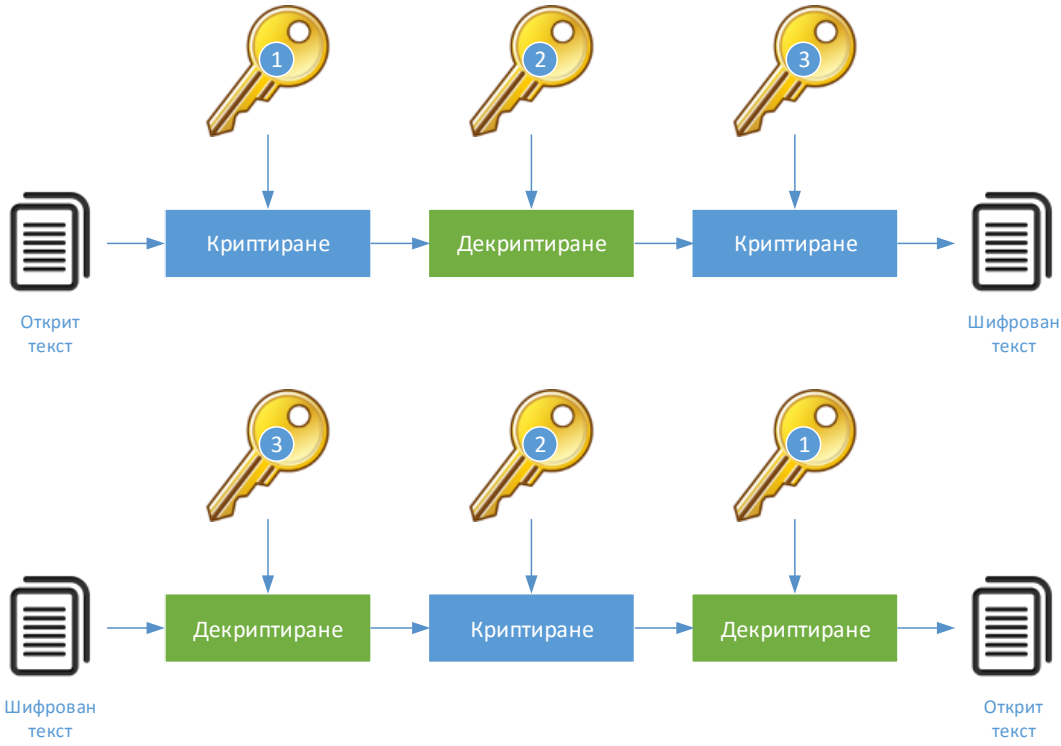
- DES е криптографски ненадежден;
- Необходима е честа смяна на ключовете за превенция на атаки, базирани на метода на грубата сила;
- Препоръчително е да се използва режим на работа CBC;
- Необходимо е проверка за слаби ключове (открити са 4 слаби и 12 полу-слаби);
- Не използвайте DES при възможност за прилагане на по-надеждните 3DES или AES.

3DES

Алгоритъмът 3DES използва три последователни операции с DES, върху отделните блокове на входящите данни, с три различни ключа. За разлика от DES, 3DES към момента се счита за криптографски надежден, което се дължи и на резултатите от над 35 години изследвания и проверки.

За 3DES може да се обобщи:

- Стандартизиране през 1977 година;
- Симетричен криптографски алгоритъм;
- Ключовете могат да са с дължина 112 или 168 бита;
- Има ниска производителност и средно използване на системните ресурси.



Фиг. 11.17 Алгоритъм 3DES

AES

Advanced Encryption Standard (AES) е стандарт, обявен от National Institute of Standards and Technology (NIST) през 2001 година. Той е резултат на инициатива, стартирана през 1997 година, която цели да се разработи и публикува криптографски алгоритъм, който да замени вече несигурния DES. Постъпват 15 предложения, като AES се базира на алгоритъма, предложен от Rijndael.

AES е проектиран от Daemen и Rijmen, като едно от неговите предимства е, че може да използва блокове и ключове с променлив размер. AES работи на принципа на итерации на криптографски операции, извършвани върху блокове, като резултатът се получава след няколко трансформации. AES позволява да се използват ключове с дължина 128, 192 или 256 бита, които да криптират 128, 192 или 256 битови блокове, като е възможна всяка една от деветте комбинации.

От гледна точка на производителност AES е бърз, като използваните системни ресурси са сравнително малко. Задълбочените криптографски анализи сочат, че AES е достатъчно надежден и може да се използва като заместващ алгоритъм на 3DES, дори при шифроване на данни с висока степен на конфиденциалност.

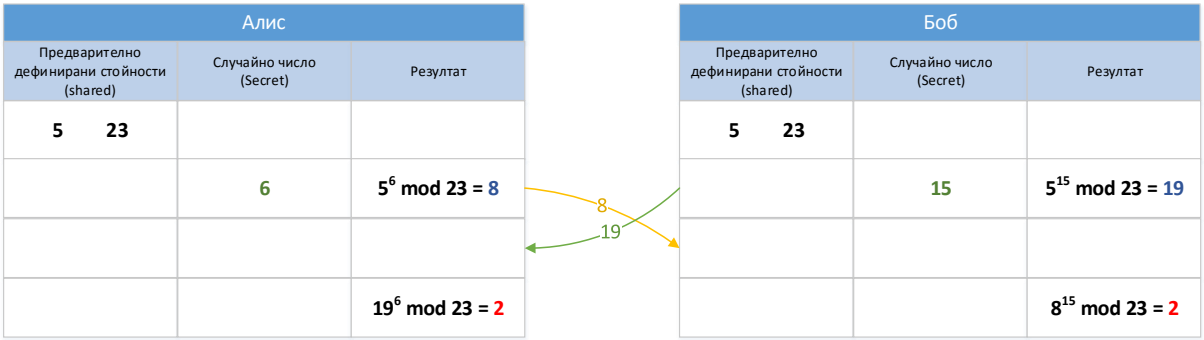
Алгоритъм Diffie-Hellman

Алгоритъмът на Дифи и Хелман (Diffie-Hellman) е асиметричен криптографски алгоритъм, който позволява генериране на надеждни ключове на две системи през несигурен комуникационен канал. Идеята е реализиране през 1976 година и към момента се прилага като база за много криптографски системи, които имат необходимост от автоматично генериране на ключове.

Алгоритъмът не се прилага за криптиране на данни, а пример за принципа на неговата работа е следния:

1. Алис и Боб се уговарят да използват числото p = 23 за делене с остатък и числото q = 5 за основа
2. След това Алис избира едно целочислено число a = 6 и изпраща на Боб ключ A = q^a(mod p)
   A = 5^6 mod 23 = 8
3. Боб също избира целочислено число b=15 и изпраща на Алис ключ B = q^b(mod p)
   B = 5^15 mod 23 = 19
4. Алис изчислява общият ключ s = B^a(mod p)
   19^6 mod 23 = 2
5. Боб изчислява общият ключ s = A^b(mod p)
   8^15 mod 23 = 2

След приключване на пресмятането и двете страни получават еднаква стойност, която не може да се изчисли, ако трета страна прихване обменяната информация.



Фиг. 11.18 Пример за изчисляване на споделен ключ чрез DH

Криптография с публичен ключ

Асиметричните криптографски алгоритми използват двойка ключове – един за шифриране и втори за дешифриране на данните. Този тип алгоритми се прилагат при мрежови технологии като:

- Internet Key Exchange (IKE);
- Secure Socket Layer (SSL);
- Secure Shell (SSH);
- GNU Privacy Guard (GNUPG);
- Pretty Good Privacy (PGP).

При асиметричните криптографски алгоритми ако дължината на ключ е по-малка от 1024 (типични дължини са 2048, 4096), ключа може да се счита за недостатъчно надежден.

Криптографските методи с публичен ключ могат да се използват както за шифроване на данните с цел конфиденциалност на пренасяната информация, така и за автентификация.

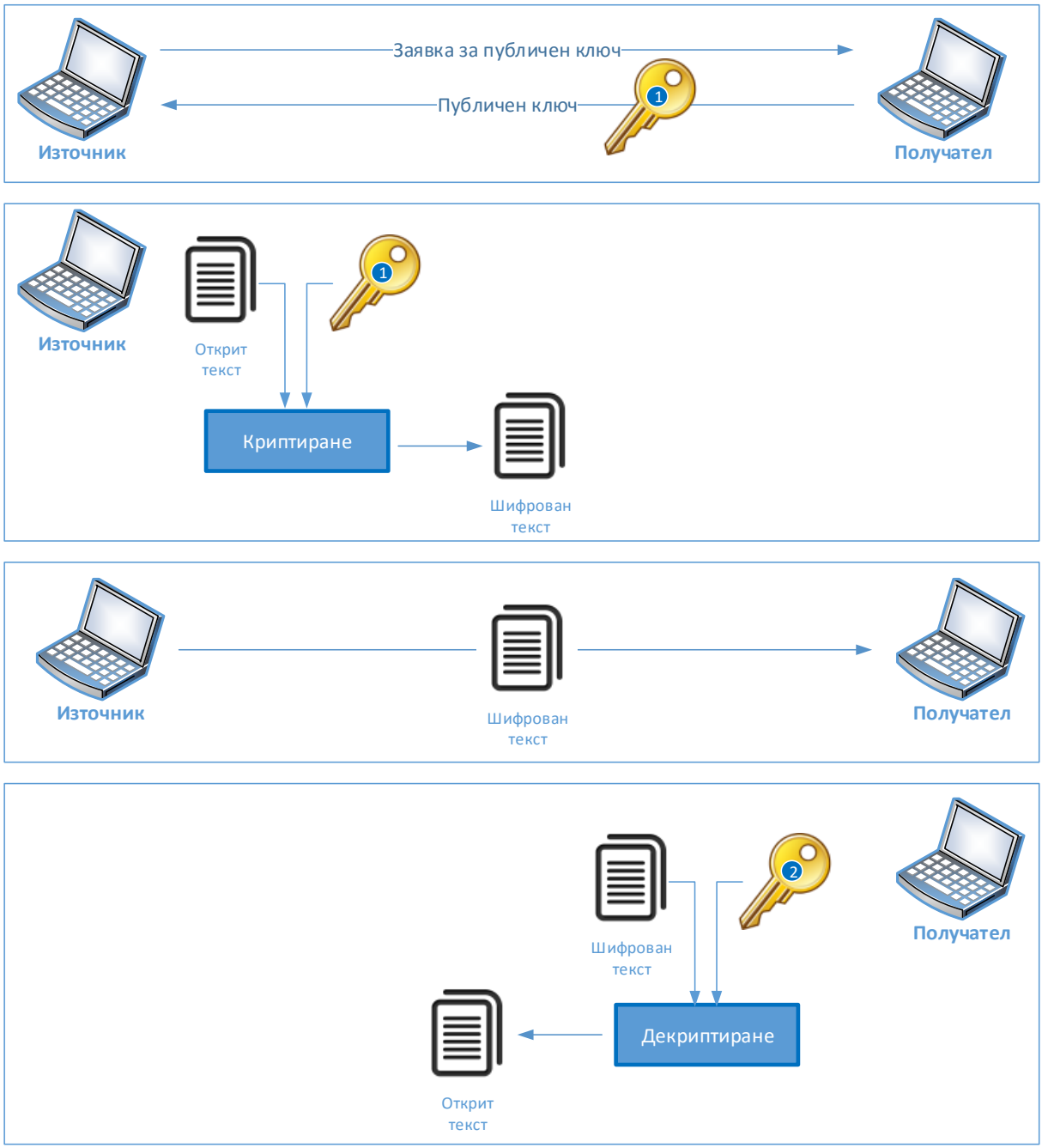
Криптирането на мрежовия трафик между две системи чрез асиметричен криптографски алгоритъм се свежда до следните основни етапи:

1. Източника изпраща заявка към получателя, с която изисква публичния ключ на отдалечената система;
2. Получателя изпраща своя публичен ключ на заявителя;
3. Източника шифрира данните с публичния ключ и ги предава в кодиран вид на получателя;
4. След като се получат кодираните данни получателя може да ги дешифрира със своя секретен ключ.

Комуникацията е с висока степен на надеждност (при използване на ключове с дължина над 1024 бита) и дори трафика да бъде прихванат от трети лица кодираната информация не може да бъде декриптирана само със секретния ключ от дадената двойка.

В този случай публичният ключ се използва за шифроване на информацията, а секретния – за декодирането и.

Шифроване на данни с асиметричен алгоритъм при мрежова комуникация

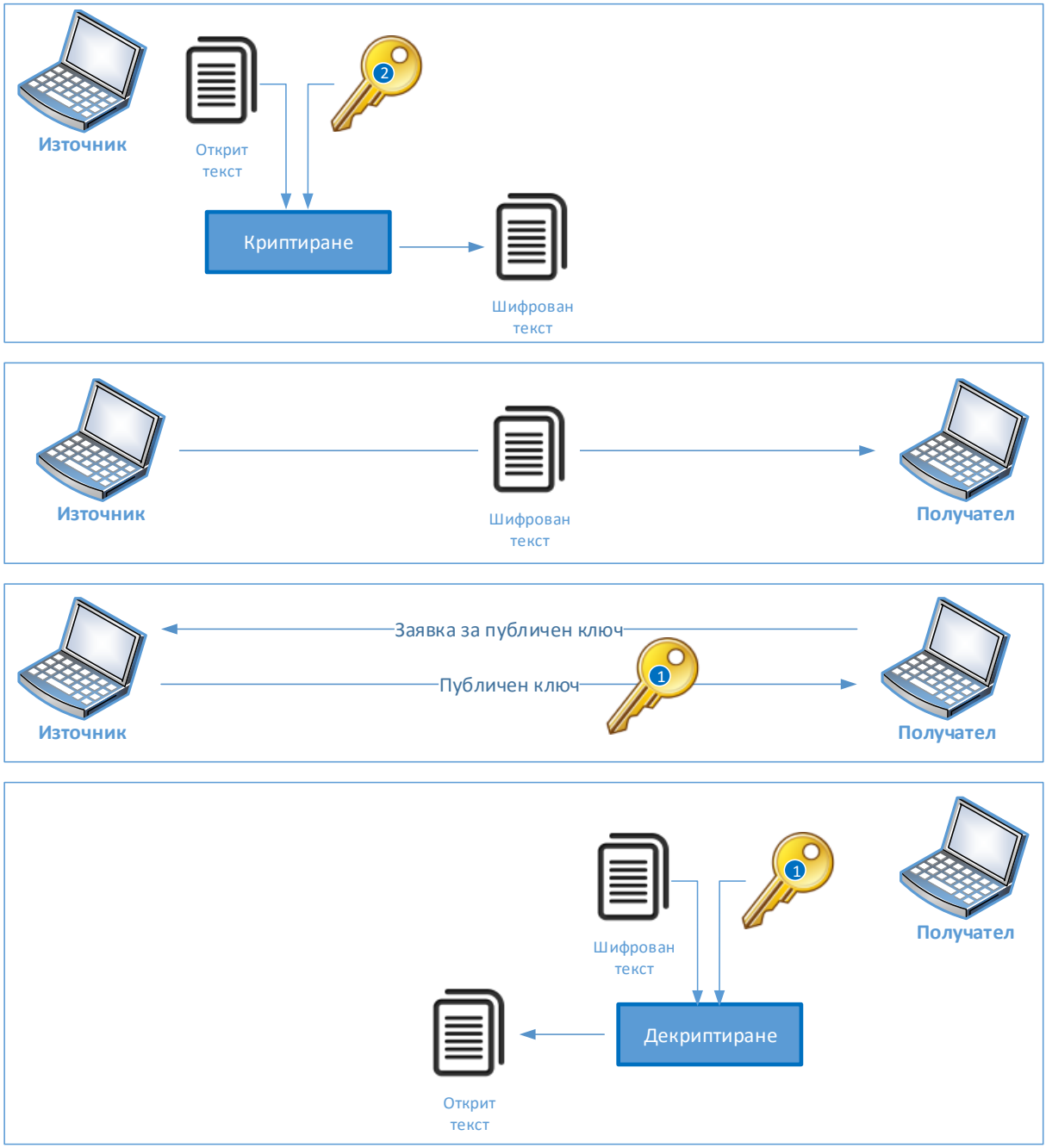


Фиг. 11.19 Приложение на асиметрични криптографски алгоритми за шифриране и дешифриране на данни

За да се извърши автентификация на съседна система чрез асиметричен криптографски алгоритъм се използват следните стъпки:

1. Източникът шифрова съобщение със своя секретен ключ и го изпраща на получателя;
2. Получателя изисква от източника публичен ключ;
3. След получаване на ключа (публичен) данните могат да се декриптират и сравнят с други с цел потвърждаване на изпращача.

Автентификация с асиметричен алгоритъм при мрежова комуникация



Фиг. 11.20 Приложение на асиметрични криптографски алгоритми за автентификация

Някой от най-често използваните асиметрични алгоритми са:

- DH – прилага се за генериране на споделени ключове през несигурен комуникационен канал. Сигурността се базира на предположението, че повдигането на число на степен е лесна операция, но откриването на стойността на степента при възстановяването на изходната стойност – не. Дължината на ключовете е 1024 бита, 2048 бита и др.;
- DSS/DSA – Стандарт, разработен от NIST, DSA е алгоритъм с публичен ключ, който като производителност при шифроване е близък до RSA, но декриптирането е приблизително 40 пъти по-бавно. Използваните ключове са с дължини от 512 бита до 1024 бита;

- RSA – Разработка на Ривест, Шамир и Адлеман. Един от най-често използваните асиметрични алгоритми, които се базират на големи числа. По същество това е първият алгоритъм, който може да се използва както за шифроване, така и за подписване (интегритет). Дължината на ключовете е от 512 бита до 4096 бита;
- ElGamal – Алгоритъмът се базира на DH и е разработен от Таер Елгамал през 1984 година. Използва се от GNU Privacy Guard, PGP и други криптографски системи. Един от основните недостатъци, е че криптираното съобщение е с голям размер, като най-често е приблизително 2 пъти обема на открития текст. Поради тази причина този алгоритъм се използва най-вече при данни с малък обем. Дължината на ключовете е 512 бита до 1024 бита;
- ECC – Elliptic Curve Cryptography е открита от Нил Коблиц и Виктор Милър в средата на 80^{те} години на миналия век. Този метод може да се използва с различни криптографски алгоритми, сред които DH, ElGamal и др. Основното предимство е, че размерът на надеждните ключове е малък, особено при сравнение с този при асиметричните алгоритми.

Цифрови сертификати

Един от най-често използваните методи за доказване на съгласие, собственост и др. е нашият собственоръчно поставен подпис върху документите, който е уникален за всеки човек и в общия случай трудно може да бъде преправен или точно копиран. Цифровите сертификати (Digital Certificate) могат да предоставят подобна функционалност, свързани с цифровите документи и съобщения, която може да се обобщи като:

1. Автентификация – след прилагането на цифровите сертификати може да се провери източника на съобщението;
2. Интегритет – използването на цифров подпис (сертификат) гарантира, че данните не са били модифицирани след тяхното изпращане;
3. Доказване на изпращането – ако данни, които са подписани с цифров сертификат се предоставят на трето лице, източникът не може да отрече, че той е изпращача.

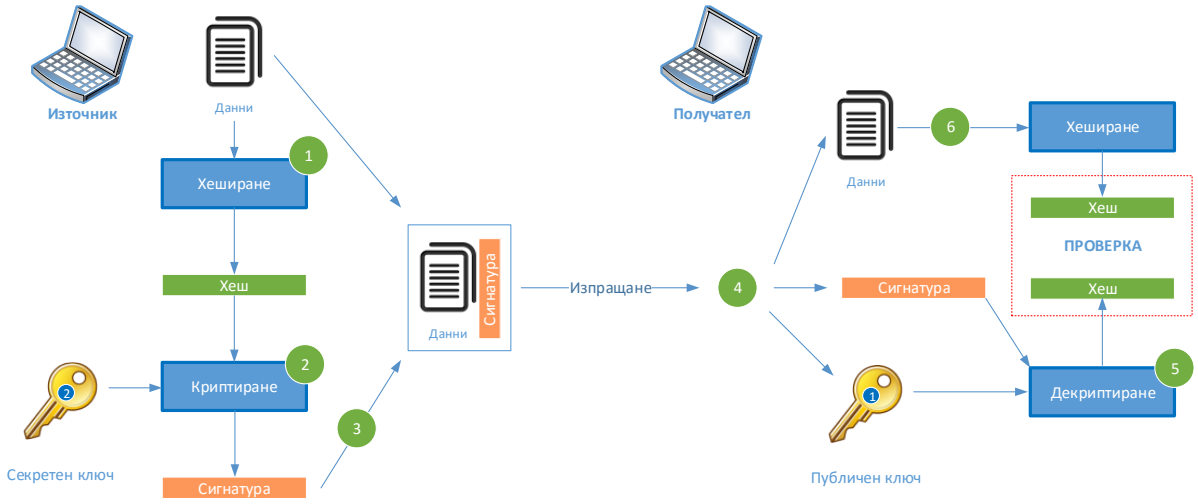
Цифровите сертификати имат някои специфични характеристики, които спомагат за тяхното приложение за автентификация:

1. Сигнатурата не може да се използва за нов подпис след като е генерирана;
2. Сигнатурата е автентична и не може да се фалшифицира;
3. Сигнатурата не може да бъде преправена;
4. От правна гледна точка сигнатурата има тежест на нормален подпис.
   При мрежовата комуникация цифровите сертификати се използват при:
   - IPsec VPN;
   - SSL;
   - VoIP и др.;

   Използването на цифрови сертификати може да се опише със следите стъпки:

1. Изпращащото (подписващото) устройство изчислява хеш, на базата на документа;
2. Стойността на хеша се шифрира със секретния ключ на изпращача;
3. Криптираният хеш, наречен още сигнатура се добавя към документа;
4. След получаване на данните отдалечената система изисква от изпращача неговия публичен ключ;
5. Стойността на хеша се дешифрира;

6. Изчислява се хеш стойност за документа. Полученият резултат се сравнява с изпратения хеш. Ако двете стойности съвпадат се гарантират интегритета на съобщението, а също така и кой е изпращача.



Фиг. 11.21 Приложение на електронен подпис

Алгоритъм DSA

Алгоритъмът Digital Signature Algorithm (DSA) е одобрен от FIPS за приложения, свързани с цифрови подписи. Разработката е предложена от NIST през 1991 година, а през 1994 година DSA е избран за Digital Signature Standard (DSS).

DSA се базира на дискретни логаритми и може да се използва единствено за подписване на данни, което е и една от неговите най-обсъждани слаби страни. Други недостатъци са, че проверката на сигнатурите е бавна, а NIST избират DSA по непрозрачен за обществеността начин. За да се коригират тези проблеми към момента DSS предоставя възможност за използване на RSA и ECC.

PKI

Терминът Public Key Infrastructure (PKI) на български се среща като „инфраструктура на публичния ключ“, „инфраструктура с публичен ключ“, „публична ключова идентификация“. По същество това е технология за проверка на автентичността на електронен документ с помощта на публичен ключ. PKI е съвкупността от хардуер, софтуер, физически и юридически лица, политики и процедури, необходими за издаването, управлението, разпределението, използването, съхранението и отнемането на цифрови сертификати.

В криптографията PKI е споразумението, което свързва определен публичен ключ с идентичността на неговия собственик (титуляр) с помощта на сертифициращ орган (Certificate Authority или CA). Еднозначността на свързването се гарантира от сертифициращия орган чрез строго установен процес на регистрация и издаване на цифровия сертификат, които е описан в т.нар. политики за предоставяне на удостоверителни услуги. Органът, осигуряващ тази еднозначна свързаност, се нарича регистриращ орган (Registration Authority или RA) и представлява звено на сертифициращия орган (това може и да е упълномощена външна организация), извършващо дейностите по приемане, проверка, одобряване или отхвърляне на исканията за издаване на сертификати.

Друг участник в PKI е проверяващият орган (Vertification Authority или VA). В издадения от сертифициращия орган сертификат с публичен ключ са кодирани редица атрибути като

идентичност на титуляря, самият публичен ключ, тяхната връзка, условията за валидност и др. по начин, който гарантира че не могат да бъдат фалшифицирани.

Фиг. 11.22 Инфраструктура с публичен ключ (PKI)

PKI се използва за автентикация на участника в транзакцията — дали той е този, за който се представя. Това е от особено значение за комуникацията през Интернет, тъй като там липсва стандартен механизъм за проверка на идентичността на участниците. Чрез използването на PKI е възможно въвеждането на концепция за признаване на Интернет-базирани транзакции.

Изводи

Криптографията намира широко приложение в информационните и комуникационни технологии. Чрез криптографските алгоритми може да се извърши автентификация, шифриране на данни, изпращани през несигурен канал, както и да се гарантира интегритета на съобщението.

Криптографските алгоритми са два основни типа – симетрични и асиметрични, като и при двата сигурността се гарантира от използваните ключове. Асиметричните алгоритми изискват по-дълги ключове в сравнение със симетричните. Изборът на алгоритъм е от съществено значение за производителността и надеждността на комуникацията.

Източници

1. http://lubbopitko.blogspot.com/2012/03/blog-post_21.html
2. http://bg.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F
3. http://bg.wikipedia.org/wiki/%D0%95%D0%BD%D0%B8%D0%B3%D0%BC%D0%B0_(%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0)
4. http://golubev.com/files/ighashgpu/readme.htm

Глава 12. Конфигуриране на виртуални частни мрежи (VPN) с EFW CE

Виртуални частни мрежи

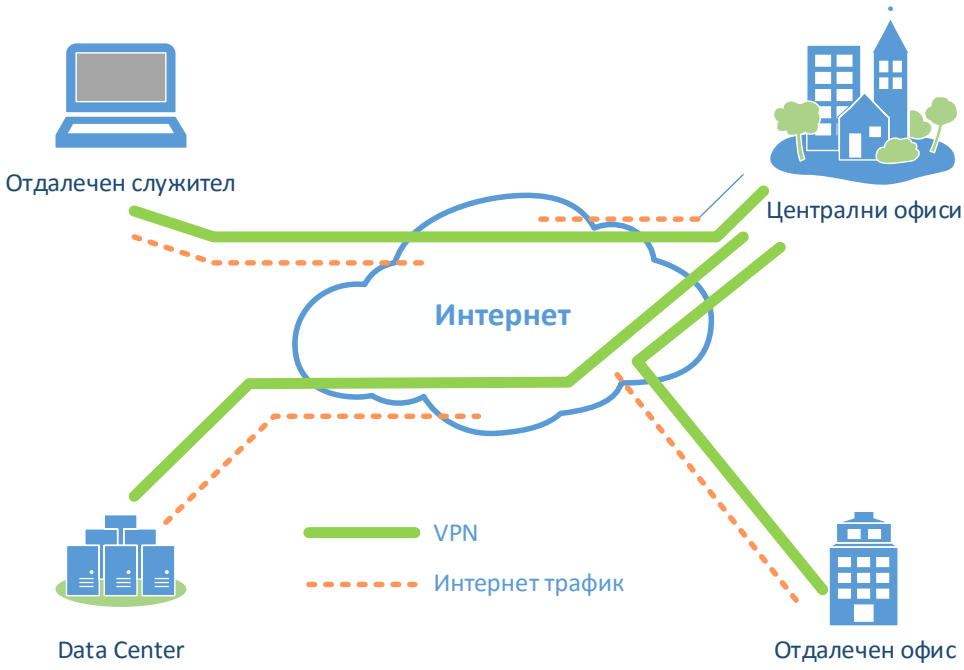
Много често се налага да се пренасят данни, включително с конфиденциален характер, през несигурни информационни канали като Интернет. Типичен пример е малка фирма, която има офиси в няколко града и за която заплащането на таксата за наета линия към телеком би надхвърлило съответните бюджети. В тази ситуация едно от най-добрите решения изграждане на свързаност чрез осигурения достъп до Интернет във всеки един офис, но сериозен недостатък на този подход е ниската степен на сигурност на пренасяните данни.

Технологията за изграждане на виртуални честни мрежи (VPN – Virtual Private Networks) предоставя възможност за създаване на криптиран тунел между две системи, като данните се пренасят през потенциално несигурни мрежи. Ако се върнем на предходния пример конфигурирането на VPN ще бъде гъвкаво и сигурно решение за свързване на отдалечените офиси на разглежданата фирма, като недостатък ще бъдат редуцираната скорост на трансфер на данни през тунела и необходимостта от допълнителни конфигурации.

В исторически аспект първите VPN протоколи единствено са изграждали тунели, но не са шифрирали пакетите, което към момента е едно основно изискване към тази технология.

Терминът **“виртуална” (virtual)** идва от факта, че се постига аналог на наета линия между използваните системи, но реално данните се пренасят през публични мрежи, а **“частна” (private)** - от криптирането на пренасяните пакети.

Важно е да се отбележи, че при VPN се поставя стриктното изискване комуникиращите системи да принадлежат към единна група (например организация, фирма и др.).



Фиг. 12.1 Обобщен пример за VPN тунел

Някои от по-съществените предимства на VPN технологията са:

- **Редуциране на разходите** – в много случаи изграждането и поддържането на един или няколко VPN тунела е в пъти по-евтино от реализирането на същата топология с наета линия или друга WAN технология;
- **Повишаване на сигурността** – използването на криптографски надеждни алгоритми за HMAC и шифриране на данните позволяват да се постигне изключително висока степен на защита на пренасяната информация от неоторизиран достъп и от промяна на нейното съдържание;
- **Гъвкавост** – промяна на конфигурацията на крайните точки, както и добавянето и премахването на VPN тунели рядко изискват промяна на физическата топология;
- **Съвместимост** – почти всяка модерна операционна система поддържа VPN.

Най-общо под VPN тунел се разбира свързване на две отдалечени системи чрез специализиран протокол, който изгражда тунел през несигурна или публична комуникационна инфраструктура. Логически VPN може да бъде конфигуриран с технологии, работещи на каналното или на мрежовото ниво на OSI референтния модел, като при повечето протоколи се налага добавяне на допълнителен хедър към пренасяните пакети или рамки.

Някои от най-често използваните VPN технологии, които работят на мрежовото ниво са:

- Generic Router Encapsulation (GRE) – протокол, разработен от Cisco Systems® и описан в RFC 1701, който по същество позволява пренос на пакетите през тунел, но не и криптиране на трафика. Силна страна е възможността за пренос на всякакъв вид пакети, включително и мултикаст трафик;
- MPLS - Multiprotocol Label Switching (MPLS) е специален телекомуникационен протокол (RFC 3031) за комутиране на пакети на база на къси етикети, избягвайки сравнително по-дългите адреси и по този начин увеличавайки производителността на мрежовите устройства;
- IPsec – IP Security е набор от протоколи (RFC 2401-2412), които не са обвързани с определена технология за криптиране и хеширане и които позволяват да се изградят тунели с криптиран трафик (или само с автентификация и проверка на интегритета). Към момента това е една от най-често използваните VPN технологии.

IPsec

IPsec е една от най-често прилаганите технологии за изграждане на VPN и това е само една от многото причини всеки мрежови специалист да се запознае с нея.

Технология

IPsec е стандартизиран от IETF в RFC 2401 до RFC 2412, като този голям брой стандарти биха довели до извода, че IPsec е сложен протокол, но след като се вникне в неговия замисъл и начин на конфигуриране изграждането и поддръжката на VPN тунели с него са лесни.

Важно е да се наблегне на факта, че IPsec не е обвързан с точно определени алгоритми за криптиране, хеширане и управление на ключовете, както и с методите за автентификация. Това прави IPsec платформата много гъвкава и приложима в редица случаи.

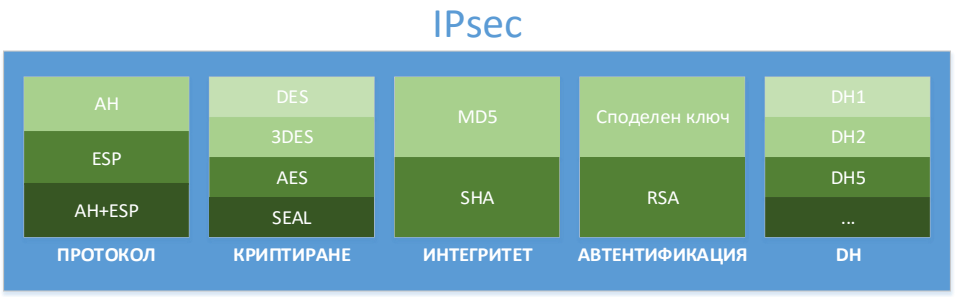
От гледна точка на OSI референтния модел IPsec работи на мрежовото (3) ниво, като този тип виртуални частни мрежи позволяват да се извърши автентификация, а пакетите да се пренасят шифровани с гарантирана липса на промени по време на тяхното транспортиране между комуникиращите системи. Теоретично IPsec защитава трафика от мрежовото до

приложното ниво, като един от недостатъците е, че пакетите могат да са единствено IP и не се поддържат мултикаст и броадкаст заявки.

Компоненти

IPsec има 5 основни компонента:

1. **Протокол** – възможностите са да се използва Encapsulation Security Payload (ESP) или Authentication Header (AH);
2. **Алгоритъм за осигуряване на конфиденциалност** – най-често поддържаните криптографски алгоритми са DES, 3DES, AES и SEAL;
3. **Проверка на интегритета** – избор на хеширащ алгоритъм, като най-често е възможно да се посочи MD5 или SHA;
4. **Генериране на ключовете** – използване на цифрови сертификати или RSA;
5. **DH алгоритъм**.



Фиг. 12.2 Основни компоненти на IPsec

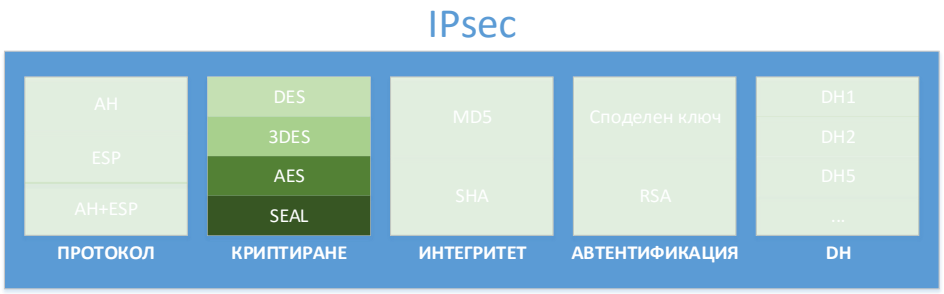
Конфигурирането на всеки отделен тунел изисква избор на съответните параметри, като двете участващи системи трябва да имат еднотипна конфигурация, например:

- Протокол – ESP;
- Криптиране – AES;
- Интегритет – SHA;
- Автентификация - споделени ключове;
- DH – група 7 (DH7).

Конфиденциалността на данните се гарантира от протокола ESP и съответния криптографски алгоритъм за шифриране. Повечето IPsec системи предоставят възможност да се посочи един от следните методи:

- DES – симетричен алгоритъм, използващ 56 битови ключове, който е с ниска криптографска надеждност и не се препоръчва да се използва в системи, които имат стриктни изисквания към сигурността;
- 3DES – алгоритъм, базиран на DES, но използващ три последователни операции с DES и три различни ключа. В сравнение с DES има значително по-висока степен на сигурност;
- AES – симетричен алгоритъм по-сигурен от DES и по-бърз от 3DES;
- SEAL – поточен симетричен криптографски алгоритъм.

Изборът на алгоритъм зависи най-вече от подържаните методи и от необходимата степен на защита на потока от данни.

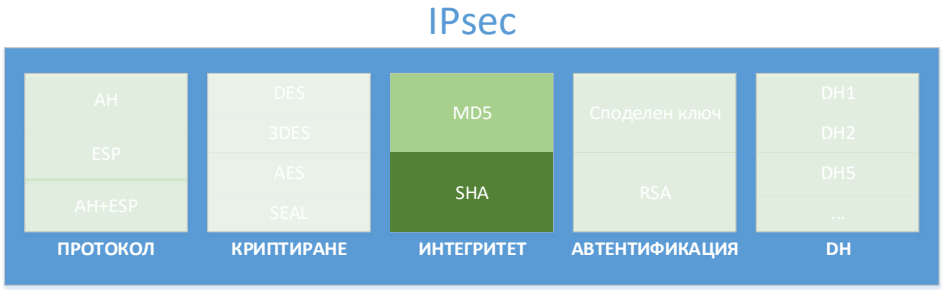


Фиг. 12.3 Конфиденциалност при IPsec

IPsec използва HMAC (Hashed Message Authentication Code) за да се гарантира интегритета на пренасяните данни. Най-често поддържаните HMAC алгоритми са:

- HMAC-MD5 – използва 128 битов ключ и хеширане чрез MD5, като резултата е 128 битова стойност;
- HMAC-SHA256 – на базата на 256 ключ и SHA хеширане се изчисляват 160 бита хеш резултат.

От криптографска гледна точка в сравнение с MD5, SHA е с по-висока степен на надеждност.

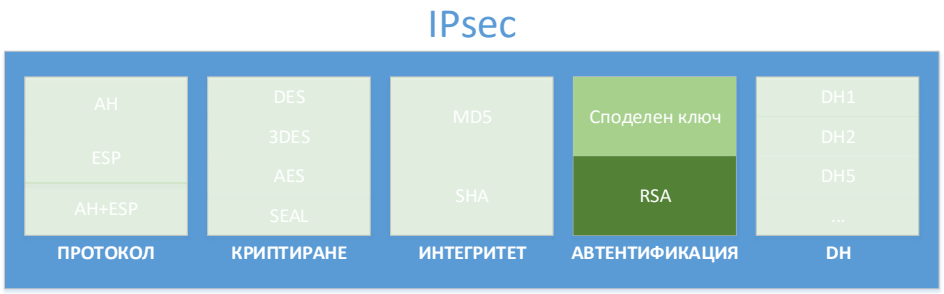


Фиг. 12.4 HMAC при IPsec

Един от най-важните етапи при изграждането на IPsec тунел е автентификацията на комуникиращите системи. По този начин се гарантира, че изграденият надежден тунел ще бъде между две легитимни устройства и данните няма да бъдат пренасочвани към трети лица.

Двата варианта за автентификация при IPsec са:

- Споделени ключове (PSK) – двете системи имат предварително конфигурирани от администраторите “пароли” за изграждане на тунела. Споделената стойност се комбинира с други данни с цел еднозначна автентификация на отдалечената страна. Този метод е сравнително лесен за конфигуриране, но поставя изисквания към начина на определянето и споделянето на PSK, както и води до липса на гъвкавост при голям брой тунели;
- RSA – използването на асиметрични криптографски алгоритми и цифрови сертификати значително подобрява сигурността и точността на автентификацията, но е по-трудно за конфигуриране и изисква допълнителни проверки (CA).

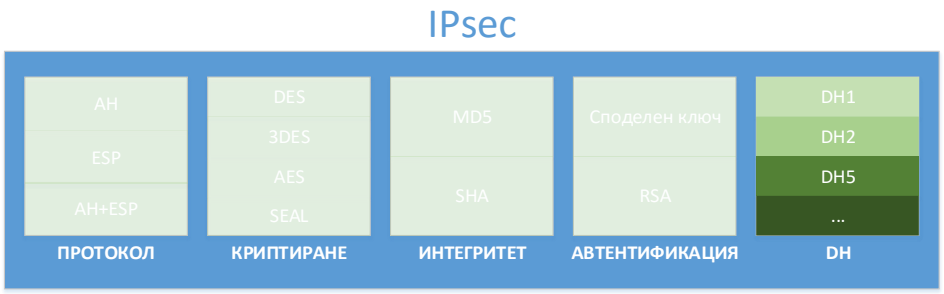


Фиг. 12.5 Автентификация на страните при IPsec

Приложението на симетричните алгоритми DES, 3DES, AES, както и на HMAC при IPsec изисква споделен ключ. Вместо тази стойност да бъде конфигурирана от администраторите (някои системи позволяват това) се използва DH протокола за надеждно генериране на криптографски ключове през несигурен канал за обмяна на пакети.

DH групите при IPsec най-често са:

- DH групи 1, 2 и 5 – използват се за създаване на ключове с размери 768, 1024 и 1536 бита, като не се препоръчва те да бъдат използвани;
- DH групи 14, 15 и 16 – генерираните ключове са с дължини от 2048, 3072 и 4096 бита;
- DH групи 19, 20 и 24 – използват ECC и поради спецификата на тези алгоритми се генерират надеждни ключове с малка дължина, респективно 256, 384 и 2048 бита. Препоръчва се използването на DH24, ако системите позволяват.



Фиг. 12.6 DH при IPsec

Както вече беше споменато двата основни IPsec протокола са Authentication Header (AH) и Encapsulation Security Payload (ESP).

AH предоставя възможност за:

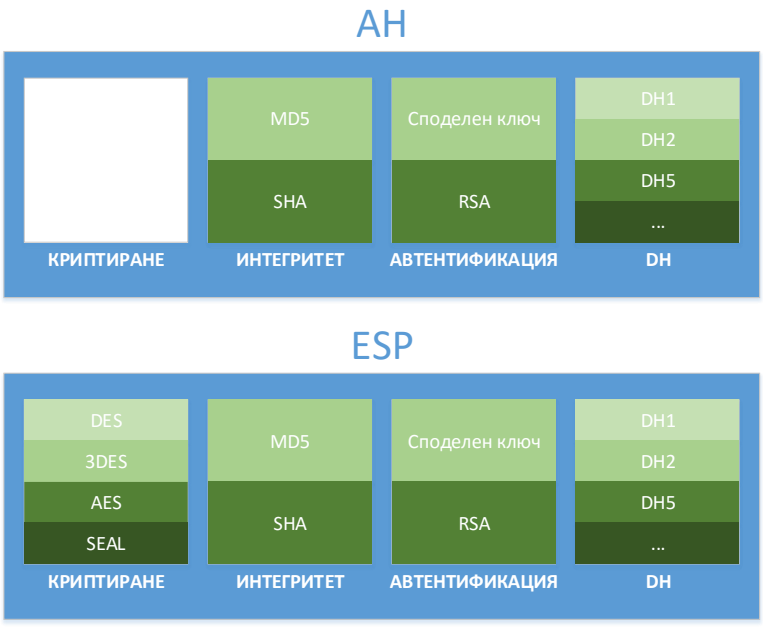
- Автентификация на страните;
- Интегритет на данните.

ESP се използва за:

- Криптиране на потока от информация;
- Автентификация на страните;
- Интегритет на данните.

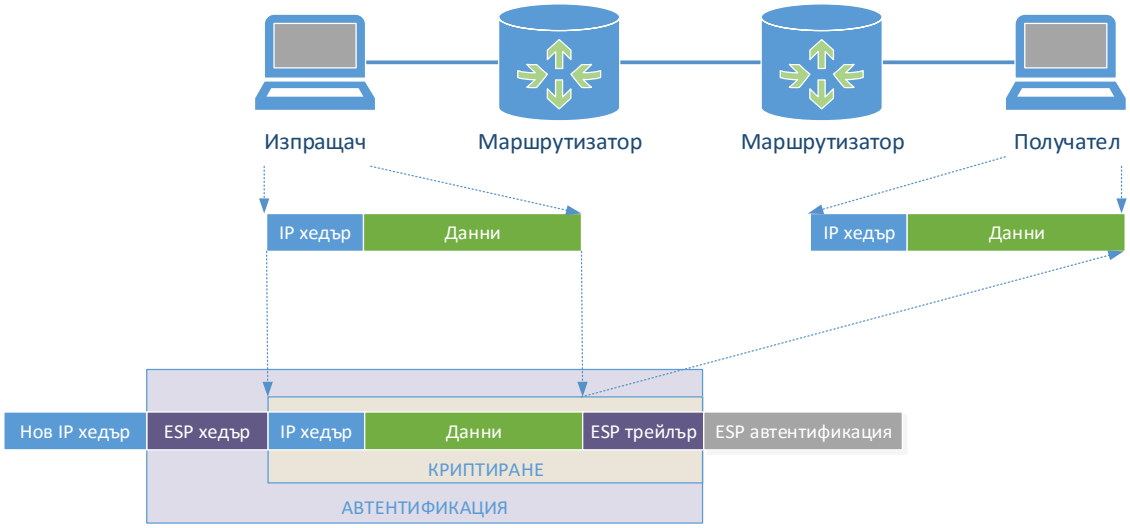
AH е IP протокол с номер 51 и за разлика от ESP (IP протокол 50) не може да шифрира данните. Ако IPsec VPN използва единствено AH нивото на сигурност на комуникацията като цяло е ниско. При ESP първоначално се криптира пренасяния пакет, след което получения резултат се

използва като входни данни за HMAC алгоритъм. AH и ESP поддържат специална функционалност предпазваща данните в тунела от тяхното повторно изпращане от трета страна – т.нар. “anti-replay” техника.



Фиг. 12.7 Сравнение на параметрите при AH и ESP

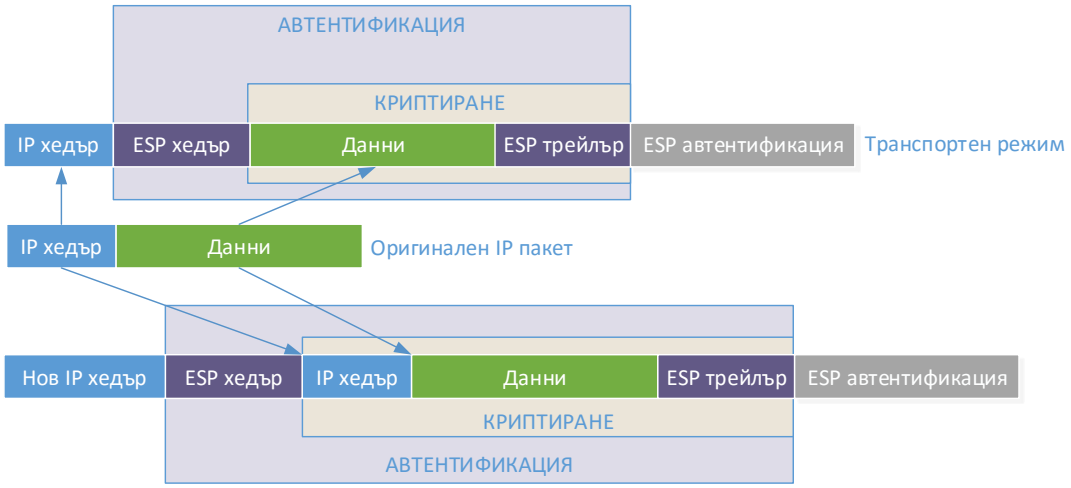
След криптирането на оригиналния IP пакет ESP генерира нов IP хедър, който се използва при маршрутизирането на данните.



Фиг. 12.8 Енкапсулиране на данните при ESP

Тунелите при IPsec могат да бъдат конфигурирани в два режима (показани на фиг. 12.9):

1. **Транспортен** – запазва се оригиналния IP хедър, но се криптира пренасяната информация;
2. **Тунелен** – целият оригинален пакет се шифрова, което изисква добавянето на нов IP хедър.



Фиг. 12.9 Транспортен и тунелен режим при IPsec

IKE

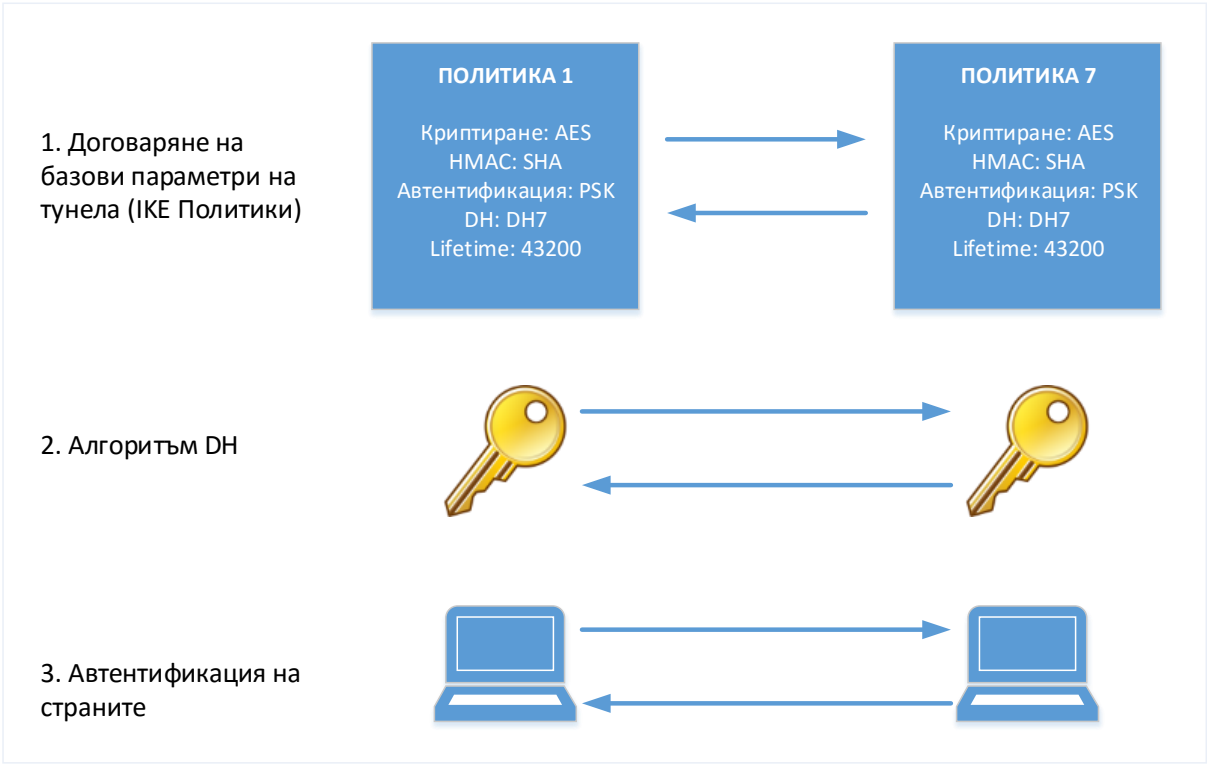
Някои от криптографските алгоритми изискват да има налични споделени ключове, които за да се генерират надеждно през несигурния комуникационен канал се прилага алгоритъма DH.

При IPsec се използва протоколът “Internet Key Exchange” (IKE), при който на базата на няколко обменени пакета между двете системи изграждащи тунела се изчисляват споделени ключове. IKE е описан в RFC 2409 и използва транспортен протокол UDP и порт 500. Най-общо за IKE може да се каже, че това е хибриден протокол, който обхваща “Internet Security Association and Key Management Protocol” (ISAKMP) и механизма за обмяна на ключове “Oakley and Skeme”.

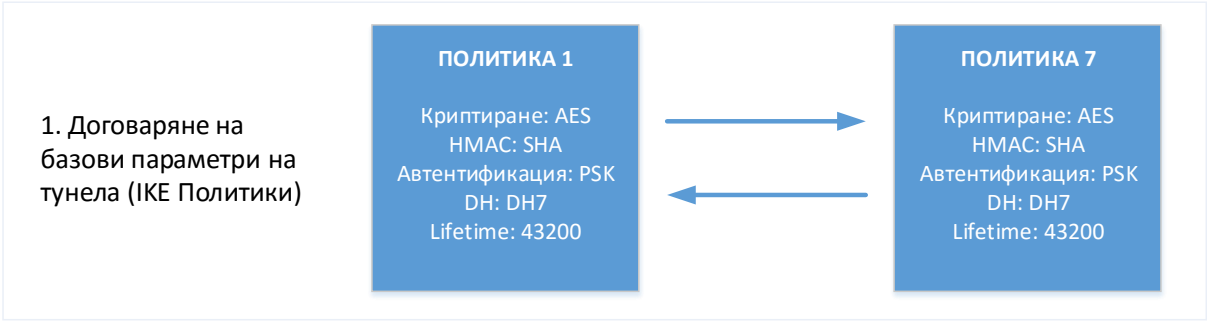
IKE работи в две фази:

1. **IKE фаза 1** – първоначално определяне на основните настройки за подсигуряване на тунела, включващи автентификация на системите, криптографски алгоритми, генериране на ключове и др.;
2. **IKE фаза 2** – след успешно преминаване на фаза 1 при необходимост от предоговаряне на параметрите се стартира значително по-бързата фаза 2.

IKE фаза 1



IKE фаза 2



Фиг. 12.10 Фази 1 и 2 при IKE

Предимства и недостатъци на IPsec

Виртуалните частни мрежи, базирани на IPsec имат следните предимства пред други VPN технологии:

- Използват набор от свободни стандарти, без да се обвързват с точно определени криптографски алгоритми;
- Сигурността на шифроването на данните, които се пренасят през тунела може да бъде много висока (към момента при използване на AES);
- Получените пакети са с гарантирана липса на модификации след тяхното изпращане;
- Възможно е да се реализират сложни топологии, които да бъдат икономически изгоден заместител на WAN свързаност;
- Значително се редуцира цената, най-вече при сравнение с наети линии;
- Почти всяка модерна операционна система поддържа IPsec;
- Възможно е да се използват хардуерни модули за ускоряване на криптографските алгоритми.

Като недостатък могат да се посочат:

- Необходимост от първоначално запознаване с технологията преди нейното прилагане на практика;
- Трудно отстраняване на възникнали проблеми, от администратори с малко опит;
- Необходимост от внимателен избор на криптографски алгоритми.

SSL VPN

За разлика от IPsec тунелите при SSL VPN не е необходимо да бъдат инсталирани допълнителни софтуерни пакети. Тунелът може да се изгради през стандартен Web браузър, което води до значително улеснение за потребителите.

Протоколът SSL поддържа различни криптографски алгоритми и методи, сред които автентификация на клиента и на сървъра, криптиране на трафика, цифрови сертификати и др.

Някои от най-основните предимства на SSL VPN са:

- Достъп до корпоративни данни и приложения през криптиран тунел директно от браузър, без да е необходимо инсталиране на допълнителен софтуер;
- Висока степен на гъвкавост;
- Намаляване на комуникационните разходи;
- Лекота при работа от страна на потребителите.

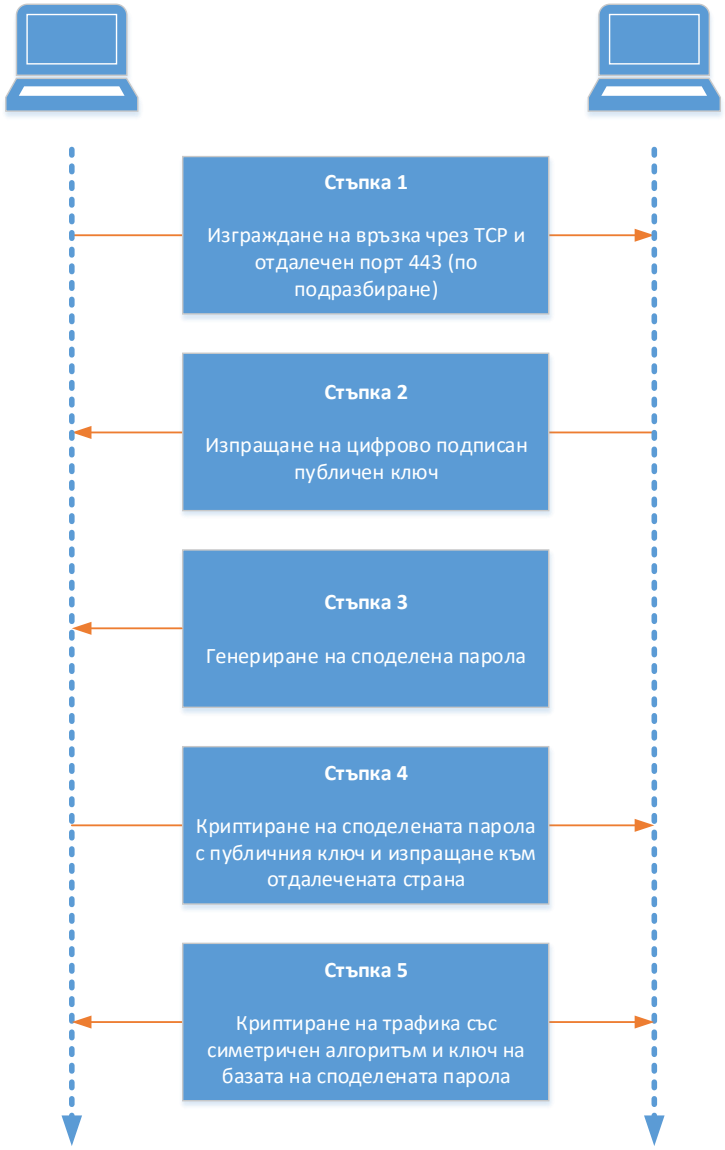
Фиг. 12.11 Достъп до SSL VPN през стандартен браузър (източник Интернет)

Ако се сравнят SSL и IPsec виртуалните частни мрежи, се вижда, че не всички приложения могат да използват пълноценно SSL VPN, както и че IPsec предоставя по-висока степен на защита на трафика, но е по труден за работа от страна на потребителите (най-вече при софтуерни клиенти с неинтуитивен интерфейс).

Основните етапи при изграждането на SSL VPN са следните:

1. Изграждане на връзка чрез TCP и отдалечен порт 443 (по подразбиране);
2. Изпращане на цифрово подписан публичен ключ;
3. Генериране на споделена парола;

4. Криптиране на споделената парола с публичния ключ и изпращане към отдалечената страна;
5. Криптиране на трафика със симетричен алгоритъм и ключ на базата на споделената парола.



Фиг. 12.12 Етапи при изграждане на SSL VPN

OpenVPN

"OpenVPN Technologies" е частна компания, регистрирана в Калифорния, САЩ, която разработва специализиран софтуер за мрежова комуникация. Най-популярния продукт с отворен код на тази компания е едноименния OpenVPN, който позволява да се изградят най-често използваните VPN топологии (point-to-point, site-to-site и remote-access) лесно и бързо, като защитата на информацията е на изключително високо ниво.

OpenVPN използва специално разработен протокол за генериране на ключовете, който се базира на SSL/TLS. Автентификацията на страните може да се извърши чрез споделени пароли, цифрови сертификати или комбинацията потребител/парола. Криптирането на данните

използва функциите на библиотеката OpenSSL, чрез която се шифроват и контролните пакети. Интегритета може да се гарантира чрез HMAC.

OpenVPN може да се използва с повечето от най-актуалните операционни системи, сред които Linux, Microsoft Windows, Android, BSD, MacOS и др.

Endian Firewall поддържа както IPsec, така и OpenVPN за изграждане на виртуални частни мрежи.

Конфигуриране на VPN с EFW CE

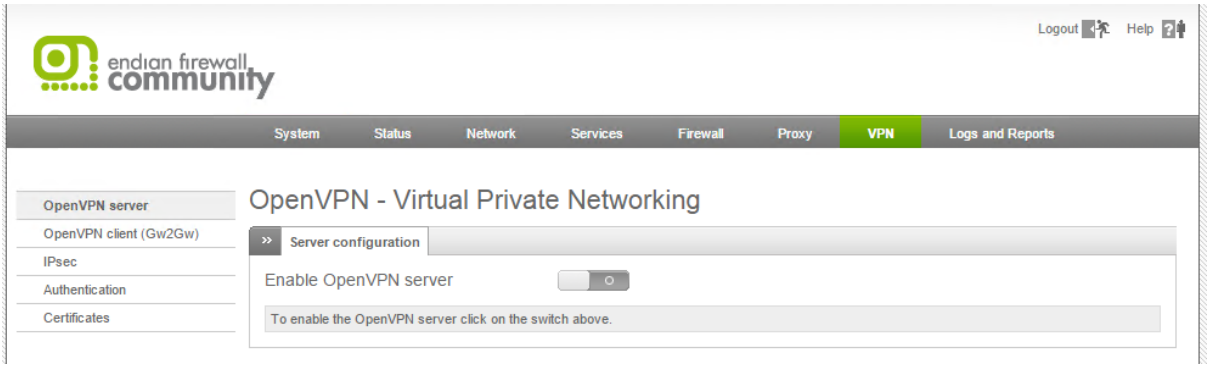
За създаване на виртуални частни мрежи Endian Firewall предлага да се конфигурират две от най-разпространените към момента технологии – IPsec и OpenVPN. Тяхната поддръжка и от най-разпространените операционни системи прави приложението им лесно.

Едно допълнение, което Endian предоставят е специално разработен софтуерен пакет, който е клиент за OpenVPN и който може да се изтегли от сайта на компанията.

EFW може да работи като OpenVPN сървър, клиент или едновременно като клиент и сървър.

Конфигурирането на VPN се извършва от едноименното меню VPN, което е разделено на следните подменюта:

- OpenVPN server – включва конфигурационни параметри за активиране на необходимите функции на OpenVPN за да може EFW CE да бъде използван като сървър за VPN;
- OpenVPN client (GW2GW) – от това подменю може да бъде направена необходимата конфигурация дадената EFW система да работи като клиент при реализиране на OpenVPN тунел;
- IPsec – съдържа необходимите настройки за изграждане на IPsec виртуални частни мрежи;
- Authentication – автентификация на страните;
- Certificates – управление на цифрови сертификати.



Фиг. 12.13 Меню VPN при EFW CE

EFW OpenVPN сървър

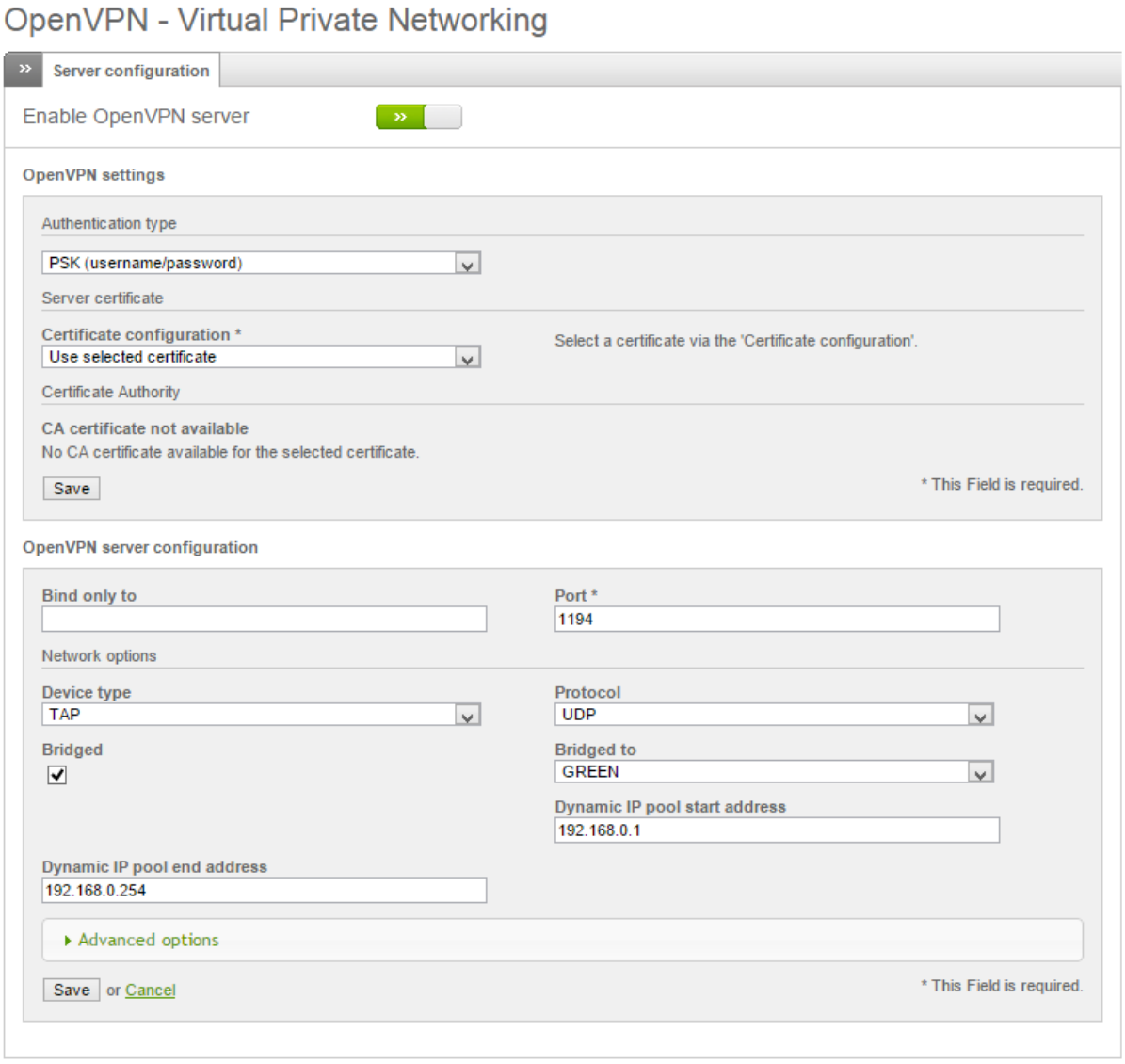
Когато EFW бъде конфигуриран в режим на OpenVPN сървър през подсигурена тунелирана свързаност на uplink интерфейс отдалечени клиенти могат да изградят VPN към защитната стена. От версия 3.0 на EFW е възможно да се стартират няколко паралелни OpenVPN сървърни процеса, като всеки от тях използва отделен порт, а при наличие на процесор с няколко ядра и пренасочване на натоварването към някое от тях. Възможно е да се стартират и няколко OpenVPN сървъра на едноядрен процесор, но това ще доведе до редуциране на

производителността на системата като цяло и на пропускателната способност на тунела. Тази функционалност е ограничена при EFW CE, което е и една от разликите между свободната и платената версия на продукта.

Конфигурационната страница OpenVPN сървър включва:

- Server configuration – параметри за настройка на OpenVPN сървъра;
- Client download (не е налична при EFW CE) – изтегляне на специално разработен от Endian OpenVPN клиент.

Стартирането на OpenVPN сървъра се извършва чрез преместване на бутона “Enable OpenVPN Server” в активна позиция.



Фиг. 12.14 Конфигуриране на OpenVPN сървър при EFW CE

Конфигуриране

Настройките на OpenVPN сървъра са разделени на две части – “OpenVPN settings” и “OpenVPN server configuration”, като първата включва:

- Authentication type – избор на метод за автентификация. Възможните варианти са споделена парола (PSK), X.509 цифров сертификат или двуфакторна автентификация (X.509 в комбинация с PSK). Ако се използва единствено X.509 всеки клиент, който има валиден сертификат ще може успешно да се автентифицира към EFW;
- Certificate configuration – позволява да се посочи метода за приложение или генериране на новите цифрови сертификати. Първата опция “Generate a new certificate” е валидна единствено, ако на защитната стена няма генерирани цифрови сертификати и след нейния избор се визуализират полета, свързани със създаването на нов сертификат. Втората опция “Use selected certificate” позволява на администраторите да посочат кой съществуващ сертификат да бъде използван. “Upload certificate” може да се използва, когато е необходимо да се изпрати наличен цифров сертификат към EFW. Последната опция “Upload a certificate signing request” позволява да се изпрати към EFW заявка от тип CSR (Certificate Signing Request). Всяка една опция изисква допълнителни конфигурационни параметри.

За да се активират направените настройки, свързани с автентификацията и сертификатите се натиска бутона “Save” в съответната група конфигурационни параметри.

Втората част на страницата “OpenVPN server” съдържа основните параметри за сървърния процес:

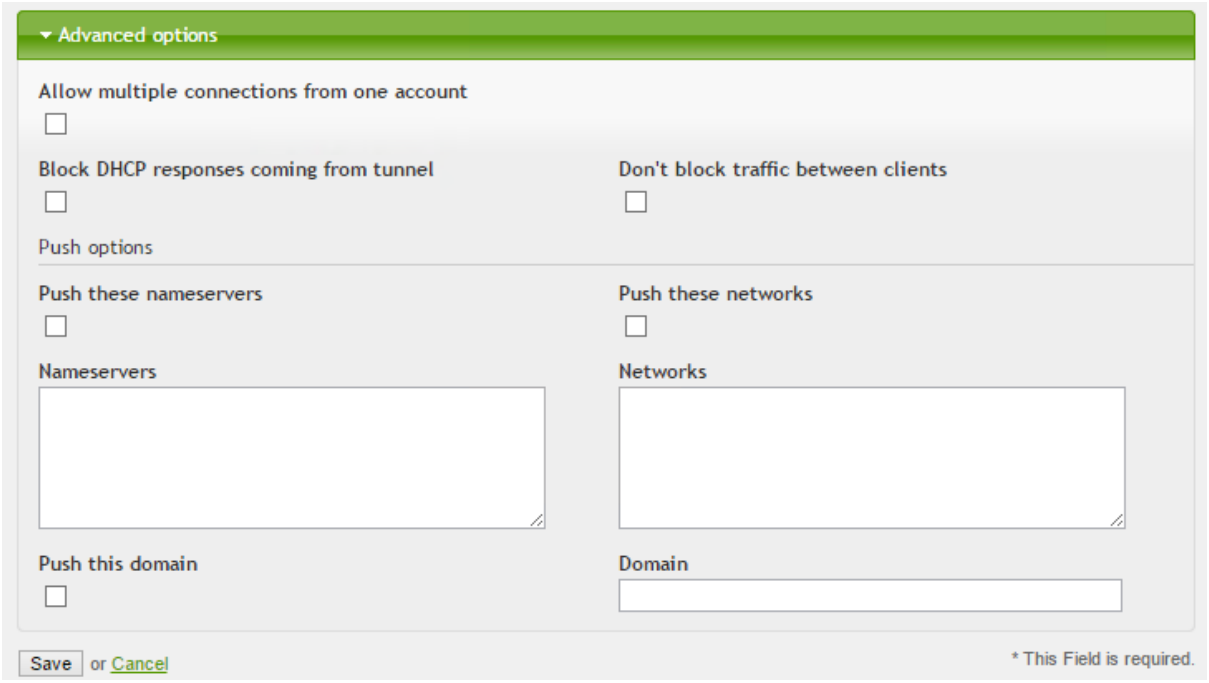
- Bind only to – IP адресът, на който да работи OpenVPN сървъра;
- Port – портът, на който сървърния процес очаква да се получи заявка от страна на клиент;
- Device type – тип на тунела. При TUN се изисква пакетите да се маршрутизират и не се позволява да се използва опцията за изграждане на мостова свързаност, която е налична при TAP;
- Protocol – транспортен протокол (TCP или UDP);
- Bridged – активира мостова свързаност при тип на тунела TAP;
- Bridged to – ако типа на тунела е TAP се посочва към кой интерфейс на EFW да се направи мостова свързаност.
- Dynamic IP pool start address – начален адрес от съответната мрежа, който да се използва от OpenVPN клиента;
- Dynamic IP pool end address – последен адрес от съответната мрежа, който да се използва от OpenVPN клиента.

След въвеждане на параметрите е необходимо те да се запишат чрез бутана “Save” от текущата секция.

Разширени настройки (Фиг. 12.15) на OpenVPN сървъра могат да бъдат променени от връзката “Advanced options”, като те включват:

- Allow multiple connections from one account – разрешава даден клиент да се свърже към EFW OpenVPN сървъра повече от един път и от различни места. При тази ситуация правилата на VPN защитната стена не се използват, което може да се посочи и като недостатък;
- Block DHCP responses coming from tunnel – блокират се всички отговори на DHCP заявки, които се получават през тунела. Тази опция позволява да се избегне DHCP конфликт с локален сървър;
- Don't block traffic between clients – разрешава трафика между отделните VPN клиенти;
- Push these nameservers – към клиента се изпращат описаните в списъка DNS сървъри;
- Nameservers – списък с DNS сървъри;

- Push these networks – към маршрутизиращата таблица на клиента се добавят описаните мрежи;
- Networks – списък с мрежи;
- Push this domain – към клиента се изпраща посоченото име на домейн;
- Domain – име на домейн.
  Направените настройки се активират след натискане на бутона “Save”
  Част от тези конфигурационни параметри се различават между EFW и EFW CE.



Фиг. 12.15 Разширени настройки на OpenVPN сървър при EFW CE

Изтегляне на VPN клиент

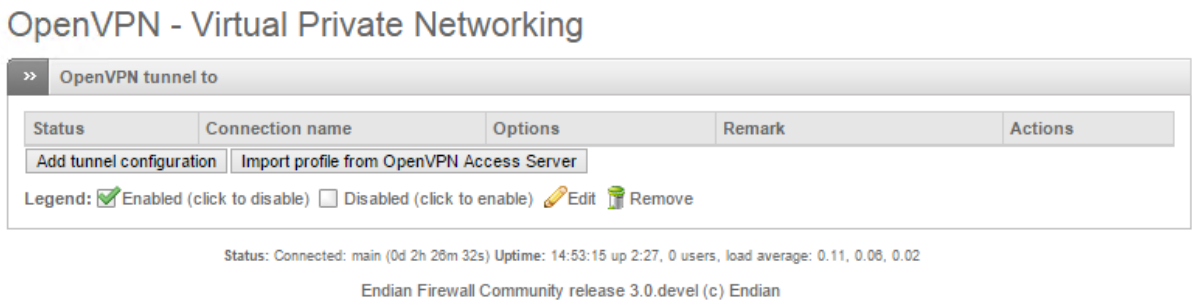
Едно от ограниченията при EFW CE е липсата на специално разработения от EFW VPN клиент. Това ограничение изисква да се използва стандартния OpenVPN клиент, който може да бъде изтеглен от www.openvpn.net и който има версии за Microsoft Windows, Mac OS, Android, iOS и Linux.

Настройките на OpenVPN клиента се извършват през текстови конфигурационни файлове, като примерен набор от параметри е:

```
client
dev tap
proto udp
remote efw.ict-academy.bg
resolve-retry infinite
nobind
persist-key
persist-tun
ca efw.pem
pska12 alex.p12
auth-user-pass
comp-lzo
```

EFW OpenVPN клиент (Gw2Gw)

EFW CE може да се използва и като OpenVPN клиент, като този му режим на работа се нарича Gateway-to-Gateway (Gw2Gw). Тази функционалност позволява изграждане на сложни VPN топологии, а настройките се извършват от подстраницата “OpenVPN client (Gw2Gw)” на VPN.



Фиг. 12.16 Настройки на OpenVPN в режим Gw2Gw при EFW CE

В таблицата са включени всички конфигурирани VPN връзки към отдалечени OpenVPN сървъри. Статус “closed” означава, че тунела не е активен, “established” – тунелът работи, а “connecting”, че в момента дадената свързаност се изгражда.

Аналогично на останалите таблици при EFW от колоната “Actions” може да се променя състоянието, да се редактира или изтрие дадена конфигурация.

Конфигуриране на тунел

Конфигурирането на нов тунел в режим Gw2Gw може да се извърши по два начина:

1. Добавяне на необходимите параметри през интерфейса на EFW (Add tunnel configuration);
2. Чрез импортиране на настройките от OpenVPN сървър (Import profile from OpenVPN Access Server).

При първият подход е необходимо да бъдат конфигурирани следните параметри:

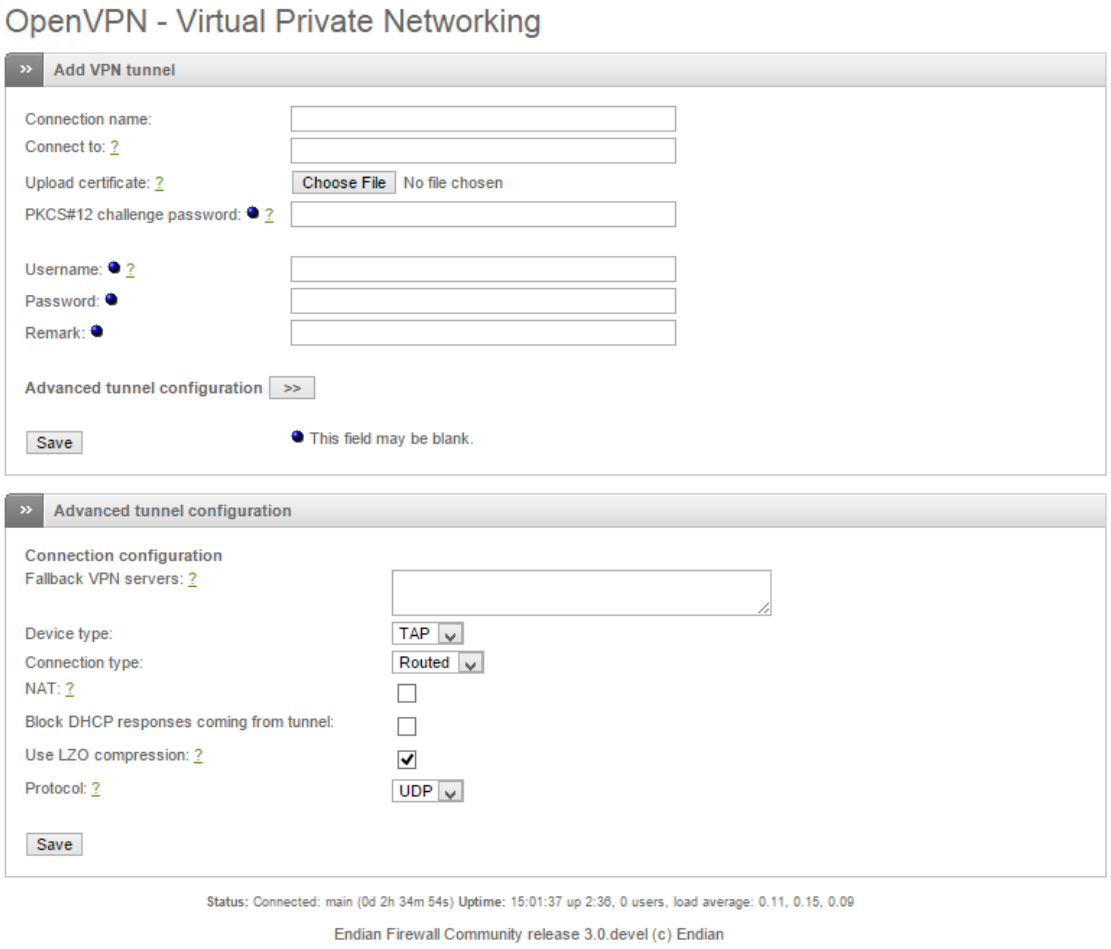
- Connection name – име на връзката;
- Connect to – FQDN на отдалечения сървър (например vpn.test.com:port:protocol). По подразбиране протоколът е UDP, а използвания порт е 1194. Важно е да се отбележи, че при задаване на протокола трябва да се използват малки букви;
- Upload certificate – изпращане на сертификата на сървъра, който е необходим за изграждане на тунела. От бутонът Browse може да се избере файла;
- PKCS#12 challenge password – паролата, за достъп до PKCS ключа;
- Username – потребител (опционално при двуфакторна автентификация);
- Password – парола (опционално при двуфакторна автентификация);
- Remark – коментар.

Връзката “Advanced tunnel configuration” дава възможност за конфигуриране на допълнителни параметри, които рядко се налага да бъдат променяни:

- Fallback VPN servers – Един или няколко VPN сървъра, които да се използват, ако основният не е достъпен;
- Device type – тип на използваното устройство TAP или TUN;
- Connection type – тип на връзката, която може да бъде зададена като маршрутизирана или като мостова;

- NAT – ако връзката е от маршрутизиран тип, тази опция позволява клиентите да се намират зад NAT процес, спрямо тунела;
- Block DHCP responses coming from tunnel – забраняване на DHCP отговори, постъпващи през тунела;
- Use LZO compression – компресиране на данните, които се изпращат към тунела;
- Protocol – транспортен протокол, който може да бъде UDP или TCP.

Направените конфигурационни настройки се записват чрез бутоните “Save”.



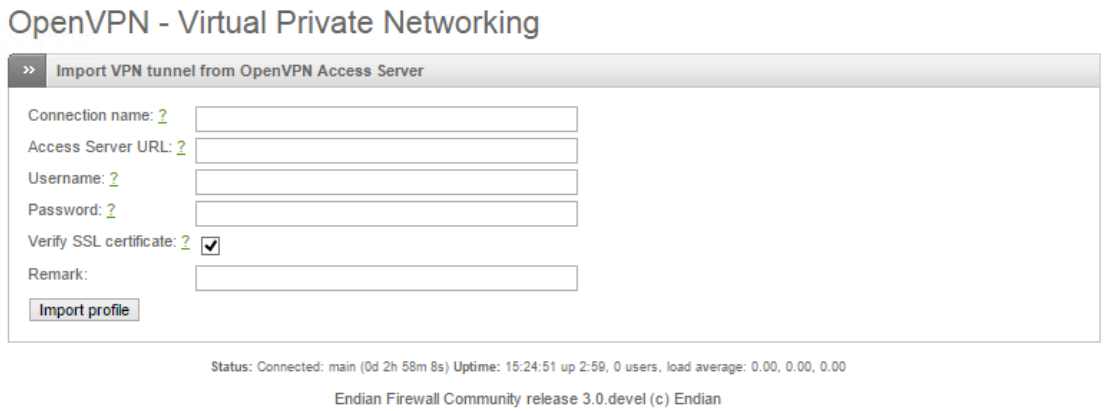
Фиг. 12.17 Настройки на OpenVPN в режим Gw2Gw посредством интерфейса на EFW CE

Импортирането на настройките от OpenVPN сървър се извършва чрез бутона “Import profile from OpenVPN Access Server”.

Необходимо е да бъдат посочени:

- Connection name – име на връзката;
- Access Server URL – URL, от който чрез XML-RPC да бъде получена необходимата конфигурация;
- Username – потребителско име;
- Password – парола;
- Verify SSL certificate – ако тази опция е активирана, и сървъра използва подсигуряване на връзката с SSL, използвания сертификат ще бъде проверен за валидност;
- Remark – опционален коментар.

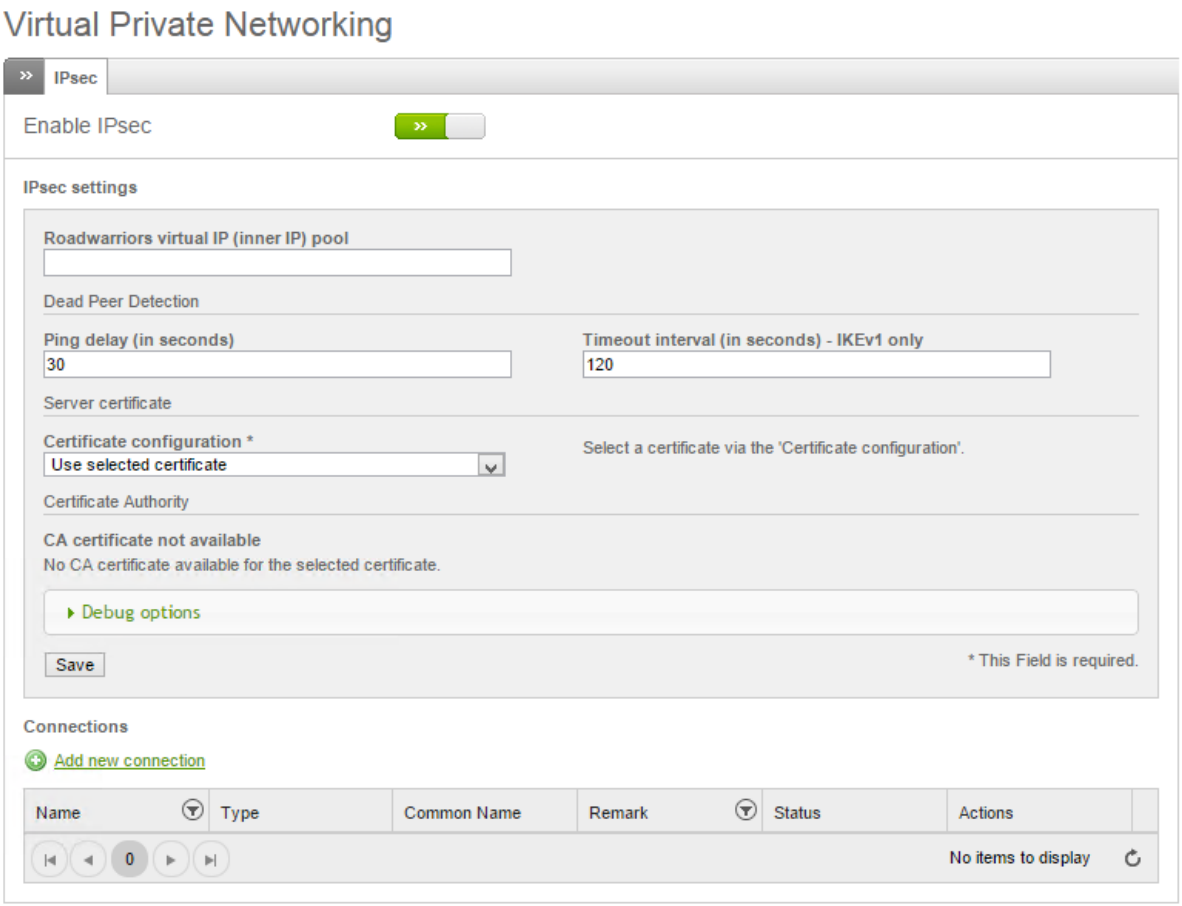
Импортирането на данните се стартира при натискане на бутона “Import profile”.



Фиг. 12.18 Импортиране на настройки на OpenVPN в режим Gw2Gw

IPsec

EFW CE поддържа изграждане на IPsec VPN, като достъп до конфигурационните параметри може да се получи от менюто “VPN” и подстраницата “IPsec”, която е разделена на две части – настройки (IPsec settings) и връзки (Connections).



Фиг. 12.19 настройки на IPsec VPN

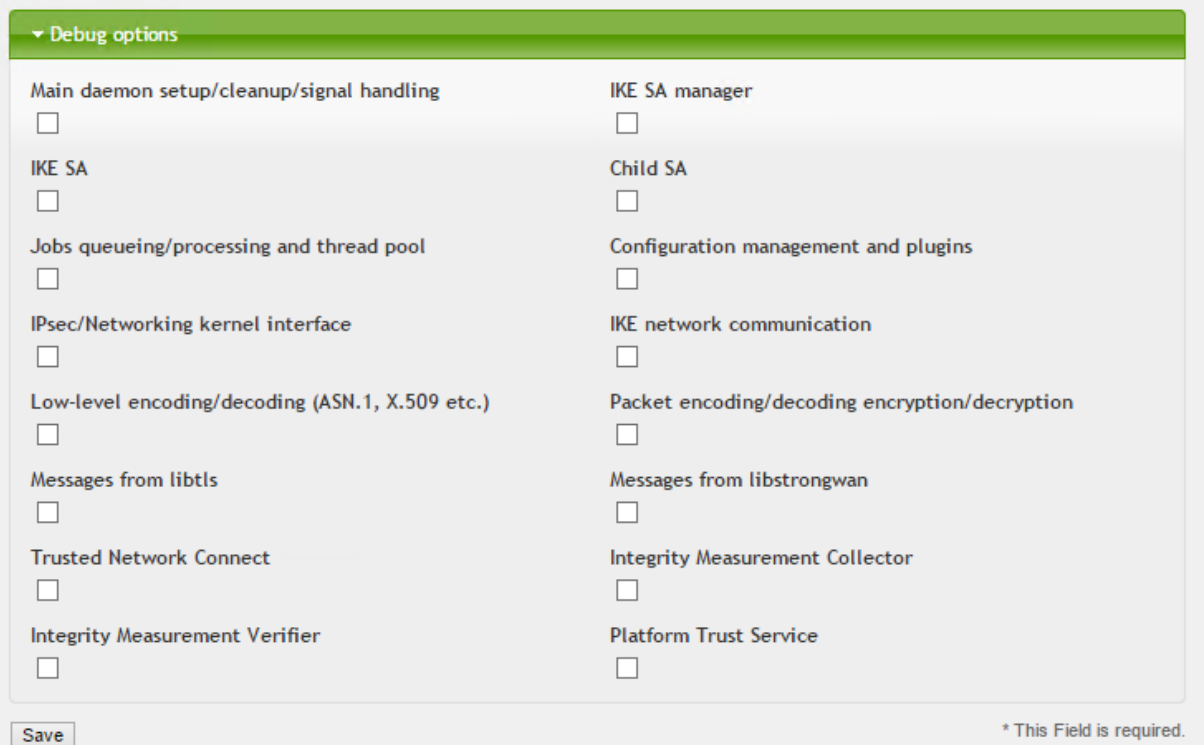
Основните настройки, свързани с IPsec са:

- Roadwariors virtual IP (inner IP) pool – адресите, които клиентите ще получават след успешно свързване през IPsec тунел;

- Ping delay (in seconds) – интервал от време, зададен в секунди, който се използва за изпращане на ICMP Echo-request (ping) с цел проверка на тунела;
- Timeout interval (in seconds) - IKEv1 only – максимален интервал от време, зададен в секунди, който се използва за приключване на функциите на протокола IKEv1;
- Certificate configuration – управление на цифровите сертификати, необходими за изграждане на IPsec VPN тунела. Принципа на тяхното конфигуриране е аналогичен на използвания при OpenVPN.

Когато необходимите параметри са въведени конфигурацията се записва чрез натискане на бутона “Save”.

При отстраняване на проблеми, свързани с работата на IPsec тунел могат да бъдат активирани т.нар. “Debug” опции. Те генерират допълнителни съобщения в журналите и са полезни при откриване на неточности в конфигурацията или други причини за проблеми в работата на IPsec.



Фиг. 12.20 Debug опции за анализ на IPsec VPN

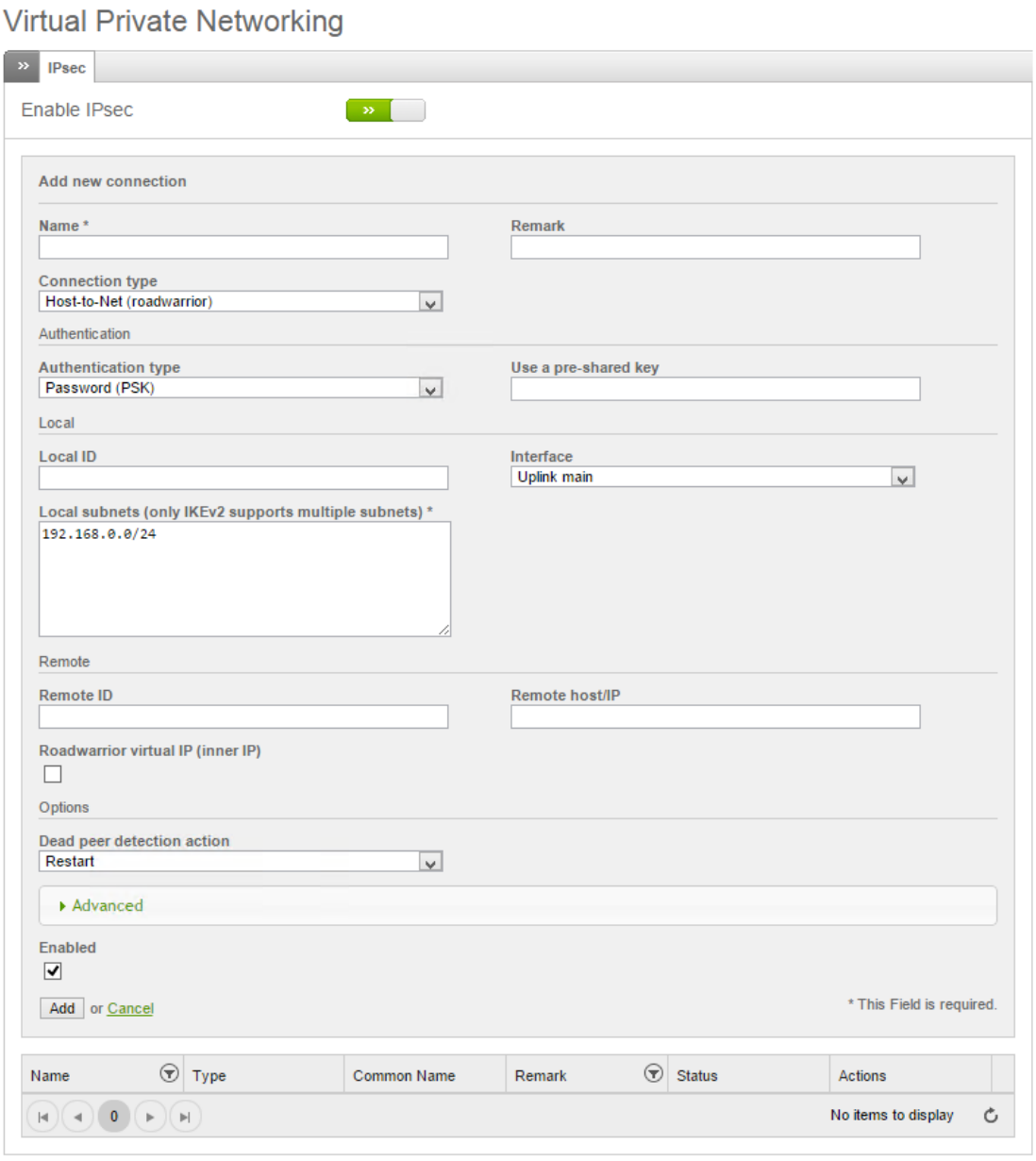
Добавянето на нова IPsec комуникационна линия се извършва от връзката “Add new connection”. Необходимо е да бъдат конфигурирани:

- Name – име на връзката;
- Remark – опционален коментар;
- Connection type – тип на връзката. При “Host-to-Net” един клиент се свързва към IPsec сървъра, което е типичен пример за т.нар. “remote-access” VPN, при “Net-to-Net” се изгражда свързаност между два мрежови сегмента, а при “XAuth Host-to-Net” отново един клиент се свързва към сървъра, но автентификацията е през XAuth[86];

[86] Тази опция не е налична при EFW CE

- Authentication type – тип на автентификацията, като в зависимост от направения избор са налични и допълнителни конфигурационни опции;
- Local ID – име, което еднозначно определя клиента в локалната мрежа;
- Interface – мрежови интерфейс, който се използва от тунела;
- Local subnets – локалните мрежи или подмрежи, които ще са достъпни от страна на клиента. Възможно е да се добавят повече от един запис единствено, при IKEv2;
- Remote ID – идентификатор за отдалеченото устройство;
- Remote host/IP – IP или FQDN на отдалечената система;
- Roadwarrior virtual IP – посоченият IP адрес ще бъде изпратен на клиента;
- Dead peer detection action – действието, което ще се извърши, ако клиента се изключи. За всяка избрана опция е необходимо да се конфигурират и допълнителни параметри;

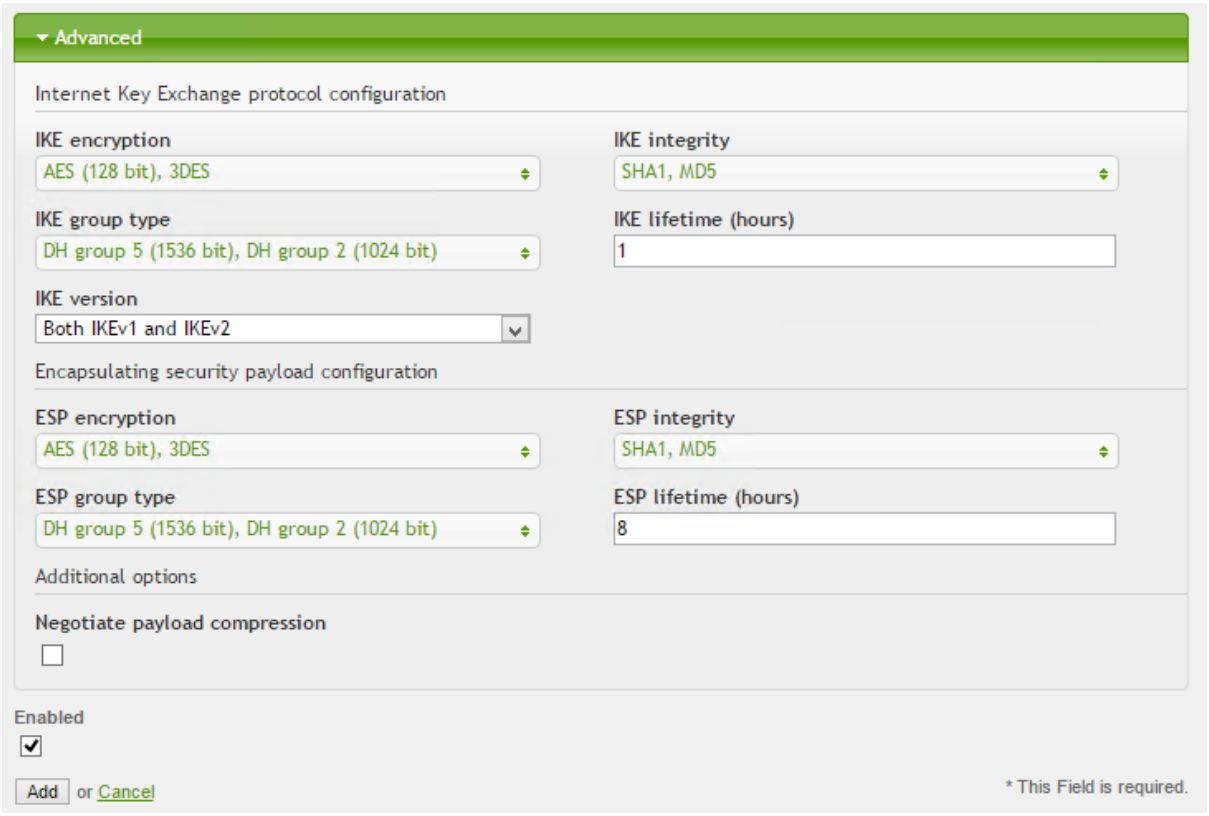
Добавянето на връзката се осъществява от бутона “Add”.



Фиг. 12.21 Добавяне на нова IPsec VPN линия (тунел)

Разширените настройки, свързани с дадена връзка са:

- IKE encryption – алгоритъм за шифриране, който ще се използва от IKE;
- IKE integrity – алгоритъм за интегритет, използван от IKE;
- IKE group type – DH група;
- IKE lifetime – интервала от време в часове, за който IKE пакетите са валидни;
- IKE version – версия на IKE;
- ESP encryption – алгоритъм за шифриране , който ще се използва от ESP;
- ESP integrity – алгоритъм за интегритет, използван от ESP;
- ESP group type – DH група;
- ESP lifetime – интервала от време в часове, за който ESP ключа е валиден;
- Negotiate payload compression – активира компресия на пренасяните данни.

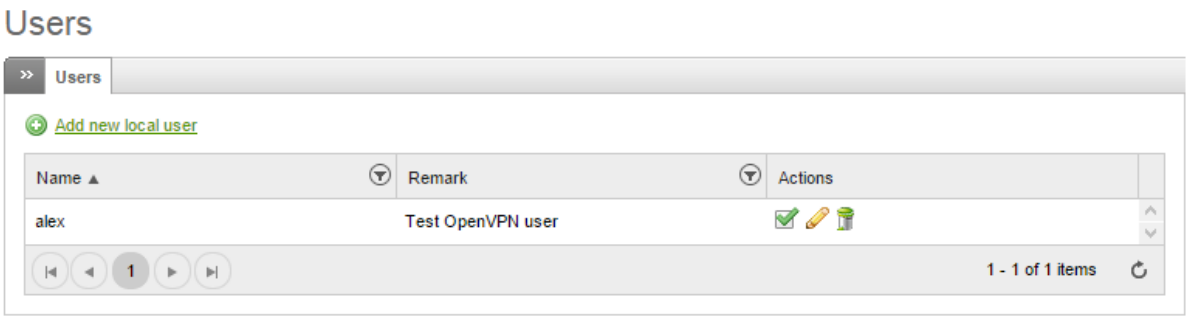


Фиг. 12.22 Разширени IPsec VPN настройки

Автентификация

Подменюто “Authentication” при “VPN” дава достъп до функции за създаване или редактиране на локални потребители.

Аналогично на повечето страници от графичния интерфейс на EFW списъка с потребителите е представен в табличен вид, а от колоната “Actions” може да се избере действие.

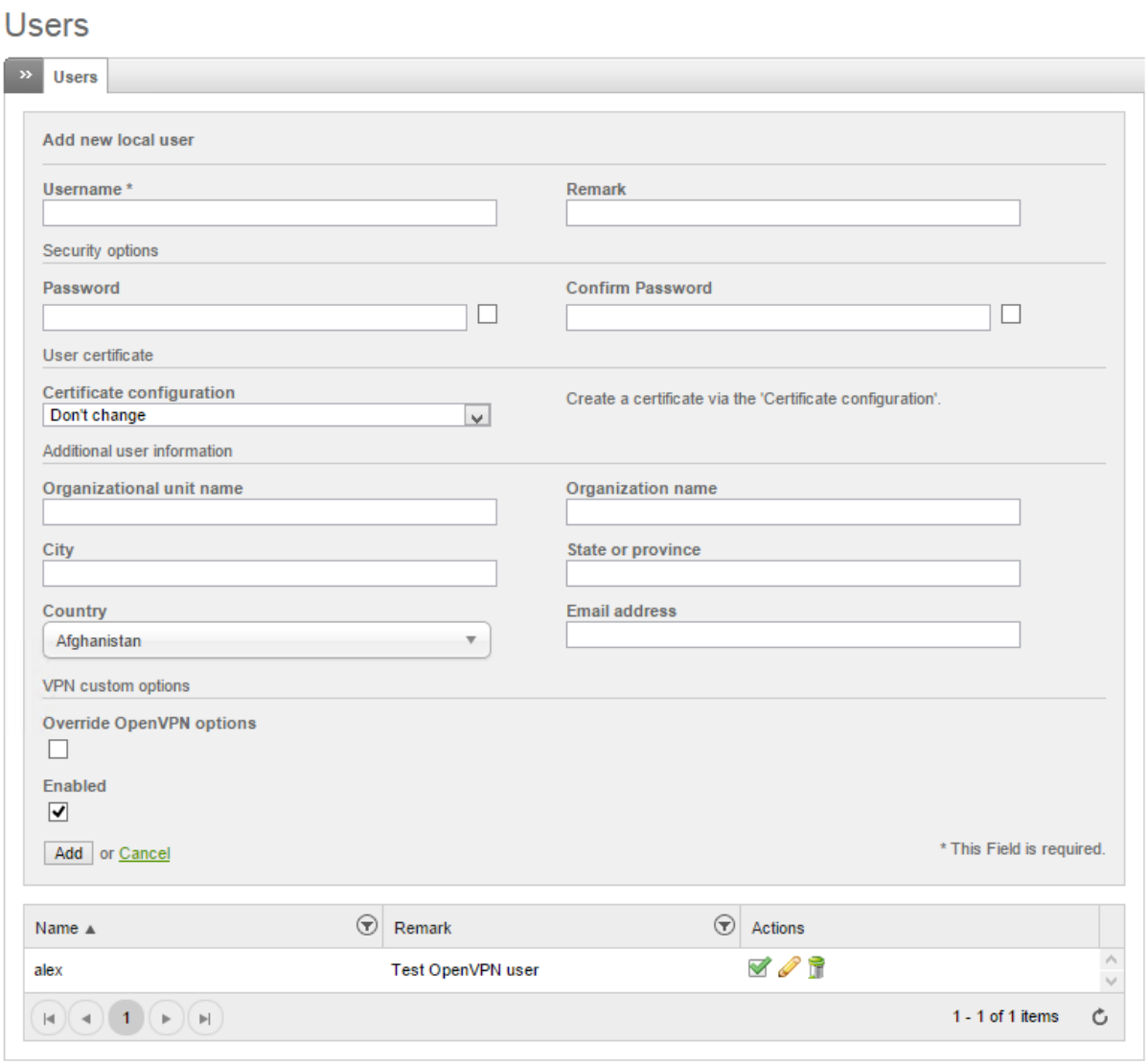


Фиг. 12.23 Управление на VPN потребителите

Добавянето на нов локален потребител (При EFW CE не се поддържат отдалечени) е сравнително лесно и изисква да бъдат въведени следните параметри:

- Username – потребителско име;
- Remark – опционален коментар;
- Password – парола, като е препоръчително винаги да се спазват правилата за надеждни пароли.
- Confirm Password – потвърждение на въведената парола;
- Certificate configuration – управление на сертификатите. Възможните опции са да не се използва цифров сертификат, да се генерира нов, да се изпрати съществуващ или CSR;
- Organizational unit name – организация, към която е причислен сертификата;
- Organization name – отдел, към който принадлежи потребителя;
- City – град;
- State or province – щат или община;
- Country – държава;
- Email address – адрес за електронна поща на потребителя;
- Override OpenVPN options – ако тази опция е активна се визуализират допълнителни конфигурационни полета, свързани с потребителя и OpenVPN;
- Enabled – активира или деактивира профила на потребителя;

Добавянето на новия потребител се извършва от бутона “Add”.



Фиг. 12.24 Параметри за добавяне на нов локален VPN потребител

Цифрови сертификати

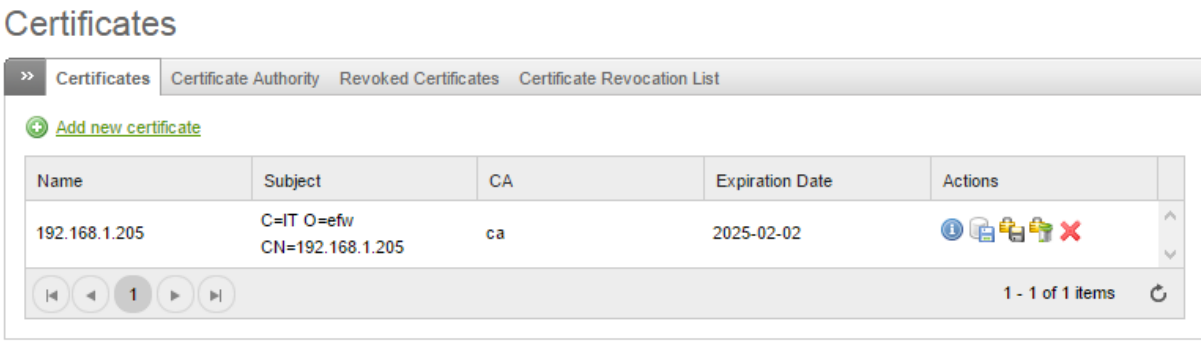
Подменюто “Certificates” от “VPN” предоставя възможност за управление на цифровите сертификати, които се използват от EFW.

Страницата е разделена на четири части:

1. **Certificates** – включва списък със сертификати, функция за добавяне на нов, изтегляне, информация и изтриване за всеки един;
2. **Certificate Authority** – управление на CA;
3. **Revoked Certificates** – списък с изтеглените от употреба сертификати;
4. **Certificate Revocation List** – управление на качените списъци със сертификати, които са изтеглени от употреба.

Основните параметри за сертификатите са:

- Serial – уникален сериен номер;
- Name – име на сертификата;
- Subject – описание на сертификата;
- Expiration Date – последна дата на валидност.



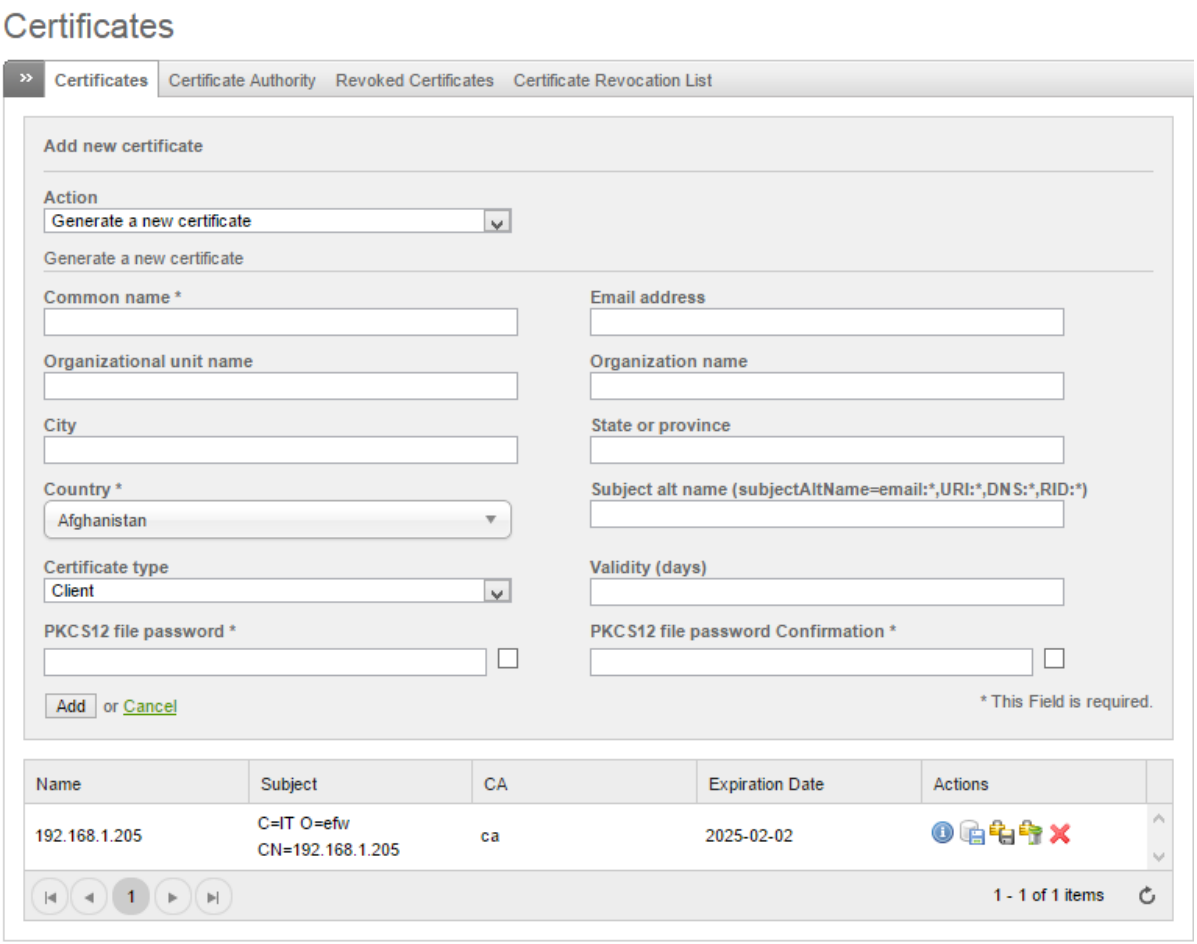
Фиг. 12.25 Управление на цифрови сертификати при EFW CE

Генерирането на нов цифров сертификат се извършва от връзката “Add new certificate”, която е в групата “Certificates”. Конфигурационните параметри са:

- Action – действие, което позволява генериране на нов сертификат или изпращане на съществуващ;
- Common name – име (CN) на собственика на сертификата;
- Email address – адрес за електронна поща на собственика на сертификата;
- Organizational unit name – организация, към която е причислен сертификата;
- Organization name – отдел, към който принадлежи сертификата;
- City – град;
- State or province – щат или община;
- Country – държава;
- Subject alt name – алтернативно име;
- Certificate type – тип на сертификата. Възможно е да се посочи дали да е за клиент (client) или да се използва на ниво сървър (server);
- Validity – валидност, посочена в брой дни;
- PKCS12 file password – парола за достъп до PKCS12 файла;
- PKCS12 file password confirmation – потвърждение на паролата за достъп до PKCS12 файла.

Новият сертификат се добавя след натискане на бутона “Add“.

EFW позволява да се инсталира вече съществуващ сертификат, като към системата се изпрати (upload) PKCS12 или PEM файл, като при необходимост трябва да се въведе и паролата за достъп до сертификата.

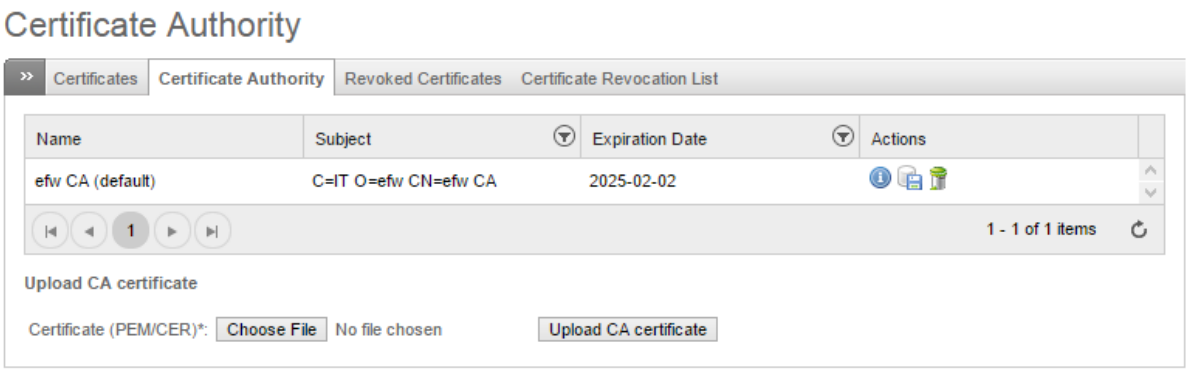


Фиг. 12.26 Генериране на нов цифров сертификат при EFW CE

За да може VPN (в частност OpenVPN) да функционира правилно е необходимо да има конфигуриран Certificate Authority (CA). Това се извършва от втората група функции – “Certificate Authority”.

При създаване на нов CA трябва да се въведат:

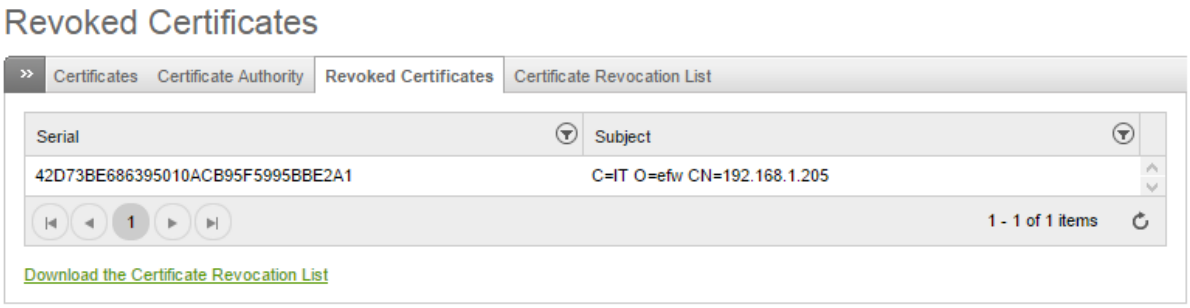
- System hostname – името на системата, което ще се използва при CN на сертификата;
- Email address – адрес за електронна поща на отговорника за системата (администратор);
- Organizational unit name – организация, към която е причислен сертификата;
- Organization name – отдел, към който принадлежи сертификата;
- City – град;
- State or province – щат или община;
- Country – държава;
- Subject alt name – алтернативно име;
- Validity – валидност, посочена в брой дни.



Фиг. 12.27 Създаване на нов CA

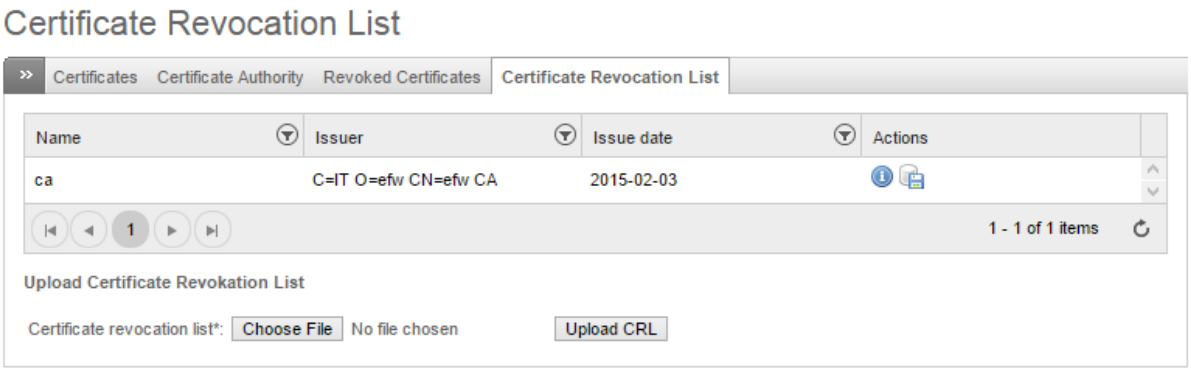
Изтеглените от употреба сертификати са описани в табличен вид в третата група – “Revoked certificates” на страницата “Certificates”.

От втората икона в колоната “Actions” може да се изтегли списъка с излезлите от употреба сертификати.



Фиг. 12.28 Излезли от употреба сертификати при EFW CE

Последната група от страницата “Certificates” е свързана с управление на списъците с изтеглени от употреба сертификати.



Фиг. 12.29 Списъци с излезли от употреба сертификати при EFW CE

VPN Firewall

VPN защитната стена (VPN Firewall) позволява да се ограничи трафика, който постъпва от или се генерира към OpenVPN потребителите. По подразбиране тази функционалност е изключена и за да се стартира е необходимо от менюто “Firewall” да се избере “VPN Firewall” и да се премести бутона “Enable VPN firewall” в активна позиция.

Конфигурираните правила са описани в табличен вид, като важно напомняне е, че техния ред (позиция) е от съществено значение при филтрирането на пакетите.

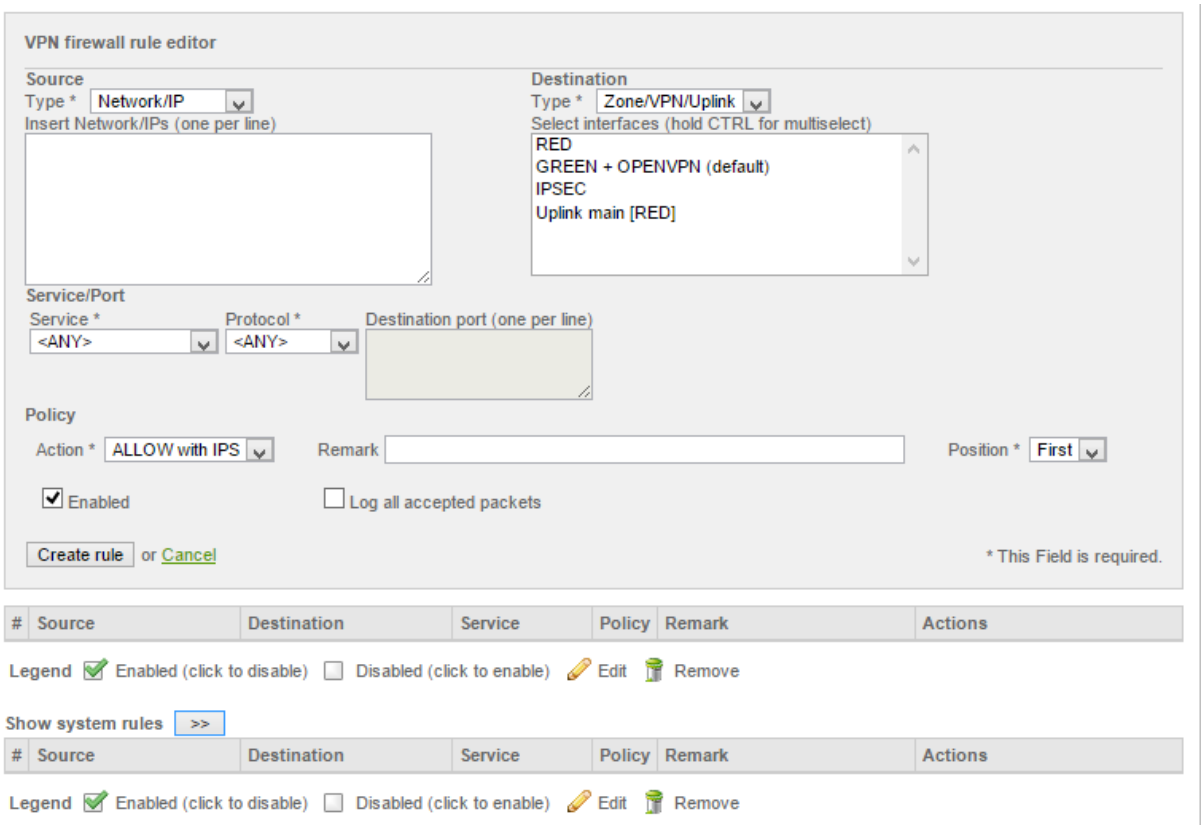
Направените промени в конфигурацията се записват с бутона “Save”, като е необходимо промените да се потвърдят с “Apply”, след което някои от системните услуги се рестартират, което може да отнеме известен интервал от време.

Фиг. 12.30 Основна конфигурация на VPN защитна стена при EFW CE

Добавянето на ново правило се извършва от “Add a new VPN firewall rule”, като параметрите са следните (аналогични на другите правила при защитната стена):

- Source – източник на трафика;
- Destination – получател на пакетите;
- Service – услуга, като е възможно да се избере предварително конфигурирана от списъка или да се посочи комбинация транспортен протоколи порт/портове;
- Policy – действие, като по подразбиране трафика се пропуска, но се проверява от IPS системата (ALLOW with IPS);
- Remark – опционален коментар;
- Position – позиция на новото правило;
- Enabled – активира или деактивира правилото;
- Log accepted VPN connections – в журнала се добавя информация за одобрената VPN връзка.

В групата “System access” са включени правилата, които разрешават административен достъп до EFW системата през VPN тунел.



Фиг. 12.31 Правила при VPN защитна стена на EFW CE

Заключение

Технологията на виртуалните частни мрежи предоставя висока степен на защита при пренос на данни през несигурни комуникационни канали, като изграждане топологии могат да бъдат гъвкави и се явяват алтернатива на WAN свързаност;

Изборът на криптографски алгоритми е важен параметър за постигане на максимално ниво на сигурност и производителност на VPN;

EFW CE поддържа OpenVPN и IPsec VPN, а тяхното конфигуриране и използване е интуитивно.

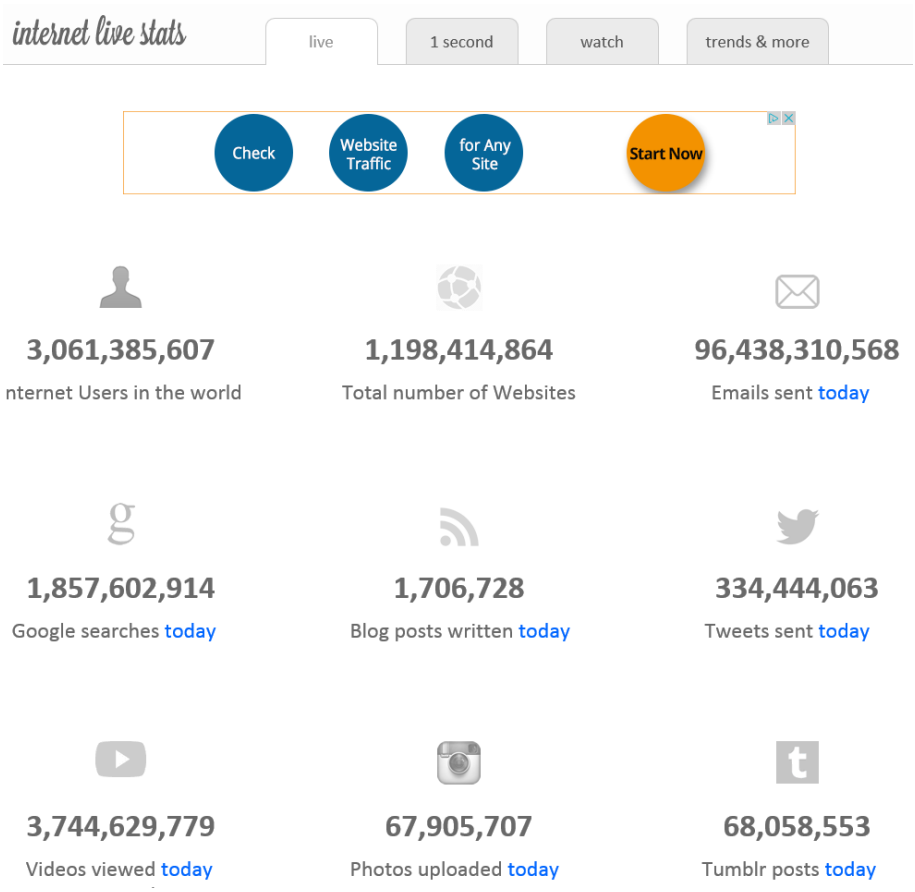
Източници

1. https://tools.ietf.org/html/rfc1701
2. https://www.ietf.org/rfc/rfc3031.txt
3. https://www.ietf.org/rfc/rfc2401.txt

Глава 13. Защита на електронна поща с EFW CE

Електронната поща е една от най-често използваните услуги в Интернет, като тя води своето начало от 1961 година, когато MIT разработват Compatible Time-Sharing System (CTSS). CTSS позволява няколко потребителя едновременно да се свържат от своите терминали през модеми към централизиран компютър, на който да съхраняват и споделят файлове и съобщения. През 1971 е изпратено първото електронно писмо през ARPANET, което поставя началото на развитието на стандартизираните протоколи и софтуерни клиенти.

Една интересна статистика, която е представена на сайта www.internetlivestats.com показва, че само за деня са изпратени над 96 милиарда електронни писма, което сравнено с останалите услуги това е е най-големия брой.



Фиг. 13.1 Статистика от www.internetlivestats.com

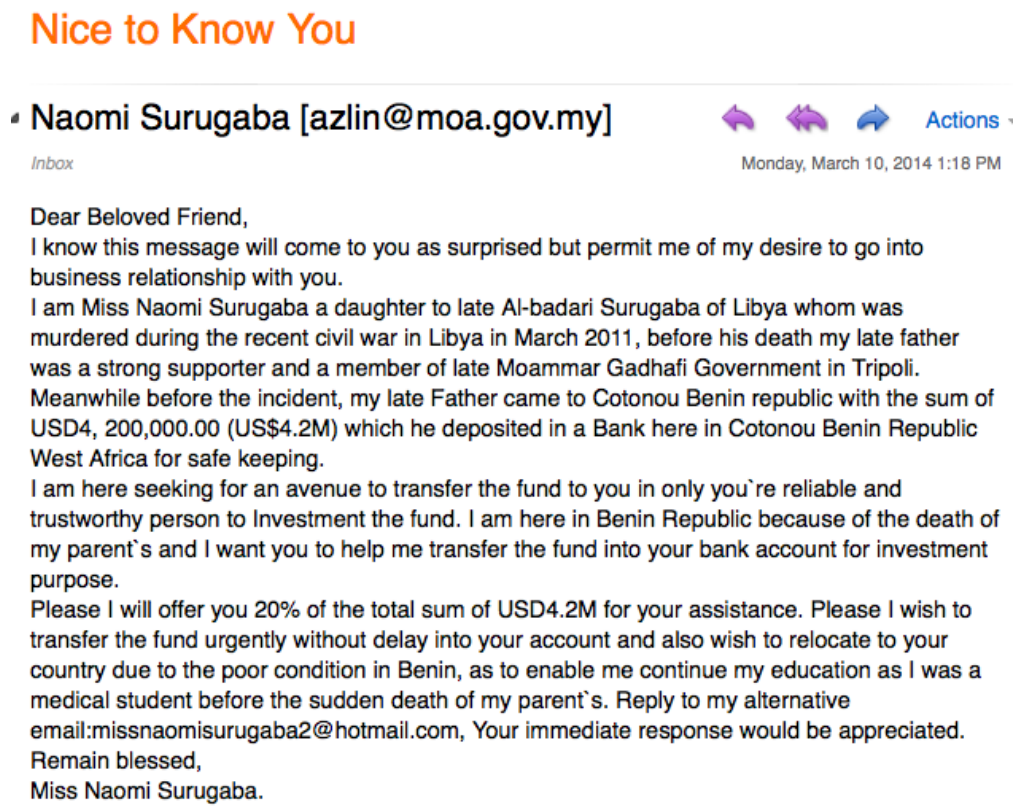
Заплахи при електронната поща и използваните протоколи

Електронната поща е една от най-използваните услуги в Интернет, което логично води и до нейното приложение като основен инструмент за атаки, насочени към потребителите. Анализи показват, че през 2011 година честотата но този тип атаки се увеличава, а финансовите негативи при успешно провеждане могат да бъдат значителни. Интересен показател е, че намалява броя на масовите изпращания на електронна поща (mass e-mail), за сметка на директно подбрани получатели.

Въпреки, че след глобална кампания за инсталиране на специализирани филтри за анализ и оценка на електронните писма броят на нежеланите (SPAM) значително се редуцира ежедневно все още се получават поне няколко съобщения от този тип. Най-често SPAM съдържа

не-поискана реклама или връзки, които водят към сайтове или файлове с инсталиран зловреден код.

"Nigerian scam" е тип SPAM, при който съобщението съдържа информация за "изключително изгодно" предложение от Нигерийски гражданин, който ней-често е със "знатни корени" и който търси финансова подкрепа да си върне позицията, след което ще се "отблагодари".



Nice to Know You

Naomi Surugaba [azlin@moa.gov.my] Actions

Inbox Monday, March 10, 2014 1:18 PM

Dear Beloved Friend,
I know this message will come to you as surprised but permit me of my desire to go into business relationship with you.
I am Miss Naomi Surugaba a daughter to late Al-badari Surugaba of Libya whom was murdered during the recent civil war in Libya in March 2011, before his death my late father was a strong supporter and a member of late Moammar Gadhafi Government in Tripoli.
Meanwhile before the incident, my late Father came to Cotonou Benin republic with the sum of USD4, 200,000.00 (US$4.2M) which he deposited in a Bank here in Cotonou Benin Republic West Africa for safe keeping.
I am here seeking for an avenue to transfer the fund to you in only you`re reliable and trustworthy person to Investment the fund. I am here in Benin Republic because of the death of my parent`s and I want you to help me transfer the fund into your bank account for investment purpose.
Please I will offer you 20% of the total sum of USD4.2M for your assistance. Please I wish to transfer the fund urgently without delay into your account and also wish to relocate to your country due to the poor condition in Benin, as to enable me continue my education as I was a medical student before the sudden death of my parent`s. Reply to my alternative email:missnaomisurugaba2@hotmail.com, Your immediate response would be appreciated.
Remain blessed,
Miss Naomi Surugaba.

Фиг. 13.2 Пример за Nigerian scam (източник Интернет)

Може да се обобщи, че към момента най-честите атаки към електронната поща са:

- SPAM – изпращане на нежелани съобщения;
- E-mail hijacking – нерегламентирана употреба (кражба) на нечия електронна поща;
- DoS.

SMTP и POP3 защита с EFW CE

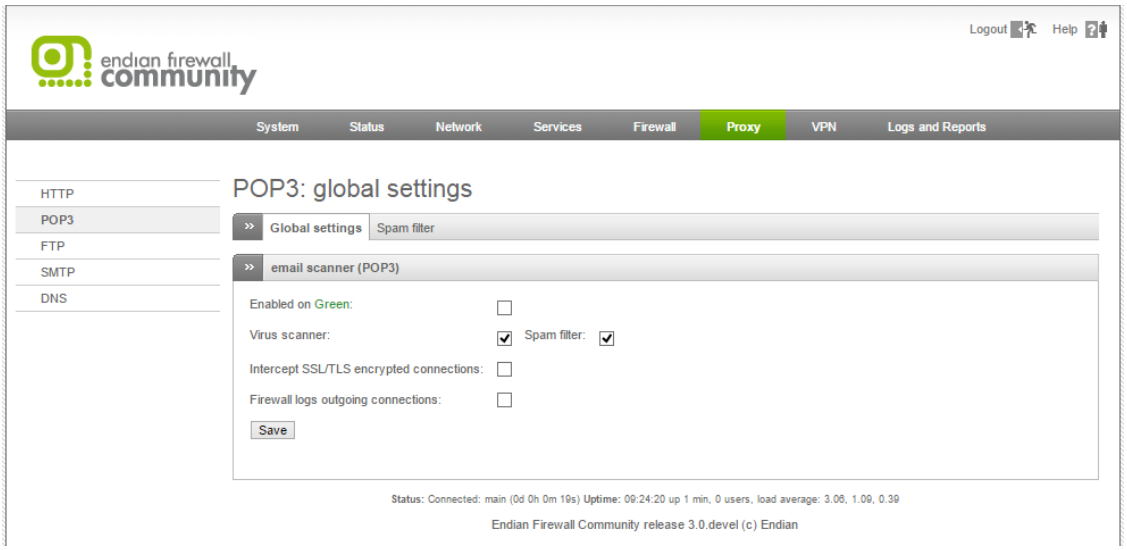
EFW е UTM система, която позволява да се изгради надеждна защита на кореспонденцията с електронна поща. За целта са интегрирани два специализирани прокси сървъра – един за POP3 и втори за SMTP протокола.

POP3 прокси се базира на SpamAssassin[87] един от най-популярните SPAM филтри с отворен код, който към момента е част от Apache Foundation.

Като недостатък на EFW, свързан с анализа и предпазването на електронната поща може да се посочи, че в момента не се поддържа инспектиране чрез прокси на протокола IMAP.

[87] spamassassin.apache.org

Настройките на двата прокси сървъра се извършват от менюто “Proxy” и съответно подменютата “POP3” и “SMTP”.



Фиг. 13.3 POP3 прокси при EFW CE

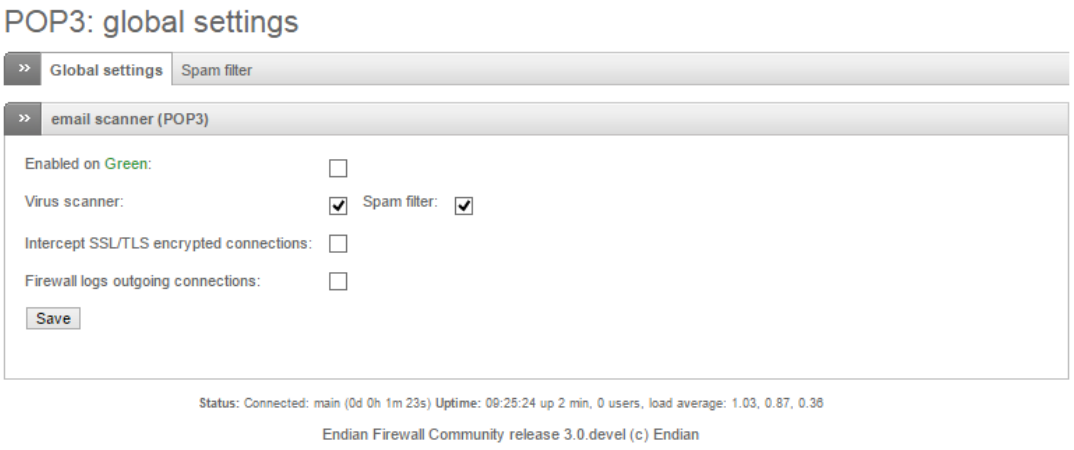
POP3 прокси

Страницата за конфигурирането на POP3 прокси сървъра е разделена на две части:

1. Global settings – основни настройки;
2. Spam filter – настройки на филтрирането на SPAM съобщенията.

Базовите конфигурационни параметри са:

- Enabled on Green (Blue/Orange) – активиране на проверката на електронната поща на съответния интерфейс, като зоните могат да бъдат избрани само ако са конфигурирани;
- Virus scanner – активиране на проверка за зловреден код в електронните писма;
- Spam filter – филтриране за SPAM;
- Intercept SSL/TLS encrypted connections – проверка за зловреден код на връзки, осъществени през SSL/TLS;
- Firewall logs outgoing connections – добавяне на информация в журнала за изходящите връзки, свързани с обмяната на електронни писма.

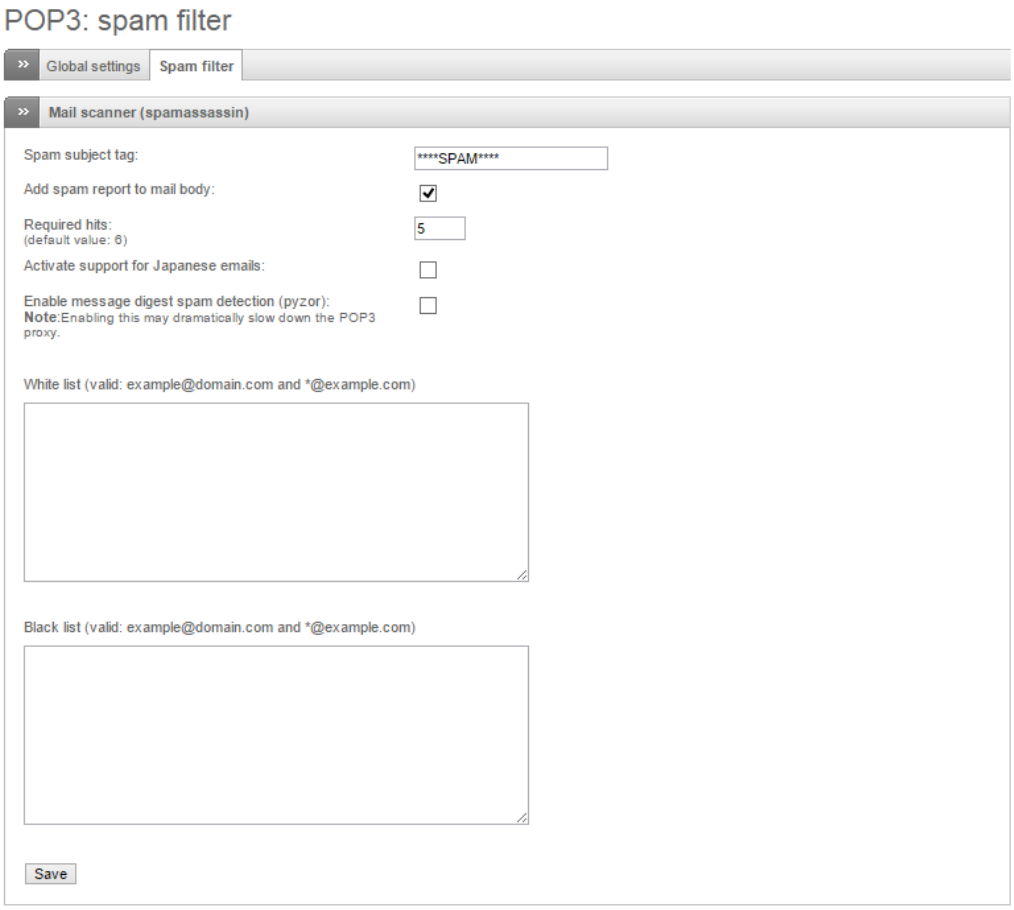


Фиг. 13.4 Основни настройки на POP3 прокси при EFW CE

Втората група с функции при POP3 прокси сървъра включва настройки за управление на SPAM филтъра:

- Spam subject tag – посоченият текст ще бъде добавен към темата на писмото, ако то е класифицирано като SPAM;
- Add spam report to mail body – ако тази опция е активна, съдържанието на писмото ще бъде заменено от доклада на SpamAssassin (информация която описва защо филтъра е определил даденото писмо да попада в категория SPAM и др.);
- Required hits – броят на оценките на SpamAssassin, който ако е надвишен, писмото попада в категорията SPAM. Стойността по подразбиране е 6;
- Activate support for Japanese emails – поддръжката на кореспонденция на японски език;
- Enable message digest spam detection (pyzor[88]) – активиране на анализа на съобщенията чрез pyzor – генериране на уникални данни, които позволяват по-лесен анализ на последващи подобни съобщения. Активирането на тази опция може значително да редуцира производителността на POP3 прокси функциите;
- White list – списък с адреси или домейни, които не се анализират за SPAM;
- Black list – списък с адреси или домейни, които винаги се считат, че генерират SPAM съобщения.

Направените настройки се записват от бутна “Save”.



Фиг. 13.5 Настройки на SPAM филтриране за POP3 прокси при EFW CE

[88] pyzor.readthedocs.org/en/release-1-0-0/

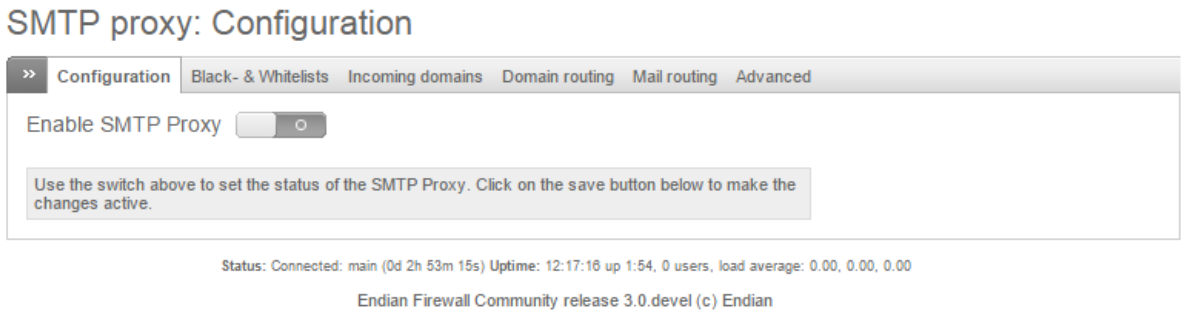
SMTP прокси

SMTP прокси функциите позволяват да се анализира трафика на електронната поща, когато писмата се изпращат от клиента към сървъра (обратна функционалност спрямо POP3 прокси). Основната цел на този сървър е да се оптимизира SMTP трафика, като се предпазят потребителите от потенциални заплахи и атаки.

Конфигурирането се извършва от менюто “Proxy” и подменюто “SMTP proxy”, а страницата е разделена на следните части:

- Configuration – основни настройки;
- Black- & Whitelist – филтриращи списъци;
- Incoming domains – входящи домейни;
- Domain routing – маршрутизиране на трафика между домейни;
- Mail routing – пренасочване на електронните писма;
- Advanced – разширени настройки.

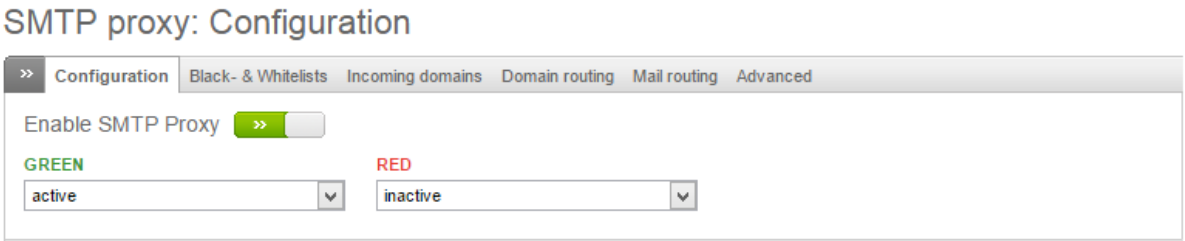
Активирането на SMTP прокси процеса е чрез преместване на бутона “Enable SMTP Proxy” в активна позиция.



Фиг. 13.6 SMTP прокси при EFW CE

След активиране на SMTP прокси е необходимо да се посочи неговото функционално състояние на определените интерфейси, което може да бъде:

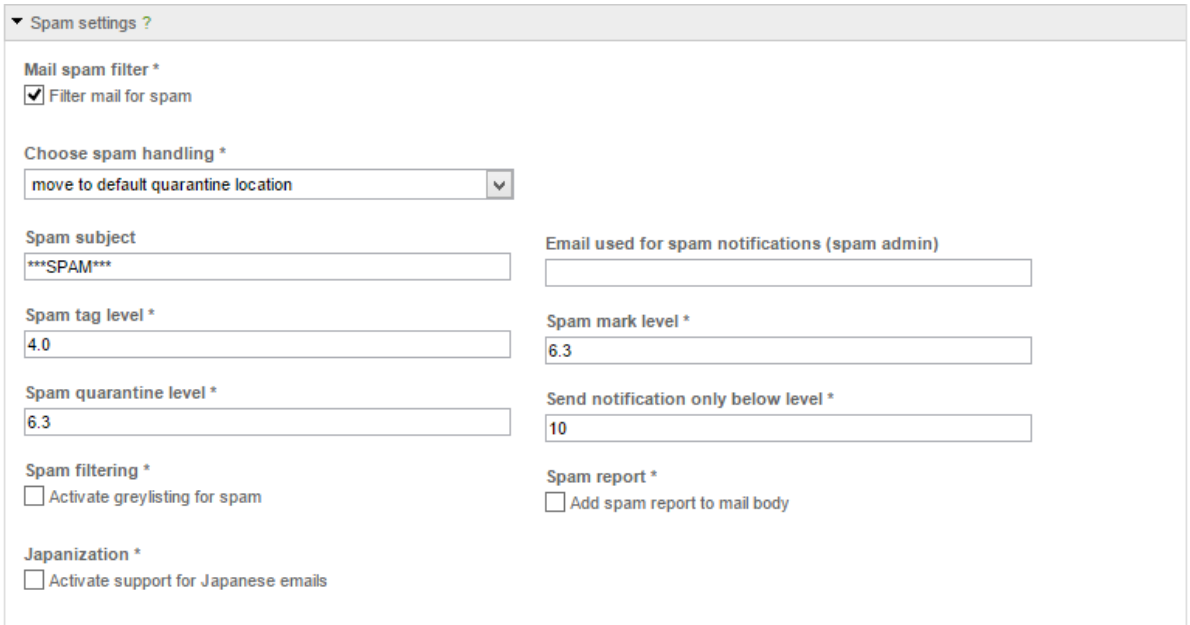
1. Active – функциите следят за постъпващи заявки на TCP порт 25. Необходимо е да се направи и промяна в конфигурацията на софтуерните клиенти, използвани от потребителите;
2. Transparent – заявките на SMTP порта се прихващат и препращат към посочения сървър, но трафика се анализира. Тази функционалност не може да се използва на червения (RED) интерфейс;
3. Inactive – SMTP прокси не се използва на дадения интерфейс.



Фиг. 13.7 Активиране на SMTP прокси на определен интерфейс на EFW CE

Настройките на функциите на EFW да разпознават SPAM се задават от частта “Spam setting” и съдържат:

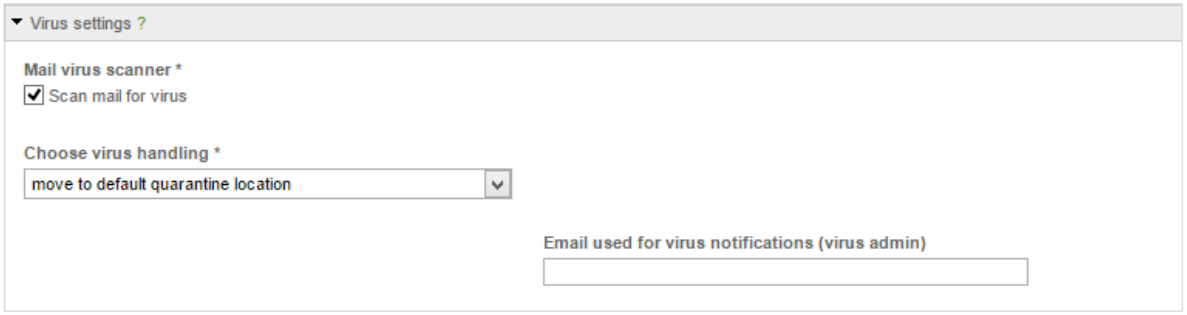
- Filter mail for spam – активиране или деактивиране на сканирането на електронните писма за SPAM;
- Choose spam handling – избор на действието, което ще се извърши, ако писмото е определено като SPAM. Първият вариант за действие е “move to default quarantine location”, което премества писмото мястото за карантина, което е зададено по подразбиране. “Send to quarantine email address” препраща писмото към специален адрес за получаване на SPAM трафика. Опцията “mark as spam” маркира писмото като SPAM и го пренасочва към следващата система. Последното възможно действие е “drop email”, което незабавно изтрива писмото;
- Spam Subject – текст (префикс), който се добавя преди темата на писмото, ако след филтрирането то е оценено като SPAM;
- Email used for spam notifications (spam admin) – адрес за електронна поща, към който ще се изпращат данни за всяко писмо, което е класифицирано като SPAM;
- Spam tag level – ако оценката от SpamAssassin надвишава тази стойност се добавят хедърите X-Spam-Status и X-Spam-Level към оригиналното писмо;
- Spam mark level - ако оценката от SpamAssassin надвишава тази стойност към оригиналното писмо се добавят хедърите “Spam subject” и X-Spam-Flag;
- Spam filtering – активира или деактивира използването на т.нар “spam graeylist”. Ако тази функционалност се включи е необходимо да се направят и допълнителни настройки на съответния списък;
- Spam report – ако опцията е активна към съдържанието на писмото, което е класифицирано като SPAM се добавя и доклад;
- Japanization – активира или деактивира функциите за анализ на електронна поща на японски език.



Фиг. 13.8 SMTP прокси – настройки за анализ на SPAM

EFW позволява да се извършва сканиране на SMTP трафика за наличие на зловреден код, като настройките на тази функционалност се извършват от частта “Virus settings”:

- Scan mail for virus – активира или деактивира сканирането за вируси и друг зловреден код;
- Choose virus handling – избор на действие, което да се извърши, ако е открит зловреден код. Първата възможност е “move to default quarantine location” – писмото се препраща към карантинната зоната по подразбиране. При “send to quarantine email address” писмото се препраща към карантинната електронна поща. Опцията “pass to recipient (regardless of bad contents)” пренасочва писмото към получателя, аналогично на останалата кореспонденция. Последната възможност е “drop email”, при която писмото се изтрива.
- Email used for virus notifications (virus admin) – адрес за електронна поща, използван за уведомяване за открити писма със зловреден код.

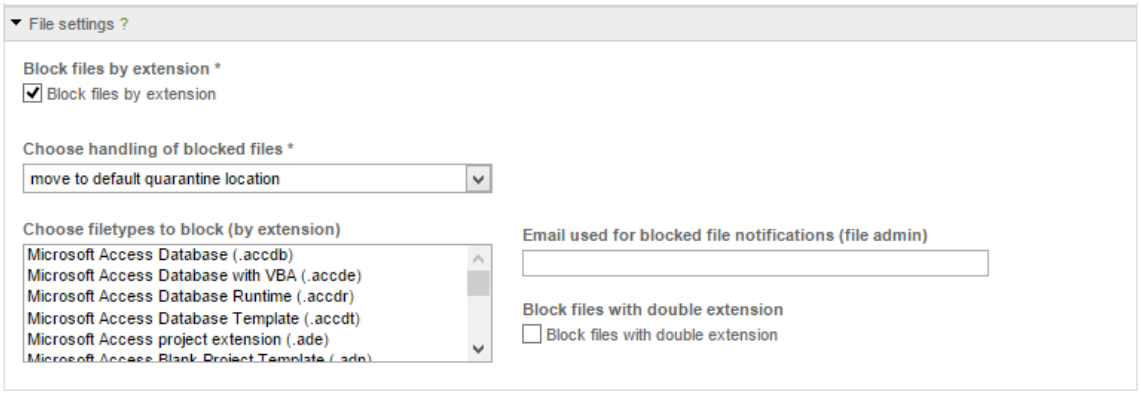


Фиг. 13.9 Проверка на SMTP трафика за наличие на зловреден код

Групата “File settings” включва настройки, свързани с проверката на прикачените файлове към електронната поща.

Конфигурационните параметри са:

- Block files by extension – ако тази опция е активна се извършва допълнително сканиране на файловете, които са част от електронно писмо, единствено ако тяхното разширение е включено в списъка;
- Choose handling of blocked files – определя извършваното действие, което зависи от типа на файла. Възможните параметри са “move to default quarantine location”, който пренасочва писмото към карантинното местоположение по подразбиране, “send to quarantine email address” – изпраща писмото към специален адрес за електронна поща, “pass to recipient” – писмото се пренасочва към получателя.
- Choose filetypes to block (by extension) – списък с файлови разширения, които да се блокират;
- Email used for blocked file notifications (file admin) – адрес за електронна поща, на който да се изпращат блокираните писма;
- Block files with double extension – блокиране на файлове, които имат две разширения, например “funny_cat.jpg.exe”.



Фиг. 13.10 Проверка на SMTP прикачени файлове за наличие на зловреден код

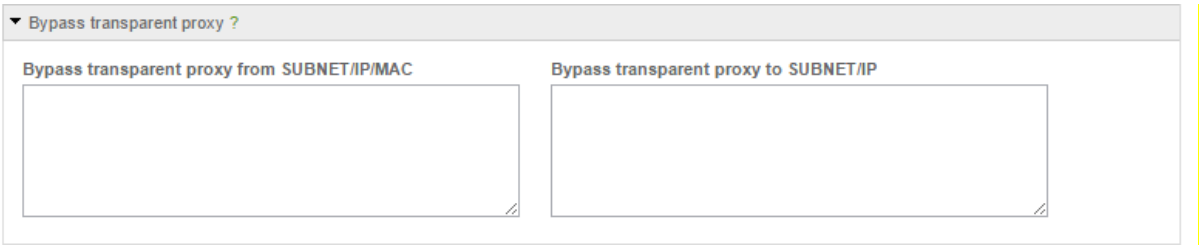
В секцията “Quarantine settings” може да се конфигурира една единствена опция “Quarantine retention time (in days)”, която определя интервала в дни, през който дадено писмо ще бъде съхранявано в съответното местоположение за карантина.

След изтичане на този период писмото се изтрива.

Фиг. 13.11 Настройки на карантината при EFW CE

Последната група от конфигурационните параметри съдържа два списъка:

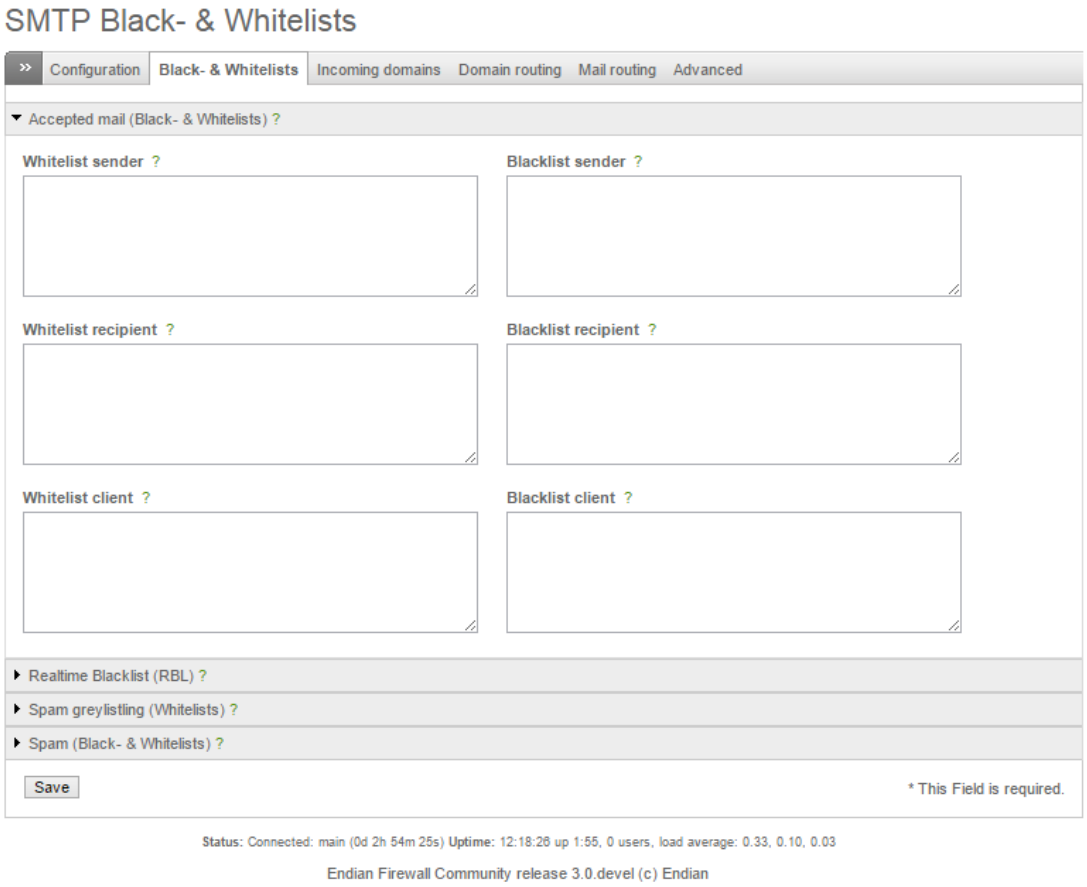
- Bypass transparent proxy from SUBNET/IP/MAC – включените в списъка мрежи, IP или MAC адреси няма да бъдат проверявани от прозрачното SMTP прокси;
- Bypass transparent proxy to SUBNET/IP – електронна поща, изпратена към посочените мрежи или адреси няма да бъде проверявана от прозрачното SMTP прокси.



Фиг. 13.12 Списъци за “заобикаляне” на прозрачното прокси

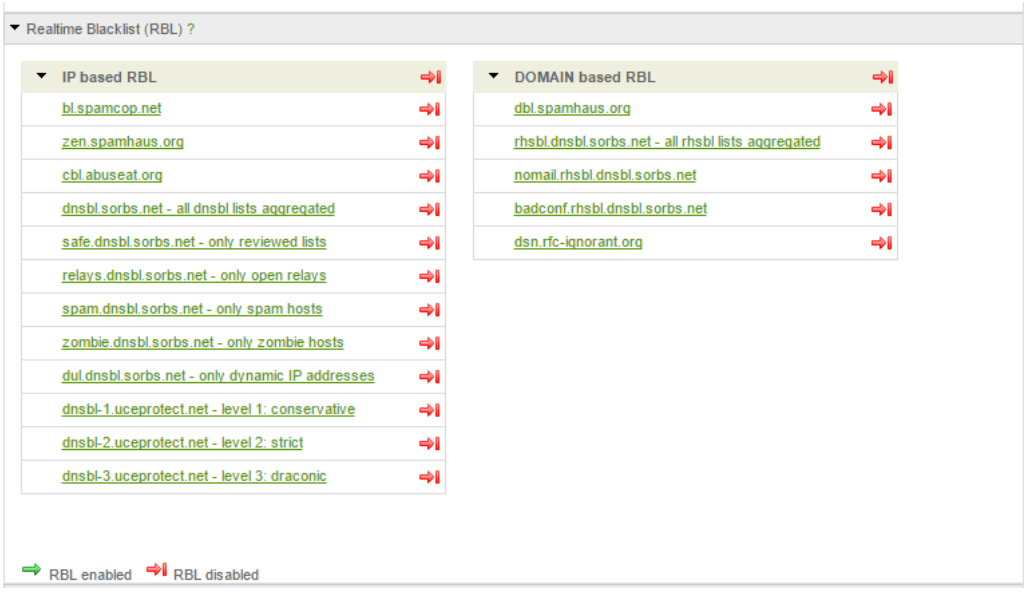
Втората група функции на страницата “SMTP proxy” е “Black- & Whitelists”, от където е възможно да се създадат списъци за блокиране (blacklist) и разрешаване (whitelist) на потоци от електронни писма. В първата част “Accept mail (Black- & Whitelists)” могат да се зададат параметрите:

- Whitelist sender – списък с разрешени изпращачи;
- Blacklist sender – списък със забранени изпращачи;
- Whitelist recipient – списък с разрешени получатели;
- Blacklist recipient – списък със забранени получатели;
- Whitelist client – списък с разрешени клиенти;
- Blacklist client – списък със забранени клиенти.



Фиг. 13.13 Списъци за блокиране и разрешаване на изпращачи и получатели на електронна поща

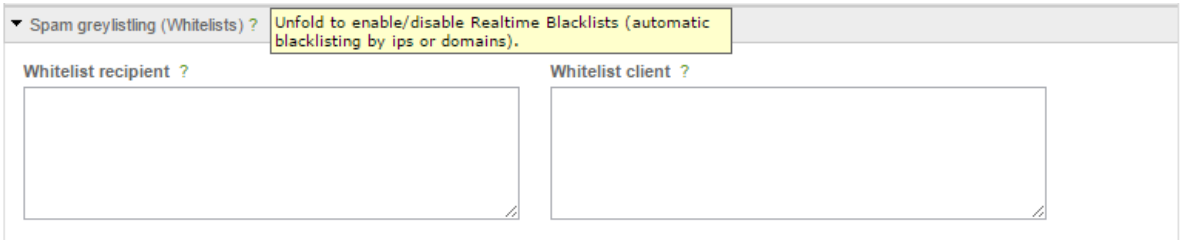
Един често използван подход при борбата със SPAM е използването на специално разработени списъци за блокиране и разрешаване, които се обновяват и поддържат от редица компании и организации. В частта “Realtime Blacklist (RBL)” могат да бъдат посочени кои списъци да се използват за анализ по IP адреси или по домейни. Активирането или деактивирането се извършва от иконата със стрелка след името на съответния списък.



Фиг. 13.14 Списъци за блокиране и разрешаване на изпращачи и получатели на електронна поща в реално време

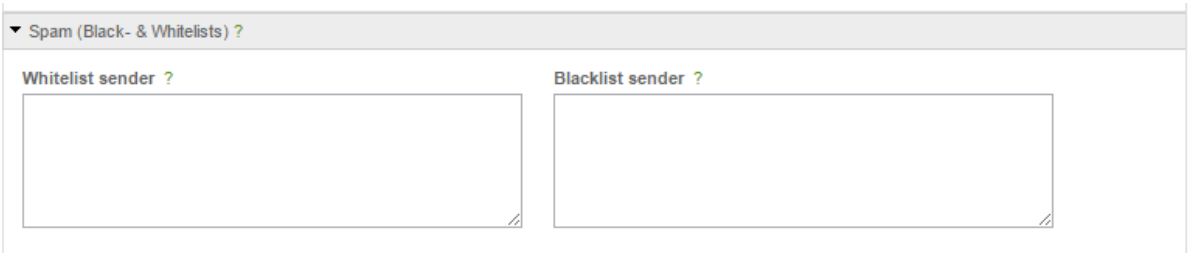
“Greylisting” е подход, който се използва при пренасянето на електронна поща, при който първоначално писмото се отхвърля и се изчаква повторна заявка от същия адрес. Ако писмото не се изпрати повторно, то се счита за SPAM. Идеята е, че повечето SPAM приложения не изпращат едни и също писмо двукратно.

Конфигурацията на “Spam greylistng (Whitelists)” съдържа два списъка – един с разрешени получатели и втори с разрешени клиенти.



Фиг. 13.15 Конфигуриране на “Greylisting” при EFW CE за защита от SPAM

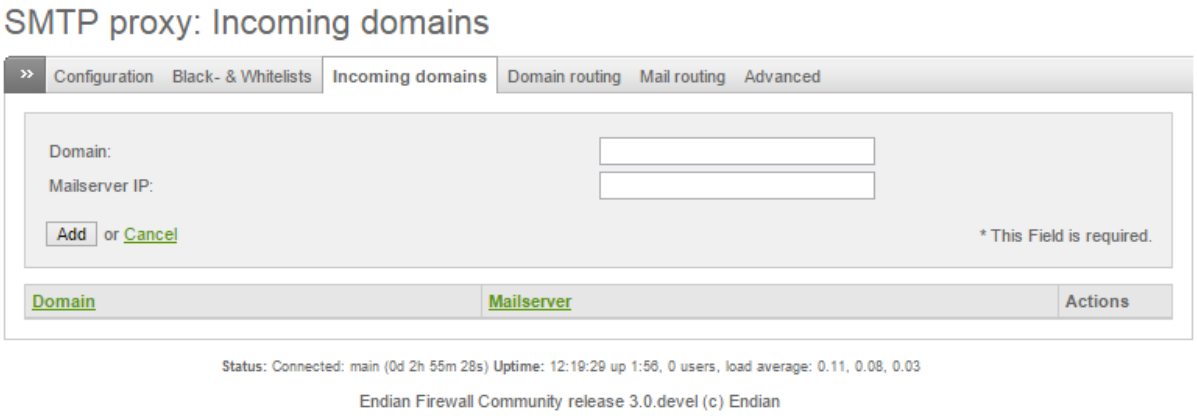
Втората част на страницата “Spam greylistng (Whitelists)” включва два списъка с винаги разрешени и блокирани изпращачи.



Фиг. 13.16 Списъци с блокирани и разрешени изпращачи при анализ на SPAM

Ако са добавени домейни в “Incoming domains”, отдалечени системи, които са свързани към червения интерфейс (RED) могат да използват SMTP сървър, намиращ се в мрежовия сегмент на определен интерфейс на EFW (най-често в DMZ зоната) за изпращане на електронна поща. Параметрите, които трябва да бъдат зададени са:

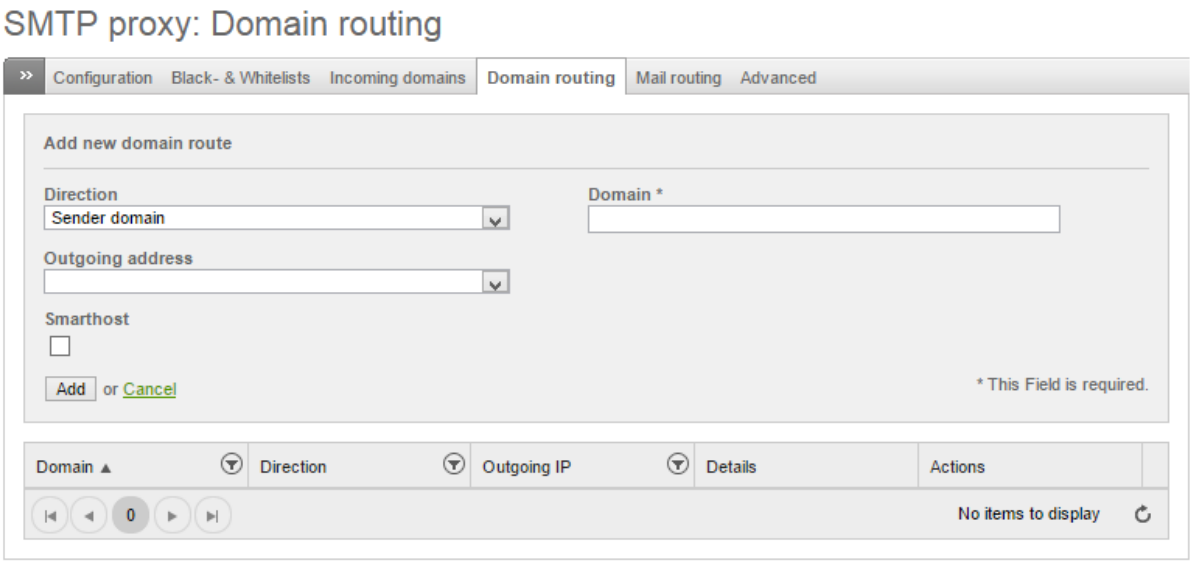
- Domain – домейна, за който отговаря сървъра за електронна поща;
- Mailserver IP – IPv4 адрес на пощенския сървър.
  Добавянето на настройките се извършва от бутона “Add”.



Фиг. 13.17 Конфигуриране на “incoming domainя” при SMTP прокси

“Domain routing” е нова функционалност, която е налична от EFW версия 3.0. Тя позволява да се конфигурира пренасочване на домейни, като конфигурационните параметри са:

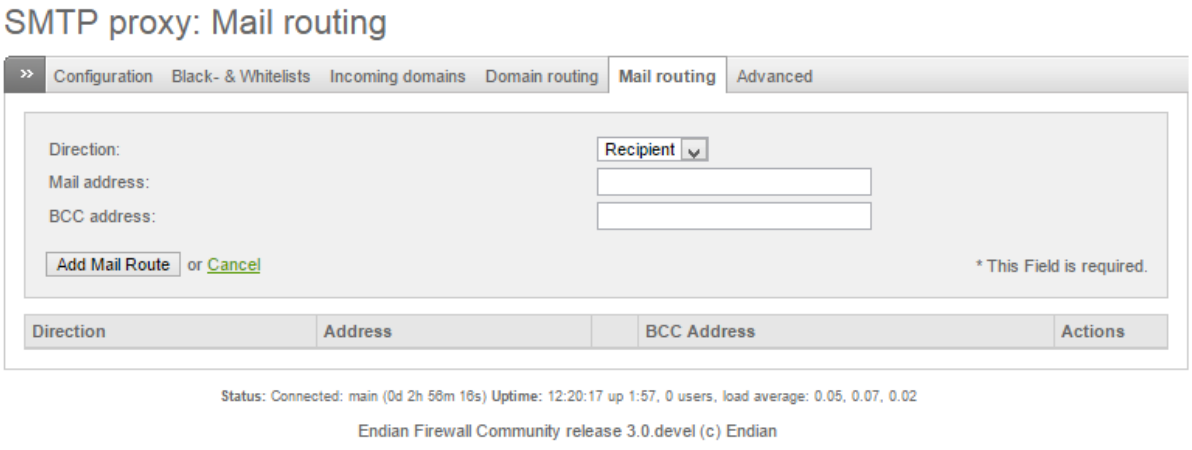
- Direction – посока на пренасочване, която се определя на базата на получател или на изпращач;
- Domain – домейн, за който ще се използва посочения сървър;
- Outgoing address – адрес на сървъра, през който ще се препращат електронните писма;
- Smarthost – специално устройство, което позволява пренасочване на трафика на електронните писма към междинна система, а не директно към следващия SMTP сървър.



Фиг. 13.18 Конфигуриране на пренасочване на електронната поща (domain routing) при EFW CE

Настройките в страницата “Mail routing” позволяват да се изпрати BCC към посочен адрес, като това се отнася или за всички писма или за точно определени изпращачи. Конфигурационните параметри са:

- Direction – определя типа на пътя, като изпращащ или приемащ;
- Mail address – в зависимост от типа на пътя това е или адрес на получателя или адрес на изпращача;
- BCC address – адрес на електронна поща, където ще се пренасочват BCC писмата.



Фиг. 13.19 Конфигуриране на автоматично изпращане на BCC (mail routing) при EFW CE

Допълнителните (разширени) настройки на SMTP прокси сървъра са включени в страницата “Advanced” и са групирани в четири части. В първата част “Smarthost configuration” може да се посочи дали да се използва smarthost при преноса на електронните писма, като трябва да се конфигурира:

- Smarthost for delivery – активира използването на smarthost;
- Smarthost address – адреса на smarthost системата;
- Smarthost port – порта, на който работи smarthost процеса на отдалеченото устройство;
- Smarthost authentication – посочва дали се използва автентификация;
- Smarthost username – потребител;
- Smarthost password – парола;
- Choose authentication method – метод за автентификация.

Втората група е “IMAP Server for SMTP authentication”, в която са включени необходимите параметри, ако сървъра ще използва механизъм за автентификация чрез IMAP при изпращането на електронните писма. Параметрите, които могат да бъдат зададени са:

- SMTP authentication – активира или деактивира автентификацията на SMTP сървъра;
- Choose number of authentication daemons –дефинира броя на паралелните включвания през EFW системата;
- IMAP authentication server – IP адрес на IMAP сървър;
- IMAP authentication port – port на IMAP сървър.

Третата група с параметри е “Mail server setting”, в която са включени:

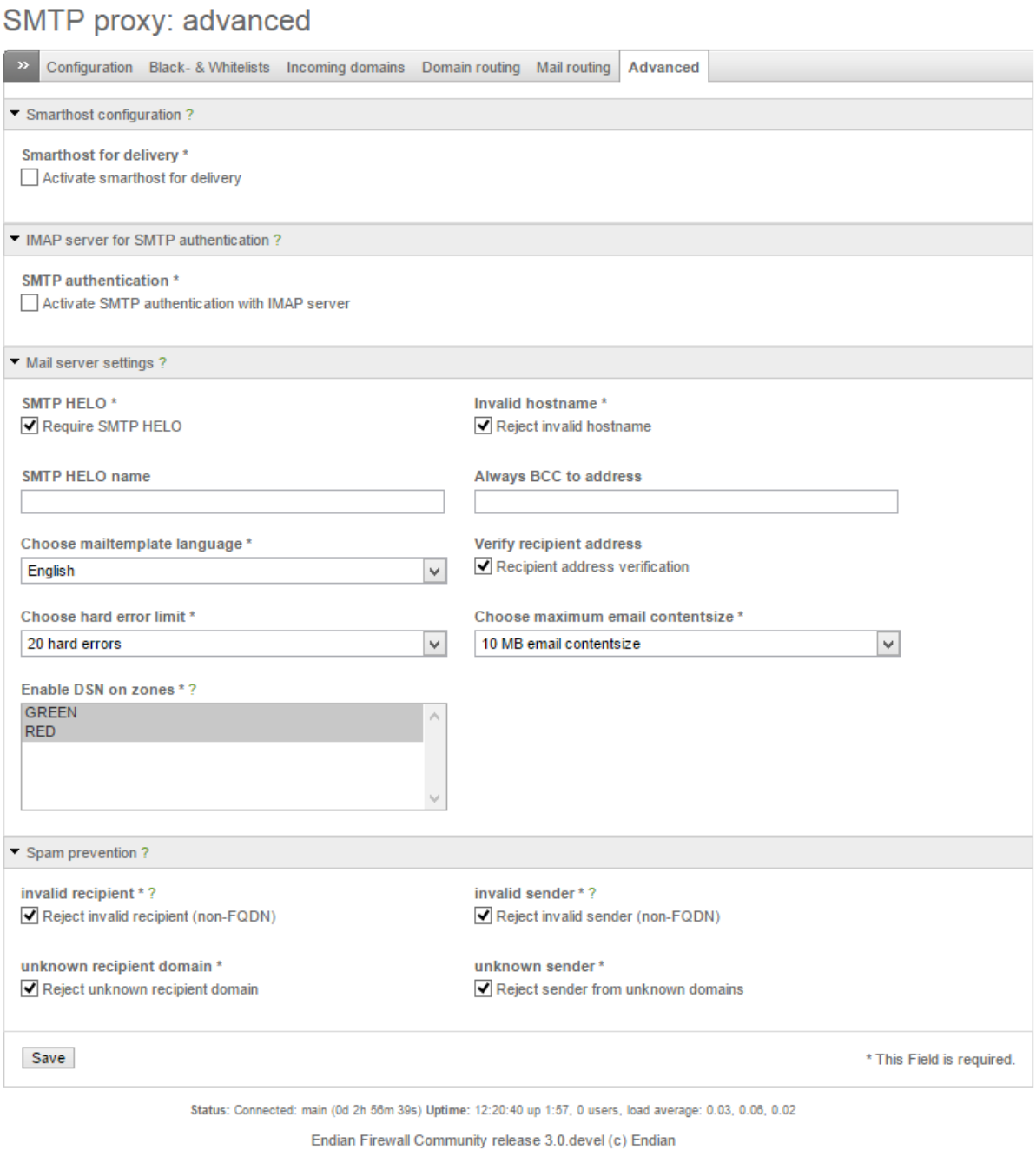
- SMTP HELO – ако тази опция е активирана клиентът трябва да изпрати HELO или EHLO преди да се изгради SMTP сесията;
- Invalid hostname – връзката се прекъсва, ако при изпращането на HELO или EHLO е посочен невалиден хост (извършва се допълнителна проверка);
- SMTP HELO name – името на хоста, което се изпраща при HELO или EHLO;
- Always BCC to address – винаги се извършва препращане (BCC) към посочения адрес;
- Choose mailtemplate language – избор на език за шаблоните за електронните писма;
- Verify recipient address – посочва дали преди да се изпрати писмото се извършва проверка за валидност на адреса на получателя;
- Choose hard error limit – максимално допустим брой грешки преди да се прекъсне връзката;
- Choose maximal email contentsize – посочва максималния размер на съдържанието на електронната поща;
- Enable DSN on zones – избор на зони, но които се изпращат DNS съобщения при проблеми с доставката на електронна поща.

Последните конфигурационни параметри са включени в групата “Spam prevention” и съдържат:

- Invalid recipient – заявката се отхвърля, ако RCPT TO адресът не е FQDN, което се изисква от RFC 821[89];

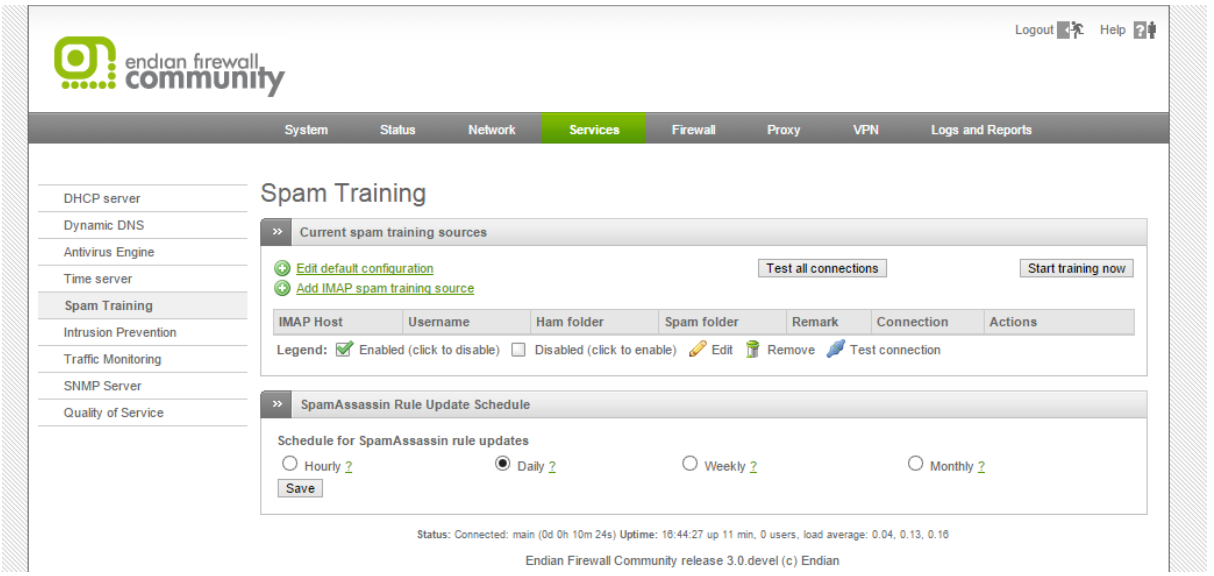
[89] james.apache.org/server/rfclist/smtp/rfc0821.txt

- Invalid sender – заявката се отхвърля, ако името на хоста, което е изпратено с HELO или EHLO не е FQDN (RFC 821);
- Unknown recipient domain – заявката се отхвърля, ако домейна на получателя няма валиден A или MX DNS запис;
- Unknown sender – заявката се отхвърля, ако домейна на изпращача няма валиден A или MX DNS запис.



Фиг. 13.20 Разширени настройки на SMTP прокси при EFW CE

EFW използва SpamAssassin като основен инструмент за филтриране на SPAM писмата. Въпреки, че има висока степен на точно класифициране по някога е необходимо SpamAssassin да бъде допълнително ”обучен” да разпознава нежеланата кореспонденция. Допълнителните настройки, свързани с работата на SpamAsssasin са достъпни от менюто “Services” и подменюто “Spam Training”.



Фиг. 13.21 Допълнително обучение на SpamAssassin при EFW CE

Базовите параметри на обучението на SpamAssassin са достъпни от връзката “Edit default configuration” и включват:

- Default IMAP host – IMAP хост, на който са достъпни папките, съдържащи данните за обучението;
- Default username – потребителско име за достъп до IMAP сървъра;
- Default password – парола за достъп до IMAP сървъра;
- Default ham folder – име на директория, която съдържа нормални писма (не SPAM);
- Default spam folder – име на директория, в която има единствено SPAM електронни писма;
- Schedule an automatic spam filter training – дефинира интервала от време между две последователни автоматични обучения. Възможните параметри са ежечасно, ежедневно, всяка седмица или на всеки месец.

При натискане на бутонът “Test all connections” се извършва проверка на всички направени настройки, свързани с обучението на SpamAssassin. Стартирането на процеса на обучение е чрез изчакване на следващия планиран момент от време или при натискане на бутона “Start training now”.

Добавянето на допълнителни IMAP източници е възможно след натискане на “Add IMAP spam training source”.

Необходимо е да бъдат конфигурирани параметрите:

- Enabled – активира или деактивира използването на конфигурирания източник;
- Remark – опционален коментар;
- IMAP host – IMAP хост, на който са достъпни папките, съдържащи данните за “обучението”;
- Delete processed mails – ако тази опция е активна, след приключване на анализа електронните писма ще бъдат изтрити;
- Username – потребителско име за достъп до IMAP сървъра;
- Password – парола за достъп до IMAP сървъра;

- Ham folder – име на директория, която съдържа нормални писма (не SPAM);
- Spam folder – име на директория, в която има единствено SPAM електронни писма.

Добавянето на източника за обучение е от “Add Training Source”. Аналогично на повечето менюта при EFW направените комбинации от настройки са описани в табличен вид, а от колоната “Actions” те могат да бъдат активирани или деактивирани, редактирани изтривани и др.

Заключение

Активирането на POP3 и SMTP прокси функциите на EFW води до по-висока степен на защита на електронните писма и редуцира нежелания SPAM.

При необходимост SpamAssassin може да бъде допълнително “обучен” да разпознава по-прецизно SPAM кореспонденцията.

Глава 14. Наблюдение и създаване на отчети

Наблюдението на работата на мрежовите системи и периодичното анализиране на журналите (log) е задължително за да се гарантира правилната работа на устройствата и максималната защита на трафика и потребителите. EFW CE предоставя възможност за обобщено наблюдение през менюто “System” (разгледани по-рано), както и задължителните журнали.

Важно е да се отбележи, че някои функции могат да генерират голям обем от записи в журналните файлове или бази данни, което изисква да се използват автоматизирани подходи за тяхното филтриране и периодично архивиране..

Older Newer

Time	data
Feb 6 12:17:58	clamd[6430]: clamd daemon 0.97.8 (OS: linux-gnu, ARCH: i386, CPU: i586)
Feb 6 12:17:58	clamd[6430]: Running as user clamav (UID 1001, GID 108)
Feb 6 12:17:58	clamd[6430]: Log file size limited to 2097152 bytes.
Feb 6 12:17:58	clamd[6430]: Reading databases from /var/signatures/clamav
Feb 6 12:17:58	clamd[6430]: Not loading PUA signatures.
Feb 6 12:17:58	clamd[6430]: Bytecode: Security mode set to "TrustSigned".
Feb 6 12:18:03	clamd[6430]: Loaded 1096223 signatures.
Feb 6 12:18:04	clamd[6430]: TCP: Bound to address 127.0.0.1 on port 3310
Feb 6 12:18:04	clamd[6430]: TCP: Setting connection queue length to 30
Feb 6 12:18:04	clamd[6545]: Limits: Global size limit set to 52428800 bytes.
Feb 6 12:18:04	clamd[6545]: Limits: File size limit set to 31457280 bytes.
Feb 6 12:18:04	clamd[6545]: Limits: Recursion level limit set to 5.
Feb 6 12:18:04	clamd[6545]: Limits: Files limit set to 1000.
Feb 6 12:18:04	clamd[6545]: Archive support enabled.
Feb 6 12:18:04	clamd[6545]: Algorithmic detection enabled.
Feb 6 12:18:04	clamd[6545]: Portable Executable support enabled.
Feb 6 12:18:04	clamd[6545]: ELF support enabled.
Feb 6 12:18:04	clamd[6545]: Mail files support enabled.
Feb 6 12:18:04	clamd[6545]: Mail: RFC1341 handling enabled.
Feb 6 12:18:04	clamd[6545]: OLE2 support enabled.

Фиг. 14.1 Съдържание на журнал при EFW CE

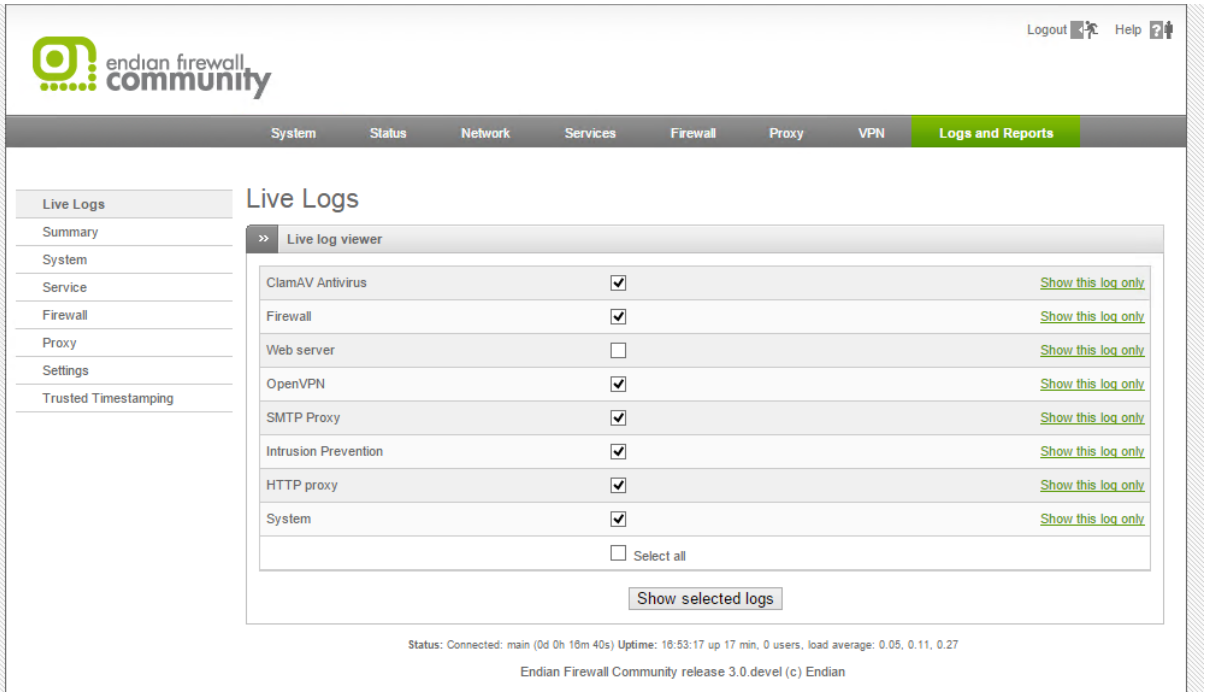
Журнали и доклади

Журналите и докладите при EFW CE са достъпни от менюто “Logs and Reports”, което включва следните подменюта:

- Live logs – визуализиране в реално време на данни от журналите;
- Summary – обобщена информация от журналите;
- System – данни, свързани с работата на EFW системата;
- Service – информация, генерирана в журналите от някои от най-важните услуги на EFW;
- Firewall – информация получена, по време на работата на защитната стена;
- Proxy – данни от прокси сървърите, които са конфигурирани и работят;
- Settings – настройки на журналите;
- Trusted Timestamping – процес, при който журналните файлове се проверяват дали не са били модифицирани неправомерно.

Първото подменю е “Live Logs”, като от него могат да бъдат преглеждани в реално време следните журнали:

- ClamAV Antivirus;
- Firewall;
- Web server;
- OpenVPN;
- SMTP Proxy;
- Intrusion Prevention;
- HTTP proxy;
- System.

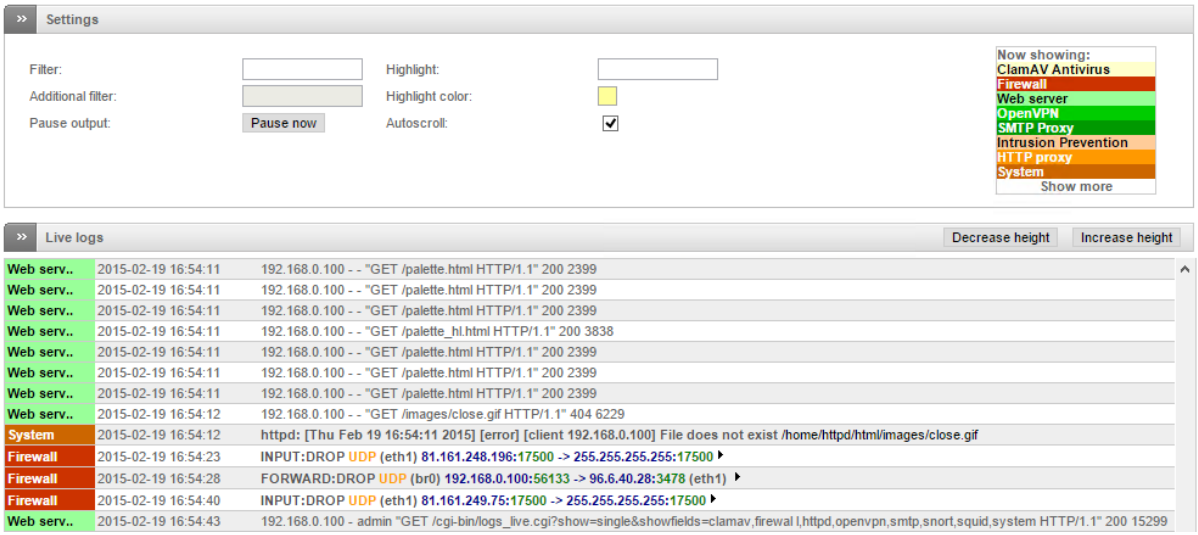


Фиг. 14.2 Подменю “Live Logs”

При отваряне на тази страница е възможно да се посочат кои журнали да се визуализират, като под таблицата е налична и опцията “Select all”, която активира всички възможности едновременно. В дясно от името на журнала е поместена връзка “Show this log only”, при натискането на която се визуализира само посочената категория.

Данните за журналите се визуализират в реално време (с минимално забавяне) в нов прозорец и в табличен вид. Над таблицата в страницата е секцията “Settings” в която може да бъдат посочени филтър, селектиране, цвят на селектиране, както и да се направи временно спиране на обновяването (бутон Pause now). В най-дясната част на тази група е и списък с отделните журнали, които са били избрани и тяхното цветово означение в таблицата.

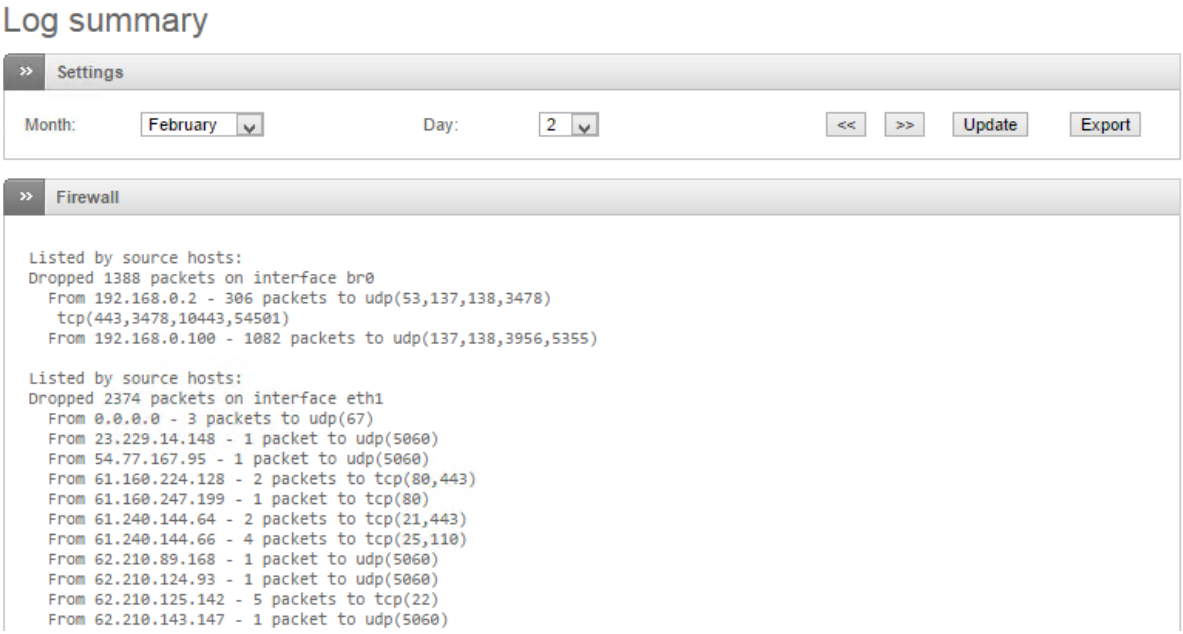
“Live Logs” на EFW е една изключително удобна функционалност, особено при анализиране на работата на защитните функции и на системата.



Фиг. 14.3 Журнални данни за работата на системата, визуализирани в реално време

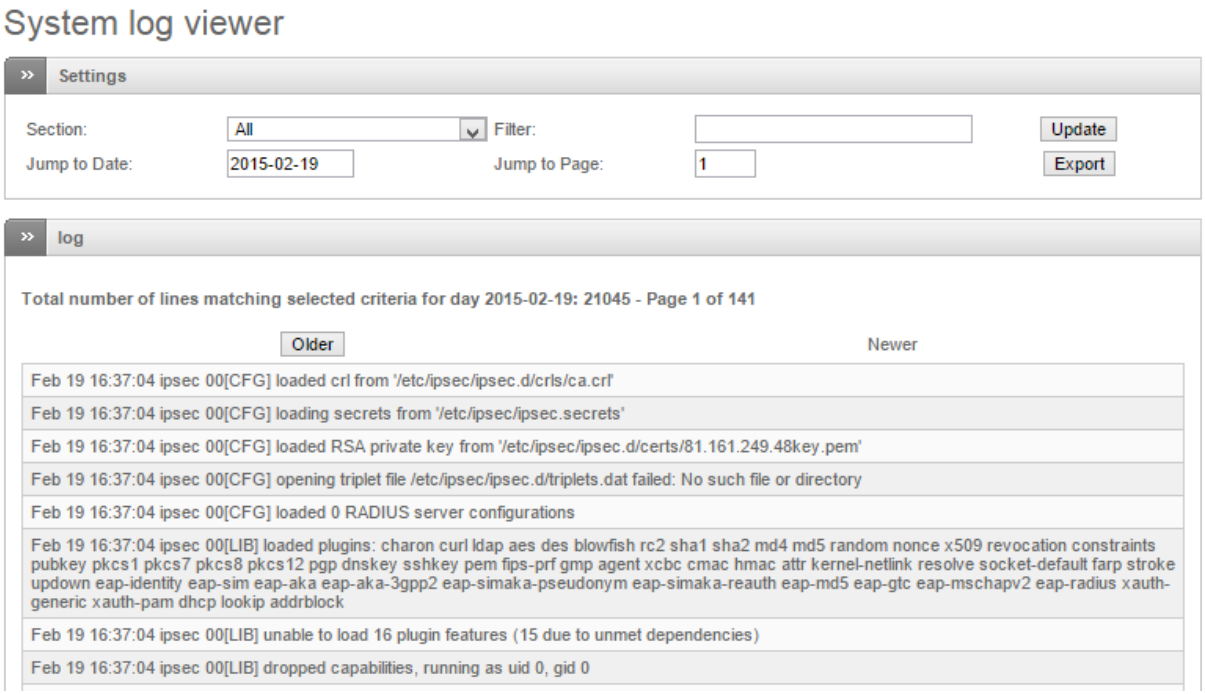
Второто подменю на “Logs and Reports” е “Summary” и в тази страница са включени резултатите от пакета “logwatch”, който анализира журналите. В групата за настройки може да бъде посочен месеца и деня, както и са налични бутони за преместване с един ден напред (>>) и назад (<<).

Бутонът “Update” опреснява данните, а „Export” генерира текстова версия на информацията, която може да бъде съхранена локално.



Фиг. 14.4 Обобщени журнални данни

В подменюто “System” се визуализират данни, свързани с работата на системата. От групата “Settings” може да бъде избрана секцията с информация, да се посочи желаната дата и филтър, както и да се опресни информацията чрез бутона “Update”. Чрез бутонът “Export” данните могат да бъдат съхранени в локален файл.



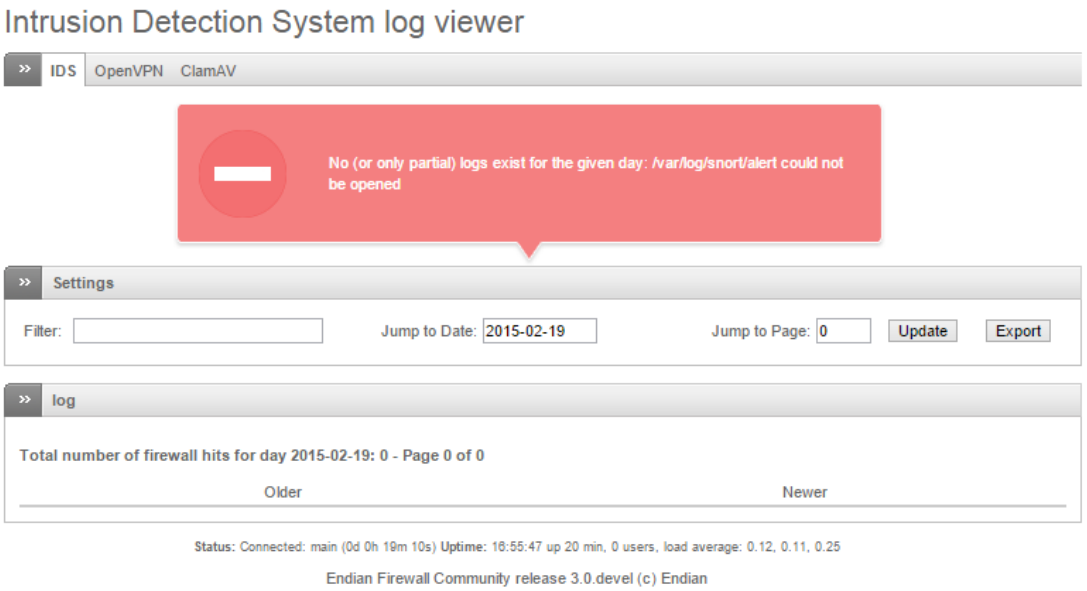
Фиг. 14.5 Системен журнал

От подменюто “Services” могат да бъдат прегледани три от най-важните журнала:

1. IDS;
2. OpenVPN;
3. ClamAV.

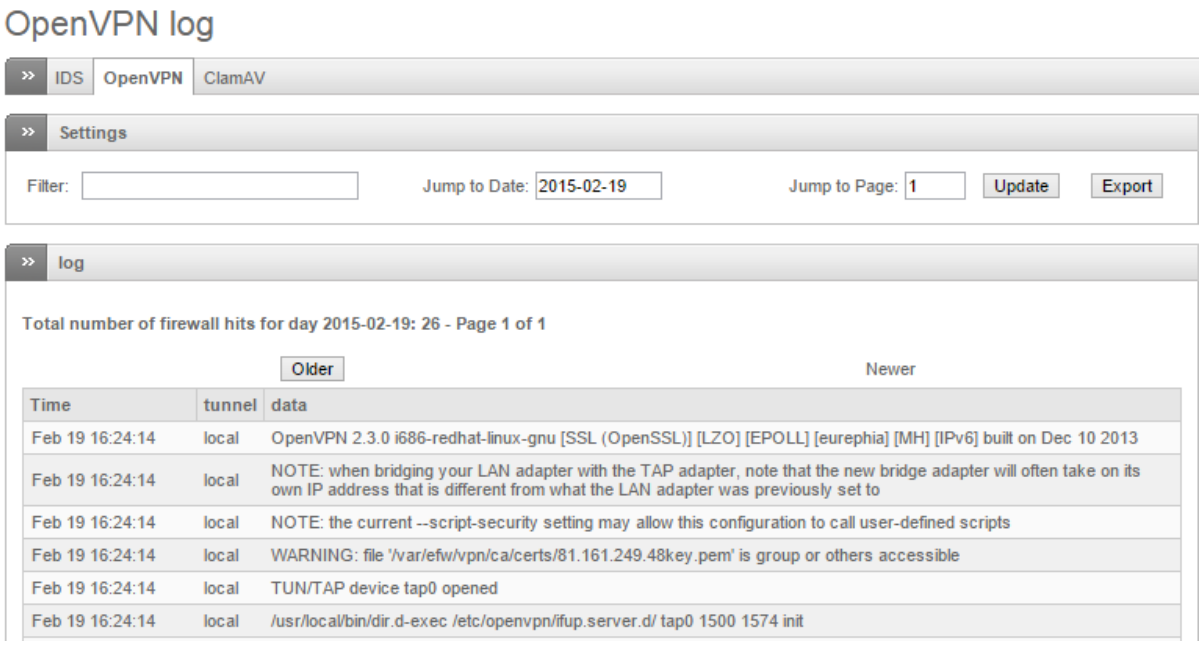
И при трите достъпни стрваници от групата “Settings” е възможно да се избере желаната дата за анализ, да се посочи филтър, както и данните да бъдат опреснени и експортирани.

На фигура 14.6 е показана информацията за IDS, но поради факта, че тази функционалност не е активна на използваната система, журналите са празни и е визуализирано съобщение за липса на генерирани данни.



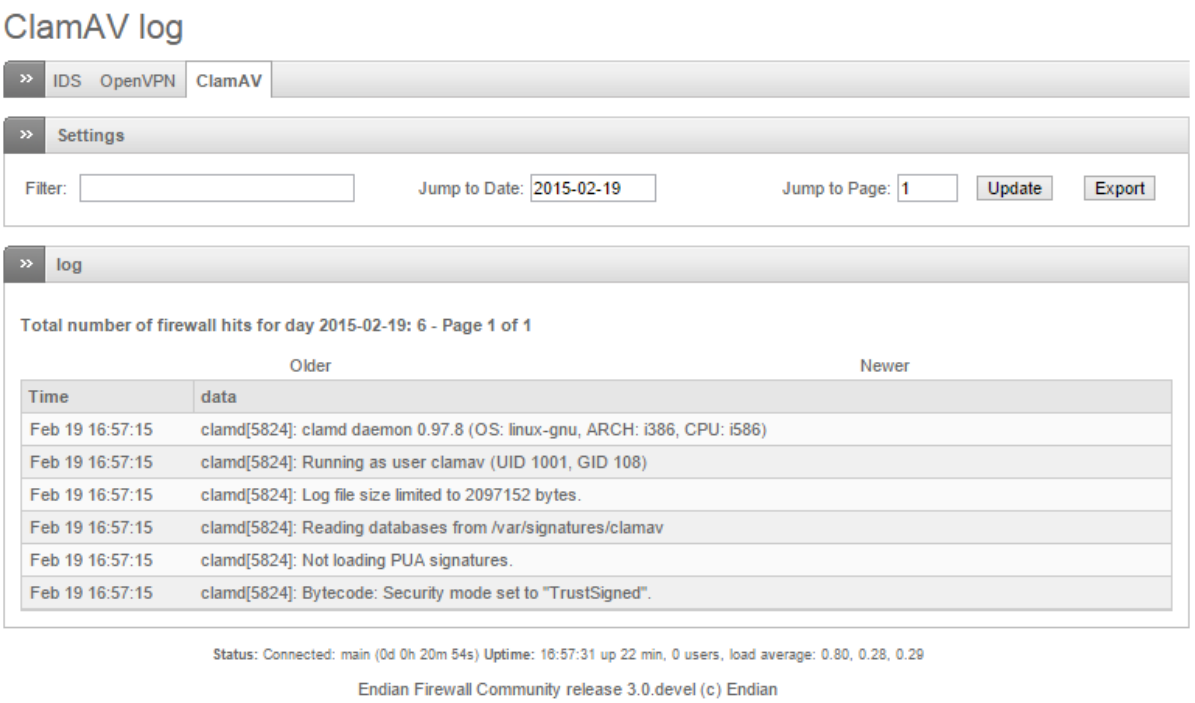
Фиг. 14.6 Журнал IDS (при неактивна IDS/IPS функционалност)

В подстраницата OpenVPN в табличен вид са обобщени данните за работата на OpenVPN сървъра. Препоръчително е информацията да бъде анализирана на база на типа на съобщенията – NOTE, WARNING и др.



Фиг. 14.7 OpenVPN журнал

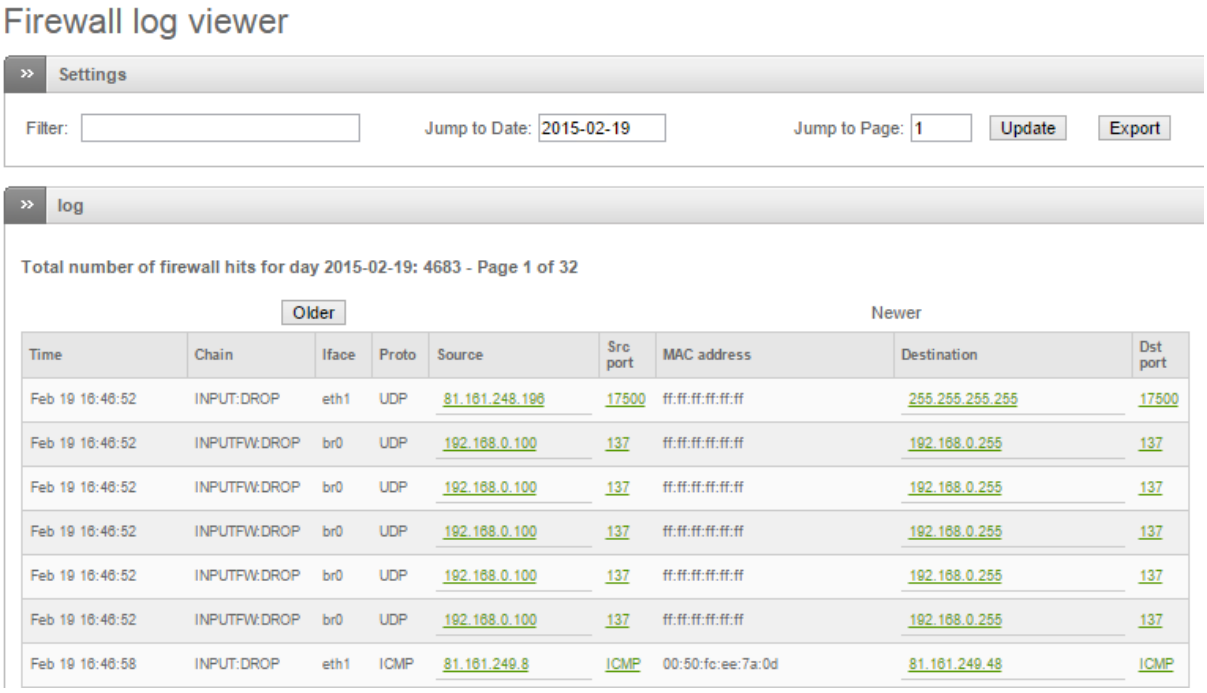
Последният журнал, включен в “Summary” е ClamAV и данните от него съдържат важна информация за работата на антивирусните функции на EFW CE.



Фиг. 14.8 ClamAV журнал

В страницата “Firewall” може да се прегледа информация, свързана с работата на защитната стена. Налични са стандартните функции за избор на дата, филтриране, обновяване и експортиране. В табличен вид са показани следните параметри:

- Time – дата и час на генериране на съобщението;
- Chain – верига (chain) през която е обработен пакета;
- Iface – интерфейс, през който е преминал пакета;
- Proto – пренасян протокол;
- Source – IP адрес на източника;
- Src port – порта на приложението, източник на пакета;
- MAC address – MAC адрес на интерфейса;
- Destination – IP адрес на получателя на пакета;
- Dst port – порт на получаващото приложение;



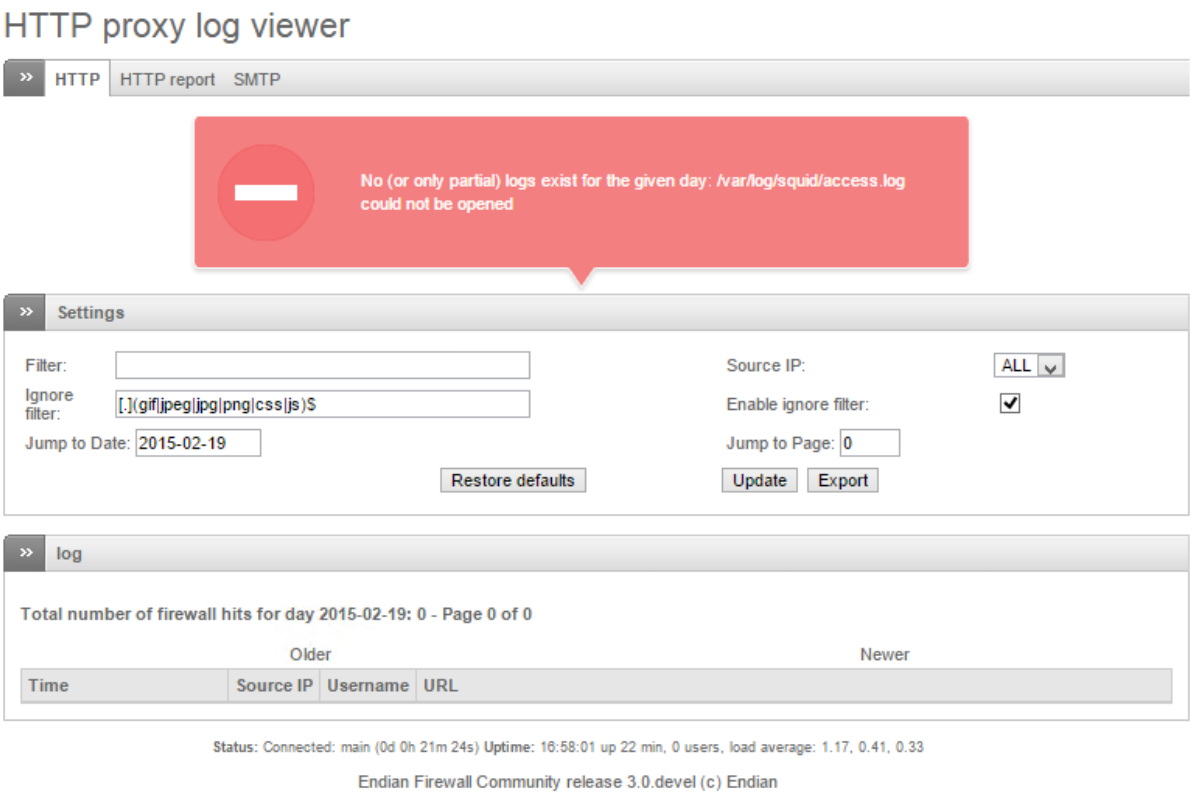
Фиг. 14.9 Журнални данни за защитната стена

От подменюто “Proxy” могат да бъдат анализирани следните данни:

- HTTP;
- HTTP report;
- SMTP.

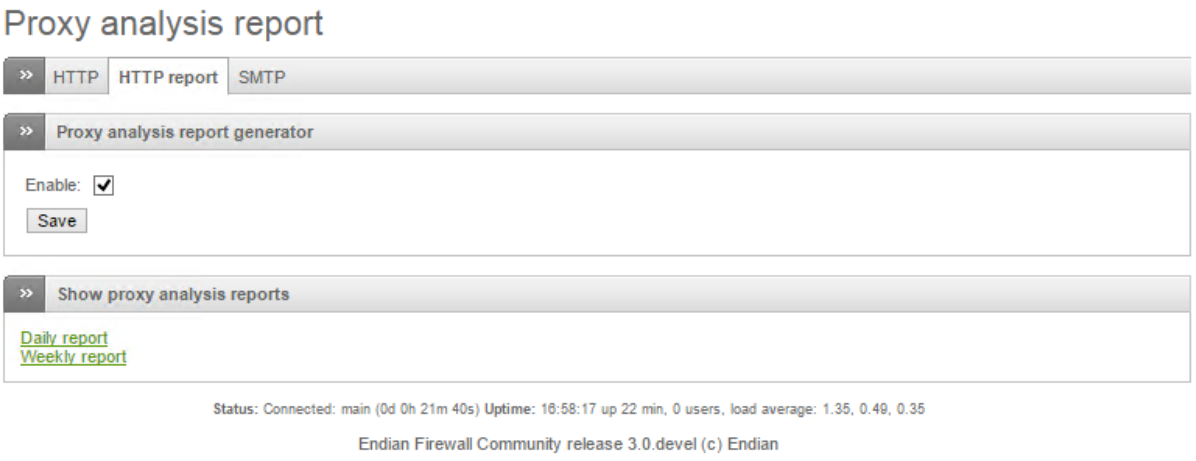
В първата страница “HTTP” освен стандартните функции за филтриране, опресняване и експортиране са налични и следните допълнителни възможности:

- Source IP – данните се визуализират единствено ако адреса на източника съвпада с посочения;
- Ignore filter – Regex израз, който се използва за филтриране по съдържание на записите от журнала. Информацията, която отговаря на филтъра се игнорира;
- Enable ignore filter – активиране на филтъра по съдържание;
- Restore defaults – при натискане на този бутон се възстановяват настройките по подразбиране, свързани с анализа на HTTP журнала.



Фиг. 14.10 Журнални данни за HTTP прокси функциите на EFW CE

В страницата “HTTP report” е налична една единствена опция, която позволява да се активира или деактивира генерирането на доклади, съдържащи резултат от анализа на работата на HTTP прокси сървъра.

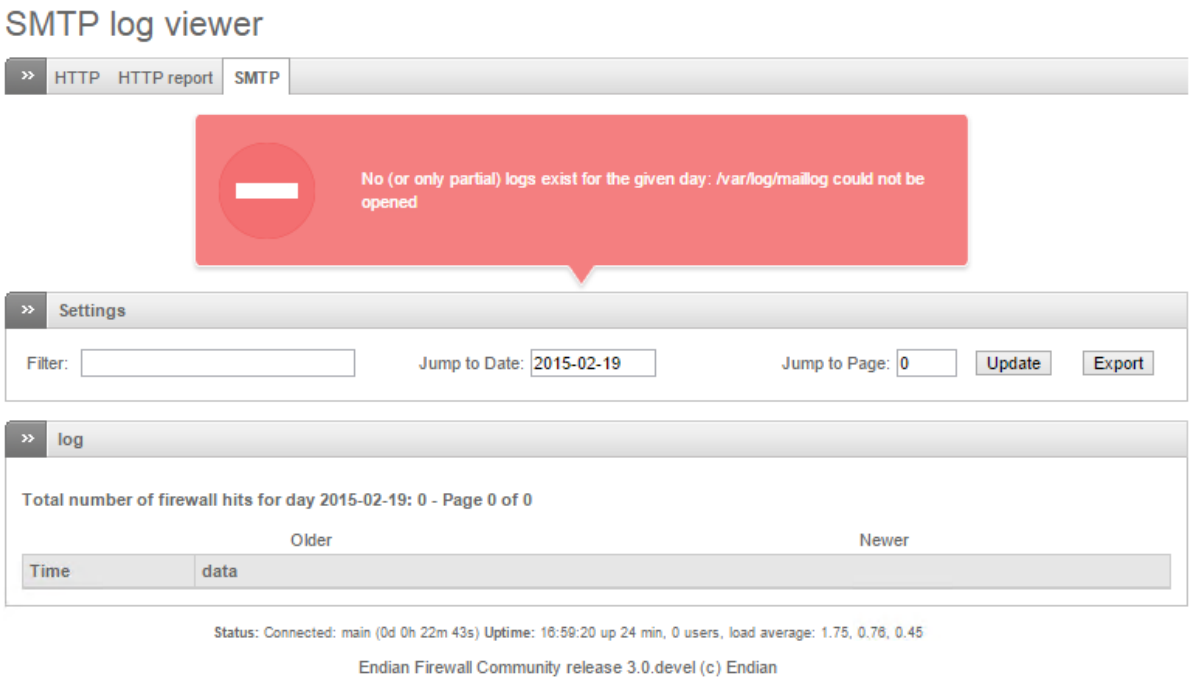


Фиг. 14.11 Подстраница “HTTP report” при EFW CE

На фигура 14.12 е показан доклада, съдържащ обобщените данни за работата на HTTP прокси функциите на EFW CE.

Фиг. 14.12 Обобщени данни за работата на HTTP прокси функциите на EFW CE

В подстраницата SMTP е включена стандартната група за настройки и информация за postfix демона.



Фиг. 14.13 SMTP журнал

Страницата “Log settings” включва глобалните опции, които могат да бъдат зададени с цел оптимално конфигуриране на журналите при EFW. Съдържанието е разделено на четири секции, като първата “Log viewing options” съдържа:

- Number of lines to display – брой редове, които се визуализират по подразбиране за всяка страница, която съдържа данни от определен журнал;
- Sort in reverse chronological order – ако тази опция е активирана първо се извеждат най-новите записи.

Втората секция “Log summaries” обединява следните два параметъра:

- Keep summaries for – интервал от време, посочен в дни, през който се съхраняват обобщените данни от журналите;
- Detail level – ниво на описание на данните в обобщените журнали. Колкото това ниво е по-ниско (Low), толкова по-малко информация се съхранява.

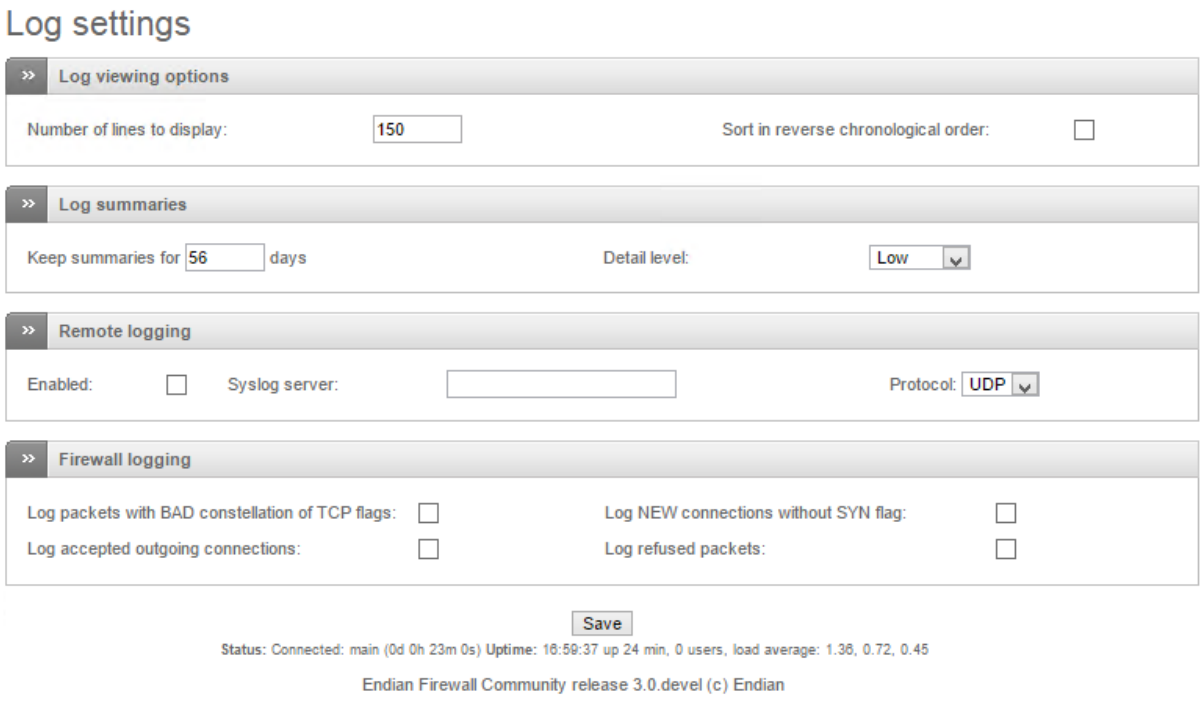
В секцията “Remote logging” е възможно да се посочат необходимите параметри за пренасочване на журналите към отдалечен сървър. Конфигурационните параметри включват:

- Enabled – активира или деактивира отдалеченото съхранение на журналните записи;
- Syslog server – IPv4 адрес на Syslog сървър;

- Protocol – транспортен протокол.

Последната четвърта секция е “Firewall logging”. Тя обединява следните конфигурационни параметри:

- Log packets with BAD constellation of TCP flags – в журналът се записва информация за пакетите с грешни или неизползваеми комбинации от TCP флагове (например всички флагове са едновременно активни);
- Log NEW connections without SYN flag – съхранява се информация за нови сесии, при които не е открит първоначален сегмент с активен флаг SYN[90];
- Log accepted outgoing connections - в журналът се записва информация за всички одобрени изходящи сесии;
- Log refused packets – ако тази опция е активна всеки отхвърлен пакет се описва в журнала на защитната стена.



Фиг. 14.14 Настройки на журналите при EFW CE

Поради фактът, че журналните файлове могат да нараснат значително по подразбиране EFW всяка вечер архивира данните от предходния ден. Въпреки това е препоръчително да се следи свободното пространство на файловата система, на която се съхраняват журналите.

“Trusted Timestamping” е процес, който анализира журналите с цел да се провери за наличие на извършени промени както от неоторизирани лица и процеси, така и от съответните автори (включително програми).

По подразбиране тази функционалност не е активирана, но тя лесно се стартира чрез преместване на бутона “Enable trusted timestamping” в активна позиция.

Конфигурационните параметри са:

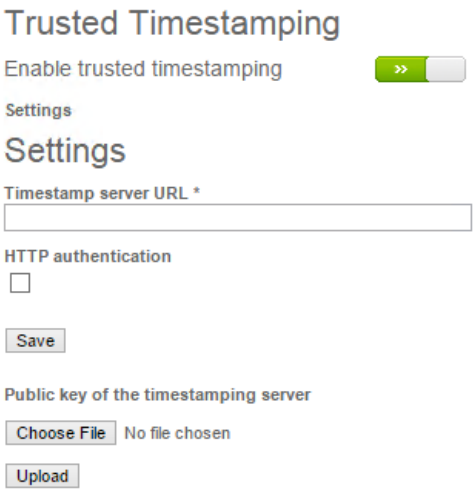
- Timestamp server URL – задължително поле, което посочва сървъра (т.нар. TSA), който ще се използва за цифрово подписване на журналните файлове.

[90] За справка TCP 3-way handshake

- HTTP authentication – ако TSA използва HTTP автентификация трябва тази опция да се активира и да се въведат съответно потребителско име и парола;
- Public key of the timestamping server – за де се улесни процеса за автентификация с TSA е възможно да се импортира неговия публичен ключ.

Направените конфигурационни промени се активират чрез бутона “Save”.

Допълнителна информация за Timestamp протокола можете да откриете в RFC 3161[91].



Фиг. 14.15 “Trusted Timestamping” настройки при EFW CE

Конфигуриране на SNMP сървър с EFW CE

"Протоколът за управление на опростена мрежа" (SNMP) е протокол, който се използва за централизирано конфигуриране и наблюдение на сложни TCP/IP мрежи. SNMP може също да се използва за наблюдение на производителността на мрежата и откриване на проблеми.

SNMP се състои от следните основни компоненти:

1. Управлявано или наблюдавано устройство (Managed Device);
2. Агент (agent) – специализиран софтуер, който работи на управляваното устройство;
3. Мрежова станция за управление (Network Management Station) – централизирано софтуерно приложение за наблюдение на управляваните устройства.

SNMP не определя формата и типа на данните, които могат да се извлекат или изпратят към отделните устройства. За целта се използва MIB (Management Information Base) – специализирана база данни, съдържаща йерархично описание на отделните обекти и техните свойства.

EFW може да се наблюдава отдалечено през протокола SNMP. Необходимите конфигурационни параметри се задават от менюто “Services” и подменюто “SNMP Server”.

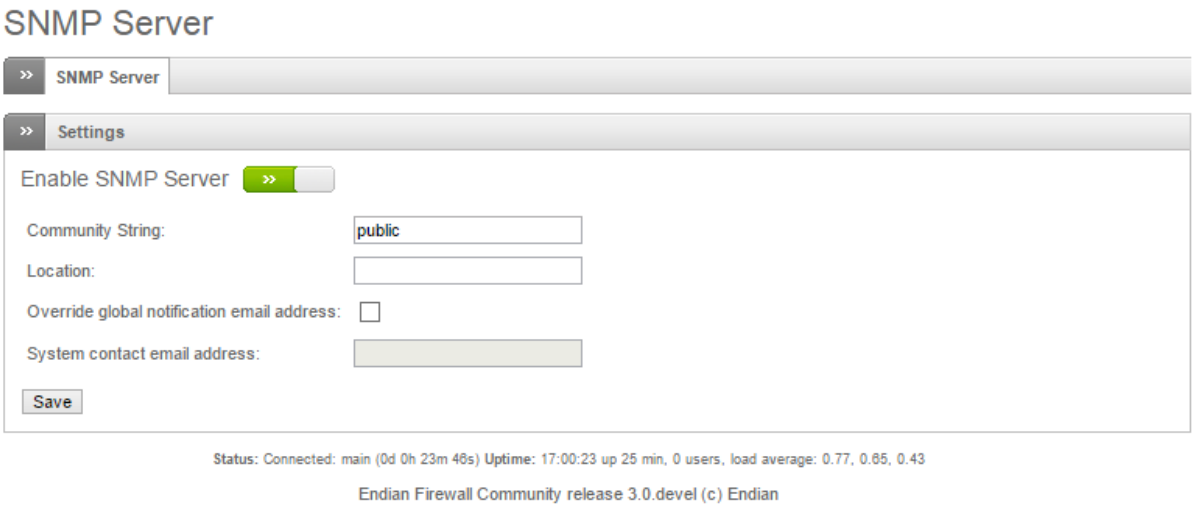
По подразбиране тази функционалност не е активирана и за да се стартира SNMP демона трябва бутона “Enable SNMP Server” де се премести в активна позиция.

[91] https://www.ietf.org/rfc/rfc3161.txt

Фиг. 14.16 Активиране на SNMP сървър при EFW CE

След активирането на SNMP сървъра трябва да се зададат:

- Community String – парола за достъп до SNMP сървъра и извличане на данни. Стойността по подразбиране “public” е задължително тя да бъде заменена с надеждна;
- Location – текстово описание на местоположението на EFW системата;
- Override global notification email address – ако тази опция е активна не се използва общия адрес за изпращане на съобщения от EFW към администраторите. Необходимо е да се въведе “System contact email address”.



Фиг. 14.17 Настройки на SNMP сървъра при EFW CE

Отдалеченото наблюдение и конфигуриране на EFW през SNMP не е обект на разглеждане в тази книга.

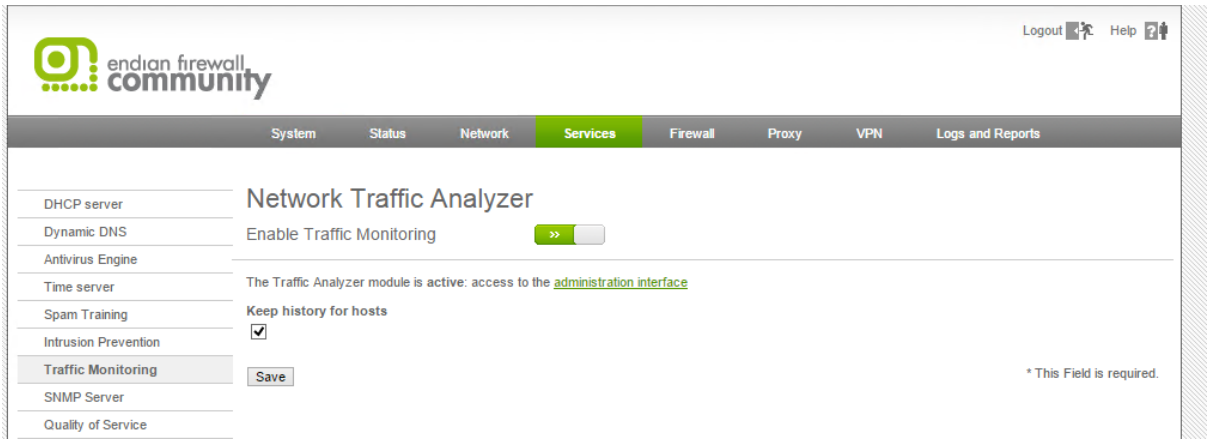
Network Traffic Analyzer

EFW използва мощният инструмент ntop-ng[92] за анализ на трафика през устройството. Тази функционалност е изключително полезна при наблюдението на поведението на мрежата и при проверка за нелегитимни приложения. Активирането на анализирането на трафика се извършва от менюто “Services” и подменюто “Traffic Monitoring”.

[92] www.ntop.org/products/traffic-analysis/ntop

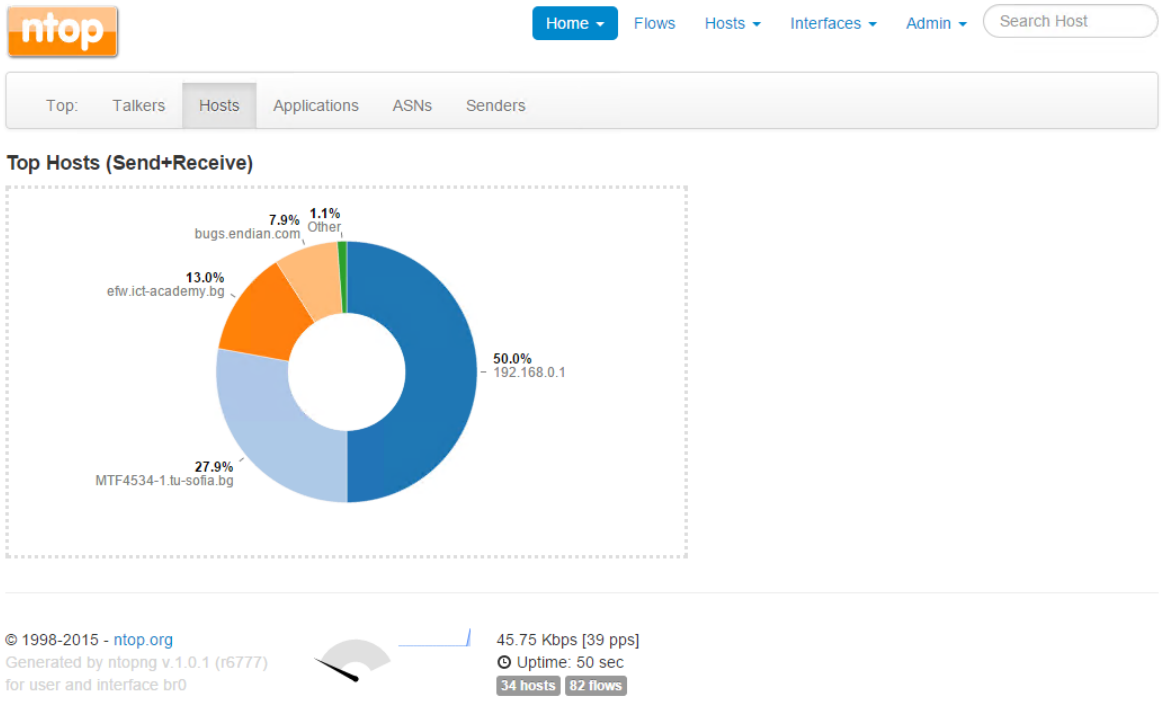
След стартирането на ntop-ng се генерират необходимите статистически данни, които се визуализират в отделен административен интерфейс, който може да се достъпи от препратката “administration interface”.

Опцията “Keep history for hosts” генерира журнални данни за всеки отделен хост.



Фиг. 14.18 Конфигуриране на ntop-ng при EFW CE

На фигура 14.19 е показан административния интерфейс на ntop-ng. От менюто в горната част на страницата могат да се изберат кои данни да се визуализират.



Фиг. 14.19 Административен интерфейс на ntop-ng при EFW CE

Заключение

Активирането на журналните функции на EFW е задължително, като е силно препоръчително отделните журнали периодично да се преглеждат и анализират.

С цел да се предотврати препълване на файловата система за журнали EFW всяка вечер архивира данните от предходния ден.

Визуализирането на журналните данни в реално време е изключително полезно при анализ на атаки или при проверка на правилната работа на EFW.

Вграденият SNMP сървър позволява EFW системата да се наблюдава отдалечено.

Чрез ntop-ng администраторите на EFW могат да получат достъп до статистическа информация, свързана с пренасяните протоколи, комуникиращите хостове, приложения и др.

Източници

1. http://windows.microsoft.com/bg-bg/windows-vista/what-is-simple-network-management-protocol-snmp
2. http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

Списък с фигури