СУ “Св. Климент Охридски”
Факултет по Математика и Информатика
София

# Курсова работа

по

Компютърна Сигурност

Тема: Изграждане на защитна стена с pfSense

Изготвил:
Щеряна Шопова М23681
Маг.Програма: ЗИКСМ
СУ “Св. Климент Охридски”, ФМИ

Януари, 2012
София

# Съдържание

# Въведение

В резултат на бурното развитие на компютърните технологии и в частност на Глобалната Мрежа, на практика всеки офис днес, колкото и малък да е, разполага с поне един, или обикновено няколко компютъра, свързани помежду си и интернет връзка. Бизнесът все повече разчита на компютърните и мрежови технологии за по-ефективно осъществяване на ежедневната си работа. За съжаление, обаче Глобалната мрежа не е безопасно място и за всеки бизнес, колкото и малък да е, възниква необходимост да осигури надежден достъп до данните си за всички оторизирани потребители, като в същото време съхрани целоста и конфиденциалността им. Широко разпространени в последните години станаха така наречените защитни стени (firewalls), които играят ролята на входна точна за данните, които една локална мрежа обменя с Глобалната – Интернет и обратното. В повечето случаи защитните стени са специални и скъпоструващи устройства, които по-малките бизнеси не могат или не намират за необходимо да заплатят.

Настоящата разработка ще опише едно подходящо за този случая решение, което се базира изцяло на безплатни софтуерни продукти, инсталирани и конфигурирани върху обикновен персонален компютър.

## *Цели*

Най-важните цели при поставянето на такава защитна стена бяха определени по следния начин:

- Всеки персонален компютър в офиса да има безопасен достъп до уеб-базирана електронна поща и различни сайтове, необходими за осъществяване на дейността на офиса
- Да се намали риска от достъп до потенциално опасни сайтове
- Да бъде предоставен отдалечен достъп до персоналните компютри на група служители, което да им позволи да работят при необходимост, макар и да не се намират физически във офиса
- Да се намали риска от неоторизиран външен достъп до компютрите в локалната мрежа
- Да се минимизират необходимите разходи за закупуване, внедряване и поддръжка на софтуер и хардуер
- Да се минимизира необходимото време за закупуване на необходимия софтуер и хардуер и внедряване на защитната стена, както и времето за обучение на системния администратор
- Реализацията на защитната стена да е достатъчно гъвкава, че да позволи разширяване на офиса и осъществяване на връзка с отдалечени офиси в бъдеще с минимални промени

## *Избор на технологии и продукти*

При така поставените цели, изборът логично беше сведен до безплатни продукти с отворен код, като първоначално бяха разгледани няколко софтуерни проекта за защитна стена, а именно – PF/OpenBSD, pfSense, m0n0wall, IPCop и IPFire. В крайна сметка се пристъпи към реализация на

защитната стена с pfSense поради сигурността и възможностите, които продуктът предлага, бързата и лесна инсталация и употреба, и не на последно място – знанията за продукта и предишният опит на системния администратор със  pfSense.

# pfSense

## *Кратко описание*

pfSense е безплатна операционна система с отворен код, създадена специално за защитни стени и маршрутизатори. Проектът стартира през 2004 като първоначално ползва за основа операционната система Linux, а в последствие  FreeBSD – проект наследник на Berkley Unix. pfSense е гъвкава дистрибуция, която намира широко приложение при изграждането на различни мрежови елементи като

- Защитна стена (включително и конфигурации със основна и заместваща защитна стена, в случай на отпадане на първата)
- Малък/домашен маршрутизатор (рутер)
- Маршрутизатор за средни/големи мрежи
- Wireless Access Point (Входна точка за безжичен достъп)
- VPN устройство
- VoIP устройство
- Устройство със специално предназначение – DNS, DHCP сървър, подслушване на мрежовия трафик, и други

Най-новата стабилна версия на проекта е pfSense 2.0.1 (от Декември, 2011).

## *pfSense – системни изисквания*

Операционната система поддържа голямо разнообразие от хардуер – от малки устройства със специално предназначение и персонални компютри до сървърни системи. Все пак в зависимост от необходимата производителност на защитната стена, системата има минимални изисквания за хардуера, на който ще се инсталира. Най-слабата /Най-старата система, на която успешно се стартира embedded версията на pfSense е със 100Mhz Pentium CPU, 128MB RAM, 1GB хард диск/512MB компакт флаш карта. В зависимост от необходимата пропускливост на системата pfSense изисква

- 10-20Mbps – поне 266Mhz CPU
- 21-50Mbps – поне 500Mhz CPU
- 51-200Mbps – поне 1GHz CPU
- 201-500Mbps – поне 2.0Ghz CPU, PCI-e мрежови адаптери
- 501Mbps+ - сървърна система, поне 3.0Ghz CPU, PCI-X/PCI-e мрежови адаптери

Допълнително в зависимост от използваните възможности и инсталирани софтуерни пакети се

изисква

- за VPN – допълнителен CPU ресурс и/или хардуерни модули за криптиране, поне 500Mhz CPU за 10Mbps пропускливост на IPSec трафик
- Captive Portal - CPU ресурс
- Големи таблици със състояния на връзките – 1Kb RAM памет за всяка връзка
- Инсталирани пакети - snort, ntop и други - допълнително RAM памет

## *pfSense2.0 – възможности*

### Възможности на защитната стена

pfSense използва пакетния филтър pf от OpenBSD. В общи линии опростеният синтаксис на правилата на филтъра може да се представи по следния начин -

```
action [direction] [log] [quick] [on interface] [af] [proto protocol] \
   [from src_addr [port src_port]] [to dst_addr [port dst_port]] \
   [flags tcp_flags] [state]
```

***Фигура 1. Синтаксис на правило на pf***

където

- *action* – е *pass* (пропуска пакета) или *drop* (не пропуска пакета)
- *direction – in* (входящи пакети) или *out* (изходящи пакети)
- *log* – при наличието на тази ключова дума в правилото, за всеки пакет, който удовлятворява правилото, се генерира запис в журналния файл на пакетния филтър
- *quick* – по подразбиране пакетния филтър оценява всички правила като се предприема действието според последното правило, което удовлетворява пакета; при наличието на тази ключова дума, действието на първото правилото удовлетворено от пакета се предприема веднага, т.е пакета се пропуска или не и се прекратява оценката на останалите правила на филтъра спрямо текущия пакет
- *interface* – име или група от мрежови интерфейси в системата
- *af* – тип на мрежовия адрес, inet за IPv4 или inet6 за IPv6
- *protocol* -номер или има на транспортен протокол, напр. *tcp, udp, icmp, icmp6*
- *src_addr, dst_addr* – Source и Destination IP адреси на пакета
- *src_port, dst_port* - Source и Destination портове на пакета
- *tcp_flags* – TCP флагове във пакета

- *state - keep state* (по подразбиране), *modulate state* или *synproxy state;* тази ключова дума указва дали ще се пази състояние на връзката за пакети за пакетите, които удовлетворяват правилото

Накратко, част от останалите възможности на пакетния филтър са следните:

- Филтрация на IP, TCP и UDP пакети по различни критерии със следене на състоянието на връзката (Stateful Packet Inspection)
- Ограничение на едновременно установените връзки за всяко правило
- Избирателни журнални записи за пакетите които удовлетворяват дадено правило
- Policy Routing
- Използване на псевдоними (aliases) и таблици за групиране на IP-та, портове и мрежи
- Прозрачно филтриране на етернет ниво
- Нормализиране на трафика
- Ограничения и алгоритми за оптимизации на таблицата със състоянията на установените връзки
- Филтриране по критерий тип на операционна система, изпратила пакета
- Синхронизация на таблиците на установените връзки между основна и заместваща защитна стена чрез мрежови протоколи CARP и pfsync
- Netwok Address Translation (транслиране на IP адреси)
- Възможност за разпределяне на трафика между няколко мрежови интерфейса (Load Balancing)
- Филтриране на ниво 7 (приложно ниво)
- EasyRule – възможност за добавяне на нови правила на базата на отделни записи от журналния файл на пакетния филтър
- Възможност за активиране/деактивиране на правила според часа от денонощието

В *Приложение 1* е даден пълният списък от правила, които са използвани за реализация на защитната стена, включително и NAT правилата, необходими за OpenVPN сървър.

## Управление на достъпа до системата и потребителски акаунти

Вградените възможности на системата позволяват три метода за аутентикация на потребители -

1. Чрез създаване на акаунти в локална база данни
2. Аутентикация чрез външен LDAP сървър
3. Аутентикация чрез външен RADIUS сървър

Под потребител в случая имаме предвид системен администратор или оператор, който управлява настройките на защитната стена или извършва диагностика. Ще разгледаме накратко само първия метод. Важно е да се отбележи, че достъпът до системата се осъществява по два начина –

чрез терминал/SSH, където потребителите имат достъп до UNIX шел (bin/sh) и може да изпълняват всички команди, които системата поддържа; и чрез WebUI интерфейс, разделен на менюта и подстраници. Част от страниците позволяват промяна на системните настройки чрез различни форми, докато други предостявят само информация за състоянието на системата. Трети позволяват изпълнението на различни приложения за диагностика, като използването на предоставените възможности може да се отрази на текущото състояние на системата (например използва се повече мрежови ресурс при стартиране на командата iperf за проверка на пропускливостта на комуникационен канал), но не се отразява дългосрочно на услугите предоставени на потребителите в локалната мрежа зад защитната стена. WebUI интерфейсът е много удобен и интиутивен за ползване дори и от начинаещи потребители – когато потребителят се опита да запази несъвместима или непълна конфигурация, обикновено системата дава съобщение за грешка; докато достъпът до системата чрез терминал/SSH изисква много добри познания на UNIX шел-а, възможностите на системата и начина по който работи. Предполага се, че достъп до терминал/SSH има само един или няколко много опитни системни администратори и такъв достъп се налага само когато даден проблем не може да бъде разрешен през WebUI интерфейса (например, няма връзка от локалната мрежа до защитната стена, а пакетния филтър е настоен да блокира всички заявки до HTTPS/HTTP порт-а на защитната стена от външния интерфейс). В описанието на системата оттук нататък, ще предпочитаме възможностите на WebUI интерфейса, като където е необходимо ще описваме и еквивалентните команди в UNIX шел-а.

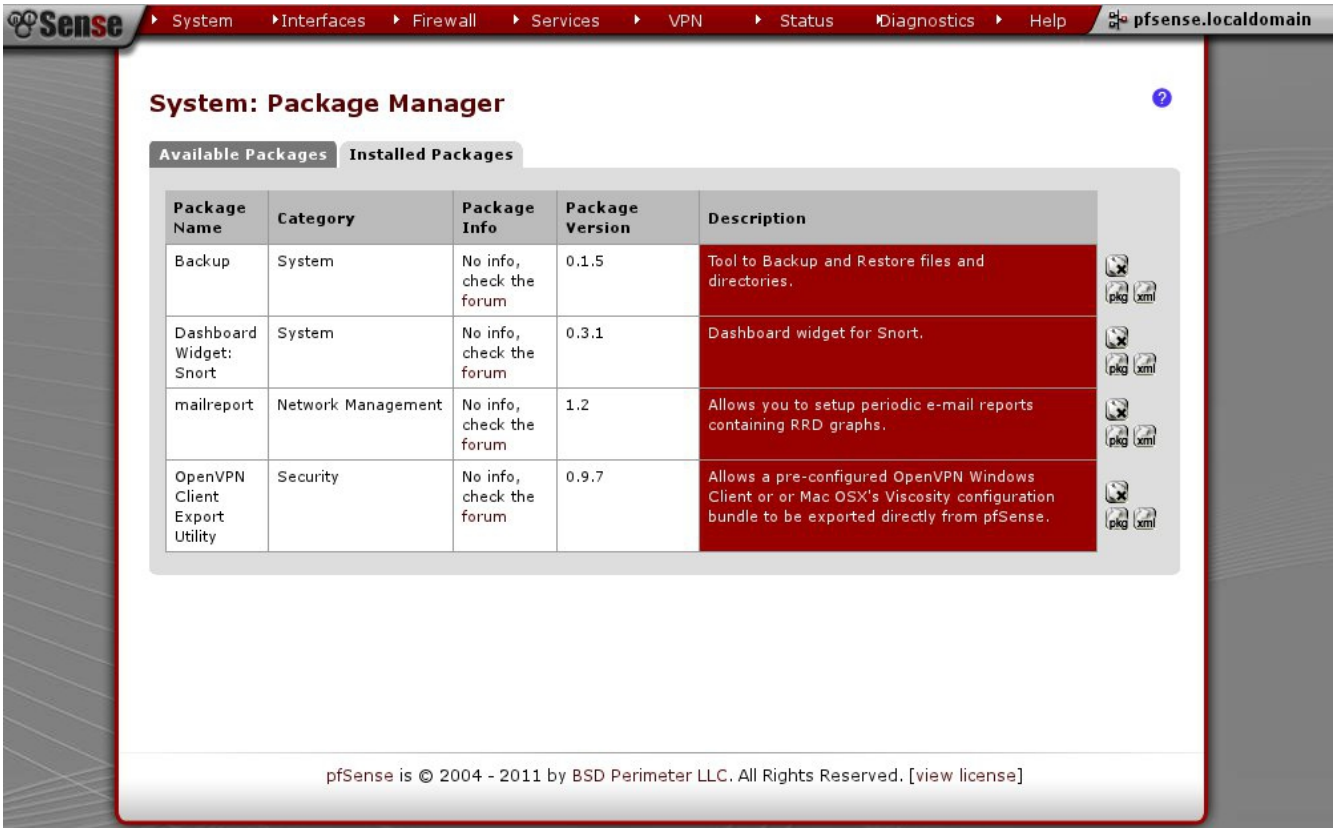
Системата позволява създаването на групи от потребители и списъци за управление на достъпа до различни части от WebUI интерфейса за всяка група. Всеки потребител наследява правата за достъп дадени на групата, към която принадлежи, като могат да му бъдат делегирани нови права за достъп или наложени допълнителни ограничения. Всеки потребител трябва да е член на поне една група, а системата позволява даден потребител да е член на няколко групи едновременно. Всеки потребител се характеризира със следните полета (незадължителните полета са маркирани със *)

- Потребителско име, парола и пълно име на потребителя
- Срок на изтичане на валидността на акаунта (възможно е да има акаунти, чиято валидност никога не изтича)
- Списък от групи, на които потребителят е член
- Списък от ефективни права за достъп до различните части на WebUI интерфейса
- Списък от сертификати на потребителя *
- Списък от ауторизирани SSH ключове * (когато на съответния потребител трябва да бъде предоставен SSH достъп до системата)
- IPsec Pre-Shared Key*

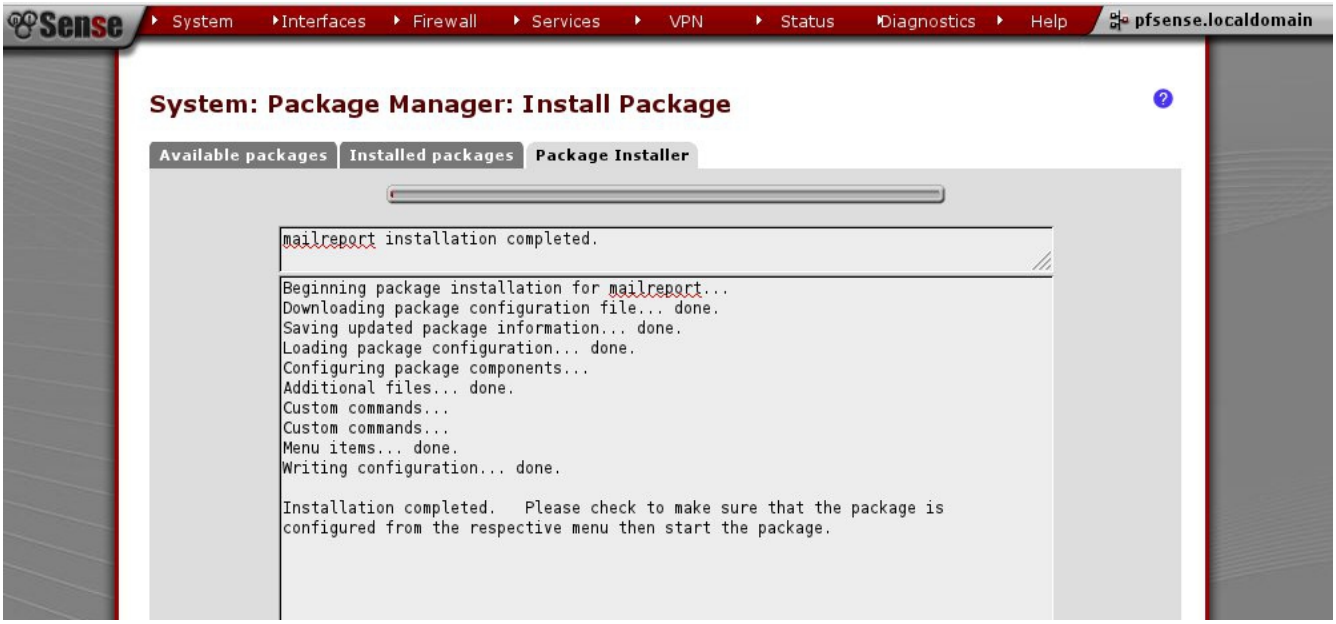
Паролите на потребителите се съхраняват в глобален конфигурационен файл като паролите биват еднопосочно криптирани с MD5 хеш функция. Системата позволява използането на повече от един метод за аутентикация едновременно. Ако за даден потребител има конфигурирани сертификати, то има възможност за интегрирането им при настройките на VPN достъп за този потребител.

## Управление на софтуерни пакети

Системата позволява разширяване на вградените възможности чрез инсталиране на допълнителни софтуерни пакети. WebUI интерфейсът предоставя специална страница за управление на пакети с възможност за преглеждане на всички налични пакети и добавяне на нови.




***Фигура 2. Спъсък на инсталираните софтуерни пакети***




***Фигура 3. Инсталиране на софтуерен пакет***

Управлението на пакети в pfSense се базира на пакетната система на FreeBSD, т.е софтуерните пакети се инсталират от предварително подготвени бинарни файлове, достъпни от сървърите на проекта, като специфичното при pfSense е, че за всеки инсталиран пакет, конфигурацията на инсталирания пакет се запазва в XML формат в глобален за системата конфигурационен файл, вместо в отделен файл в /etc директорията на файловата система, както е традиционно за UNIX операционните системи. Освен това, за всеки инсталиран пакет се добавят една или няколко нови страници в WebUI интерфейсът на системата, които позволяват на потребителите да управляват настройките и диагностицират работата на инсталирания софтуерен пакет.

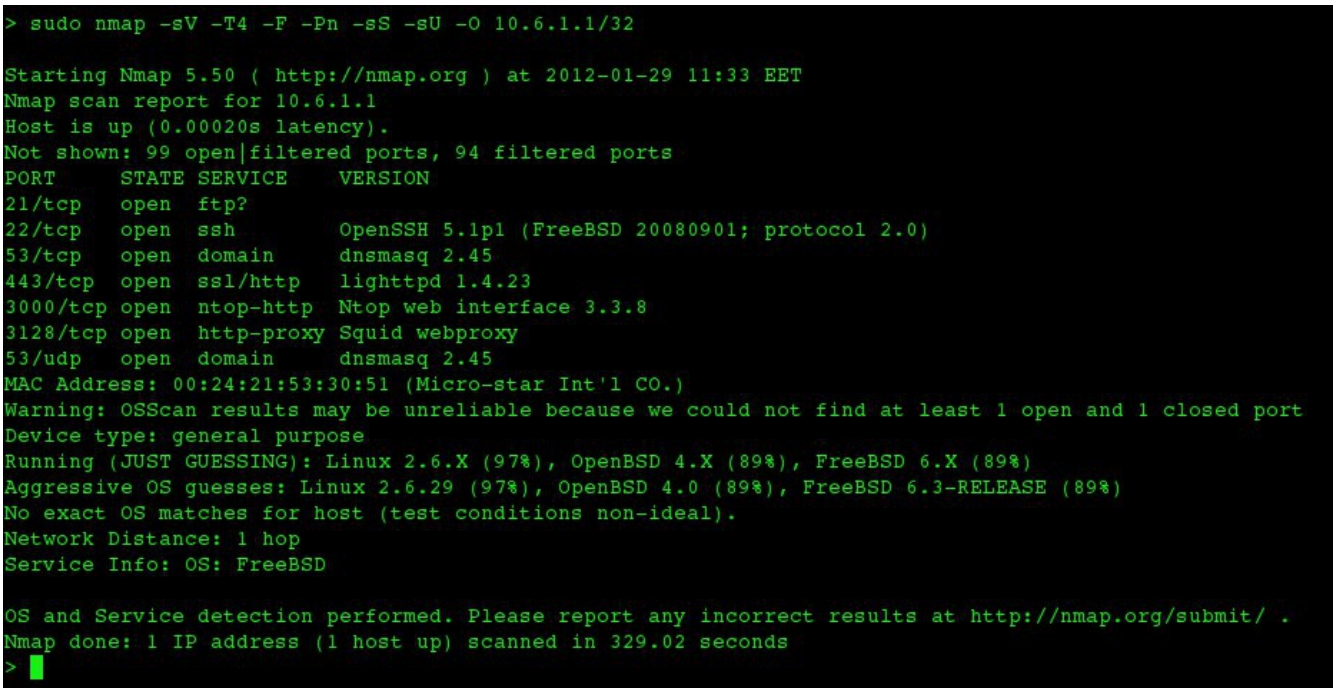
Към момента pfSense 2.0.1 поддържа около 75 софтуерни пакета, разделени в няколко категории. Ще опишем накратко по-интересните от тях.

### *Сигурност на мрежата (Security)*

- ***Snort*** – Snort е безплатна реализация на система за откриване и предотвратяване на нарушения; най-общо системата работи като подслушва целия мрежови трафик на интерфейсите, на които е стартирана, и на база на извършения анализ на трафика, съобщава за евентуални атаки и типа им. Съществува допълнителна опция, освен извеждане на съобщения за евентуални атаки, хостовете от които идват атаките, да бъдат блокирани от защитната стена.


***Фигура 4. Snort съобщения по време на Port Scan атака***

- ***nmap*** **–** Nmap е приложение за анализ на мрежи и извършване на одити на сигурността на мрежите. Използва се за откриване на активни хостове чрез ping проби, за осъществяване на различни техники за реализиране на Port Scan - определяне на предоставяните мрежови услуги, включително какви приложения ги обслужват, както и версиите им; както и за TCP/IP fingerprinting с цел отгатване на операционната система и типа на отдалеченото устройство. Освен като софтуерен пакет за тестване на сигурност на защитната стена, *nmap* от отдалечена машина може да се ползва за тестване на сигурността на самата защитна стена.




***Фигура 5. Резултати от nmap Port Scan тест за сигурност срещу интерфейса към локалната мрежа на защитна стена***

### *Управление и следене състоянието на мрежата (Network Management)*

- ***Zabbix агент*** **–** Zabbix агентът събира информацията за състоянието и работата на системата и мрежоватата и активност и я изпраща към отдалечена система за управление на мрежата – Zabbix NMS (Network Monitoring System), където тази информация да бъде анализирана като част от цялостните характеристики и работа на мрежата, а не само на отделен неин елемент

### *Допълнителни услуги (Services)*

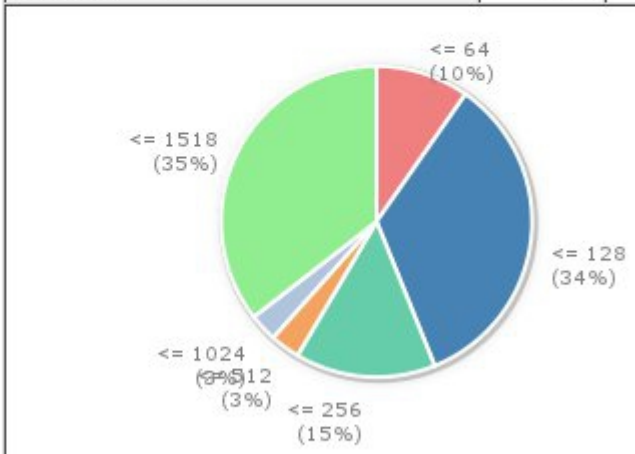
- ***Dansguardian*** **–** филтър за WEB съдържание
- ***squid, squidGuard, Lightsquid*** **–** приложения за кеширане на WEB страници, филтриране на URL и отчет на посетени WEB страници
- ***Varnish*** - HTTP accelerator - HTTP прокси сървър, който обикновено се ползва за обслужване на динамични WEB страници с много съдържание; позволява да се увеличи скоростта, с която един WEB сървър предоставя съдържанието на дадена страница от 300

до 1000 пъти, което доста затруднява реализирането на DoS атаки срещу самия WEB сървър

- ***HAVP antivirus -*** HTTP прокси сървър със антивирусен скенер
- ***imspector*** – прозрачно прокси за приложения за обмяна на мигновени съобщения; поддържа следните протоколи - MSN, AIM, ICQ, Yahoo и IRC; има възможност за отчет на обменените съобщения
- ***FreeSWITCH, siproxd*** – платформа за телефония, използва се при реализирането на системи за обмен на гласови и мултимедийни съобщения като софтуерни телефони и централи
- ***freeradius*** – сървър за аутентикация, ауторизация и отчетност (AAA) чрез RADIUS протокол
- ***dnsserver –*** DNS сървър

### *Пакети за диагностика (Diagnostics)*

- ***arping, arpwatch***- приложения за изпращане на ARP пакети и за следене и отчет на промяната на двойки от етернет(MAC) и IP адреси в ARP таблицата на системата; помага за засичане на ARP spoofing атаки в локалната мрежа
- ***bandwidthd*** – приложение, ковто следи натоварването на TCP/IP подмрежите на системата и на база на събраната информация предоставя диаграми със различни статистики
- ***iperf –*** е приложение за измерване на производителността/пропускливостта на комуникационните канали
- ***ntop –*** приложение, което събира допълнителни статистики за активността на мрежата и генерира графики на базата на различни характеристики
- ***mailreport*** – приложение, което позволява изпращането на и-мейли със периодични отчети и графики

| | | | |
|---|---|---|---|
| Packets | Shortest | | 60 bytes |
| | Average Size | | 546 bytes |
| | Longest | | 1,514 bytes |
| | Size <= 64 bytes | 9.7% | 6,713,563 |
| | 64 < Size <= 128 bytes | 34.2% | 23,723,992 |
| | 128 < Size <= 256 bytes | 14.6% | 10,128,129 |
| | 256 < Size <= 512 bytes | 3.1% | 2,164,512 |
| | 512 < Size <= 1024 bytes | 2.9% | 1,990,313 |
| | 1024 < Size <= 1518 bytes | 35.5% | 24,645,045 |
| | Size > 1518 bytes | 0.0% | 0 |
| |  | | |
| | Packets too long [> 1514] | 0.0% | 0 |
| | Bad Packets (Checksum) | 0.0% | 0 |

***Фигура 6. Ntop – Графика на процентното разпределение на различните големини на пакети***

## Обновяване на системата

Използването на нови/актуални версии на софтуера е важен фактор за сигурността на всяка система. В зависимост от избраната политика за обновяване на системата, има възможност за автоматично или ръчно обновяване на pfSense. Ръчното обновяване може да се извърши през терминал/SSH или през WebUI интерфейса на системата, при условие че потребителят разполага с новата версия, към която да премине системата. Автоматичното обновяване проверява периодично за наличието на нови версии на определени URL (обикновено сървърите които съхраняват новите версии на pfSense, като всяка версия е дигитално подписана) и при наличието на такава дава възможност на потребителя да стартира изтеглянето и преминаването към версията. Потребителят може да посочи собствен URL, различен от официалните сървъри на pfSense - недостатъкът в този случай е, че не се извършва проверка за цялостта и автентичността на новата версия. Силно препоръчително е, преди да се премине към новата версия, на изолирана машина да бъдат детайлно изтествани както самата процедура по преминаване към новата версия, така и новата версия за евентуални функционални проблеми и

проблеми със сигурността. Дори и при така проверена нова версия на софтуера, добра практика е преди да се премине към актуализация, да се направи пълен архив (Backup) на системата, както и план/алгоритъм за възстановяване на старото състояние на системата (Backup plan), ако нещо все пак се обърка. С развитието на виртуализационните технологии, особено при невъзможност да се осигури физичека машина за изолирани тестове, разпространена практика в последните нялкоко години е новите версии да се тестват на виртуална машина. Популярни софтуерни пакети за виртуализация са *VMWare*, както и безплатните *qemu* и *VirtualBox*.

## Хронология на промените в конфигурацията и backup/restore

Част от вградените възможности на системата е поддържането на пълна хронология на промените в конфигурацията на всяка нейна част, както и на добавянето или премахването на софтуерни пакети. Чрез WebUI интерфейса, ауторизираните потребители имат възможност да прегледат всички направени промени от първоначалното инсталиране на системата, включително ден и час на промяната, име на потребител, който е извършил промяната, както и кратко описание. Има възможност да се видят разликите между два приозволни записа в хронологията (под формата на Unix diff формат), системата да бъде върната до произволна стара конфигурация или да бъде направен външен архив на конфигурацията на системата в момента на произволен запис от хронологията. WebUI интерфейсът също дава възможност за автоматично създаване и изтегляне на архиви (включително и криптирани) на произволни части на системата - обикновено това са глобалния конфигурационен файл, журнални файлове, репорти и статистики за работата на системата и инсталираните софтуетни пакети.

## VPN и управление на сертификати

Обикновено защитната стена е входна точка на комуникацията на една локална мрежа с външния свят (Интернет), тя е мястото, което е най-подходящо за избор на крайна точка на тунел (криптирана връзка) между локалната мрежа и отдалечени машини или мрежи , които трябва да обменят защитена информация с локална мрежа чрез т.нар Вирутални Частни Мрежи (VPN). pfSense поддържа няколко вградени технологии за изграждане на VPN

- IPSec – реализира се директно на ниво 3 от OSI модела; основните части на Secure IP протокола са - удостоверяващо начало (Authentication Header - AH), вграден закодиран товар (Encapsulated Security Payload - ESP), както и разменен интернет ключ (Internet Key Exchange - IKE) за размяна на сесийните ключове
- L2TP/PPTP – използват се средствата на протокол от ниво 2 от OSI модела; например PPTP е метод за изграждане на виртуални частни мрежи, който използва контролен канал над TCP и GRE тунели за капсулиране на пакети на PPP
- OpenVPN – OpenVPN е софтуерен пакет за изграждане на VPN, който ползва свой собствен протокол за установяване на защитената връзка; OpenVPN може да изгради контролен и/или канал за данни върху UDP или TCP връзка (т.е на ниво 4 от OSI модела); контролният /каналът за данни се криптират с помощта на OpenSSL библиотеката, което позволява използването на различни шифъри; аутентикацията може да се извършва на база на споделени ключове (pre-shared keys), сертификати или потребителско име и парола. В системата могат да бъдат стартирани един или няколко самостоятелни OpenVPN сървъра или клиента. По време на самия процес на конфигурация на OpenVPN

сървър се дава възможност да бъдат генерирани необходимите за сървъра сертификати, а впоследствие от менюто за добавяне/управление на потребители, да бъдат добавени и подписа съответните сертификати на потребителите, които ще ползват VPN-а. Последно ще спомена, че като допълнителен софтуерен пакет, pfSense предоставя и т.нар. “OpenVPN Client Export Utility “ - той позволява автоматично да бъде създаден пакет - .т.е. архив съдържащ изпълним файл на OpenVPN клиент и необходимите сертификати. Клиентските пакети засега поддържат само различните версии на Windows и Mac OSX.

## Управление на мрежови услуги

Както вече казахме, освен вградените в системата мрежови услуги, pfSense поддържа и други услуги чре допълнително инсталираните софтуерни пакети. WebUI интерфейсът предоставя удобен и бърз начин за настройване на всяка мрежова услuга чрез менюто *Services*.


***Фигура 5. Меню Services – Избор на подпрозорец за настройка на предлаганите мрежови услуги***

Накратко ще опишем всяка от вградените в pfSense услуги.

- ***Captive Portal –*** Обикновено се излолзва, когато системата с pfSense е входна точка за безжичен достъп до Интернет (Wireless Access Point). Captive Portal-ът предоставя на безжичните клиенти достъп до първоначална страница, в която да въведат своето потребителско име и парола или код за достъп. Интернет достъпът е разрешен за всички клиенти аутентикирали се успешно. Услугата предоставя множество допълнителни настойки като филтриране по MAC адрес, период на неактивност, след който да се изисква повторна аутентикация от страна на клиента, ограничения по отношение на изтегленото и каченото за секунда количество информация (в Kbit/s) и други.
- ***DHCP Server/Relay*** – DHCP е мрежов протокол за автоматична конфигурация на хостове в една IP мрежа. В най-простия случай хостовете в локалната мрежа получават IP настройките си от DHCP сървъра (IP адрес и маска, адрес на маршрутизатора в локалната мрежа, адресите на DNS сървъри и д.р.). В други случаи разширенията на DHCP протокола позволяват на сървър-а да указва на клиентите допълнителни настройки като име или адрес на SMTP сървър или дори име/адрес на сървър, от който клиентът да изтегли файл с операционната система, която ще стартира. Тъй като част от съобщенията, които клиентът и сървърът обменят са броудкаст пакети, които биват блокирани от защитните стени и маршрутизаторите, се налага DHCP сървърът и клиентът да са в една локална мрежа (броудкаст домейн). Когато DHCP сървърът и клиентите са в различни броудкаст домейни и се използва различен от вградения в pfSense системата, за да се преодолее това ограничение се използва DHCP Relay агентът на pfSense.

- ***DNS Forwarder*** – използва се за препращане на DNS заявки от локалната мрежа към външни DNS сървъри
- ***Dynamic DNS –*** услугата се ползва когато външният IP адрес на pfSense системата може да се променя (например се взима от DHCP сървър в мрежата на доставчика)
- ***IGMP Proxy*** – се ползва например за доставяне на видео и IP телевизия в локалната мрежа, когато IGMP е протоколът, който хостовете използват за заявка за получаване на даден канал/стрийм
- ***OLSR*** – това е протокол за маршрутизациия, който често се използва при изграждането на безжични мрежи (Wireless Mesh Routing)
- ***OpenBGPD, OpenOSPFD*** – процеси, които обслужват BGP и OSPF протоколи за маршрутизация, които когато pfSense системата се използва за маршрутизатор в средни и големи мрежи
- ***RIP*** – един от първите протоколи за маршрутизация, заради недостатъците си в днешно време ползва се най-вече в малки мрежи предимно с учебна цел
- ***OpenNTPD*** – NTP сървър; Network Time Protocol се използва за синхронизиране на системното време на хостовете в една мрежа; тази услуга е важна тъй като коректната работа на много приложения (напр. общ календар, отчети и т.н) разчита на това хостовете да имат (почти) една и съща идея за системната дата и време
- ***SNMP*** - SNMP е стандартен протокол за следене и управление на мрежи. SNMP сървърът се използва когато е необходимо отдалечен хост за събира информация за състоянието на pfSense системата, или да бъде уведомяван за критични събития като отпадане на мрежова карта или системен диск, достигане на определен праг на натоварване и други
- ***Wake on LAN*** - позволява определени хостове в локалната мрежа да бъдат включени или "събудени" от специални мрежови пакет - “магически” пакет; изисква се специална хардуерна поддръжка на протокола от страна на системата, която трябва да бъде събудена. Услугата е полезна при отдалечено администриране на хостовете в локалната мрежа.

## Състояние на системата и журнални файлове

Едно от важните условия за коректна работа на защитната стена, успешно предотвратяване на мрежови атаки или засичането на вече осъществени пробиви е постоянното следене на различна информация за състоянието на системата и записването на тази информация в журнални файлове или под друга форма. Различни страници на WebUI интерфейса предоставят информация за текущото състояние на мрежовите интерфейси, заетите адреси от DHCP клиенти (т.нар. DHCP Leases), активните gateways, активните IPSec и OpenVPN виртуални частни мрежи, журналните записи на инсталираните софтуерни пакети.


***Фигура 8. Табло с текущото състояние на системата***

Най-важните страници, на които ще обърнем внимание са две

- Журнални записи на системата и настройка на журнални логове
- Страница със RRD графики

Журналните записи на системата са разделени на няколко категории – записи за цялостната работа на системата (вход/изход на потребителите в системата, стартиране на критични за системата процеси и други), записи на пакетния филтър – блокирани или пропуснати пакети от защитната стена, записи за работата на активните VPN-и, PPP, DHCP, и NTPD сървърите, както и Captive Portal-а. За съжаление броя на записите, които могат да бъдат съхранени на файловата система, както и тези които могат да бъдат изобразени в WebUI интерфейса, е крайно ограничен. Стойността по подразбиране е 50, а разрешения максимум на записите е 2000, като по-новите записи изместват по-старите. Затова pfSense предоставя възможност журналните записи да се изпращат към отдалечен (един или няколко) *syslogd* сървъра. Силно препоръчително е тази опция да се използва, не само за да е достъпна информация за работата на системата с месеци и години назад, а и поради възможността при евентуален пробив на защитната стена, данните и журналните записи, които се съхраняват на локалната файлова система, да бъдат компроменирани и недостъпни за по-нататъшен анализ.

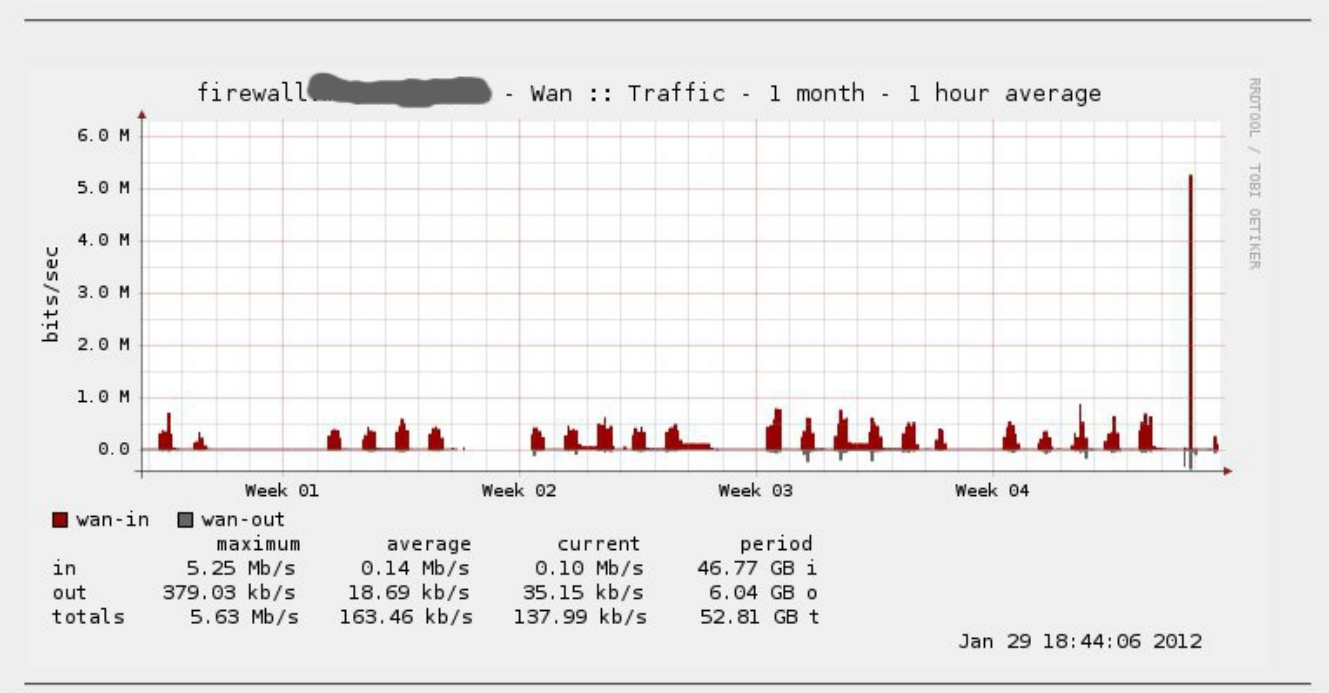
В **П*риложение 2*** е дадена малка извадка от логовете на защитната стена, във текстов формат така както са достъпни на файловата система на отдалечения *syslogd* сървър. От извадката се виждат проби за достъпността на различни услуги като Telnet (порт 23), HTTP (порт 80, както и понякога 8080), SAMBA over TCP (порт 445), Remote Desktop Server (порт 3389) и други.

Изключително важно за засичането на извънредни ситуации в една система или мрежа, е да се познава добре работата на системата или мрежата при нормални обстоятелства. Това обикновено става чрез събиране и анализ на статистики /броячи/ за различните части на системата. По подразбиране pfSense събира такива статистики и предоставя графики на страница та с RRD графиките за параметри на системата, включително

- натовареност на процесора(процесорите) в %
- брой на състояния в талицата на състояния на връзките
- обща пропускливост на мрежовия трафик през системата в kbs
- входящ и изходящ трафик за всяка мрежова карта на системата в kbs
- брой входящ и изходящ пакети за всяка мрежова карта на системата в pps
- качество на канала за всяка мрежова карта – загуба на пакети (в %) и време за достигане

до определена точка и обратното (Roundtrip time, в ms)

Графиките се чертаят за периоди от 4 и 16 часа, 2 дни, 1 и 6 месеца и 1 година със средни стойности взети съответно за периоди от 1 минута, 5 минути, 1 и 12 часа.




***Фигура 9. Графика на трафика на външния интерфейс за период от 1 месец***

На фигурата по-горе е даден пример за такава графика. Обикновено входящият трафик е около 0.5 Mbit/s за дневните часове на работните дни, и почти няма потребление през нощните часове (малките "дупки" между стълбовете) и събота и неделя (големите "дупки"). Ясно се вижда аномалията в графиката – потребление около 10 пъти над обичайното в края на периода, което е индикация за необходима по-нататъшна проверка на журнални записи и друга данни с цел да се установи причината за аномалията/възможен пробив или атака.

## Диагностика

За диагностициране и отстраняване на проблеми при настройка и работа, системата предоставя меню с подстраници за изпълнение на добре познати UNIX/FreeBSD команди за диагностика на TCP/IP - top, arp, nslookup, ping, traceroute, tcpdump, netstat, route, ntop, pftop. pftop е специфична за pfSense команда, която по аналог на top показва състоянието на пакетния филтър.


***Фигура 10. Изход от командата pftop***

Други команди, които нямат отделна страница в менюто за Диагностика могат да бъдат изпълнени в UNIX шел-а или страницата за изпълнение на команди на WebUI интерфейса. Менюто за диагностика също така дава възможност да се види таблицата на състоянията на осъществените връзки, да се спре или рестарира система и пакетния филтър.

## Контекстни помощни менюта и документация

Една от новостите в последните версии на pfSense са новите контекстни помощни менюта и меню с връзки към различни ресурси с документация на проекта в WebUI интерфейса, а също връзки към дискусионните форуми на проекта и към портала за комерсиална помощ на pfSense. Ще споменем и че проектът поддържа активен IRC канал в мрежата Freenode - ##pfsense където нови и опитни потребители могат да получат безплатна помощ и ценни съвети.

## Заключение

**pfSense** е гъвкава дистрибуция, която намира широко приложение като защитна стена или специално устройство осигуряващо сигурност на малки и средни мрежи, независимо от разнообразните изисквания. pfSense предоставя възможности за защита от широк спектър от мрежови атаки и при правилна конфигурация и експлоатация, може да бъде евтино решение, осигуряващо необходимото ниво на сигурност на повечето малки и средни бизнес и домашни потребители. Едно от големите предимства на дистрибуцията е лесната и употреба и изключително краткото време, необходимо за въвеждането и в експлоатация.

## ***Приложение 1: Списък от правила на пакетния филтър***

```
myip="{ N/A }"

vpnports="{ N/A }"

pass in quick on re0 proto tcp from any to ! (re0) port = http flags S/SA keep
state

pass in quick on re0 proto tcp from any to ! (re0) port = 3128 flags S/SA keep
state

anchor "ftpsesame/*" all

anchor "firewallrules" all

block drop quick proto tcp from any port = 0 to any

block drop quick proto tcp from any to any port = 0

block drop quick proto udp from any port = 0 to any

block drop quick proto udp from any to any port = 0

block drop quick from <snort2c> to any label "Block snort2c hosts"

block drop quick from any to <snort2c> label "Block snort2c hosts"

block drop in quick inet6 all

block drop out quick inet6 all

anchor "loopback" all

pass in quick on lo0 all flags S/SA keep state label "pass loopback"

pass out quick on lo0 all flags S/SA keep state label "pass loopback"

pass quick inet proto icmp from $myip to any keep state

pass in quick on re0 inet proto udp from any port = bootpc to 255.255.255.255 port
= bootps keep state label "allow access to DHCP server on LAN"

pass in quick on re0 inet proto udp from any port = bootpc to 10.6.1.1 port =
bootps keep state label "allow access to DHCP server on LAN"

pass out quick on re0 inet proto udp from 10.6.1.1 port = bootps to any port =
bootpc keep state label "allow access to DHCP server on LAN"

block drop in log quick on rl0 inet proto udp from any port = bootps to 10.6.1.0/24
port = bootpc label "block dhcp client out wan"

block drop in on ! re0 inet from 10.6.1.0/24 to any

block drop in inet from 10.6.1.1 to any

block drop in inet from 192.168.1.1 to any

block drop in on re0 inet6 from fe80::224:21ff:fe53:3051 to any

pass out quick on re0 proto icmp all keep state label "let out anything from
firewall host itself"

pass out quick on rl0 proto icmp all keep state label "let out anything from
firewall host itself"

pass out quick on rl0 all flags S/SA keep state (tcp.closed 5) label "let out
```

```
anything from firewall host itself"

anchor "firewallout" all

pass out quick on rl0 all flags S/SA keep state label "let out anything from
firewall host itself"

pass out quick on re0 all flags S/SA keep state label "let out anything from
firewall host itself"

pass out quick on rl1 all flags S/SA keep state label "let out anything from
firewall host itself"

pass out quick on tun0 all flags S/SA keep state label "let out anything from
firewall host itself openvpn"

pass in quick on tun0 all flags S/SA keep state label "let out anything from
firewall host itself openvpn"

pass out quick on tun1 all flags S/SA keep state label "let out anything from
firewall host itself openvpn"

pass in quick on tun1 all flags S/SA keep state label "let out anything from
firewall host itself openvpn"

pass out quick on tun2 all flags S/SA keep state label "let out anything from
firewall host itself openvpn"

pass in quick on tun2 all flags S/SA keep state label "let out anything from
firewall host itself openvpn"

pass in quick on re0 inet from any to 10.6.1.1 flags S/SA keep state label "anti-
lockout web rule"

block drop in log quick proto tcp from <sshlockout> to any port = ssh label
"sshlockout"

pass in log quick on rl0 inet proto udp from any to any port = $vpnports keep state
label "USER_RULE: Pass OpenVPN on WAN Interface"

pass in log quick on rl0 inet proto tcp from <AdminIP> to any port = ssh flags S/SA
keep state label "USER_RULE: Admin Allow SSH from WAN w/ log"

block drop in quick on rl0 inet proto tcp from any to any port = ssh label
"USER_RULE: Block all other SSH from WAN"

block drop in quick on rl0 inet from any to 224.0.0.0/4 label "USER_RULE: Block
Multicast from WAN entering the local subnet"

pass in quick on re0 inet proto icmp from 10.6.1.0/24 to any keep state label
"USER_RULE: Allow pings from inside"

pass in quick on re0 inet proto udp from 10.6.1.0/24 to <DNSServers> port = domain
keep state label "USER_RULE: Allow DNS from Local subnet to DNS servers"

pass in quick on re0 inet proto udp from 10.6.1.0/24 to <NTPServers> port = ntp
keep state label "USER_RULE: NTP updates from Local subnet"

block return in log quick on re0 inet proto udp from 10.6.1.0/24 to any port =
domain label "USER_RULE: Block all other DNS requests"

block drop in quick on re0 inet from any to 224.0.0.0/4 label "USER_RULE: Don't
leak Multicast from local subnet"

block drop in quick on re0 inet from <notweb> to ! 10.6.1.0/24 label "USER_RULE:
Deny all access to outer nets"

pass in quick on re0 proto tcp from <webgroup> to any port = https flags S/SA keep
```

```
state label "USER_RULE: Allow HTTPS WEB access"

pass in quick on re0 proto tcp from <webgroup> to any port = http flags S/SA keep
state label "USER_RULE: Allow WEB access"

pass in quick on re0 proto tcp from <web> to any port = ssh flags S/SA keep state
label "USER_RULE: Allow SSH"

pass in quick on re0 proto tcp from <web> to any port = 8080 flags S/SA keep state
label "USER_RULE: Allow port 8080 so that xxxx site works works"

pass in quick on re0 proto tcp from <web> to any port = xfer flags S/SA keep state
label "USER_RULE: Allow Custom web ports"

pass in quick on re0 proto tcp from <web> to any port = ftp flags S/SA keep state
label "USER_RULE: Allow FTP port"

pass in quick on re0 inet from 10.6.1.0/24 to 10.6.1.0/24 flags S/SA keep state
label "USER_RULE: Pass OpenVPN traffic comming from local subnet"

block drop in log quick all label "Default deny rule"

block drop out log quick all label "Default deny rule"

nat-anchor "natrules/*" all

nat on rl0 inet from 10.6.1.0/24 to any -> (rl0) port 1024:65535 round-robin

nat on re0 inet from 172.16.1.0/24 to 10.6.1.0/24 -> (re0) port 1024:65535 round-
robin

nat on re0 inet from 172.16.2.0/24 to 10.6.1.0/24 -> (re0) port 1024:65535 round-
robin

no rdr on re0 inet proto tcp from any to 172.16.0.0/12 port = http

no rdr on re0 inet proto tcp from any to 10.0.0.0/8 port = http

rdr on re0 inet proto tcp from any to ! (re0) port = http -> 127.0.0.1 port 80
```

## ***Приложение 2: Част от журнален файл на пакетния филтър***

```
N/A
```

## ***Списък на използваните фигури***

## Използвани материали:

1. Официален уебсайт на проекта pfSense - http://pfsense.org
2. Wikipedia - http://en.wikipedia.org/wiki/PfSense
3. Настолна книга за FreeBSD - http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook
4. Пакетна система на FreeBSD - http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports.html
5. FreeBSD Manual pages - http://www.freebsd.org/cgi/man.cgi
6. Christopher M. Buechler, Jim Pingle - pfSense: The Definitive Guide; Reed Media Services, 2009 ISBN-10: 0979034280
7. W. Richard Stevens, TCP/IP Illustrated, Volume 1; Addison-Wesley 1994, ISBN 0201633469
8. pfSense Documantation wiki - http://doc.pfsense.org/index.php/Main_Page
9. PF: The OpenBSD Packet Filter - http://www.openbsd.org/faq/pf
10. Публична инфраструктура за състоянието на pfSense - http://redmine.pfsense.org/
11. Chris Buechler, Scott Ullrich - pfSense: 2.0 and beyond, BSDCan 2009 - http://www.bsdcan.org/2009/schedule/events/130.en.html
12. Изграждане на IPSec VPN с pfSense - http://doc.pfsense.org/index.php/VPN_Capability_IPsec
13. Документация на OpenVPN - http://openvpn.net/index.php/access-server/docs.html
14. m0n0wall - http://m0n0.ch/wall/
15. IPCop - http://www.ipcop.org/
16. IPFire - http://www.ipfire.org/
17. nmap - http://nmap.org

18. Snort - http://www.snort.org
19. Zabbiz - http://www.zabbix.com
20. Varnish - https://www.varnish-cache.org
21. RRDTool - http://oss.oetiker.ch/rrdtool/index.en.html