

КУРС LL-103

Сетевое администрирование Linux

Yuriy Lebedev

SOFTLINE EDUCATION

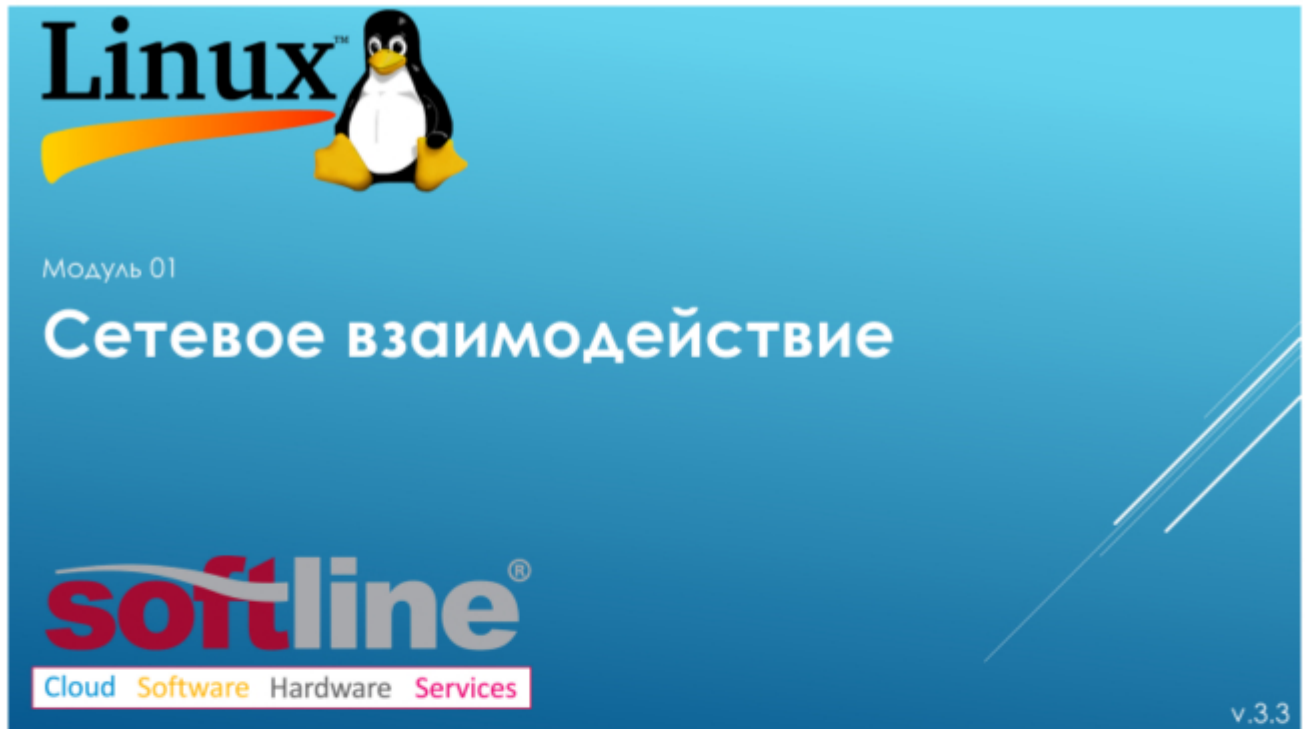
<http://edu.softline.ru/>

2017

Оглавление

Модуль 01: Сетевое взаимодействие.	2
Лабораторная работа.....	24
Модуль 02: DNS и сервер BIND.	33
Лабораторная работа.....	53
Модуль 03: Электронная почта.	61
Лабораторная работа.....	76
Модуль 04: Протокол HTTP.	88
Лабораторная работа.....	116
Модуль 05: Доступ к файловой системе.	128
Лабораторная работа.....	141
Модуль 06: Авторизация и аутентификация.....	152
Лабораторная работа.....	181
Модуль 07: Основы безопасности.....	198
Лабораторная работа.....	213
Модуль 08: Диагностика и безопасность сети	225
Лабораторная работа.....	246
Модуль 09: Системный мониторинг.	250
Лабораторная работа.....	260
Модуль 10: Безопасность сетевых сервисов.....	264
Лабораторная работа.....	273

Модуль 01: Сетевое взаимодействие.

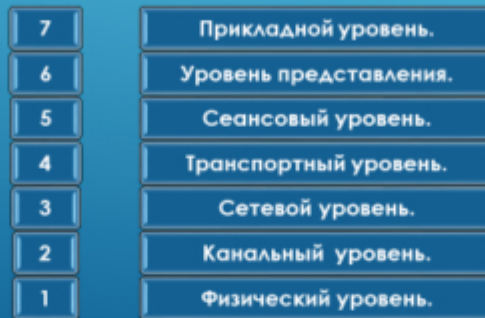


Сетевая модель OSI.

- В модуле представлены:
- Сетевая модель OSI.
- стек протоколов TCP/IP.
- Настройка сети (**ip**).
- Локальное разрешение имён.
- Маршрутизация средствами Linux.
- DHCP.

Сетевая модель OSI

- OSI - Open Systems Interconnection Basic Reference Model.
- Абстрактная модель для разработки сетевых протоколов.
- Уровневый подход, каждый уровень отвечает за свою часть процесса сетевого взаимодействия.
- 7 уровней:



<http://edu.softline.ru/>



Сетевая модель OSI (Open Systems Interconnection Reference Model) – модель взаимосвязи открытых систем). Эта модель является абстрактной, предлагая уровневую модель построения сети. Такая модель позволяет более структурно подходить к разработке сетевого программного обеспечения. Несмотря на это существует мнение, что модель очень сложна в реализации и является неудачной.

Прикладной уровень (Application Layer) – уровень, который обеспечивает взаимодействие пользователя с сетевой системой. именно на этом уровне работают конечные приложения, предлагающие высокий уровень абстракции. Типичный пример приложения, работающего на 7-м уровне модели OSI – веб-браузер. Обеспечивает формирование запросов к 6-му уровню.

Уровень представления (Presentation Layer) – уровень, обеспечивающий кодирование данных. Данные, полученные от приложений, преобразуются в формат для передачи по сети. Данные, полученные из сети, преобразуются в формат для передачи приложениям 7-го уровня.

Сеансовый уровень (Session Layer) – 5-й уровень модели OSI обеспечивает поддержание сеанса связи, что позволяет устанавливать соединения на длительное время. Синхронизация обеспечивается с помощью контрольных точек.

Транспортный уровень (Transport Layer) – данный уровень обеспечивает непротиворечивость передаваемых данных. На этом уровне происходит фрагментация данных, определение последовательности и т.д.

Сетевой уровень (Network Layer) – на сетевом уровне модели OSI определяются средства передачи данных. Определение маршрута доставки пакетов, преобразование имён и логических адресов в физические адреса и многое другое.

Канальный уровень (Data Link Layer) – этот уровень является последним программным. Предназначен для обеспечения взаимодействия сетей на физическом уровне. На 2-м уровне обеспечивается определение кадров, проверка их целостности. С точки зрения программного обеспечения, канальный уровень представлен в виде драйвера сетевого устройства.

Физический уровень (Physical Layer) – нижний уровень модели OSI предназначен для непосредственной передачи данных с помощью электрических/оптических сигналов и их приёма.

Стек протоколов TCP/IP

OSI	TCP/IP	Стек протоколов TCP/IP
Прикладной	Прикладной	HTTP DNS
Представительский		FTP POP3
Сеансовый		SMTP SNMP
Транспортный	Транспортный	TCP UDP
Сетевой	Сетевой	IPv4 IPv6
Канальный	Канальный	Ethernet Wi-Fi Мобильные сети
Физический		

<http://edu.softline.ru/>



Стек протоколов TCP/IP – набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет. Название TCP/IP происходит из двух наиважнейших протоколов семейства – Transmission Control Protocol (TCP) и Internet Protocol (IP), которые были разработаны и описаны первыми в данном стандарте.

Протоколы работают друг с другом в стеке (Stack) – это означает, что протокол, располагающийся на уровне выше, работает «поверх» нижнего, используя механизмы инкапсуляции. Например, протокол TCP работает поверх протокола IP.

Стек протоколов TCP/IP включает в себя четыре уровня:

- Прикладной уровень (Application Layer),
- Транспортный уровень (Transport Layer),
- Сетевой уровень (Internet Layer),
- Канальный уровень (Link Layer).

Протоколы этих уровней полностью реализуют функциональные возможности модели OSI. На стеке протоколов TCP/IP построено всё взаимодействие пользователей в IP-сетях. Стек является независимым от физической среды передачи данных.

Протокол IP.

Протокол IP находится на сетевом уровне сетевой модели OSI и является основой семейства протоколов TCP/IP и предназначен для обеспечения межсетевого маршрутизируемого взаимодействия. Основной особенностью является то, что протокол IP не даёт гарантии доставки данных.

На сегодняшний день основное развитие получила четвёртая версия протокола IP (IPv4). Уже в ближайшие годы стоит ожидать повсеместного внедрения протокола версии IPv6, это связано с тем, что при проектировании IPv4 не учли бурного роста сети Internet и адресное пространство стремительно заканчивается.

Протокол TCP.

TCP – один из основных протоколов сети Internet, созданный для управления передачей данных.

Основная особенность протокола TCP – работа с предварительной установкой соединения, что даёт гарантию целостности передаваемых данных. Протокол TCP обеспечивает контроль над процессом передачи информации, осуществление повторов передачи, в случае, когда часть данных была утеряна, исключение дублирования пакетов и т.д. Гарантия целостности данных ведёт за собой повышенное потребление ресурсов, а потому, в случае, когда требуется передавать данные быстро, например, голосовые, используют другие протоколы.

Протокол UDP.

В отличие от протокола TCP, требующего установления соединения, протокол UDP просто отправляет пакеты данных по одному без предварительных договорённостей, более того, удалённый узел не должен подтверждать принятие пакета. В протоколе UDP отсутствует всяческий контроль за правильностью доставки пакета.

Такая особенность делает передачу по протоколу UDP наиболее «дешевой», но и этой дешевизне есть своя цена. В случае, когда всё же нужен контроль за доставкой пакетов, его должно обеспечить само приложение.

Протокол ICMP.

Основная функция протокола ICMP-передача сообщений об ошибках и других исключительных ситуациях, возникающих при сетевом взаимодействии. Дополнительно, ICMP может использоваться для выполнения сервисных функций.

Содержимое ICMP-пакета составляют тип сообщения, код сообщения и само сообщение, зависящее от кода и типа.

Управление сетью (ip)

- **ip** – утилита командной строки в Linux из пакета **iproute**.
- Синтаксис:
 - **ip** [ОПЦИИ] ОБЪЕКТ { КОМАНДА | **help** }
- Примеры объектов:
 - Адреса IPv4/IPv6: **address**, **a**, **addr**.
 - Сетевые адаптеры: **link**, **l**.
 - Кэш ARP или NDISC: **neighbor**, **n**, **neigh**.
 - Таблица маршрутизации: **route**, **r**.
- Получение справки по объектам:
 - **# ip** ОБЪЕКТ **help**

<http://edu.softline.ru/>



ip – утилита командной строки в Linux из пакета **iproute**. Позволяет выполнять настройку сетевой подсистемы и является заменой таких утилит, как **ifconfig**, **route**, **arp**.



ПРИМЕЧАНИЕ:

В состав CentOS 7 (RHEL 7) входит **iproute-3** (CentOS 7.1.1503 – **iproute-3.10.0-21.el7.x86_64**).
В состав CentOS 6 (RHEL 6) входит **iproute-2** (CentOS 6.7 – **iproute-2.6.32-45.el6.x86_64**).

Синтаксис:

```
ip [ ОПЦИИ ] ОБЪЕКТ { КОМАНДА | help }
```

Список объектов команды **ip**, за полным списком обратитесь к документации:

Объект	Аббревиатура	Команды	Описание
address	a , addr	add, del, show, flush	Адреса IPv4/IPv6.
link	l	set, show	Сетевые адаптеры.
neighbor	n , neigh	add, del, change, replace, show, flush	Кэш ARP или NDISC
route	r	list, add, del, flush	Таблица маршрутизации

Получение справки по объектам:

```
# ip address help
# ip r help
```

Примеры использования команды **ip**.

Отображение информации по всем сетевым интерфейсам:

```
# ip a
```

Отображение только IPv4:

```
# ip -4 a
```

Отображение только IPv6:

```
# ip -6 a
```

Отображение информации по выбранным интерфейсам:

```
# ip a show eth0
# ip a list eth0
# ip a show dev eth0
```

Назначение IP-адреса на интерфейс:

```
# ip a add 192.168.0.1/255.255.255.0 dev eth0
# ip a add 192.168.0.1/24 dev eth0
```

Удаление IP-адреса с интерфейса:

```
# ip a del 192.168.1.200/255.255.255.0 dev eth0
# ip a del 192.168.1.200/24 dev eth0
```

Включение и отключение интерфейса:

```
# ip link set dev eth0 down
# ip link set dev eth0 up
```

Отображение ARP-кэша:

```
# ip n show
192.168.1.1 dev eth0 lladr 00:15:5d:46:8b:0a STALE
...
```

Последнее поле отображает состояние записи, возможные состояния:

- **STALE** – запись корректная, но будет проверена перед первой передачей.
- **DELAY** – пакет был отправлен и ожидается подтверждение.
- **REACHABLE** – запись корректна.



ПРИМЕЧАНИЕ:

В некоторых сценариях устранения неисправностей может пригодиться возможность внесения изменений в записи ARP-кэша:

Добавление записи:

```
ip neigh add 192.168.1.5 lladdr 00:1a:30:38:a8:00 dev eth0 nud perm
```

Удаление записи:

```
ip neigh del 192.168.1.5 dev eth1
```

Изменение состояния записи:

```
ip neigh chg 192.168.1.100 dev eth1 nud reachable
```

Очистка ARP-кэша:

```
ip -s -s n f 192.168.1.5
ip -s -s n flush 192.168.1.5
```

Отображение таблицы маршрутизации:

```
# ip r
# ip r list
# ip r list 192.168.10.0/24
```

Добавление нового маршрута:

```
# ip route add 192.168.10.0/24 via 192.168.1.2
```

Удаление маршрута:

```
# ip route del 192.168.10.0/24 dev eth0
```

Локальное разрешение имён

- Разрешение имён необходимо для нормального функционирования сети.
- `/etc/hosts` – локальная база данных привязок IP-адресов к символическим именам:
 - По умолчанию проверяется перед запросом к DNS-серверу.
 - Используется в малых сетях.
 - По умолчанию всегда прописан `localhost`.
- `/etc/resolv.conf` – список DNS-серверов, к которым необходимо обращаться:

```
nameserver      192.168.1.1
```

<http://edu.softline.ru/>



Система разрешения имён сделала работу в сети удобной и массовой. Без разрешения имён не обходится практически ни один запрос.

Как и в операционной системе MS Windows, Linux имеет файл `/etc/hosts`, в котором указывается соответствие имён IP-адресам. Естественно, что использование такого файла в качестве основной системы разрешения имён вызывает огромные сложности, именно поэтому файл `/etc/hosts` используется только для разрешения имени `localhost`:

```
# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1      c7-server01 localhost.localdomain  localhost
```

Учитывая, что ОС Linux является системой, очень сильно ориентированной на сетевую работу, то можно смело предложить, что она умеет работать и с сервером имён, называемым DNS (Domain Name System), чтобы указать системе адрес сервера имён, необходимо внести изменения в файл `/etc/resolv.conf`:

```
# cat /etc/resolv.conf
nameserver 10.10.10.10
nameserver 192.168.1.2
```


Маршрутизация средствами Linux

- **Quagga** – пакет обеспечивающий протоколы динамической маршрутизации IP.
Поддерживаемые протоколы динамической маршрутизации:
 - Routing Information Protocol (RIP): v1, v2, v3.
 - Open Shortest Path First (OSPF): v2, v3.
 - Border Gateway Protocol (BGP): v4.
 - Intermediate System to Intermediate System (IS-IS).
 - Protocol Independent Multicast (PIM, только PIM-SSM).
- **Quagga** — это усовершенствованная версия **GNU Zebra**.

<http://edu.softline.ru/>



Quagga — пакет свободного программного обеспечения, поддерживающий протоколы динамической маршрутизации IP. Компьютер с установленным и сконфигурированным пакетом Quagga становится способен использовать любые из нижеследующих протоколов динамической маршрутизации:

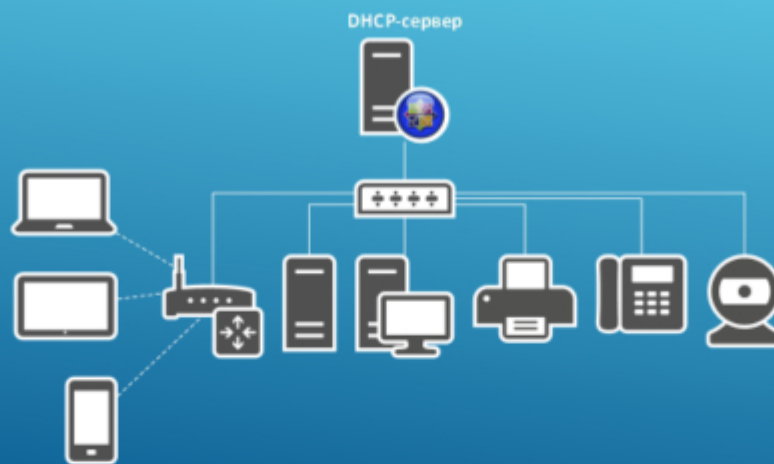
- Routing Information Protocol (RIP): v1, v2, v3.
- Open Shortest Path First (OSPF): v2, v3.
- Border Gateway Protocol (BGP): v4.
- Intermediate System to Intermediate System (IS-IS).
- Protocol Independent Multicast (PIM, только PIM-SSM).

Пакет Quagga может быть установлен на UNIX-подобные операционные системы.

Quagga — это усовершенствованная версия GNU Zebra, компьютерной программы, развитие которой остановилось в 2005 году.

DHCP

- Dynamic Host Configuration Protocol (DHCP) – это сетевой протокол автоматической настройки IP-адресов клиентов.



Dynamic Host Configuration Protocol (DHCP) – это сетевой протокол, который автоматически назначает информацию TCP/IP клиентским машинам. Каждый DHCP-клиент подключает к централизованному DHCP-серверу, который возвращает сетевые настройки (включая IP-адрес, шлюз и DNS-серверы) клиенту.

DHCP полезен для автоматической настройки сетевых интерфейсов клиентов. В момент настройки клиентской системы вы можете выбрать DHCP вместо указания IP-адреса, сетевой маски, шлюза и DNS-серверов. В таком случае клиент получит информацию от DHCP-сервера. Также DHCP-сервер может быть полезен, в случае, когда необходимо изменить IP-адреса на большом числе систем или изменить список и порядок DNS-серверов вашей организации. Когда вы перезапускаете сеть и перезагружаете систему на клиенте изменения применяются к клиенту.

Если в организации корректно настроен и функционирует DHCP-сервер, пользователи ноутбуков и мобильных ПК могут легко перемещаться между офисами каждый раз автоматически получая корректные сетевые настройки при подключении к сети офиса.

Установка DHCP-сервера



- Пакет **dhcp**, содержит Internet System Consortium (ISC) DHCP-сервер.
- Установка сервера DHCP:

```
# yum install dhcp
```
- Конфигурационный файл:
 - `/etc/dhcp/dhcpd.conf`
- Пример конфигурационного файла:
 - `/usr/share/doc/dhcp-версия/dhcpd.conf.example`

<http://edu.softline.ru/>



Пакет **dhcp**, содержит Internet System Consortium (ISC) DHCP-сервер. Для установки потребуются привилегии пользователя root:

```
# yum install dhcp
```

В процессе установки пакета **dhcp**, будет создан файл `/etc/dhcp/dhcpd.conf`, который практически пуст:

```
!!!# cat /etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#
```

Пример конфигурационного файла можно найти `/usr/share/doc/dhcp-версия/dhcpd.conf.example`. Вы можете скопировать его на место конфигурационного файла `/etc/dhcp/dhcpd.conf` и затем его отредактировать.

Настройка сервера DHCP



- Конфигурационный файл: `/etc/dhcp/dhcpd.conf`.
- Выражения в конфигурационном файле:
 - Параметры (Parameters).
 - Глобальные параметры.
 - Параметры декларации.
 - Декларации (Declarations).
 - Подсети (**subnet**).
 - Хосты (**host**).
 - Общие сети (**shared-network**).
 - Группы (**group**).

<http://edu.softline.ru/>



Первый шаг в настройке DHCP-сервера – это создание конфигурационного файла, который хранит сетевую информацию для клиентов. Используйте этот файл для определения опций для клиентских систем.

Конфигурационный файл может содержать отступы и пустые строки для упрощения форматирования. Ключевые слова не чувствительны к регистру и строки, начинающиеся с решетки `<#>` являются комментариями.

Конфигурационный файл поддерживает два типа выражений в конфигурационном файле:

- Параметры (Parameters) – определяет, как выполнять задачу, в каком случае выполнять задачу и какие опции настройки передавать клиенту.
- Декларации (Declarations) – Описывают топологию сети, предоставляют адрес клиентам, применяют групповые параметры к группе деклараций.

Параметры, которые начинаются с ключевого слова **option** также называются опциями (Options). Эти опции управляют параметрами DHCP, в то же время они могут быть обязательными и управлять поведением DHCP.

Параметры (в том числе опции) определенные до фигурных скобок (`{ }`) называются глобальными параметрами. Глобальные параметры применяются ко всем секциям ниже.



ПРИМЕЧАНИЕ:

После внесения изменений в конфигурационный файл, для того чтобы параметры вступили в силу

```
# systemctl restart dhcpd
```

**ПРИМЕЧАНИЕ:**

Вместо внесения изменений в конфигурационный файл DHCP и перезапуска службы, можно использовать команду **omshell**, которая предоставляет интерактивные возможности запросов и внесения изменений в параметры DHCP-сервера. При использовании **omshell** все изменения вступают в силу в то время как сервер остается запущенным. За дополнительной информацией обратитесь к странице руководства **man**.

Рассмотрим пример определения подсети.

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option subnet-mask 255.255.255.0;
    option domain-search "ll-100.local";
    option domain-name-servers 192.168.0.11;
    range 192.168.0.101 192.168.0.199;
}
```

Опции **routers**, **subnet-mask**, **domain-search**, **domain-name-servers**, and **time-offset** используются для всех выражений **host** описанных ниже.

Для каждой подсети, которая будет обслуживаться и для каждой подсети, к которой подключен DHCP-сервер должна быть определена одна декларация **subnet**, которая сообщает демону DHCP как распознавать адреса из этой подсети. Декларация **subnet** должна быть определена для всех подсетей, даже если в эту подсеть не будут выдаваться IP-адреса.

В примере выше определены глобальные параметры для всех клиентов DHCP в подсети и определен параметр **range**. Клиенты будут получать IP-адреса из диапазона, определенного в параметре **range**.

Рассмотрим пример определения глобальных параметров.

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
option domain-name-servers 192.168.0.11, 192.168.0.12;
option domain-search "ll-100.local";
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.101 192.168.0.199;
}
```

Для настройки DHCP-сервера, который выдает IP-адреса системам без указания параметров для отдельной подсети можно использовать глобальные параметры, они определяют время аренды и максимальное время аренды, а также параметры сети, которые будут передаваться клиентам в диапазоне от **192.168.0.101** до **192.168.0.199**.

Для назначения IP-адресов клиентам в зависимости от MAC-адреса сетевого интерфейса используйте параметр **hardware ethernet** вместе с декларацией **host**.

```
host c7-client01 {
    option host-name "c7-client01.ll-100.local";
    hardware ethernet 00:15:5d:46:86:24;
    fixed-address 192.168.0.103;
}
```


Как показано в примере выше декларация **host c7-client01** указывает, что сетевой интерфейс с MAC-адресом **00:15:5d:46:86:24** всегда будет получить IP-адрес **192.168.0.103**.



ПРИМЕЧАНИЕ:

CentOS 7 (RHEL7) поддерживает интерфейсы IPoIB (InfiniBand) и может выдавать им IP-адреса при помощи DHCP-сервера, а также использовать резервирование при помощи декларации host.

```
host apex.0 {
    option host-name "apex.example.com";
    hardware ethernet 00:A0:78:8E:9E:AA;
    option dhcp-clientidentifier=
        ff:00:00:00:00:00:02:00:00:02:c9:00:00:02:c9:03:00:31:7b:11;
    fixed-address 172.31.0.50,172.31.2.50,172.31.1.50,172.31.3.50;
}
host apex.1 {
    option host-name "apex.example.com ";
    hardware ethernet 00:A0:78:8E:9E:AB;
    option dhcp-clientidentifier=
        ff:00:00:00:00:00:02:00:00:02:c9:00:00:02:c9:03:00:31:7b:12;
    fixed-address 172.31.0.50,172.31.2.50,172.31.1.50,172.31.3.50;
}
```

Все подсети, которые разделяют одну физическую сеть должны быть определены при помощи декларации **shared-network**. Параметры, переданные внутри **shared-network**, но за пределами деклараций подсетей, являются глобальными параметрами. Имя, назначенное **shared-network**, должно быть описательным названием, таким как «**test-lab**» для описания всех подсетей в тестовом лабораторном окружении.

Рассмотрим пример определения общей сети.

```
shared-network lab-stand {
    option domain-search "ll-100.local";
    option domain-name-servers 192.168.0.11, 192.168.0.12;
    option routers 192.168.0.1;
    #more parameters for EXAMPLE shared-network
    subnet 192.168.0.0 netmask 255.255.255.128 {
        #parameters for subnet
        range 192.168.0.2 192.168.0.127;
    }
    subnet 192.168.0.128 netmask 255.255.255.128 {
        #parameters for subnet
        range 192.168.0.130 192.168.0.254;
    }
}
```

Декларация **group** используется для применения глобальных параметров к группе деклараций. Например, общие сети, подсети и хосты могут быть сгруппированы.

Рассмотрим пример определения группы.

```
group {
    option routers 192.168.0.1;
    option subnet-mask 255.255.255.0;
    option domain-search "ll-100.local";
    option domain-name-servers 192.168.0.11;
    option time-offset -18000; # Eastern Standard Time
    host c7-client01 {
        option host-name "c7-client01.ll-100.local";
        hardware ethernet 00:A0:78:8E:9E:AA;
        fixed-address 192.168.0.51;
    }
    host raleigh {
```



```
option host-name " c7-client02.ll-100.local ";  
hardware ethernet 00:A1:DD:74:C3:F2;  
fixed-address 192.168.0.52;  
}  
}
```

**ПРИМЕЧАНИЕ:**

Вы можете использовать предоставленный пример конфигурационного файла в качестве отправной точки для добавления опций настройки. Для этого скопируйте файл в соответствующее расположение, при помощи следующей команды:

```
# cp /usr/share/doc/dhcp-версия/dhcpd.conf.example /etc/dhcp/dhcpd.conf
```

Где версия – это версия DHCP.

Полный список опций и их описаний можно найти на странице руководства **`dhcp-options(5)`**.

База данных аренд



- База данных аренд клиентов DHCP:
 - `/var/lib/dhcpd/dhcpd.leases`
- База данных аренд использует Coordinated Universal Time (UTC).
- База данных аренд периодически пересоздается:
 1. Файл `dhcpd.leases` переименовывается в `dhcpd.leases~`.
 2. Временная база данных аренд переписывается в `dhcpd.leases`.

<http://edu.softline.ru/>



На сервере DHCP, файл `/var/lib/dhcpd/dhcpd.leases` хранит базу данных аренд клиентов DHCP. Этот файл не нужно править. Информация об аренде DHCP для каждого назначенного IP-адреса автоматически сохраняется в базе данных аренд. Информация включает в себя продолжительность аренды, получателя, клиента, которому был назначен IP-адрес, даты начала и окончания аренды, MAC-адрес сетевой карты (NIC), которая была использована для получения аренды.

Все записи о времени в базе данных аренд хранятся в формате Coordinated Universal Time (UTC), а не в локальном.

База данных аренд периодически пересоздается, чтобы не быть слишком большой. В первую очередь все известные аренды сохраняются во временной базе данных аренд. Файл `dhcpd.leases` переименовывается в `dhcpd.leases~` и временная база данных аренд переписывается в `dhcpd.leases`.

Демон DHCP может быть убит или система может развалиться после переименования базы данных аренд в файл резервной копии до того, как новый файл будет записан. Если это случится, файл `dhcpd.leases` не будет существовать, но будет требоваться для запуска службы. Не нужно создавать новый файл аренд, так как это приведет к потере всех старых аренд и множеству других проблем. Правильным решением будет переименование базы данных файла резервной копии `dhcpd.leases~` в `dhcpd.leases` и запуск демона.

Запуск и остановка сервера



- Запуска службы DHCP:
`# systemctl start dhcpd.service`
- Остановки службы DHCP:
`# systemctl stop dhcpd.service`
- Настройки автоматического запуска службы DHCP:
`# systemctl enable dhcpd.service`

<http://edu.softline.ru/>



ПРИМЕЧАНИЕ:

Когда сервер DHCP запускается в первый раз, он может выдать ошибку если файл **`dhcpd.leases`** не существует. Можно воспользоваться командой **`touch /var/lib/dhcpd/dhcpd.leases`** для создания файла. Если этот же сервер используется для запуска сервера BIND в качестве сервера DNS, этот шаг не обязателен, так как служба **`named`** автоматически проверяет наличие файла **`dhcpd.leases`**.

Для запуска службы DHCP используйте следующую команду:

```
# systemctl start dhcpd.service
```

Для остановки службы DHCP используйте следующую команду:

```
# systemctl stop dhcpd.service
```

Для настройки автоматического запуска службы DHCP используйте следующую команду:

```
# systemctl enable dhcpd.service
```

Отладка DHCP-сервера



- Настройка службы `dhcpcd.service`:
 - `/etc/systemd/system/dhcpd.service`
- Настройка просушки интерфейсов:
 - `ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid интерфейс (интерфейсы)`
- Опции командной строки `/usr/sbin/dhcpd`:
 - `-p номер_порта` – указывает номер UDP-порта.
 - `-f` – запуск демона на переднем плане.
 - `-d` – ведение журнала в стандартный дескриптор ошибок.
 - `-cf имя_файла` – расположение конфигурационного файла.
 - `-lf имя_файла` – расположение файла базы данных аренд.
 - `-q` – не выводить сообщение об авторском праве.

<http://edu.softline.ru/>



Если более одного сетевого интерфейса подключено к системе, но сервер DHCP должен прослушивать запросы только на одном из них, необходимо настроить сервер DHCP на просушку только этого интерфейса. Демон DHCP прослушивает только те интерфейсы, для которых он нашел декларации подсетей в файле `/etc/dhcp/dhcpd.conf`.

Это удобно для систем с двумя сетевыми интерфейсами, выполняющих функции межсетевого экрана. Одна сетевая карта может быть использована клиентом DHCP для получения IP-адреса для доступа к сети интернет. Вторая сетевая карта может быть использована сервером DHCP для выдачи IP-адресов клиентам из внутренней сети. Указание только сетевой карты, подключенной к внутренней сети делает систему более безопасной так как пользователи не могут подключиться к демону через интернет.

Для указания опций командной строки, скопируйте и отредактируйте файл `dhcpd.service` от имени пользователя `root`.

```
# cp /usr/lib/systemd/system/dhcpd.service /etc/systemd/system/
# vi /etc/systemd/system/dhcpd.service
```

Отредактируйте строки в секции **[Service]**:

```
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd
--no-pid интерфейс (интерфейсы)
```

Затем перезапустите службу:

```
# systemctl --system daemon-reload
# systemctl restart dhcpd
```

Опции командной строки могут быть дописаны в строку `ExecStart=/usr/sbin/dhcpd` в файле юнита `/etc/systemd /system/dhcpd.service` внутри секции **[Service]**.

Поддерживаются следующие опции:

- `-p номер_порта` – указывает номер UDP-порта, который должен прослушивать **dhcpd**. По умолчанию это порт **67**. Сервер DHCP передает ответы клиентам DHCP на порты выше

указанного на единицу. Например, если используется порт по умолчанию 67 для запросов, ответы отправляются клиенту на порт 68.

- **-f** – запускает демон в качестве процесса на переднем плане. Этот параметр как правило используется для отладки.
- **-d** – ведет журнал демона сервера DHCP в стандартный дескриптор ошибок. Этот параметр как правило используется для отладки. Если параметр не указан события записываются в **/var/log/messages**.
- **-cf имя_файла** – указывает расположение конфигурационного файла. По умолчанию используется **/etc/dhcp/dhcpd.conf**.
- **-lf имя_файла** – указывает расположение файла базы данных аренд. Если файл базы данных аренд существует, очень важно чтобы этот же файл использовался при каждом запуске сервера DHCP. Настоятельно рекомендуется использовать эту опцию только в отладочных целях в непроизводственной среде. Расположение по умолчанию - **/var/lib/dhcpd/dhcpd.leases**.
- **-q** – не выводить сообщение об авторском праве при запуске демона.

Агент трансляции DHCP



- Конфигурационный файл:
 - `/etc/systemd /system/dhcrelay.service`
- Синтаксис:
 - `ExecStart=/usr/sbin/dhcrelay -d --no-pid сервер (ы) -i интерфейс (ы)`
- Активации внесенных изменений:
 - # `systemctl --system daemon-reload`
 - # `systemctl restart dhcrelay`



<http://edu.softline.ru/>



DHCP Relay Agent (**dhcrelay**) позволяет перенаправлять запросы DHCP и BOOTP из подсети без сервера DHCP одному или более серверам DHCP в других подсетях.

Когда клиент DHCP запрашивает информацию, DHCP Relay Agent пересылает запрос списку серверов DHCP указанному при запуске DHCP Relay Agent. Когда сервер DHCP возвращает ответ, DHCP Relay Agent воспроизводит широкоэвещательно или адресно ответ в сети, которая отправляла оригинальный запрос.

DHCP Relay Agent для IPv4 (**dhcrelay**), прослушивает DHCPv4 и BOOTP запросы на всех интерфейсах, до тех пор пока интерфейсы не указаны в `/etc/sysconfig/dhcrelay` при помощи директивы **INTERFACES**.

dhcrelay может быть запущен как агент трансляции DHCPv4 и BOOTP (по умолчанию) или как агент трансляции DHCPv6 (при помощи аргумента `-6`). Для получения информации по использованию вызовите команду **dhcrelay -h**.

Настройка **dhcrelay** в качестве агента трансляции DHCPv4 и BOOTP.

Для запуска **dhcrelay** в режиме DHCPv4 и BOOTP, необходимо указать серверы, на которые необходимо пересылать запросы. Для этого нужно скопировать и отредактировать файл **dhcrelay.service** от имени пользователя root:

```
# cp /lib/systemd /system/dhcrelay.service /etc/systemd /system/
# vi /etc/systemd /system/dhcrelay.service
```

Отредактируйте опцию **ExecStart** в секции **[Service]** и добавьте один или несколько адресов IPv4 в конец строки, например:

```
ExecStart=/usr/sbin/dhcrelay -d --no-pid 192.168.0.1
```

Если необходимо указать интерфейсы, которые должен прослушивать DHCP Relay Agent, добавьте аргумент `-i` в строку **ExecStart** (иначе будут прослушиваться все интерфейсы), например:

```
ExecStart=/usr/sbin/dhcrelay -d --no-pid 192.168.0.1 -i eth0
```


Для получения информации о других опциях воспользуйтесь страницей руководства **dhcrelay(8)**.

Для активации внесенных изменений перезапустите службу от имени пользователя root:

```
# systemctl --system daemon-reload
# systemctl restart dhcrelay
```

Дополнительные ресурсы по **dhcpcd**



- Страницы руководства:
 - **dhcpcd(8)** – описывает принцип работы демона DHCP.
 - **dhcpcd.conf(5)** – описывает настройку конфигурационного файла DHCP.
 - **dhcpcd.leases(5)** – описывает постоянную базу данных аренд.
 - **dhcp-options(5)** – описывает синтаксис декларируемых опций DHCP.
 - **dhcrelay(8)** – описывает DHCP Relay Agent и опции его настройки.
- **/usr/share/doc/dhcp-version/** – содержит примеры файлов, файлы README и примечания о релизе к текущим версиям службы DHCP.

<http://edu.softline.ru/>



Установленная документация:

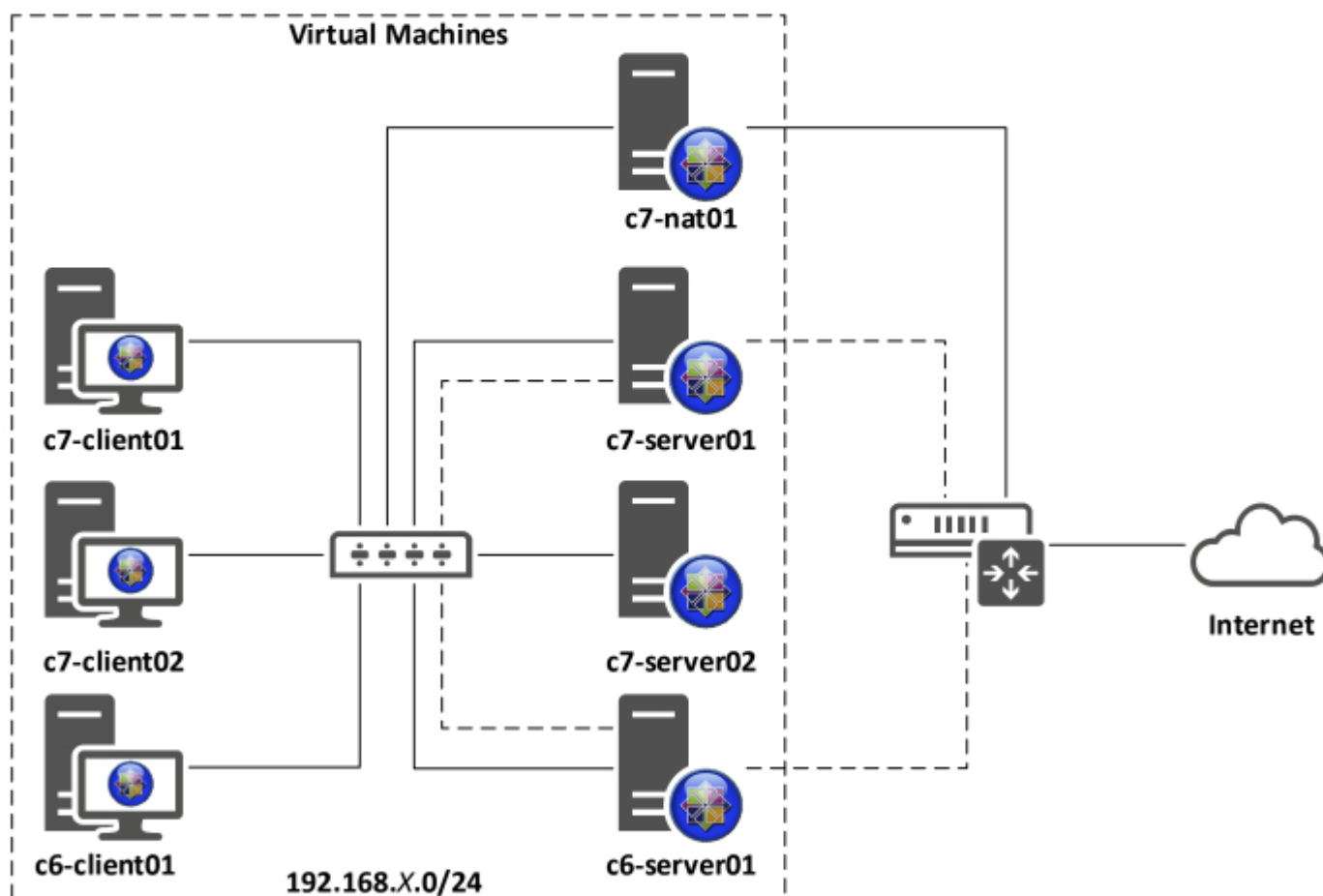
- Страница руководства **dhcpcd(8)** – описывает принцип работы демона DHCP.
- Страница руководства **dhcpcd.conf(5)** – описывает настройку конфигурационного файла DHCP с примерами.
- Страница руководства **dhcpcd.leases(5)** – описывает постоянную базу данных аренд.
- Страница руководства **dhcp-options(5)** – описывает синтаксис декларируемых опций DHCP в конфигурационном файл **dhcpcd.conf** с примерами.
- Страница руководства **dhcrelay(8)** – описывает DHCP Relay Agent и опции его настройки.
- **/usr/share/doc/dhcp-version/** – содержит примеры файлов, файлы README и примечания о релизе к текущим версиям службы DHCP.

Лабораторная работа

Лабораторная работа 1

- **Упражнения:**
 - Настройка сетевого интерфейса.
 - Работа со справочной информацией.
 - Установка и настройка сервера DHCP.
- **Необходимые машины:**
 - c7-nat01.
 - c7-server01.
 - c7-server02.
 - c7-client01 или c7-client02.
- **Учетные данные:**
 - Пользователь: root.
 - Пароль: Pa\$\$w0rd.

<http://edu.softline.ru/>





Упражнение – 1: Настройка сетевого интерфейса.

Выполняйте данное упражнение на машине **c7-server01**.

Задание 1: Управление IP адресами и интерфейсами.

1. Просмотрите конфигурацию IPv4.
2. Просмотрите конфигурацию сетевых интерфейсов **eth0** и **eth1**.
3. Отключите интерфейс **eth0**.
4. Активируйте интерфейс **eth1**.
5. Добавьте статические IP-адреса **192.168.0.61** и **192.168.0.62** интерфейсу **eth1**.
6. Удалите IP-адрес **192.168.0.62** у интерфейса **eth1**.
7. Выполните проверку сетевого взаимодействия с машиной **c7-nat01** (**ping**).

Задание 2: Управление локальной маршрутизацией:

1. Просмотрите таблицу маршрутизации.
2. Добавьте маршрут во внешнюю сеть через **c7-nat01** (**192.168.0.1**).



ПРИМЕЧАНИЕ:

Уточните конфигурацию внешней сети у инструктора, либо посмотрите ее на машине **c7-nat01**.

3. Выполните проверку сетевого взаимодействия с машиной во внешней сети (**ping**).
4. Удалите маршрут во внешнюю сеть и назначьте **c7-nat01** (**192.168.0.1**) шлюзом по умолчанию.
5. Выполните проверку сетевого взаимодействия с машиной в глобальной сети (**ping**).



Ответы к упражнению – 1: Настройка сетевого интерфейса.

Выполняйте данное упражнение на машине **c7-server01**.

Задание 1: Управление IP адресами и интерфейсами.

8. Просмотрите конфигурацию IPv4:

```
# ip -4 a
```

9. Просмотрите конфигурацию сетевых интерфейсов **eth0** и **eth1**:

```
# ip a show eth0  
# ip a show eth1
```

10. Отключите интерфейс **eth0**:

```
# ip l set dev eth0 down
```

11. Активируйте интерфейс **eth1**:

```
# ip l set dev eth1 up
```

12. Добавьте статические IP-адреса **192.168.0.61** и **192.168.0.62** интерфейсу **eth1**:

```
# ip a add 192.168.0.61/24 dev eth1  
# ip a add 192.168.0.62/24 dev eth1  
# ip a show eth1
```

13. Удалите IP-адрес **192.168.0.62** у интерфейса **eth1**:

```
# ip a del 192.168.0.51/24 dev eth1  
# ip a show eth1
```

14. Выполните проверку сетевого взаимодействия с машиной **c7-nat01** (**ping**):

```
# ping 192.168.0.1 -c4
```

Задание 2: Управление локальной маршрутизацией.

6. Просмотрите таблицу маршрутизации:

```
# ip r
```

7. Добавьте маршрут во внешнюю сеть через **c7-nat01** (**192.168.0.1**):

```
# ip r add 10.0.0.0/24 via 192.168.0.1  
# ip r
```

**ПРИМЕЧАНИЕ:**

Уточните конфигурацию внешней сети у инструктора, либо посмотрите ее на машине **c7-nat01**.

8. Выполните проверку сетевого взаимодействия с машиной во внешней сети (**ping**):

```
# ping 10.0.0.251 -c4
```

9. Удалите маршрут во внешнюю сеть и назначьте **c7-nat01** (192.168.0.1) шлюзом по умолчанию:

```
# ip r add default via 192.168.0.1
```

10. Выполните проверку сетевого взаимодействия с машиной в глобальной сети (**ping**):

```
# ping 64.233.161.197 -c4
```



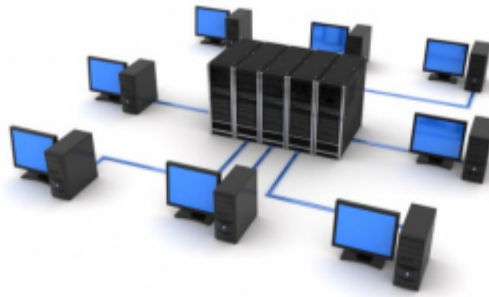

Упражнение 2 – Работа со справочной информацией.

Выполняйте данное упражнение на машине **c7-client01** или **c7-client02**.

1. Откройте файл **/etc/services** и ознакомьтесь с кратким описанием сетевых портов.
2. Произведите поиск сервисов **exec** и **biff**. Как вы можете объяснить то, что два сервиса используют одинаковые сетевые порты?

```
# grep ^exec /etc/services  
# grep ^biff /etc/services
```

3. Ознакомьтесь с содержимым файла **/etc/protocols**, содержащим информацию о сетевых протоколах.



Упражнение – 3: Установка и настройка сервера DHCP.

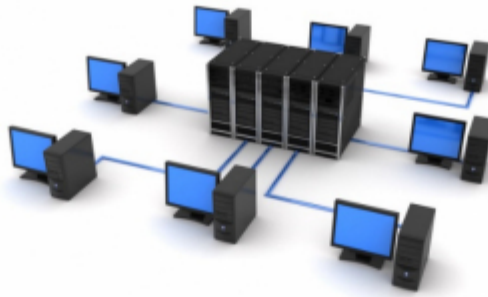
1. Остановите службу DHCP на машине **c7-nat01**.
2. Настройте статическую конфигурацию IP-адреса (192.168.0.21/24) на интерфейсе **eth0** машины **c7-server01**.
3. Установите пакет сервера DHCP (**dhcp**)
4. Создайте конфигурационный файл, либо скопируйте и отредактируйте пример конфигурационного файла (**/usr/share/doc/dhcp-версия/dhcpd.conf.example**)
5. Определите глобальные опции сервера DHCP:
 - Домен для поиска: **ll-100.local**.
 - Сервер DNS: **172.20.70.251**.



ПРИМЕЧАНИЕ:

Уточните адрес DNS-сервера у инструктора.

6. Назначьте сервер официальным для локальной сети.
7. Определите подсеть со следующими параметрами:
 - Маска подсети: **255.255.255.0**.
 - Диапазон IP-адресов: **192.168.0.100 до 192.168.0.199**.
 - Шлюз по умолчанию: **192.168.0.1**.
 - Широковещательный адрес: **192.168.0.255**.
 - Время аренды по умолчанию: **600**.
 - Максимальное время аренды: **7200**.
8. Запустите службу сервера DHCP (**dhcpd**).
9. Проверьте получение IP-адреса на машине **c7-client01**.
10. Добавьте резервацию IP-адреса **192.168.0.150** для машины **c7-client01**.
11. Перезапустите службу сервера DHCP (**dhcpd**).
12. Проверьте получение зарезервированного IP-адреса на машине **c7-client01**.
13. По аналогии создайте резервацию для машины **c7-client02**.

**Ответы к упражнению – 3: Установка и настройка сервера DHCP.**

14. Остановите службу DHCP на машине **c7-nat01**.

```
# systemctl stop dhcpd.service
```

15. Настройте статическую конфигурацию IP-адреса на интерфейсе **eth0** машины **c7-server01**.

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
...
BOOTPROTO=static
...
DEVICE=eth0
ONBOOT=yes
IPADDR=192.168.0.21
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
DNS1=172.20.70.251
DOMAIN=ll-100.local
# systemctl restart network
```

**ПРИМЕЧАНИЕ:**

Вывод файла **ifcfg-eth0** содержит только добавленные и измененные строки.

**ПРИМЕЧАНИЕ:**

Уточните адрес DNS-сервера у инструктора.

16. Установите пакет сервера DHCP (dhcp)

```
# yum install dhcp -y
```

17. Создайте конфигурационный файл, либо скопируйте и отредактируйте пример конфигурационного файла (**/usr/share/doc/dhcp-версия/dhcpd.conf.example**)

```
# vi /etc/dhcp/dhcpd.conf
```

18. Определите глобальные опции сервера DHCP:

- Домен для поиска: **ll-100.local**.
- Сервер DNS: **172.20.70.251**.

```
option domain-name "ll-100.local";
option domain-name-servers 172.20.70.251;
```

**ПРИМЕЧАНИЕ:***Уточните адрес DNS-сервера у инструктора.*

19. Назначьте сервер официальным для локальной сети.

```
authoritative;
```

20. Определите подсеть со следующими параметрами:

- Маска подсети: 255.255.255.0.
- Диапазон IP-адресов: 192.168.0.100 до 192.168.0.199.
- Шлюз по умолчанию: 192.168.0.1.
- Широковещательный адрес: 192.168.0.255.
- Время аренды по умолчанию: 600.
- Максимальное время аренды: 7200.

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.100 192.168.0.199;
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    default-lease-time 600;
    max-lease-time 7200;
}
```

21. Запустите службу сервера DHCP (**dhcpcd**).

```
# systemctl start dhcpcd
```

**ПРИМЕЧАНИЕ:***Если конфигурационный файл сконфигурирован верно, то ошибок выдано не будет.*

22. Проверьте получение IP-адреса на машине **c7-client01**.

```
# systemctl restart network
# ip a
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 00:15:5d:46:da:1e brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.100/24 brd 192.168.0.255 scope global dynamic eth0
        valid_lft 604sec preferred_lft 604sec
    inet6 fe80::215:5dff:fe46:da1e/64 scope link tentative
        valid_lft forever preferred_lft forever
...
```

23. Добавьте резервацию IP-адреса **192.168.0.150** для машины **c7-client01**.

```
host c7-client01 {
    hardware ethernet 00:15:5d:46:da:1e;
    fixed-address 192.168.0.150;
}
```

**ПРИМЕЧАНИЕ:**

Проверьте MAC-адрес при создании резервации.

24. Перезапустите службу сервера DHCP (**dhcpcd**).

```
# systemctl start dhcpcd
```

25. Проверьте получение зарезервированного IP-адреса на машине **c7-client01**.

```
# systemctl restart network
# ip a
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 00:15:5d:46:da:1e brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.100/24 brd 192.168.0.255 scope global dynamic eth0
        valid_lft 604sec preferred_lft 604sec
    inet6 fe80::215:5dff:fe46:dade/64 scope link tentative
        valid_lft forever preferred_lft forever
...
```

26. По аналогии создайте резервацию для машины **c7-client02**.

Модуль 02: DNS и сервер BIND.



В модуле представлены:

- Основные положения и концепции DNS.
- DNS-сервер BIND.
- Настройка параметров DNS-сервера BIND.
- Настройка зон DNS-сервера BIND.
- Вспомогательные утилиты.

Разрешение имён

- DNS (Domain Name System) была разработана в 1983 году Полом Мокапетрисом.
- Распределённая система преобразования имён.
- Иерархическая структура.
- Основа жизнедеятельности глобальной сети Интернет:
 - Преобразование IP-адресов в символьные имена.
 - Преобразование символьных имён в IP-адреса.
 - Средства маршрутизации электронной почты.
 - Механизм поиска сервисов в сети.

<http://edu.softline.ru/>



С момента появления сетевого IP-протокола прошло много времени, а следовательно, сетью начали пользоваться не только программисты и системные администраторы, но и обычные люди, далёкие от технических тонкостей реализации сетевого взаимодействия. IP-адрес становился неудобным, так как он тяжело запоминался, а если сервер изменял свой сетевой адрес, то его и вовсе могли потерять. Проблема была решена в 1983 году, когда впервые появилась система доменных имён (DNS). Основной задачей, ставившейся перед DNS, было преобразование IP-адресов в символьные, удобочитаемые, адреса. Сегодня для рядового пользователя использование этой системы стало прозрачным, порой, многие и не подозревают, что за любимым адресом mail.ru стоит ip-адрес.

ОСНОВНЫЕ ПОНЯТИЯ

- Зона – логическое звено в дереве имён, находящееся под управлением соответствующего сервера.
- Домен – название зоны, хранящее в себе сведения о всей иерархии имён.
- Субдомен – имя подчинённой зоны, может быть передано под управление третьих лиц.
- DNS-запрос – запрос от клиента к серверу с целью получения необходимого преобразования.
- DNS-сервер – программная реализация для обработки DNS-запросов.
- DNS-клиент – ПО, позволяющее пользователю передавать DNS-запросы серверу.

<http://edu.softline.ru/>



Зона – логическое звено в иерархии DNS. Управление зонами может быть передано третьим лицам. Родительская зона может ничего не знать о ресурсах дочерней зоны, ей достаточно лишь знать её адрес.

Домен – имя зоны, читается справа налево. Изначально существует определённое число доменов, выделяемых странам, крупным ведомствам, выделяемых в соответствии с требованиями бизнеса. Такие домены называются доменами первого уровня. Корневым доменом является точка. Таким образом доменное имя **www.example.com** можно разделить на следующие составляющие:

- **com** – домен первого уровня.
- **example** – домен второго уровня.
- **www** – домен третьего уровня.

Субдомен – имя подчинённой зоны. Например, **www** является субдоменом домена **example**.

DNS-запрос – запрос от клиента серверу с целью получения необходимого преобразования. Запросы бывают двух типов:

Рекурсивные запросы позволяют переложить всю работу на DNS-сервер. Если сервер не знает ответ на поступивший запрос, то он обратится к корневым серверам для получения адреса авторитетного, для искомой зоны, DNS-сервера. Таким образом DNS-сервер возьмёт на себя все обязательства по поиску необходимой информации;

Нерекурсивный запрос в той же ситуации вернёт адрес корневого сервера подразумевая, что DNS-клиент сам выполнит дальнейший поиск

Рекурсивные запросы используются наиболее часто, но они требуют значительных затрат ресурсов, а потому системный администратор должен чётко определить, кто может делать рекурсивные запросы, а кто нет.

Существует ряд зарезервированных доменных имён (RFC 2606):

- **.test.**
- **.example, example.com, example.net, example.org.**

- `.invalid.`
- `.localhost.`

Иерархия DNS

- Каждый домен или субдомен находится под управлением, как минимум, одного авторитетного сервера.
- Существует 13 корневых DNS-серверов (Root Hints), адреса которых никогда не меняются.
- Для обеспечения надёжности существует следующая схема:
 - Master – сервер, на котором хранятся основные данные о зоне.
 - Slave – сервер, на котором хранится копия данных, взятая с Master.
 - Slave всегда синхронизируется с Master в одностороннем порядке.

<http://edu.softline.ru/>



Все зоны находятся под управлением соответствующих организаций.

При регистрации домена необходимо указать, как минимум, два сервера, хранящих информацию о зоне, отвечающей за регистрируемый домен. Эти серверы называются **Master** и **Slave**. Первый содержит информацию о зоне, а второй её всего лишь к себе копирует и периодически синхронизирует изменения. Такая схема необходима для обеспечения отказоустойчивости, в случае, когда один из DNS-серверов будет недоступен. По той же причине существует 13 корневых DNS-серверов с идентичной информацией. Если учесть, что им приходится обрабатывать миллионы запросов, то распределение нагрузки и повышение отказоустойчивости являются необходимыми элементами жизнедеятельности сети Internet.

DNS-сервер типа Master является ответственным за эту зону, в случае получения запроса на разрешение имени, входящего в эту зону – сервер выдаст авторитетный ответ.

DNS-сервер типа Slave также выдаст авторитетный ответ. Единственное его отличие том, что информация о зоне определяется на Master-сервере, задача Slave-сервера получить эту зону и обеспечивать выполнение основных функций, когда первичный сервер, по какой-либо причине, не сможет отвечать на запросы.

Каждый DNS-сервер может иметь две таблицы соответствия имён:

Таблица для прямого преобразования имён необходима для преобразования типа:

`test.example.com -> 192.168.1.1`

Таблица для обратного преобразования имён необходима для преобразования типа:

`192.168.1.1 -> test.example.com`

Сервер DNS

- Сервер DNS предназначен для ответов на DNS-запросы, посылаемых DNS-клиентами.
- Процедура получения информации:
 - Сервер принимает запрос.
 - Если он знает ответ, то выдаст его DNS-клиенту, если не знает, то обратится к корневому DNS-серверу.
 - Корневой сервер предоставляет информацию о первом уровне доменного имени.
 - Сервер обращается по полученному адресу и последовательно доходит до адреса сервера, авторитетного для запрашиваемого доменного имени.
- Порты - 53 протоколов UDP и TCP. Протокол TCP используется в случае, когда размер ответа сервера превышает 512 байт.

<http://edu.softline.ru/>



Программная реализация DNS-сервера должна осуществлять прослушивание 53-го порта протоколов UDP и TCP с целью получения DNS-запросов и отправки соответствующих DNS-ответов.

Существует два основных типа запросов:

- Рекурсивные запросы передаются серверу с последующим ожиданием готового ответа;
- Нерекурсивные запросы передаются серверу с ожиданием указания DNS-сервера, имеющего большую информацию.

Для примера рассмотрим ситуацию, когда необходимо определить ip-адрес по доменному имени **one.two.org**, мы отправляем соответствующий запрос нашему DNS-серверу. Но этот сервер ничего не знает про это доменное имя, он не может дать авторитетный ответ, поэтому он должен найти того, кто сможет его дать. Такой поиск начинается с вопроса корневому серверу, - "Сообщи мне адрес доменного имени one.two.org". Но на этот вопрос корневой сервер тоже не знает ответа, ведь зона two.org лежит вне его области ответственности, но корневой сервер отвечает, - "Я не знаю его адреса, но я знаю адрес сервера, отвечающего за зону .org". Наш сервер теперь посылает запрос серверу, отвечающему за зону org, - "Сообщи мне адрес доменного имени one.two.org". Полученный ответ будет выглядеть так: "Я не знаю его адреса, но я знаю адрес сервера, отвечающего за зону two.org". Последним запросом станет запрос к серверу, отвечающему за зону two.org и последующим получением необходимого адреса.

С целью обеспечения оптимизации использования сетевых каналов была придумана система кэширования, она позволяет запоминать ответы DNS-серверов. Теперь, когда вы получили ответ на запрос и через некоторое время вам снова потребуется определение адреса того же доменного имени, то запрос к ответственному серверу сделан не будет, так как информация всё ещё хранится у вашего DNS-сервера.

BIND



- **Berkeley Internet Name Domain (BIND)** – является наиболее распространённой реализацией системы доменных имён:
 - Поддерживается Internet Systems Consortium.
 - Обеспечивает весь необходимый функционал.
 - Поставляется практически со всеми дистрибутивами Linux.
- **Поддерживает:**
 - Безопасные соединения DNSec.
 - Динамические обновления.
 - Запуск в окружении chroot.

<http://edu.softline.ru/>



BIND (Berkeley Internet Name Domain) – свободная и наиболее распространённая программная реализация DNS.

Существует три основные разновидности пакета: BIND 4, BIND 8, BIND 9. Версия 4 существовала в 80х годах и сейчас не развивается. Основными же версиями являются 8 и 9. В настоящий момент использовать версию ниже 9-й на новых серверах не имеет смысла.

Сервер имён BIND состоит из следующих компонент:

- Демон **named**, который осуществляет приём запросов;
- Библиотечные функции, используемые для обращения к другим DNS-серверам;
- Пользовательские программы **nslookup**, **host**, **dig** с помощью которых пользователи могут посылать DNS-запросы и принимать ответы.

Для установки DNS-сервера BIND, воспользуйтесь следующей командой:

```
# yum install bind -y
```


Настройка BIND



- Основной конфигурационный файл – `/etc/named.conf`:
 - Определение зон и их типов.
 - Определение директории хранения файлов зон.
 - Определение ACL (access list).
 - Определение каналов журналирования.
 - Определение различных опций.
- Файл читается при запуске демона `named`.

<http://edu.softline.ru/>



Настройка сервера BIND включает в себя создание основного файла-подсказок `named.conf` и создание необходимых описаний зон.

Файл `named.conf` включает в себя секции, определяющие конфигурацию DNS-сервера BIND:

- **acl** – заранее определённые списки доступа, список доступа может состоять из множества IP-адресов, в качестве разделителя используется точка с запятой.
- **options** – глобальные параметры конфигурации сервера BIND.
- **include** – включает в конфигурационный файл другой файл, очень часто используется при настройке системы удалённого управления сервером – `rndc`.
- **zone** – определение подчинённых зон различного типа.
- **logging** – определение каналов и категорий журналирования.
- **controls** – правила управления программой `rndc`.
- **trusted-keys** – ключи шифрования.
- **key** – определение аутентификационной информации.
- **view** – определение представлений.

Все конструкции, определяемые в файле `named.conf` должны оканчиваться точкой с запятой. Все секции группируются фигурными скобками. Принимаются комментарии нескольких типов:

- `//` – всё, что следует за слэшами – комментарий.
- `/* --- */` – всё, что находится между – комментарий, допускается перенос по строкам.
- `#` – всё, что идёт за символом решётки – комментарий.

Структура файла `named.conf`



- Списки доступа.
- Секция глобальных опций.
- Секция определения доступа с помощью `rndc`.
- Секция описания зон:
 - Зона – подсказка.
 - Зоны прямого просмотра:
 - `Master`.
 - `Slave`.
 - Зоны обратного просмотра:
 - `Master`.
 - `Slave`.

<http://edu.softline.ru/>



В конфигурационном файле администратор может задать списки доступа, определить тип DNS-сервера и, самое главное, указать необходимые зоны и файлы, содержащие описание этих зон.

Глобальные опции задаются в секции **options**:

```
options {
    directory "/var/named";           //каталог с файлами-зон
    forwarders { 192.168.2.1; };      //адрес вышестоящего DNS-сервера
    allow-query { local; };           //определение узлов, которые могут посылать запросы
    allow-query { any; };
    allow-transfer { 192.168.3.1; };  //определение узлов, которые могут получать зону
};
```

После определения глобальных параметров можно переходить к описаниям зон. Первая и обязательная зона называется зоной-подсказкой и содержит в себе адреса всех корневых серверов. Это необходимо, когда DNS-сервер получает запрос, на который не может дать авторитетного ответа, следовательно, он должен обратиться к тем серверам, которые помогут эту задачу решить.

```
zone "." {                           //имя для зоны-подсказки зарезервировано
    type hint;                       //специальный тип - hint
    file "named.ca";                //файл описания зоны, указывается относительно
                                    //параметра directory, указанного в секции глобальных
                                    //параметров
};
```

Теперь, когда необходимый минимум данных указан, можно переходить к указанию зон, необходимых для нашей сетевой инфраструктуры. Для описания всех зон существует специальная секция – **zone**:

```
zone "example.com" {                //указание домена, за который отвечает зона
    type master;                    //указание типа зоны
    file "example.hosts";           //файл описания зоны
    allow-transfer { any; };         //кто может получать всю зону
    allow-query { any; };           //определение тех узлов, которые могут посылать DNS-запросы
};
```

В случае, когда вы являетесь администратором подсети, то вполне вероятно, что вам понадобится зона обратного преобразования. В системе доменных имен поддерживается иерархия доменных имен, но не IP-адресов, именно поэтому обычная схема не будет работать. Для решения проблемы был придуман специальный домен, структура которого будет совпадать со структурой IP-адресов, назван этот домен был **IN-ADDR.ARPA** (Address and Routing Parameter Area Domain). Имена в этом домене образуют цифровую иерархию, которая соответствует IP-адресам. Запись производится в обратном порядке, например, доменное имя **server.example.com** с ip-адресом **192.168.1.254** будет описано как **254.1.168.192.in-addr.arpa**. Таким образом, определение зоны для подсети **192.168.1.0/24** будет выглядеть следующим образом:

```
zone "1.168.192.in-addr.arpa" {
    type master;
    file "example.rev";
    allow-transfer { any; };
    allow-query { local; };
};
```

Дружественная компания попросила разместить вас вторичную DNS-зону, являющуюся обязательной при регистрации нового домена. Давайте создадим соответствующее описание:

```
zone "friends.com" {                                //указание домена, за который отвечает зона
    type slave;                                       //указание типа зоны
    file "slaves/friends.hosts";                     //файл описания зоны
    allow-query { any; };                             //определение тех узлов, которые могут посылать DNS-запросы
    masters { 10.10.10.1; };
};
```

Списки контроля доступа (ACL)



- В файле `/etc/named.conf` возможно предопределить списки доступа дальнейшего использования.
- Список доступа состоит из:
 - Имени списка.
 - Перечисления ip-адресов и сетей, входящих в него.
- В дальнейшем списки доступа могут использоваться в таких директивах как:
 - `allow-query`
 - `allow-transfer`

<http://edu.softline.ru/>



Access Control Lists (`acl`) необходимы для упрощения администрирования и обеспечения защиты от ошибок при конфигурировании. Представим ситуацию, когда сервер Bind обслуживает три зоны: `example.com`, `example1.com`, `example2.com`. К `example.com` и `example1.com` должны иметь право делать запросы все узлы сетей `192.168.1.0/24` и `10.10.10.0/24`, но никто не должен иметь право трансфера зоны. К зоне `example2.com` могут делать запросы любые узлы, но трансфер зоны могут делать только узлы из сетей `192.168.1.0/24` и `10.10.10.0/24`. Администратор может прописать список адресов для каждой зоны, но это лишь загромодит конфигурационный файл, а в случае внесения изменений, потребует существенных временных затрат. Поэтому определим список доступа с именем `localnet`:

```
acl "localnet" { 192.168.1.0/24;
                 10.10.10.0/24; };
zone "example.com" {
    type master;
    file "example.hosts";
    allow-transfer { none; };
    allow-query { localnet; };
    allow-update { none; };
};
zone "example1.com" {
    type master;
    file "example1.hosts";
    allow-transfer { none; };
    allow-query { localnet; };
    allow-update { none; };
};
zone "example2.com" {
    type master;
    file "example2.hosts";
    allow-transfer { localnet; };
    allow-query { any; };
    allow-update { none; };
};
```


Описание зон



- Файл зоны содержит описание всех доменных имён зоны.
- Текстовый файл:
 - Начинается с указания **\$TTL** (Time To Live).
 - Первая и основная ресурсная запись – SOA (Start Of Authority).
 - Остальные ресурсные записи.
- Запись SOA имеет вид:

```
[zone] IN SOA origin contact (
    serial refresh retry expire minimum
)
```

<http://edu.softline.ru/>



На данном этапе мы имеем настроенный DNS-сервер BIND с указанием всех необходимых зон, остаётся единственное – создать файлы зон с соответствующими ресурсными записями (**RR**).

Начнём с зоны типа **hint**. Создавать такую зону не требуется, достаточно запросить список корневых серверов командой:

```
# dig @a.root-servers.net . ns > /var/named/named.ca
```

Все остальные описания зон типа master необходимо будет создать самостоятельно. Каждый файл-зона должен начинаться со специального параметра **\$TTL**, который указывает на то, сколько времени с момента получения данных из зоны они будут актуальны. Первая **RR** называется **SOA**

Запись **SOA** определяет начало описания зоны и состоит из нескольких элементов:

- **zone** – имя зоны. В качестве имени зоны можно использовать символ @, который будет обозначать значение параметра zone файла **/etc/named.conf**. Имя зоны должно быть обязательно определено, так как в дальнейшем все ресурсные записи будут обрабатываться относительно имени зоны.
- **origin** – доменное имя первичного сервера имён зоны. Данное имя не должно совпадать с именем зоны.
- **contact** – почтовый адрес администратора зоны. Символ @ должен быть заменён точкой.

Далее в скобках указывается ряд параметров, отвечающих за различные временные аспекты:

- **serial** – серийный номер зоны. Номер зоны может быть любым в пределах, отведённых под него 32 бит. Негласное правило устанавливает следующий формат серийного номера - **ГТТГММДДНН**, где **НН** – любой порядковый номер. Данный номер очень важен в случае, когда вторичный сервер обращается к первичному за новой версией зоны, он сравнивает серийные номера и если серийный номер зоны первичного сервера больше, то происходит обновление.

- **refresh** – время, когда зону необходимо обновить. По истечении указанного срока, вторичный сервер приступит к обновлению зоны с первичного DNS-сервера.
- **retry** – время через которое необходимо повторить попытку обновления зоны, в случае, если предыдущая не увенчалась успехом.
- **expire** – время, после которого вторичный сервер перестанет отвечать на запросы для данной зоны в случае, если он не сможет обновить зону с первичного сервера.
- **minimum** – время кэширования негативных ответов.

Но на самом деле запись **SOA** не является первой, так как самая первая строка должна содержать директиву **\$TTL**, отвечающую за время кэширования положительных ответов по умолчанию.

Все временные значения указываются в секундах, но возможны символы замены:

- **m** – минуты
- **h** – часы
- **d** – дни
- **w** – недели

Рассмотрим пример записи **SOA**:

```
$TTL 1d
@ 1d IN SOA ns.example.com. hostmaster.example.com. (
                                2008130612      ;Serial
                                3h                ;Refresh
                                15m               ;Retry
                                1w                ;Expire
                                1h )              ;Minimum
```

В первой строке определена директива **\$TTL**, которая говорит о том, что успешные запросы к этой зоне должны кэшироваться на одни сутки.

Во второй строке находится запись **SOA**:

- **@** – имя зоны.
- **1d** – время жизни **SOA**. Время, на которое **SOA** должна быть закэширована. Но **SOA** не кэшируется.
- **ns.example.com** – имя первичного сервера имён.
- **hostmaster.example.com** – почтовый адрес администратора зоны **hostmaster@example.com**.

После определения записи **SOA** можно переходить внесению в зону других ресурсных записей. Существует несколько типов записей:

Каждая ресурсная запись состоит из 5-и полей (за исключением некоторых типов):

- Доменное имя, по которому будут производиться запросы. В качестве доменного имени допускается использование символа "@", вместо которого будет подставлено имя зоны, указанное в **named.conf** для данного файла-зоны. Очень важно знать, что к каждому значению первого поля будет добавляться имя зоны. Если вы не хотите, чтобы это происходило, указывайте после имени точку. В качестве примера возьмём описание зоны **example.com**:


```

ns                -      ns.example.com
ns.example.com    -      ns.example.com.example.com
ns.example.com.   -      ns.example.com

```

- Значение **TTL**. Поле не является обязательным.
- Класс записи. Существует три основных класса **IN** (Internet), **CH** (Chaos) и **HS** (Hesiod). Chaos уже практически вымер, Hesiod используется в качестве надстройки к серверу Bind. Нас же интересует класс **IN**, который установлен по умолчанию и, в принципе, может не указываться.
- Тип записи.
- Выдаваемое значение.

Итак, первое, что мы должны указать – адрес сервера имён, отвечающего за данную зону. Запись типа NS описывает серверы имён, являющиеся авторитетными для данной зоны. В первом поле необходимо указать имя зоны, но если оно совпадает именем, указанным в записи **SOA**, то первое поле можно оставить пустым, таким образом следующие две записи являются идентичными:

```

example.com.      IN   NS   ns.example.com.
                  IN   NS   ns.example.com.

```

Теперь, когда клиенты обратятся в данной зоне в поисках сервера, за неё отвечающего, они получат ответ, что за зону example.com отвечает сервер **ns.example.com**. Остаётся маленькая неувязочка, кто же такой ns.example.com. Для этого необходимо создать ресурсную запись типа **A**.

```

ns                IN   A     192.168.1.1

```

Ресурсные записи



- Хост (A)
`<HOST> IN A <IP>`
- Псевдоним (CNAME)
`<ALIAS> IN CNAME <HOST>`
- Почтовый обменник (MX)
`IN MX <PRIORITY> <FQDN>`
- Сервер DNS (NS)
`IN NS <FQDN>`
- Сервис (SRV)
`_<SRV>._<PR> TTL IN SRV <PRI> <WEI> <PORT> <FQDN>`

<http://edu.softline.ru/>



Ресурсные записи DNS – записи о соответствии имени и служебной информации в системе доменных имён. Файлы зон поддерживают множество ресурсных записей, наиболее популярные представлены ниже:

Хост (A) [RFC 1035] – адресная запись, обеспечивает соответствие между именем и IP-адресом.

Ситаксис:

`<HOST> IN A <IP >`

Примеры:

```
c7-server01 IN A 192.168.1.10
c7-server02 IN A 192.168.1.11      # Два IP-адреса хоста.
              IN A 192.168.1.12
```

Псевдоним (CNAME) [RFC 1035] – Каноническое имя для псевдонима (одноуровневая переадресация).

Ситаксис:

`<ALIAS> IN CNAME <HOST>`

Примеры:

```
www IN CNAME c7-server01
slapd IN CNAME c7-server01.11-102.local.    # Хост из другой зоны.
```

Почтовый обменник (MX) [RFC 1035] – Адрес почтового шлюза для домена. Состоит из двух частей – приоритета (чем число больше, тем ниже приоритет), и адреса узла.

Ситаксис:

`IN MX <PRIORITY> <FQDN>`

Примеры:

```
IN MX 10 c7-server01.11-102.local.    # 10 – приоритетный хост.
```

IN MX 20 c7-server01.11-102.local. # 20 – альтернативный хост.



ПРИМЕЧАНИЕ:

Запись MX должна находиться в начале файла после записи NS, до записей A.

```
@      IN      SOA   primary.11-100-1.local. postmaster11-100-1.local. (
      ...
      )
@      IN      NS    primary.class.local.
server01  IN      MX    10 server01.11-100-1.local.
primary   IN      A     192.168.1.101
...
primary   IN      A     172.20.70.251
...
```

Сервер DNS (NS) [RFC 1035] – Адрес узла, отвечающего за доменную зону. Критически важна для функционирования самой системы доменных имён.

Ситаксис:

IN NS <FQDN>

Пример:

IN NS c7-server01.11-100-X.local.

Сервис (SRV) [RFC 2728] – определяет имя хоста и номер порта серверов для определенных служб. Некоторые Интернет-протоколы, такие как SIP и XMPP, часто требуют поддержки SRV-записей.

Ситаксис:

<SERVICE>.<PROTOCOL> TTL IN SRV <PRIORITY> <WEIGHT> <PORT> <FQDN>

Пример:

_sip._tcp.11-100-X.local 86400 IN SRV 0 5 5060 sip.11-100-X.local



ПРИМЕЧАНИЕ:

PTR записи располагаются только в зоне обратного просмотра, они описаны ниже.

Запись Start of Authority (SOA) располагается в зонах обоих типов, описана выше и ниже

Обратная зона

- Обратное преобразование – преобразование ip-адреса в доменное имя.
- У каждой сети должна быть обратная зона.
- Многие почтовые серверы не принимают почту от ip-адресов, которые не могут быть преобразованы.
- Ресурсная запись обратного преобразования имеет тип PTR.

<http://edu.softline.ru/>



Прямое образование необходимо для определения ip-адреса по доменному имени. И именно обратным преобразованием пользуются пользователи сети Интернет, часто, сами того не подозревая. Но для работоспособности сети, часто необходимо и обратное преобразование, определение доменного имени по ip-адресу.

Преобразование ip-адреса достаточно сложная задача и поэтому было придумано простое решение – создание специальной зоны, так что каждое преобразование ip-адреса в доменное имя будет выглядеть так же, как и прямое преобразование. Эта зона получила имя **in-addr.arpa**. Доменное имя адреса **1.2.3.4** будет выглядеть как:

4.3.2.1.in-addr.arpa

Как видно, адрес переворачивается. Но существует определённая сложность в том, что классификация проводится по границе байта, следовательно могут быть следующие доменные зоны:

3.2.1.in-addr.arpa

1.168.192.in-addr.arpa

Следовательно, более мелкие обратные зоны, начинающиеся с битовой маски 25, не могут быть делегированы хозяевам этих мелких подсетей. Поэтому, управление обратной зоной будет принадлежать хозяину класса сети, например, **1.2.3.0/24** или **1.2.0.0/16**.

Рассмотрим описание обратной зоны в DNS-сервере Bind. Для начала вы должны определить имя зоны и описать её в файле **named.conf**:

```
zone "1.168.192.in-addr.arpa" {  
    type master;  
    file "1.168.192.in-addr.arpa";  
};
```

Как видно, описание обратной зоны ничем не отличается от описания зоны для прямого преобразования. Осталось только определить необходимые ресурсные записи:

```
$TTL 1d
@ 1d IN SOA ns.example.com. hostmaster.example.com. (
    2008130612 ;Serial
    3h         ;Refresh
    15m        ;Retry
    1w         ;Expire
    1h )       ;Minimum

ns      IN      NS      ns.example.com.
1       IN      A       192.168.1.1
15      IN      PTR     ns.example.com.
        IN      PTR     admin.example.com.
```

Вспомогательные утилиты



- **whois** – вывод информации о владельце доменного имени:
`whois google.com`
- **nslookup** – осуществление запросов к DNS-серверу:
 - В интерактивном режиме позволяет определять тип запросов.
- **dig** – более мощная альтернатива утилите `nslookup`.
- **named-checkconf** – утилита проверки конфигурационного файла `/etc/named.conf`.
- **named-checkzone** – утилита проверки файлов-зон.



<http://edu.softline.ru/>

Иногда необходимо получить информацию о владельце доменного имени. Такого рода данные можно получить у регистратора домена с помощью утилиты **whois**:

```
# whois kremlin.ru
domain:      KREMLIN.RU
type:        CORPORATE
nserver:     ns.gov.ru.
nserver:     ns.relarn.ru.
nserver:     ns1.gov.ru.
nserver:     ns2.gov.ru.
state:       REGISTERED, DELEGATED
org:         Spetssvyaz FSO RF
phone:       +7 495 6062863
phone:       +7 495 6060333
fax-no:      +7 495 6060333
e-mail:      kozirev@gov.ru
e-mail:      davdv@gov.ru
e-mail:      noc@gov.ru
registrar:   RUCENTER-REG-RIPN
created:     1998.07.22
paid-till:   2008.08.01
source:      TC-RIPN
```



ПРИМЕЧАНИЕ:

Команда `whois` распространяется в пакете `jwhois`. В CentOS 6 и 7 `whois` по умолчанию не устанавливается. Для установки, в CentOS 6 можно воспользоваться репозиторием по умолчанию.

```
# yum install jwhois -y
```

Для установки в CentOS 7 дополнительно потребуется подключить репозиторий EPEL:

```
# wget http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
# rpm -ivh epel-release-7-5.noarch.rpm
```


Для осуществления запросов к ДНС-серверам используйте программу **nslookup**. Если в качестве аргумента ей передать доменное имя, то будет выполнено прямое преобразование, если ip-адрес, то будет выполнено обратное преобразование:

```
# nslookup kremlin.ru
Server:          195.38.45.65
Address:         195.38.45.65#53
Non-authoritative answer:
Name:   kremlin.ru
Address: 194.226.127.22
```

Используя **nslookup** в интерактивном режиме, вы сможете указывать, какую именно информацию необходимо получить, к какому ДНС-серверу адресовать запрос и т.д.

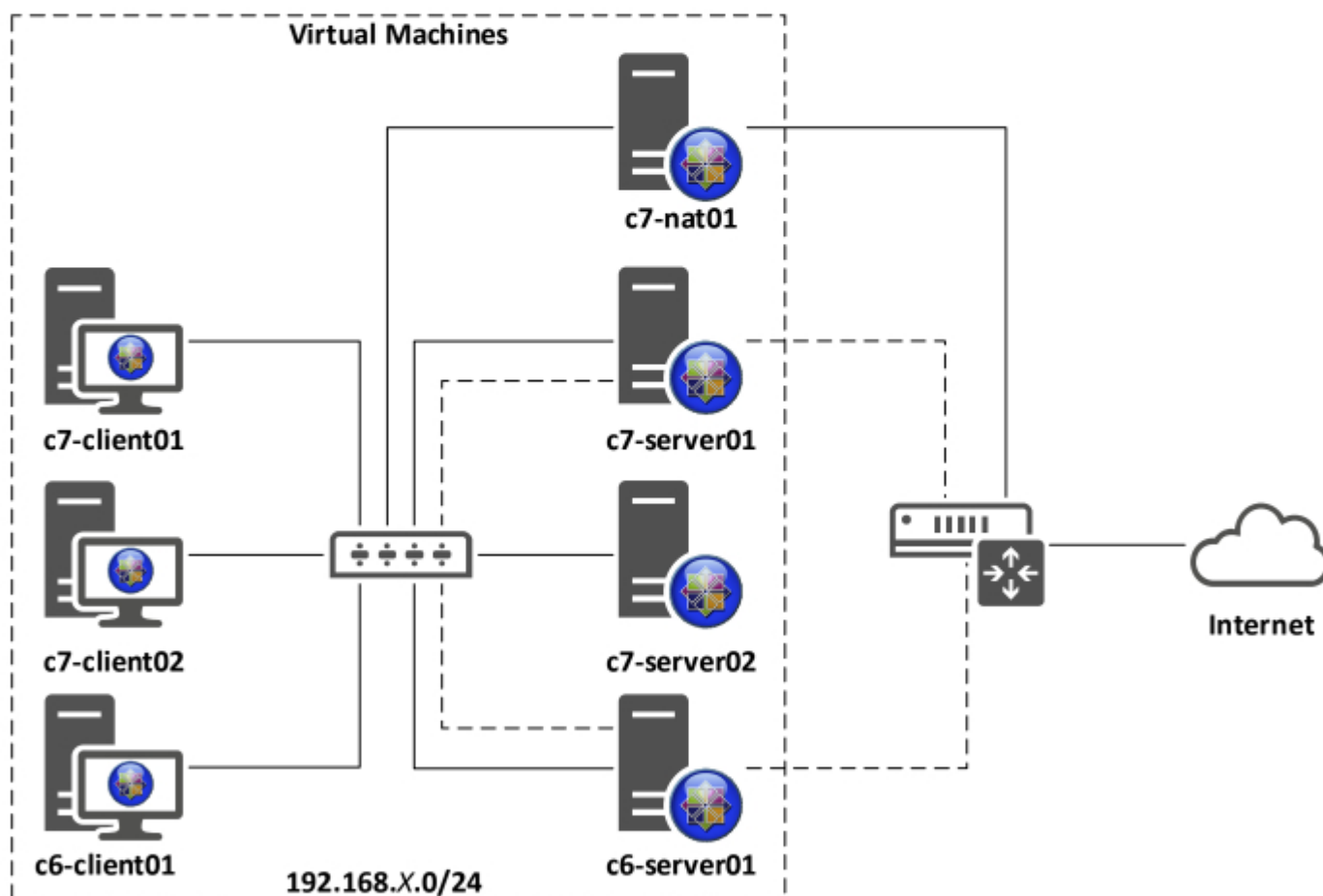
```
# nslookup
> server 10.10.10.1                # с каким сервером необходимо работать
Default server: 10.10.10.1
Address: 10.10.10.1#53
> set type=mx                      # тип запрашиваемой записи
> kremlin.ru
Server:          10.10.10.1
Address:         10.10.10.1#53
Non-authoritative answer:
kremlin.ru      mail exchanger = 15 mx0.gov.ru.
kremlin.ru      mail exchanger = 20 ns.gov.ru.
Authoritative answers can be found from:
kremlin.ru      nameserver = ns.gov.ru.
kremlin.ru      nameserver = ns.relarn.ru.
kremlin.ru      nameserver = ns1.gov.ru.
kremlin.ru      nameserver = ns2.gov.ru
```

Лабораторная работа

Лабораторная работа 2

- **Упражнения:**
 - Установка и настройка кэширующего DNS-сервера BIND.
 - Настройка зоны прямого просмотра.
 - Создание зоны обратного просмотра.
 - Репликация зон (ДОПОЛНИТЕЛЬНО).
- **Необходимые машины:**
 - c7-nat01.
 - c7-server01.
 - c7-server02.
 - c7-client01.
- **Учетные данные:**
 - Пользователь: root.
 - Пароль: Pa\$\$w0rd.

<http://edu.softline.ru/>



Упражнение 1. Установка и настройка кэширующего DNS-сервера BIND.

1. Установите пакет **bind** на машине **c7-server01**.
2. Настройте службу на автоматический запуск при старте системы.
3. Запустите службу **named**.
4. Убедитесь, что служба запустилась.
5. Настройте службу на прослушку интерфейса **eth0** и разрешите всем клиентам обращаться к DNS-серверу.
6. Добавьте исключение в **firewalld** для DNS-сервера.
7. Проверьте что исключения добавились.
8. Настройте машину **c7-client01** на использование **c7-server01** в качестве основного и единственного DNS-сервера.
9. На машине **c7-client01** проверьте разрешение внешних имен.

Ответы к упражнению 1.

1. Установите пакет **bind** на машине **c7-server01**.

```
# yum install bind
```

2. Настройте службу на автоматический запуск при старте системы.

```
# systemctl enable named
```

3. Запустите службу **named**.

```
# systemctl start named
```

4. Убедитесь, что служба **named** запустилась.

```
# systemctl status named -l
```

5. Настройте службу на прослушку интерфейса **eth0** и разрешите всем клиентам обращаться к DNS-серверу.

```
# vi /etc/named.conf
options {
    listen-on port 53 { 192.168.X.Y; };
    ...
    allow-query      { any; };
    ...
};
# systemctl restart named
```

6. Добавьте исключение в **firewalld** для порта 53 протоколов **tcp** и **udp**.

```
# firewall-cmd --permanent --add-port=53/tcp
# firewall-cmd --permanent --add-port=53/udp
# firewall-cmd --reload
```

7. Проверьте что исключения добавились

```
# iptables-save
```

8. Настройте машину **c7-client01** на использование **c7-server01** в качестве основного и единственного DNS-сервера.

```
# vi /etc/resolv.conf
nameserver 192.168.X.Y
```

9. На машине **c7-client01** проверьте разрешение внешних имен.

```
# nslookup lebedevum.blogspot.com
```

Упражнение 2. Настройка зоны прямого просмотра.

В лабораторной работе вы должны запустить dns-сервер, обслуживающий зону `11-100.local`.

1. Внесите изменения конфигурационный файл `/etc/named.conf` и определите в нём следующее:

- Директория с описаниями зон – `/var/named`.
- Сервер является авторитетным для зоны `zoneX. 11-100.local`, а описание зоны находится в файле `fwd.11-100.local.zone`.

2. Теперь, когда вы указали, что сервер является авторитетным для зоны `11-100.local`, пришло время создать описание данной зоны в файле `/var/named/ fwd.11-100.local.zone`. В соответствии со следующими требованиями:

- Время обновления зоны – 1 час.
- Время повтора обновления – 30 минут.
- Время устаревания – 1 неделя.
- Минимальное время жизни – 1 день.

После определения основных параметров зоны необходимо внести соответствующие ресурсные записи:

- Запись типа **NS** должна указывать на доменное имя `ns.fwd.11-100.local`.
- Запись типа **A** `ns.11-100.local` должна указывать на машину `C7-SERVER01`.
- Запись типа **A** `server01.11-100.local` должна указывать на машину `C7-SERVER01`.
- Запись типа **A** `client01.11-100.local` должна указывать на машину `C7-CLIENT01`.
- Запись типа **CNAME** `dns.11-100.local` должна указывать на запись `server01.11-100.local`.

3. Сохраните изменения и запустите сервер BIND.

4. С помощью команды `ping` убедитесь в правильности настройки:

5. Произведите запрос с помощью команды `nslookup`:

Ответы к упражнению 2.

1. Чтобы ограничить возможность посылать запросы вашему серверу, в секции **options** вы должны определить параметр **allow-query**:

```
options
{
    ...
    listen-on port 53 { 192.168.0.0/24;
                        127.0.0.1; };
    allow-query { 192.168.0.0/24;
                 127.0.0.1; };
    directory "/var/named";
    ...
};
```

Для указания серверу Bind на то, что он является авторитетным для **11-100.local**, необходимо определить зону типа master:

```
zone "11-100.local " {
    type master;
    file "11-100.local.dns";
};
```

2. Создайте файл-зону с именем, указанным в секции **zone** файла **/etc/named.conf** - **/var/named/11-100.local.dns**.

Первым делом определите ресурсную запись типа **SOA**, временные параметры и серийный номер:

```
@ IN SOA      ns.11-100.local.    root.11-100.local. (
                        2008010101 ;
                        1H          ;
                        30M          ;
                        1W           ;
                        1D           ;
                        )
```

Внесите необходимые ресурсные записи:

```
11-100.local. IN  A      192.168.X.Y
ns            IN  NS      ns.11-100.local.
server01      IN  A      192.168.X.Y
client01      IN  A      192.168.X.Y
dns           IN  CNAME   server01
```

3. Сохраните изменения и запустите сервер Bind

```
# systemctl restart named
```

4. С помощью команды **ping** убедитесь в правильности настройки:

```
# ping server01.11-100.local
# ping client01.11-100.local
```

5. Произведите запрос с помощью команды **nslookup**:

```
# nslookup server01.11-100.local
# nslookup client01.11-100.local
```


Упражнение 3. Создание зоны обратного просмотра.

Сделайте ваш DNS-сервер авторитетным для зоны `0.168.192.in-addr.arpa`.

1. Определите в файле `/etc/named.conf` зону `0.168.192.in-addr.arpa` типа `master`.
2. В файле `/var/named/0.168.192.zone` вы должны описать зону необходимыми ресурсными записями. Первая запись – `SOA`, для её создания используйте те же условия, что и в задании №1.
3. Создайте оставшиеся ресурсные записи:
4. Перезапустите сервер Bind.
5. Проверьте работу обратной зоны.

Ответы к упражнению 3.

Сделайте ваш DNS-сервер авторитетным для зоны `X.168.192.in-addr.arpa`.

1. Определите в файле `/etc/named.conf` зону `0.168.192.in-addr.arpa` типа `master`.

```
zone "0.168.192.in-addr.arpa" {  
    type master;  
    file "0.168.192.zone";  
};
```

2. В файле `/var/named/0.168.192.zone` вы должны описать зону необходимыми ресурсными записями. Первая запись – `SOA`, для её создания используйте те же условия, что и в задании №1.

3. Создайте оставшиеся ресурсные записи:

	<code>IN</code>	<code>NS</code>	<code>ns.11-100.local.</code>
<code>X</code>	<code>IN</code>	<code>PTR</code>	<code>server01.11-100.local.</code>
<code>Y</code>	<code>IN</code>	<code>PTR</code>	<code>client01.11-100.local.</code>

4. Перезапустите сервер Bind.

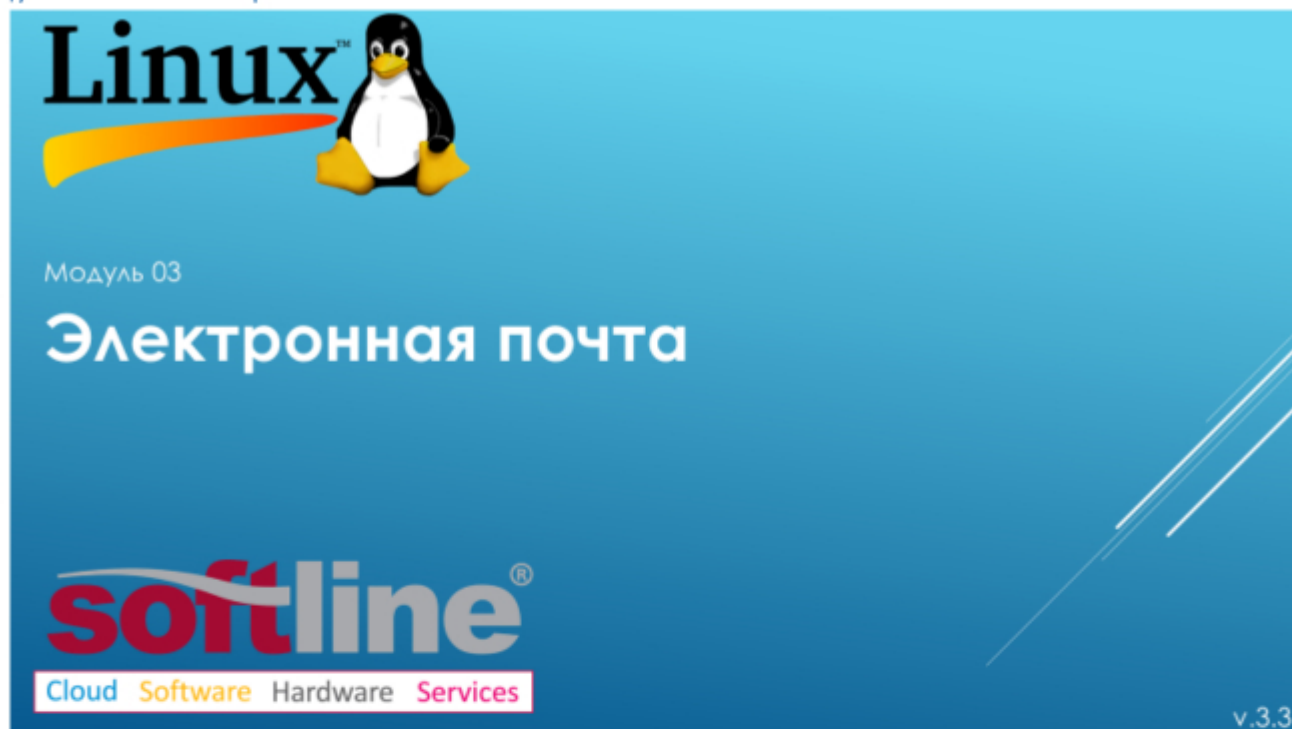
5. Запросите у вашего сервера адрес `192.168.X.Y`

```
# nslookup  
> server 192.168.X.Y  
Default server: 192.168.X.Y  
Address: 192.168.X.Y#53  
> 192.168.X.Y
```

Упражнение 4. Репликация зон (ДОПОЛНИТЕЛЬНО).

Попробуйте самостоятельно настроить репликацию зон `X.168.192.in-addr.arpa` и `11-100.local` между серверами **C7-SERVER01** и **C7-SERVER02**.

Модуль 03: Электронная почта.



В модуле представлены:

- Участники почтовой пересылки.
- Протокол SMTP.
- SMTP-сервер Postfix.
- Борьба с вирусами и спамом.
- IMAP/POP3-сервер Dovecot.
- Веб-клиенты.

Участники почтовой пересылки

- MUA (Mail User Agent) – пользовательское приложение, выполняющее передачу сообщения транспортному агенту (MTA).
- MTA (Mail Transport Agent) – приложение, обеспечивающее пересылку сообщения другому MTA.
- MDA (Mail Delivery Agent) – приложение, которое занимается непосредственной доставкой письма от MTA к пользовательскому почтовому ящику.
- MAA (Mail Access Agent) – приложение, посредством которого MUA предоставляет доступ к сообщениям.

<http://edu.softline.ru/>



Mail User Agent (**MUA**) представляет собой приложение, посредством которого пользователь имеет возможность общаться с MTA и передавать ему письма для отправки.

Mail Transport Agent (**MTA**) занимается маршрутизацией сообщения, определяет MTA, через который необходимо доставить письмо, соединяется с ним и передаёт сообщение.

Mail Delivery Agent (**MDA**) не является сетевым приложением и работает по поручению MTA. После получения сообщения, определяет пользователя, которому необходимо доставить письмо.

Mail Access Agent (**MAA**) сетевое приложение, предоставляющее интерфейс для доступа к письмам через MUA.

Протокол SMTP

- Simple Mail Transfer Protocol – простой протокол передачи почтовых сообщений.
- Предназначен для передачи сообщений, основываясь на почтовом адресе вида - user@domain.
- Программная реализация:
 - Сервер:
 - Postfix.
 - Sendmail.
 - Qmail.
 - Клиент:
 - Mozilla Thunderbird.
 - Microsoft Outlook.



<http://edu.softline.ru/>

Протокол SMTP предназначен для передачи сообщений от клиента серверу и между серверами. Для получения доступа к почтовому ящику используются протоколы IMAP и POP3.

Каждое почтовое сообщение должно содержать адрес получателя, который содержит в себе указание домена и имя почтового ящика в этом домене. За каждый домен отвечают почтовые сервера определяемые DNS-записью типа MX (Mail eXchange). В случае отсутствия записи типа MX, будет использована запись типа A.

Сервер SMTP принимает команды и возвращает коды ответа:

- **2XX** – команда успешно выполнена
- **3XX** – ожидаются дополнительные данные от клиента
- **4XX** – временная ошибка, клиент должен произвести следующую попытку через некоторое время
- **5XX** – неустраняемая ошибка

Начало сессии определяется командой **HELO** и указанием имени сервера отправителя:

```
$ telnet mail.example.com 25
220 mail.example.com ESMTP Postfix
HELO localhost
250 mail.example.com
```

Команда **MAIL** определяет отправителя почтового сообщения (поле From:):

```
MAIL FROM: admin@example.com
250 2.1.0 Ok
```

Команда **RCPT** определяет получателя почтового сообщения (поле To:):

```
RCPT TO: bill_gatez@linuxguru.com
250 2.1.5 Ok
```


Команда **DATA** определяет начало сообщения. В качестве окончания сообщения используется строка, состоящая из одной точки:

```
DATA
354 End data with <CR><LF>.<CR><LF>
```

После команды **DATA** происходит набор текста сообщения. По окончании ввода ставится одна точка:

```
Hello, Billy
How are you?
.
250 2.0.0 Ok: queued as 3B1F81000002
```

Современные почтовые серверы предоставляют множество расширений: шифрование, SMTP-авторизацию и т.д. Поэтому, согласно RFC1869, сессию необходимо начинать не командой **HELO**, а командой **EHLO**. Если почтовый сервер поддерживает расширения, то будет выдан их список, иначе – ошибка, в этом случае необходимо использовать команду **HELO**:

```
EHLO localhost
220 mail.example.com ESMTP Postfix
250-mail.example.com
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

SMTP-сервер Postfix

- **postfix** – альтернатива **sendmail**.
- Агент передачи данных MTA.
- Сервер отличают:
 - Модульность.
 - Высокий уровень гибкости.
 - Высокий уровень надёжности.
- Основной конфигурационный файл - **/etc/postfix/main.cf**:
 - Читается при запуске демона **master**.
 - **postfix reload** – перечитать конфигурацию "налету".
- Порты – 25 (smtp) протокола TCP.

<http://edu.softline.ru/>



В UNIX, в качестве MTA, изначально использовался сервер Sendmail, процедура настройки которого была очень сложной, а безопасность оставляла желать лучшего. Postfix был задуман как альтернатива и разрабатывался Вейтсом Венемой. Первые версии увидели свет в середине 1999 года. Основным преимуществом Postfix является модульность и простота настройки.

Архитектура Postfix выполнена в стиле UNIX, когда каждая программа выполняет свою конкретную функцию. Postfix не является MDA, в качестве агентов доставки почты используются отдельные программы, например, **maildrop**, **procmail**.

Конфигурационные файлы находятся в каталоге **/etc/postfix**. Основными конфигурационными файлами являются **main.cf** и **master.cf**. По умолчанию файл **main.cf** включает в себя минимальный набор параметров, определяемых как:

```
parameter = value
```

В конфигурационном файле администратор может ссылаться на определённые ранее параметры с помощью символа **\$**:

```
parameter = value
other_parameter = $parameter
```

Для обеспечения работоспособности почтового сервера вы должны определить следующие параметры:

- **myhostname** – параметр указывает имя узла почтового сервера. Именно этим именем он будет представляться другим MTA.
- **mydestination** – этот параметр должен включать в себя все имена доменов, для которых необходимо осуществлять приём почты. Имена разделяются пробелом.
- **mynetworks_style** – данный параметр определяет сетевые узлы, для которых разрешена пересылка почты. Может принимать значения **host**, **subnet**. Если указан

host, то пересылка разрешена только с этого узла. Если указано значение **subnet**, то пересылка будет разрешена для всех сетей, к которым принадлежит сервер.

- **mynetworks** – параметр определяет сети, для которых разрешена пересылка. В случае его использования, параметр **mynetworks_style** будет проигнорирован.
- **relay_domains** – параметр определяет домены, для которых разрешена пересылка почты, получаемой от анонимных узлов.

Определения вышеуказанных параметров достаточно для запуска почтового сервера postfix. Но, по умолчанию он будет принимать соединения только по адресу 127.0.0.1, что не позволит принимать

почту извне. Параметр **inet_interfaces** определяет сетевые интерфейсы, на которых необходимо принимать входящие сообщения, значение **all** укажет серверу на то, что соединения необходимо принимать на всех сетевых интерфейсах.

Рассмотрим пример конфигурации, позволяющей принимать почту для домена **example.com** и обеспечивать пересылку для сети 192.168.1.0 с маской 255.255.255.0:

```
/etc/postfix/main.cf:
myhostname = mail.example.com
mydomain = example.com
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain, localhost
mynetworks_style = subnet
mynetworks = 192.168.1.0/24
```

Если почтовый сервер не запущен, то можно запустить его, иначе применить изменения можно командой:

```
# postfix reload
```

Никогда не забывайте о том, что ваш почтовый сервер может стать жертвой спаммеров, а это не только затраты на интернет-трафик, но и существенный простой в работе, так как ваш сервер непременно попадёт в централизованный чёрный список, используемый многими администраторами, а это означает, что письма ваших сотрудников не дойдут до адресата. Перед вводом сервера в работу всегда проверяйте, как он работает. Пересылка должна быть разрешена только определённым сетям и/или пользователям. Итак, как убедиться в том, что сервер не доступен спамерам, давайте рассмотрим отправку письма по адресу postmaster@example.com на почтовый сервер, который принимает почту для этого домена.

```
telnet mail.example.com 25
Trying 1.2.3.4...
Connected to mail.example.com
Escape character is '^]'.
220 Yo, let's do it.
EHLO user.example.com
250-mail.example.com
250-PIPELINING
250-SIZE 100000000
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
```

```
250 DSN
MAIL FROM: danilamaster@cuprumhill.ru
250 2.1.0 Ok
RCPT TO: postmaster@example.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Hi there!
.
250 2.0.0 Ok: queued as 7A297208470
```

Да, письмо было помещено в очередь. Дело в том, что почтовый сервер сначала смотрит на адрес получателя, если правая часть адреса совпадает с именем домена, для которого принимает почту данный сервер, то письмо принимается. Иначе, почтовый сервер смотрит на содержимое параметра **mynetworks**, если ваш ip-адрес входит в один из диапазонов, описанных в этом параметре, то письмо будет принято и передано серверу, отвечающему за домен example.com.

Теперь представьте ситуацию, когда пересылка почты разрешена кому угодно. Любой пользователь сети Internet сможет отправлять письма через ваш сервер на любой адрес. Этого допускать нельзя. Поэтому, если мы переместимся в интернет-кафе, адреса компьютеров которого не прописаны в конфигурации Postfix, и попытаемся отправить письмо по адресу postmaster@kremlin.ru, то увидим следующее:

```
telnet mail.example.com 25
Trying 1.2.3.4...
Connected to mail.example.com
Escape character is '^]'.
220 Yo, let's do it.
EHLO user.example.com
250-mail.example.com
250-PIPELINING
250-SIZE 1000000000
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: danilamaster@cuprumhill.ru
250 2.1.0 Ok
RCPT TO: postmaster@kremlin.ru
554 5.7.1 <postmaster@kremlin.ru>: Relay access denied
```

Если ваш сервер ведёт себя точно так, будьте спокойны, он настроен верно.

Дополнительная настройка

- **Конфигурационные файлы:**
 - `/etc/aliases` – база соответствий виртуальных адресов реальным.
 - `/etc/postfix/access` – база доступа к smtp-серверу.
 - `/etc/postfix/transport` – база транспортов.
- **Утилиты:**
 - `postmap` – управление базами данных Postfix.
 - `postconf` – управление конфигурацией Postfix.

<http://edu.softline.ru/>



Почтовый сервер Postfix использует дополнительные конфигурационные файлы для более лёгкого и удобного управления различными параметрами.

База данных «псевдонимов», находящаяся в файле `/etc/aliases` призвана установить соответствие между виртуальными именами почтовых ящиков и реальными пользователями:

```
# Basic system aliases -- these MUST be present.
mailer-daemon: postmaster
postmaster:      root
# General redirections for pseudo accounts.
bin:             root
daemon:          root
adm:             root
...
root:            ivan
```

Таким образом, содержимое этого файла указывает МТА на то, что письма, предназначенные для **postmaster**, **bin**, **daemon** и т.д., необходимо отдавать пользователю **ivan**. Пользователь **ivan** будет получать все письма, предназначенные для вышеуказанных адресов.

Файл `/etc/postfix/access` призван обеспечить управление доступом к почтовому серверу. Здесь администратор может определить узлы, для которых разрешена работа с сервером:

```
10.10.10 OK
192.168.1 REJECT
```

Эти строки означают, что для сети 10.10.10.0/24 работа разрешена, а для сети 192.168.1.0/24 – запрещена.

В файле `/etc/postfix/transport` указываются узлы, которым необходимо переслать сообщение, подпадающее под определённые критерии. Например, администратор захотел настроить почтовый сервер таким образом, чтобы все письма для `director@example.com` перенаправлялись на другой МТА:

```
director@example.com      smtp:other_smtp.example.com
```

Все конфигурационные файлы должны быть проиндексированы с помощью команды `postmap`, создающей на основе этих файлов одноимённые, но с расширением «db».

```
# postmap /etc/postfix/transport
# ls -l /etc/postfix/transport*
-rw-r--r-- 1 root root 12357 Окт  4 2007 /etc/postfix/transport
-rw-r--r-- 1 root root 12288 Авг 13 02:45 /etc/postfix/transport.db
```

Утилита `postconf` предназначена для внесения изменения в конфигурацию почтового сервера Postfix. Вызванная без параметров, она выдаст список всех параметров и их значения. Для изменения параметра в файле `/etc/postfix/main.cf` необходимо использовать параметр «-e». Например, если вы захотите изменить максимальный размер почтового сообщения, то необходимо дать следующую команду:

```
# postconf -e message_size_limit=20480000
```

В случае, если вам понадобится узнать умолчательные значения, используйте параметр «-d»

Milter Manager

- Milter (Mail Filter) – расширение, позволяющее администраторам добавлять средства фильтрации к МТА.
- Позволяет реализовать фильтрацию входящих сообщений на этапе SMTP-сессии.
- Расширение поддерживается следующими МТА:
 - postfix.
 - sendmail.
- Каталог подключаемых фильтров – <http://milter-manager.sourceforge.net/>

<http://edu.softline.ru/>



Расширение Milter предоставляет разработчикам интерфейс для подключения дополнительных фильтров. При обработке сообщения, МТА, согласно конфигурации, передаёт его тому или иному фильтру. Задача фильтра заключается в обработке письма и возврату результата обработки.

Почтовый сервер Postfix включает в себя расширение Milter, что позволяет администраторам системы расширять функционал данного МТА. Основной тип используемых фильтров – SMTP Milters, что означает, что данные фильтры работают в рамках SMTP-сессии. Чтобы осуществить подключение фильтра необходимо использовать параметр `smtpd_milters`, который имеет следующий синтаксис:

```
smtpd_milters = inet:localhost:portnumber, unix:/path/to/file ...other filters...
```

В случае, когда фильтр ожидает соединения на TCP-порт, необходимо использовать параметр **inet**. Если же фильтр ожидает соединения на локальный unix-domain сервер, то необходимо использовать параметр **unix**.

Пример подключения фильтров:

```
smtpd_milters = unix:/var/run/spamassassin/spamass-milter.socket,inet:localhost:1234
```

Документация Postfix содержит описание множества дополнительных параметров, влияющих на работу Postfix с расширением Milter. Отдельно стоит отметить следующие параметры:

- **milter_default_action** – действие по умолчанию, которое будет применено в том случае, когда фильтр, по каким-то причинам, завершил работу с ошибкой.
- **milter_connect_timeout** – максимальное время, отводимое на соединение с фильтром, по прошествии которого будет применено действие **milter_default_action**.
- **milter_command_timeout** – максимальное время, отводимое на передачу команды фильтру.
- **milter_content_timeout** – максимальное время, отводимое на передачу сообщения фильтру.

Борьба со спамом

- Для борьбы со спамом существует несколько основных средств:
 - Проверка отправителя на соответствие стандартам.
 - Проверка сообщений с использованием различных фильтров:
 - Байесовский фильтр.
 - Использование чёрных списков ip-адресов, рассылающих нежелательную корреспонденцию.
 - Использование серых списков.
- Существует мнение, что "Электронную почту убьёт не спам, а борьба со спамом" – не переусердствуйте.

<http://edu.softline.ru/>



Почтовый сервер Postfix имеет все необходимые средства для эффективной борьбы со спамом. Первое, что должен сделать администратор системы, определить требования, которым должен соответствовать отправитель

Параметр `smtpd_sender_restrictions` определяет требования, предъявляемые отправителю:

```
smtpd_sender_restrictions =
    permit_mynetworks,
    reject_unknown_sender_domain,
    reject_rbl_client sbl-xbl.spamhaus.org
```

В данном примере дальнейшая работа по отправке письма будет разрешена всем узлам, входящим в параметр `mynetworks`. Те клиенты, у которых доменная часть адреса не имеет DNS-записи типа A или MX – будут отвергнуты. Также будут отвергнуты те, кто находятся в черных списках сервиса `spamhaus.org`.

Программа `Spamassassin` является воплощением целого ряда технологий, включая и мощный эвристический анализ. Система построена на огромном множестве критериев, каждый из которых определяет некоторое количество баллов. При проверке письма, к нему будут применены все критерии и каждый подходящий будет увеличивать сумму баллов для данного письма. Администратором указывается пороговое значение, после которого необходимо признавать письмо как спамовое. Основным конфигурационным файлом является `/etc/mail/spamassassin/local.cf`.

В конфигурационном файле вы можете указать используемые типы проверки. Переопределить значения различных критериев. Это необходимо, когда специфика вашей работы подразумевает получения множества полезных писем, которые подпадают под какой-то из весомых критериев и всё время определяется как спам.

```
# Включение проверок
auto_whitelist_path    /var/spool/spamassassin/auto-whitelist
bayes_path             /var/spool/spamassassin/bayes
# Минимальное количество примеров спамовых и неспамовых сообщений, при котором начинает
# работать Баесовский фильтр
bayes_min_ham_num      10
bayes_min_spam_num      10
```

```

# Пороговое значение
required_score 5
# Вносить изменение в тему письма, если оно признано спамом
rewrite_header subject *****SPAM*****
# Включать само сообщение в уведомлении
report_safe 1
# Включить Баесовский фильтр
use_bayes 1
# Включить автоматическое обучение баесовского фильтра
bayes_auto_learn 1
# Доверительные сети
trusted_networks 192.168.1/24
# Письма с указанного домена проверять не требуется
whitelist_from *@example.com
# Переопределение баллов для критерия
score NAME_OF_TEST 0

```

Теперь необходимо указать почтовому серверу Postfix использовать фильтр. Для этого существует специальный параметр **smtpd_milters** (слово Milter происходит от Mail Filter). С помощью этого параметра вы можете указать фильтры, которым необходимо передать письмо. После проверки, фильтр вернёт письмо почтовому серверу для дальнейшей доставки. Укажем, что при доставке необходимо использовать сокет, создаваемый Spamassassin:

```
smtpd_milters = unix:/var/run/spamassassin/spamass-milter.socket
```

Остаётся запустить Spamassassin и специальную программу Spamass-milter, обеспечивающую транспорт между Postfix и Spamassassin.

```

# service spamassassin start
# service spamass-milter start

```

В файле **/etc/sysconfig/spamass-milter** вы можете указать путь к файлу-сокету.

IMAP/POP3-сервер Dovecot

- Dovecot – POP3/IMAP4 сервер.
- Агент доступа к почте - MAA.
- Сервер отличают:
 - Высокий уровень безопасности.
 - Простота настройки.
 - Быстродействие.
- Основной конфигурационный файл – `/etc/dovecot.conf`.
- Порты – 110 (pop3), 143 (imap), 993 (pop3s), 995 (imaps) протокола TCP.

<http://edu.softline.ru/>



Нам удалось настроить и запустить почтовый сервер, принимающий почту. Мы знаем, что почта была получена, но где же она размещается? Почта всех пользователей, по умолчанию, помещается в каталог `/var/spool/mail` и хранится там в формате mbox (обычный текстовый файл).

Для доставки почты пользователю было создано два основных протокола:

POP3 – Post Office Protocol Version 3, позволяющий забрать письма из почтового ящика и поместить их в почтовую программу пользователя.

IMAP – Internet Message Access Protocol, предоставляющий доступ пользователя к письмам на сервере.

Протокол IMAP позволяет реализовать следующие преимущества:

- Все письма хранятся на сервере, что позволяет читать входящую корреспонденцию в любой точке планеты.
- Использование виртуальных каталогов позволяет грамотно организовать хранение почты.
- Возможно создание общих каталогов.

Программный продукт dovecot является реализацией протоколов POP3 и IMAP. Настройка достаточно проста и сводится к редактированию файла `/etc/dovecot.conf`. Поскольку сервер dovecot поддерживает разные протоколы, то первое, что необходимо определить, какие протоколы будут включены, это делается с помощью параметра:

```
protocols = imap pop3
```

Теперь достаточно запустить сервер, подключиться к нему и после прохождения авторизации, получить доступ к почтовому ящику.

Веб-клиенты

- Клиент электронной почты (MUA) с веб-интерфейсом.
 - **Squirrelmail.**
 - Клиент электронной почты с веб-интерфейсом на PHP.
 - Использует IMAP и SMTP для связи с почтовым сервером.
 - Для работы Squirrelmail необходим веб-сервер.
 - **Roundcube.**
 - Клиент электронной почты с веб-интерфейсом на PHP, CSS, XHTML, AJAX.
 - Может использовать БД: MySQL, PostgreSQL, SQLite, MSSQL или Oracle Database.
 - Использует IMAP и SMTP для связи с почтовым сервером.
 - Необходим веб-сервер с поддержкой PHP 5.3.7 или выше.



roundcube

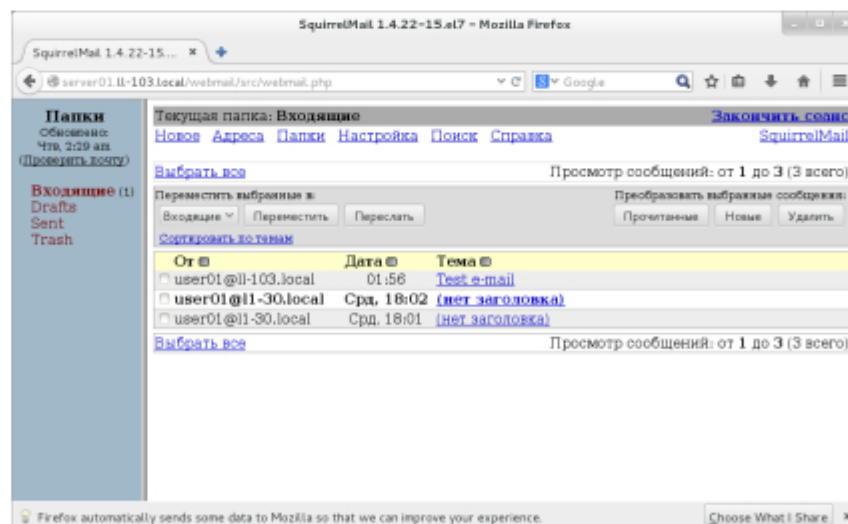


<http://edu.softline.ru/>

SquirrelMail – клиент электронной почты (MUA) с веб-интерфейсом, написанный на PHP. Разработка Натана Эрисмана (Nathan Ehresman) и Люка Эрисмана (Luke Ehresman). Приложение может быть установлено практически на любом веб-сервере, на котором установлен PHP и имеется связь с почтовым сервером по протоколам IMAP и SMTP.

SquirrelMail формирует страницы, соответствующие стандарту HTML 4.0 (без использования JavaScript), что делает его совместимым с большинством современных веб-браузеров. SquirrelMail использует плагины для разделения дополнительных функций и основного приложения.

Squirrelmail выпускается под лицензией GNU GPL, и является свободным программным обеспечением. Приложение доступно на 49 языках, в том числе на русском языке.



Для настройки Squirrelmail используйте интерактивный мастер
`/usr/share/squirrelmail/config/conf.pl:`

```
# cd /usr/share/squirrelmail/config/
# ./conf.pl
```

```

SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color off
S Save data
Q Quit

Command >> █

```

Основной конфигурационный файл: `/etc/squirrelmail/config.php`

После настройки необходимо добавить виртуальный хост (vhost) в конец конфигурационного файла Apache HTTP Server – `httpd.conf`.

```

Alias /webmail /usr/share/squirrelmail
<Directory /usr/share/squirrelmail>
    Options Indexes FollowSymLinks
    RewriteEngine On
    AllowOverride All
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>

```

Для работы Squirrelmail потребуется запустить службу httpd:

```
# systemctl start httpd
```

И настройте ее автоматический запуск:

```
# systemctl enable httpd
```

В указанном примере виртуального хоста доступ осуществляется по ссылке: <http://IP/webmail> или <http://FQDN/webmail>.

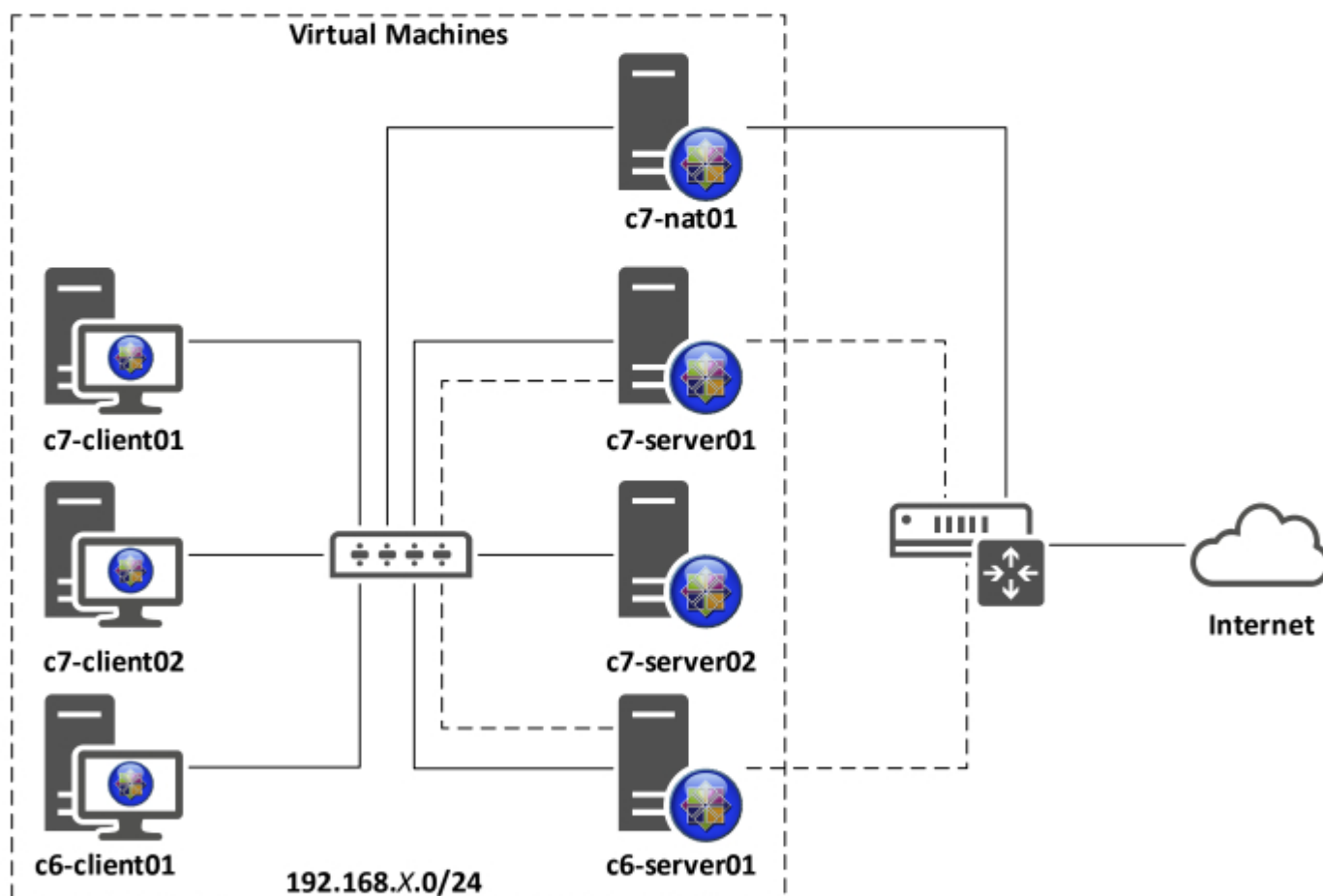
На этапе настройки и тестирования можно использовать PHP-скрипт <http://IP/webmail/src/configtest.php>, который отображает настройки Squirrelmail. Но в производственной среде доступ к нему лучше ограничить, так как с его помощью можно узнать настройки и версию SquirrelMail. Также можно в конфигурационном файле включить опцию `hide_sm_attributions`, чтобы на странице приветствия не отображалась версия.

Лабораторная работа.

Лабораторная работа 3

- **Упражнения:**
 - Подготовка сервера и окружения.
 - Установка и настройка Postfix.
 - Установка и настройка Dovecot.
 - Установка и настройка Squirrelmail (ДОПОЛНИТЕЛЬНО).
- **Необходимые машины:**
 - c7-nat01.
 - c7-server01.
 - c7-client01 или c7-client02.
- **Учетные данные:**
 - Пользователь: root.
 - Пароль: Pa\$\$w0rd.

<http://edu.softline.ru/>



Упражнение 1 – Подготовка сервера и окружения.

1. Убедитесь, что MTA sendmail не установлен.
2. В файл `/etc/hosts` добавьте запись, содержащую IP, FQDN и имя сервера почтового сервера.
3. Убедитесь что MX запись присутствует в зоне `11-100-X.local`.
4. Отключите SELinux (`/etc/sysconfig/selinux`).
5. Установите пакет `telnet`.
6. Откройте в `firewalld` порт 80 (`tcp`).
7. Добавьте пользователей `user01` и `user02` и задайте им пароль `Pa$$w0rd`.

Ответы к упражнению 1.

1. Убедитесь, что MTA sendmail не установлен.

```
# rpm -q sendmail
```

2. В файл /etc/hosts добавьте запись, содержащую IP, FQDN и имя сервера почтового сервера.

```
# vi /etc/hosts
192.168.X.Y    server01.11-100-X.local    server01
```

3. Убедитесь что MX запись присутствует в зоне 11-100-X.local.

```
# nslookup
set type=mx
11-100-X.local
```

4. Отключите SELinux (/etc/sysconfig/selinux).

```
# vi /etc/sysconfig/selinux
SELINUX=disabled
```

5. Установите пакет telnet.

```
# yum install telnet
```

6. Откройте в firewalld порт 80 (tcp).

```
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
```

7. Добавьте пользователей user01 и user02 и задайте им пароль Pa\$\$w0rd.

```
useradd user01
passwd user01
useradd user02
passwd user02
```

Упражнение 2 – Установка и настройка Postfix.

1. Установите Postfix (**postfix**).
2. Заполните следующие параметры в конфигурационном файле (`/etc/postfix/main.cf`).

Номер строки	Параметр	Значение	Комментарий.
76	<code>myhostname</code>	<code>server01.11-100-X.local</code>	
83	<code>mydomain</code>	<code>11-100-X.local</code>	
99	<code>myorigin</code>	<code>\$mydomain</code>	
113	<code>inet_interfaces</code>	<code>all</code>	
116	<code>inet_interfaces</code>	<code>localhost</code>	Закомментируйте.
119	<code>inet_protocols</code>	<code>All</code>	Для IPv4
164	<code>mydestination</code>	<code>\$myhostname, localhost.\$mydomain, localhost</code>	Закомментируйте
165	<code>mydestination</code>	<code>\$myhostname, localhost.\$mydomain, localhost, \$mydomain</code>	
264	<code>mynetworks</code>	<code>192.168.X.0/24, 127.0.0.0/8</code>	
419	<code>home_mailbox</code>	<code>Maildir/</code>	

3. Запустите службу **postfix** и настройте ее автоматический запуск.
4. Проверьте работу postfix при помощи **telnet**.
5. Проверьте что в почтовый ящик пользователя **user02** пришло письмо.
6. Проверьте содержимое письма.

Ответы к упражнению 2.

1. Установите Postfix.

```
# yum install postfix
```

2. Заполните следующие параметры в конфигурационном файле (/etc/postfix/main.cf).

```
# vi /etc/postfix/main.cf
```

Номер строки	Параметр	Значение	Комментарий.
76	myhostname	server01.11-100-X.local	
83	mydomain	11-100-X.local	
99	myorigin	\$mydomain	
113	inet_interfaces	all	
116	inet_interfaces	localhost	Закомментируйте.
119	inet_protocols	All	Для IPv4
164	mydestination	\$myhostname, localhost.\$mydomain, localhost	Закомментируйте
165	mydestination	\$myhostname, localhost.\$mydomain, localhost, \$mydomain	
264	mynetworks	192.168.X.0/24, 127.0.0.0/8	
419	home_mailbox	Maildir/	

3. Запустите службу postfix и настройте ее автоматический запуск.

```
# systemctl start postfix
# systemctl status postfix
# systemctl enable postfix
# systemctl is-enabled postfix
```

4. Проверьте работу postfix при помощи telnet.

```
# telnet localhost smtp
Trying ::1...
Connected to localhost.
Escape character is '^]'.
220 server01.11-100-1.local ESMTP Postfix
ehlo localhost
250-server01.11-100-1.local
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from:user01
250 2.1.0 Ok
rcpt to:user02
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: Test e-mail
CentOS 7 Mail Server
.
250 2.0.0 Ok: queued as 24DD820D74A0
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

5. Проверьте что в почтовый ящик пользователя **user02** пришло письмо.

```
ls /home/user02/Maildir/new/  
1443636116.Vfd00I667M50503.c7-server01
```

6. Проверьте содержимое письма.

```
# cat /home/user02/Maildir/new/1443636116.Vfd00I667M50503.c7-server01  
Return-Path: <user01@11-30.local>  
X-Original-To: user02  
Delivered-To: user02@11-30.local  
Received: from localhost (localhost [IPv6:::1])  
        by server01.11-100-1.local (Postfix) with ESMTP id 24DD820D74A0  
        for <user02>; Thu,  1 Oct 2015 00:01:20 +0600 (OMST)  
Message-Id: <20150930180129.24DD820D74A0@server01.11-100-1.local>  
Date: Thu,  1 Oct 2015 00:01:20 +0600 (OMST)  
From: user01@11-100-1.local  
CentOS 7 Mail Server
```


Упражнение 3 – Установка и настройка Dovecot.

1. Установите Dovecot (**dovecot**).
2. Внесите изменения в файл **/etc/dovecot/dovecot.conf**.

Номер строки	Параметр	Значение	Комментарий.
24	protocols	imap pop3 lmtp	

3. Внесите изменения в файл **/etc/dovecot/conf.d/10-mail.conf**.

Номер строки	Параметр	Значение	Комментарий.
24	mail_location	maildir:~/Maildir	

4. Внесите изменения в файл **/etc/dovecot/conf.d/10-auth.conf**.

Номер строки	Параметр	Значение	Комментарий.
10	disable_plaintext_auth	yes	
100	auth_mechanisms	plain login	

5. Внесите изменения в файл **/etc/dovecot/conf.d/10-master.conf**.

Номер строки	Параметр	Значение	Комментарий.
91	user	postfix	
92	group	postfix	

6. Запустите службу **dovecot** и настройте ее автоматический запуск.
7. Проверьте работу POP3 при помощи **telnet**.
8. Проверьте работу IMAP при помощи **telnet**.

Ответы к упражнению 3.

1. Установите Dovecot.

```
# yum install dovecot
```

2. Внесите изменения в файл `/etc/dovecot/dovecot.conf`.

```
# vi /etc/dovecot/dovecot.conf
```

Номер строки	Параметр	Значение	Комментарий.
24	protocols	imap pop3 lmtp	

3. Внесите изменения в файл `/etc/dovecot/conf.d/10-mail.conf`.

```
# vi /etc/dovecot/conf.d/10-mail.conf
```

Номер строки	Параметр	Значение	Комментарий.
24	mail_location	maildir:~/Maildir	

4. Внесите изменения в файл `/etc/dovecot/conf.d/10-auth.conf`.

```
# vi /etc/dovecot/conf.d/10-auth.conf
```

Номер строки	Параметр	Значение	Комментарий.
10	disable_plaintext_auth	yes	
100	auth_mechanisms	plain login	

5. Внесите изменения в файл `/etc/dovecot/conf.d/10-master.conf`.

```
# vi /etc/dovecot/conf.d/10-master.conf
```

Номер строки	Параметр	Значение	Комментарий.
91	user	postfix	Раскомментировать
92	group	postfix	Раскомментировать

6. Запустите службу `dovecot` и настройте ее автоматический запуск.

```
# systemctl start dovecot
# systemctl status dovecot
# systemctl enable dovecot
# systemctl is-enabled dovecot
```

7. Проверьте работу POP3 при помощи `telnet`.

```
# telnet localhost pop3
Trying ::1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot ready.
user user02
+OK
pass Pa$$w0rd
+OK Logged in.
retr 1
+OK 421 octets
```

```

Return-Path: <user01@11-30.local>
X-Original-To: user02
Delivered-To: user02@11-30.local
Received: from localhost (localhost [IPv6:::1])
        by server01.11-100-1.local (Postfix) with ESMTTP id 24DD820D74A0
        for <user02>; Thu,  1 Oct 2015 00:01:20 +0600 (OMST)
Message-Id: <20150930180129.24DD820D74A0@server01.11-100-1.local>
Date: Thu,  1 Oct 2015 00:01:20 +0600 (OMST)
From: user01@11-30.local

CentOS 7 Mail Server
.
quit
+OK Logging out.
Connection closed by foreign host.

```

8. Проверьте работу IMAP при помощи telnet.

```

# telnet localhost imap
Trying ::1...
Connected to localhost.
Escape character is '^'.
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE
STARTTLS AUTH=PLAIN AUTH=LOGIN] Dovecot ready.
a1 LOGIN user02 Pa$$w0rd
a1 OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE
SORT SORT=DISPLAY THREAD=REFERENCES THREAD=REFS THREAD=ORDEREDSUBJECT
MULTIAPPEND URL-PARTIAL CATENATE UNSELECT CHILDREN NAMESPACE UIDPLUS LIST-
EXTENDED I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES WITHIN
CONTEXT=SEARCH LIST-STATUS SPECIAL-USE BINARY MOVE] Logged in
a2 LIST "" "*"
* LIST (\HasNoChildren) "." INBOX
a2 OK List completed.
a3 EXAMINE INBOX
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS ()] Read-only mailbox.
* 2 EXISTS
* 0 RECENT
* OK [UNSEEN 2] First unseen.
* OK [UIDVALIDITY 1443637952] UIDs valid
* OK [UIDNEXT 3] Predicted next UID
a3 OK [READ-ONLY] Examine completed (0.000 secs).
a4 FETCH 1 BODY[]
* 1 FETCH (BODY[] {421})
Return-Path: <user01@11-30.local>
X-Original-To: user02
Delivered-To: user02@11-30.local
Received: from localhost (localhost [IPv6:::1])
        by server01.11-100-1.local (Postfix) with ESMTTP id 24DD820D74A0
        for <user02>; Thu,  1 Oct 2015 00:01:20 +0600 (OMST)
Message-Id: <20150930180129.24DD820D74A0@server01.11-100-1.local>
Date: Thu,  1 Oct 2015 00:01:20 +0600 (OMST)
From: user01@11-30.local

CentOS 7 Mail Server
)
a4 OK Fetch completed.
a5 LOGOUT
* BYE Logging out
a5 OK Logout completed.
Connection closed by foreign host.

```

Упражнение 4 - Установка и настройка Squirrelmail (ДОПОЛНИТЕЛЬНО).

1. Подключите репозиторий EPEL.
2. Установите Squirrelmail.
3. Запустите инструмент настройки Squirrelmail:

```
/usr/share/squirrelmail/config/conf.pl
```

4. Настройте параметры Squirrelmail.

Раздел		Параметр		Значение
1	Organization Preferences	1	Organization Name	11-100-X
2	Server Settings	1	Domain	11-100-X.local
2	Server Settings	3	Sendmail or SMTP	SMTP

5. Добавьте следующие строки в конец конфигурационного файла **/etc/httpd/conf/httpd.conf**.

```
Alias /webmail /usr/share/squirrelmail
<Directory /usr/share/squirrelmail>
    Options Indexes FollowSymLinks
    RewriteEngine On
    AllowOverride All
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
```

**ПРИМЕЧАНИЕ:**

В целях экономии времени можно взять данный фрагмент конфигурационного файла **httpd.conf** у инструктора:

<http://CCЫЛКА/repo/squirrelmail/httpd.conf>

6. Настройте автоматический запуск службы httpd.
7. Воспользуйтесь Webmail для проверки работы почтовой системы (<http://server01.11-100-X.local/webmail>).
8. Протестируйте создание пользователей и обмен почтой при помощи Webmail.
9. Переключите интерфейс Webmail на русский язык.

Ответы к упражнению 4.

1. Подключите репозиторий EPEL.

```
yum install epel-release
```

2. Установите Squirrelmail.

```
yum install squirrelmail
```

3. Запустите инструмент настройки Squirrelmail.

```
cd /usr/share/squirrelmail/config/
./conf.pl
```

4. Настройте параметры Squirrelmail.

Раздел		Параметр		Значение
1	Organization Preferences	1	Organization Name	ll-100
2	Server Settings	1	Domain	ll-100.local
2	Server Settings	3	Sendmail or SMTP	SMTP

```
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color off
S Save data
Q Quit

Command >> █
```

```
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Organization Preferences
1. Organization Name : ll-100
2. Organization Logo : ../images/sm_logo.png
3. Org. Logo Width/Height : (300/111)
4. Organization Title : SquirrelMail $version
5. Signout Page :
6. Top Frame : _top
7. Provider link : http://squirrelmail.org/
8. Provider name : SquirrelMail

R Return to Main Menu
C Turn color off
S Save data
Q Quit

Command >> █
```

```
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Server Settings
-----
General
1. Domain : ll-100.local
2. Invert Time : false
3. Sendmail or SMTP : SMTP

A. Update IMAP Settings : localhost:143 (uw)
B. Update SMTP Settings : localhost:25

R Return to Main Menu
C Turn color off
S Save data
Q Quit

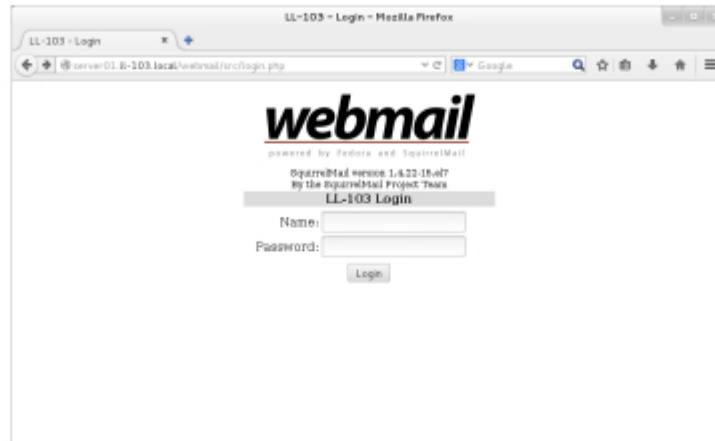
Command >> █
```

5. Добавьте следующие строки в конец конфигурационного файла `httpd.conf`.

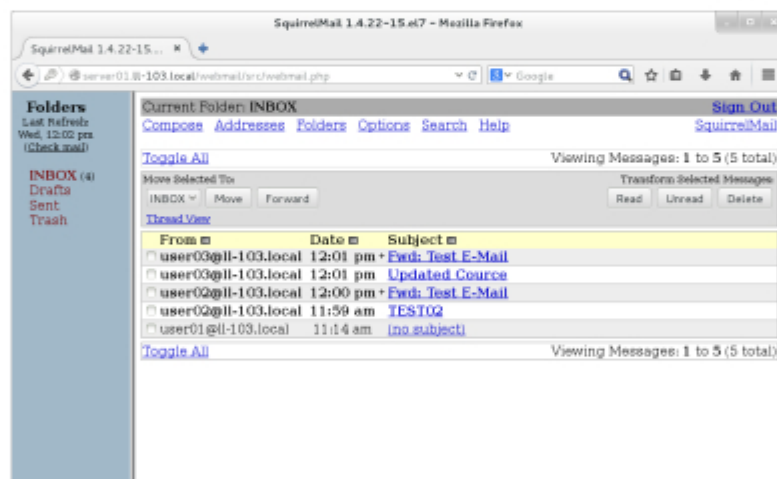
```
# vi /etc/httpd/conf/httpd.conf
Alias /webmail /usr/share/squirrelmail
<Directory /usr/share/squirrelmail>
    Options Indexes FollowSymLinks
    RewriteEngine On
    AllowOverride All
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
```

6. Настройте автоматический запуск службы httpd.

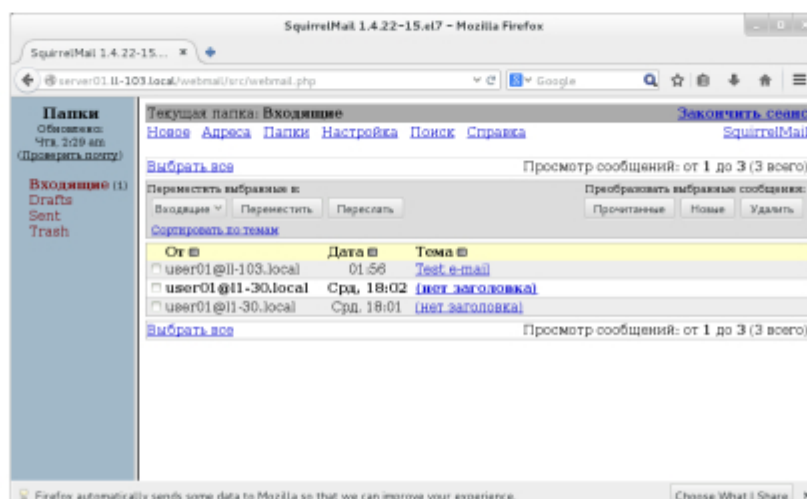
```
# systemctl start httpd
# systemctl status httpd
# systemctl enable httpd
# systemctl is-enabled httpd
```

7. Воспользуйтесь Webmail для проверки работы почтовой системы (<http://server01.ll-100-X.local/webmail>).

8. Протестируйте создание пользователей и обмен почтой при помощи Webmail.



9. Переключите интерфейс Webmail на русский язык.



Модуль 04: Протокол HTTP.



В модуле представлены:

- Общие концепции протокола HTTP.
- Apache HTTP Server.
- Модули Apache HTTP Server.
- Виртуальные узлы.
- SSL.
- Общие концепции проксирования трафика.
- Proxy-сервер **squid**.

Протокол HTTP

- HyperText Transfer Protocol – сетевой протокол передачи.
- Основной тип трафика в сети Internet.
- Основной объект – URI (Uniform Resource Identifier).
- Программная реализация:
 - Сервер:
 - Apache.
 - Nginx.
 - Lighthttpd.
 - Клиент:
 - Mozilla Firefox.
 - Microsoft Internet Explorer.

<http://edu.softline.ru/>



Протокол HTTP является основным протоколом передачи текстовой информации. Наиболее часто он используется для предоставления доступа к web-сайтам.

Основным объектом работы протокола является ресурс, адресация к которому осуществляется по URI (Uniform Resource Identifier). URI может ссылаться как на простые файлы, так и на иные объекты.

Основной версией протокола является версия HTTP/1.1. Протокол HTTP не поддерживает возможность сохранения состояния. Работа строится по схеме «запрос-ответ», где каждый «запрос» является для протокола новым.

HTTP-сообщение состоит из трёх частей:

- Стартовая строка – определяет тип сообщения.
- Заголовки сообщения – определяют параметры передачи и многие другие характеристики сообщения.
- Тело сообщения – данные.

Стартовые строки для запроса и ответа отличаются. Стартовая строка запроса имеет следующий синтаксис:

Method URI HTTP/Version

- **Method** – название запроса
- **URI** – путь к запрашиваемому документу
- **Version** – версия протокола HTTP

Стартовая строка ответа выглядит так:

HTTP/Version StatusCode Reason

- **Version** – версия протокола HTTP
- **StatusCode** – трёхзначное число определяет тип ответа и влияет на дальнейшее поведение клиента
- **Reason** – пояснение к коду. В случае ошибки, объяснение причины.

Протокол HTTP определяет следующие методы, допустимые в стартовой строке запроса:

- **OPTIONS** – данный метод позволит получить от сервера список всех доступных методов.
- **GET** – метод запрашивает содержимое ресурса.
- **HEAD** – аналогичен GET, но не выводится тело сообщения.
- **POST** – метод используется для передачи каких-либо данных на сервер.
- **PUT** – загружает указанный ресурс на сервер.
- **DELETE** – удалить ресурс.
- **TRACE** – возврат результата таким образом, что можно отследить, что изменялось в сообщении на пути следования.

Для кодов ответа предусмотрены три арабские цифры, первая из которых определяет тип ответа, а остальные его конкретизируют:

- **1XX** (Informational) – в данных кодах сообщается процесс передачи сообщения. Согласно стандарту, данные коды должны игнорироваться.
- **2XX** (Success) – данный код означает успешную обработку запроса.
- **3XX** (Redirection) – данный код ответа информирует клиента о том, что этот запрос должен быть сделан к другому URI, в поле Location передаётся необходимая ссылка.
- **4XX** (Client Error) – коды указывают на ошибку, возникшую на стороне клиента.
- **5XX** (Server Error) – ошибка возникла на стороне сервера.

Apache HTTP Server

- Apache - a patchy server, свободный сервер.
- Разрабатывается Apache Software Foundation.
 - Сервер отличают:
 - Высокий уровень надёжности.
 - Высокий уровень гибкости.
 - Высокий уровень интеграции с различными библиотеками визуализации, программирования и т.д.
- Основной конфигурационный файл – `/etc/httpd.conf`:
 - Читается при запуске демона `httpd`.
- Порты – 80 (`http`), 443 (`https`) протокола TCP.

<http://edu.softline.ru/>



Apache был создан 1996 году и был назван шутливо – "a patchy server". На сегодняшний день, apache является наиболее распространённым web-сервером, но не единственным. Основное преимущество apache – надёжность и необычайная гибкость, но, когда на первый план выходит производительность, имеет смысл обратить внимание на такие разработки как **lighttpd** и **nginx**.

Веб-сервер доступный в CentOS 7 (RHEL 7) – это демон Apache HTTP Server (**httpd**), веб сервер с открытым исходным кодом, разрабатываемый Apache Software Foundation (<http://www.apache.org/>). В CentOS 7 (RHEL 7) сервер Apache был обновлен до версии Apache HTTP Server 2.4.

В версиях Apache HTTP Server 2.2 и 2.4 имеется ряд важных отличий и если вы обновляете предыдущий релиз CentOS (RHEL), то вам нужно будет также обновить и конфигурацию службы **httpd**.

Apache HTTP Server 2.4

- Важные изменения в Apache HTTP Server 2.4:
 - Управление службой `httpd`.
 - Частный `/tmp`.
 - Вывод конфигурации.
 - Конфигурация по умолчанию.
 - Изменения настройки.
 - Модель обработки.
 - Изменения комплектации.
 - Изменения расположения на файловой системе.
 - Аутентификация, авторизация и управление доступом.
 - `suexec`.
 - Интерфейс модулей.
 - Удаленные модули.

<http://edu.softline.ru/>



Управление службой `httpd`.

С уходом от инициализирующих скриптов SysV, администраторы для управления службой `httpd` должны переключиться на использование команд `apachectl` и `systemctl`.

Команда `service httpd graceful` заменена на `apachectl graceful`.

Команда `service httpd configtest` заменена на `apachectl configtest`.

`systemd` для `httpd` имеет различия в поведении по сравнению со скриптами `init`.

- Полный перезапуск (`graceful restart`) используется по умолчанию, когда служба перезапускается.
- Полная остановка (`graceful stop`) используется по умолчанию, когда служба останавливается.

Частный `/tmp`.

Для улучшения безопасности `systemd` запускает `httpd` использует частный каталог `/tmp`, отдельно от системного каталога `/tmp`.

Вывод конфигурации.

Конфигурационные файлы, которые загружают модули теперь размещаются в каталоге `/etc/httpd/conf.modules.d`. Пакеты, такие как `php`, которые предоставляют дополнительные загружаемые модули для `httpd` будут размещаться в файлах в этом каталоге. Любой конфигурационный файл в каталоге `conf.modules.d` обрабатывается перед основным телом `httpd.conf`. Теперь конфигурационные файлы в каталоге `/etc/httpd/conf.d` обрабатываются после основного тела `httpd.conf`.

Некоторые дополнительные конфигурационные файлы предоставляются в пакете `httpd`:

`/etc/httpd/conf.d/autoindex.conf` – этот файл настраивает индексацию каталогов (`mod_autoindex`).

`/etc/httpd/conf.d/userdir.conf` – этот файл настраивает доступ к каталогам пользователей, например `http://example.com/~username`. Подобный доступ отключен по умолчанию в целях безопасности.

`/etc/httpd/conf.d/welcome.conf` – как и в предыдущих релизах, этот файл настраивает страницу приветствия, отображаемую для `http://localhost/`, когда никакого контента не предоставлено.

Конфигурация по умолчанию.

Минимальный `httpd.conf` теперь предоставляется по умолчанию. Много общих параметров настройки, таких как **Timeout** и **KeepAlive** более не настроены в явном виде в конфигурации по умолчанию. Жестко закодированные настройка по умолчанию для всех директив указаны в руководстве.

Изменения настройки.

Также присутствует некоторое количество обратно несовместимых изменений в синтаксисе настройки, которые придется поменять в случае миграции `httpd 2.2` на `httpd 2.4`.

Модель обработки.

В предыдущих релизах CentOS (RHEL) различные модели мульти-обработки (multi-processing models, MPM) были доступны в качестве различных двоичных файлов `httpd`: раздвоенная модель (forked model) «**prefork**», как `/usr/sbin/httpd`, модель на основе потоков (thread-based) «**worker**», как `/usr/sbin/httpd.worker`.

В CentOS 7 (RHEL 7), только один двоичный файл `httpd` используется и три MPM доступны в качестве подгружаемых модулей: **worker**, **prefork** (по умолчанию) и **event**. Для выбора подгружаемого модуля MPM, необходимо изменить конфигурационный файл `/etc/httpd/conf.modules.d/00-mpm.conf`.

Изменения комплектации.

Аутентификация LDAP и модули авторизации теперь предоставляются в отдельных под-пакетах `mod_ldap`. Новый модуль `mod_session` и ассоциированные модули помощники предоставляются в новом под-пакете – `mod_session`. Новые модули `mod_proxy_html` и `mod_xml2enc` предоставляются в новом под-пакете `mod_proxy_html`.

Изменения расположения на файловой системе.

Каталог `/var/cache/mod_proxy` более не доступен, вместо него в каталоге `/var/cache/httpd` располагаются подкаталоги `proxy` и `ssl`.

Контент `/var/www` перемещен в `/usr/share/httpd/`:

`/usr/share/httpd/icons`

Каталог `/var/www/icons` перемещен в `/usr/share/httpd/icons` Каталог содержит набор иконок, используется с индексами каталога. Доступны по ссылке `http://localhost/icons` в конфигурации по умолчанию, через `/etc/httpd/conf.d/autoindex.conf`.

`/usr/share/httpd/manual/`

Каталог `/var/www/manual` перемещен в `/usr/share/httpd/manual/`. Этот каталог содержит пакет `httpd-manual`, содержащий HTML-версию документации для `httpd`. Если пакет установлен, то доступен по адресу `http://localhost/manual/`, настраивается через `/etc/httpd/conf.d/manual.conf`

`/usr/share/httpd/error/`

Каталог `/var/www/error` перемещен в `/usr/share/httpd/error/`. Содержит страницы ошибок HTTP на разных языках. По умолчанию не настраивается, пример конфигурационного файла расположен по адресу `/usr/share/doc/httpd-VERSION/httpd-multilang-errordoc.conf`.

Аутентификация, авторизация и управление доступом.

Директивы настройки, используемые для управления аутентификацией, авторизацией и управление доступами существенно изменились. Существующие конфигурационные файлы использующие директивы **Order**, **Deny** и **Allow** должны быть адаптированы для использования нового синтаксиса **Require**.

suexec

Для улучшения безопасности системы, двоичный файл **suexec** более не устанавливает **setuid root**. Вместо этого он имеет биты совместимости файловой системы (file system capability bits), установка которых позволяет более ограниченный набор прав. Вместе с этим изменением двоичный файл **suexec** более не использует файл журнала **/var/log/httpd/suexec.log**. Вместо этого сообщения журнала отправляются в **syslog**. По умолчанию записываются в файл журнала **/var/log/secure**.

Интерфейс модулей.

Из-за изменений в интерфейсе модулей **httpd**, **httpd 2.4** более не совместим со сторонними двоичными модулями, разработанными для **httpd 2.2**. Такие модули должны быть скорректированы для работы с интерфейсом модулей **httpd 2.4** и пересобраны. Детальный список изменений API в версии 2.4 доступен по ссылке: http://httpd.apache.org/docs/2.4/developer/new_api_2_4.html.

Двоичный файл **apxs**, используемый для сборки модулей из исходных кодов перемещен из **/usr/sbin/apxs** в **/usr/bin/apxs**.

Удаленные модули.

Список модулей, удаленных в CentOS 7 (RHEL 7):

- **mod_auth_mysql**, **mod_auth_pgsq1**
- **mod_perl**
- **mod_perl** официально не поддерживается версией HTTPD 2.4 и старше.
- **mod_authz_ldap**

httpd 2.4 предоставляет поддержку аутентификации баз данных SQL внутри модуля **mod_auth_dbd**.

httpd 2.4 предоставляет поддержку LDAP при помощи **mod_authnz_ldap**.

**ПРИМЕЧАНИЕ:**

Дополнительную информацию по обновлению Apache HTTP Server с версии 2.0 и 2.2 до 2.4 можно получить по ссылке: <http://httpd.apache.org/docs/2.4/upgrading.html>

Установка `httpd`

- Пакет: `httpd`.

```
# yum install httpd
```
- Порт по умолчанию: 80.

```
# firewall-cmd --zone=public --add-port=80/tcp --permanent  
# firewall-cmd --zone=public --add-service=http --permanent  
# firewall-cmd --reload
```

<http://edu.softline.ru/>



Установка Apache HTTP Server

Для установки или проверки наличия Apache HTTP Server используйте следующую команду:

```
# yum install httpd
```

Обновление конфигурации.

Для обновления конфигурационных файлов от Apache HTTP Server версии 2.0 используйте следующие шаги:

1. Убедитесь, что все имена модулей корректны, так-как они могут быть изменены. Используйте директиву **LoadModule** для каждого модуля, который был переименован.
2. Пересоберите все сторонние модули, прежде чем загружать их. Обычно это требуется для модулей аутентификации и авторизации.
3. Если вы используете модуль `mod_userdir`, убедитесь, что директива **UserDir** включает имя каталога (обычно `public_html`).
4. Если вы используете Apache HTTP Secure Server, отредактируйте `/etc/httpd/conf.d/ssl.conf` для включения протокола Secure Socket Layer (SSL).

Не забудьте, что вы можете использовать следующую команду для проверки конфигурационного файла Apache HTTP Server на ошибки.

```
# service httpd configtest
```



ПРИМЕЧАНИЕ:

Дополнительную информацию по обновлению Apache HTTP Server с версии 2.0 до 2.2 можно получить по ссылке: <http://httpd.apache.org/docs/2.2/upgrading.html>.

Управление службой **httpd**

- Проверка состояния службы: `systemctl status httpd`
- Запуск службы: `systemctl start httpd`
- Остановка службы: `systemctl stop httpd`
- Перезапуск службы:
 - Полный перезапуск: `systemctl restart httpd`
 - Перезагрузка конфигурационного файла: `systemctl reload httpd`
 - Параметр `gracefull` в `systemctl` не поддерживается.
- Управление автоматическим запуском:
 - Проверка автозапуска службы: `systemctl is-enabled httpd`
 - Включение автозапуска службы: `systemctl enable httpd`
 - Отключение автозапуска службы: `systemctl disable httpd`

<http://edu.softline.ru/>



Запуск службы **httpd**

Для запуска службы **httpd**, используйте следующую команду:

```
# systemctl start httpd.service
```

Если вы хотите настроить автоматический запуск службы в процессе загрузки, используйте следующую команду:

```
# systemctl enable httpd.service
```



ПРИМЕЧАНИЕ:

Если запускается Apache HTTP Server, как безопасный сервер, пароль может потребоваться после загрузки машины, если используется зашифрованный частный ключ SSL.

Остановка службы **httpd**

Для остановки запущенной службы **httpd**, используйте следующую команду:

```
# systemctl stop httpd.service
```

Для отключения автоматического старта во время загрузки системы, используйте:

```
# systemctl disable httpd.service
```

Перезапуск службы **httpd**.

Для полного перезапуска службы используйте:

```
# systemctl restart httpd.service
```

Это останавливает службу **httpd** и немедленно запускает ее снова. Используйте эту команду после установки или удаления динамически загружающихся модулей, таких как PHP.

Для перезагрузки только конфигурации используйте:

```
# systemctl reload httpd.service
```

Это заставляет запущенную службу **httpd** перезагрузить конфигурационный файл. Любые запросы, обрабатываемые в данный момент будут сброшены, что может показать в браузере пользователей ошибку или отобразить часть веб-страницы.

Для перезагрузки конфигурации без воздействия на активные запросы используйте:

```
# service httpd graceful
```

В CentOS 7 (RHEL7) команда **service httpd graceful** заменена на **apachectl graceful**.

```
# apachectl graceful
```

Это заставляет запущенную службу **httpd** перезагрузить конфигурационный файл. Любые запросы, обрабатываемые в данный момент продолжают обрабатываться с использованием старой конфигурации.

Проверка статуса службы httpd.

Чтобы убедиться в том, что служба **httpd** запущена, используйте следующую команду:

```
# systemctl is-active httpd.service
```

Конфигурационные файлы

- Основной конфигурационный файл:
 - `/etc/httpd/conf/httpd.conf`
- Дополнительный каталог для дополнительных конфигурационных файлов:
 - `/etc/httpd/conf.d/`
- Проверка конфигурационного файла (`httpd.conf`) на наличие ошибок:

```
# service httpd configtest
```

<http://edu.softline.ru/>



Когда служба **httpd** запущена, по умолчанию, она читает конфигурационные файлы:

- `/etc/httpd/conf/httpd.conf` – основной конфигурационный файл.
- `/etc/httpd/conf.d/` - дополнительный каталог для конфигурационных файлов, включенный в основном конфигурационном файле.

Несмотря на то, что конфигурация по умолчанию подходит для многих сценариев, необходимо разбираться в наиболее популярных опциях настройки.

Обратите внимание, что для того чтобы изменения, внесенные в конфигурационный файл, вступили в силу – необходимо перезапустить службу **httpd**.

Для проверки конфигурационного файла на наличие ошибок, используйте следующую команду:

```
# service httpd configtest
```

В CentOS 7 (RHEL7) команда **service httpd configtest** заменена на **apachectl configtest**.

```
# apachectl configtest
```

Для упрощения процедуры восстановления, рекомендуется делать копию конфигурационного файла перед внесением изменений.

Работа с модулями

- Модуль – динамический общий объект (Dynamic Shared Object, DSO)
- Документация к модулям:
 - <http://httpd.apache.org/docs/2.4/mod/>
- Каталоги модулей:
 - `/usr/lib/httpd/modules`
 - `/usr/lib64/httpd/modules`
- Директива загрузки модуля:
 - `LoadModule ssl_module modules/mod_ssl.so`

<http://edu.softline.ru/>



Работа с модулями.

Будучи модульным приложением, служба **httpd** распространяется вместе с набором *Динамических общих объектов (Dynamic Shared Objects, DSO)*, которые могут быть динамически загружены или выгружены в процессе работы по мере необходимости. По умолчанию эти модули располагаются в `/usr/lib/httpd/modules` на 32-ух битной системе и в `/usr/lib64/httpd/modules` на 64-ех битной системе.

Загрузка модуля.

Для загрузки необходимого DSO используйте директиву **LoadModule**. Необходимо отметить, что модули предоставляемые в отдельных пакетах, как правило, имеют собственные конфигурационные файлы в каталоге `/etc/httpd/conf.d/`.

Загрузка DSO `mod_ssl`:

```
LoadModule ssl_module modules/mod_ssl.so
```


Разработка модулей



- Пакет: `httpd-devel`.
`# yum install httpd-devel`
- Инструмент компиляции модуля: Apache eXtenSion (`apxs`).
`# apxs -i -a -c module_name.c`

<http://edu.softline.ru/>



Для создания нового DSO модуля, потребуется установить пакет `httpd-devel`.

```
# yum install httpd-devel
```

Этот пакет содержит файлы, заголовки файлов и утилиту Apache eXtenSion (`apxs`), необходимую для компиляции модуля.

После написания вы можете создать модуль при помощи следующей команды:

```
# apxs -i -a -c module_name.c
```

Если создание пройдет успешно, вы сможете загрузить модуль также, как и все остальные модули, распространяемые вместе с Apache HTTP Server.

Виртуальные узлы

- Один web-сервер обрабатывает запросы для нескольких сайтов.
- Два типа виртуальных узлов:
 - host-based – сайт определяется ip-адресом и/или портом.
 - name-based – сайт определяется именем.
- Каждый сайт описывается отдельной секцией файла `httpd.conf`:

```
<VirtualHost *:80>
    ServerAdmin webadmin@11-100-1.local
    DocumentRoot /var/www/portal
    ServerName portal.11-100-1.local
    ErrorLog logs/portal.11-100-1.local-error_log
    CustomLog logs/portal.11-100-1.local-access_log common
</VirtualHost>
```

<http://edu.softline.ru/>

APACHE
HTTP SERVER

Встроенный в Apache HTTP Server виртуальный хостинг, позволяет предоставлять различную информацию основываясь на запрошенном IP-адресе, имени хоста или порту.

Для создания виртуального хоста на основе имени (name-based), найдите виртуальный хост, представленный в виде примера в `/etc/httpd/conf/httpd.conf`, удалите решетку (#) из начала каждой строки и настройте опции в соответствии с вашими требованиями.

Пример настройки виртуального хоста:

```
NameVirtualHost penguin.example.com:80
<VirtualHost penguin.example.com:80>
    ServerAdmin webmaster@penguin.example.com
    DocumentRoot /www/docs/penguin.example.com
    ServerName penguin.example.com:80
    ErrorLog logs/penguin.example.com-error_log
    CustomLog logs/penguin.example.com-access_log common
</VirtualHost>
```

Обратите внимание, что **ServerName** должен быть корректным DNS именем, назначенным машине. Контейнер `<VirtualHost>` очень гибок в настройке и может принимать большинство директив доступных для настройки основного сервера. Директивы не поддерживаемые данным контейнером **User** и **Group**, заменены на **SuexecUserGroup**.



ПРИМЕЧАНИЕ:

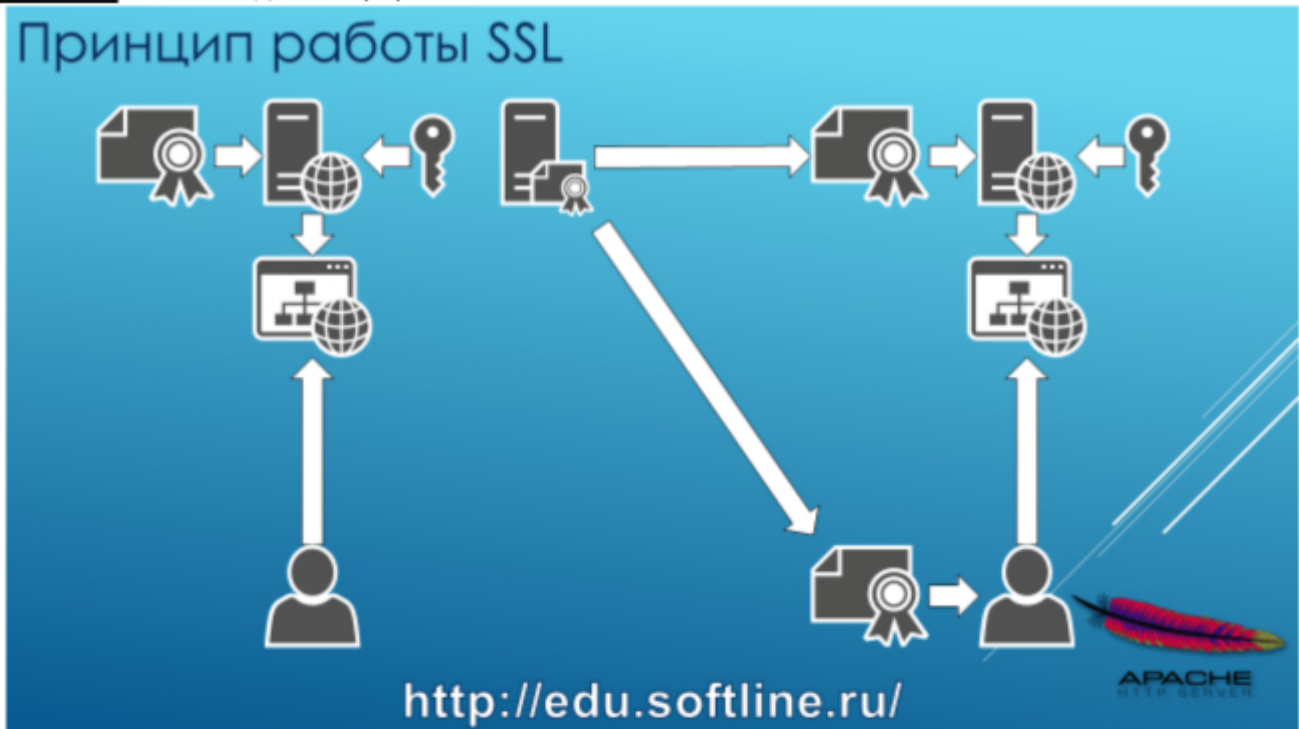
Если вы настраиваете виртуальный хост на прослушку нестандартного порта, убедитесь что вы внесли изменение в директиву **Listen** в секции глобальных настроек файла `/etc/httpd/conf/httpd.conf`



ПРИМЕЧАНИЕ:

Директива **NameVirtualHost** более не используется. Указание порта по умолчанию не требуется.

Для активации вновь созданного виртуального хоста перезапустите веб-сервер.



Настройка SSL сервера.

Secure Socket Layer (SSL) – это криптографический протокол, который позволяет серверу и клиенту и клиенту взаимодействовать безопасно. Вместе с расширенной и улучшенной версией Transport Layer Security (TLS), это обеспечивает и конфиденциальность, и целостность данных. Apache HTTP Server в комбинации с модулем `mod_ssl`, использующем набор инструментов для предоставления поддержки SSL/TLS, как правило называется SSL сервер.

В отличие от обычного HTTP соединения, которое может быть прочитано и даже модифицировано кем-то, кто способен перехватить его, использование `mod_ssl` предотвращает перехват или модификацию передаваемого контента.

В данной секции мы рассматриваются вопросы включения модуля `mod_ssl` в параметрах Apache HTTP Server и процесса генерации закрытых ключей и самоподписанных сертификатов.

Обзор сертификатов и безопасности.

Безопасные коммуникации основываются на использовании ключей. В обратимом (Conventional) или симметричном шифровании (Symmetric cryptography), обе конечные точки имеют одинаковый ключ, для шифрования каждой передачи. С другой стороны, публичное (Public) или асимметричное шифрование (Asymmetric cryptography), использует пару ключей: частный ключ (Private key), который держится в секрете и публичный ключ (Public key), который доступен всем. Когда данные шифруются при помощи публичного ключа, они могут быть расшифрованы только при помощи частного ключа и наоборот.

Для предоставления безопасного взаимодействия с использованием SSL, SSL сервер должен обладать цифровым сертификатом, подписанным при помощи центра выдачи сертификатов (Certificate Authority, CA). Сертификат содержит множество атрибутов сервера (имя сервера, имя компании, расположение и так далее) и подпись, предоставленную закрытым ключом CA. Эта подпись подтверждает, что CA выдал данный сертификат и он не был изменен.

Когда веб-браузер устанавливает новое SSL соединение, он проверяет сертификат, выданный веб-серверу. Если сертификат не содержит подписи от доверенного CA или имя, указанное в сертификате, не совпадает с именем хоста, использованным для установления соединения, это прерывает соединение с сервером и как правило предоставляет пользователю соответствующее сообщение об ошибке.

По умолчанию большинство веб-браузеров настроено на доверие широко используемым центрам выдачи сертификатов. Поэтому эти CA должны быть выбраны при настройке безопасного сервера, так как пользователи будут автоматически доверять соединению, в противном случае им будет предоставлено сообщение об ошибке и придется принимать сертификат вручную. Разрешение пользователям принимать не доверенные сертификаты, может позволить злоумышленнику перехватить соединение, поэтому необходимо выбирать доверенные CA по мере возможности.

Список доверенных CA для Mozilla Firefox: <https://wiki.mozilla.org/CA:IncludedCAs>.

В процессе настройки SSL сервера, необходимо сгенерировать запрос сертификата и частный ключ, затем отправить запрос сертификата для подтверждения компании и оплатить услуги центра выдачи сертификата. После проверки запроса сертификата и вашей идентификации, вам вышлют подписанный сертификат, который вы сможете использовать для вашего сервера. В качестве альтернативы, вы можете создать самозаверяющийся сертификат, который не будет содержать подписи CA, данный сертификат рекомендуется использовать только с целью тестирования.

Установка SSL сервера

- Пакеты: `mod_ssl` и `openssl`.
`# yum install mod_ssl openssl`
- Конфигурационный файл:
 - `/etc/httpd/conf.d/ssl.conf`
- Каталог для сертификата:
 - `/etc/pki/tls/certs/`
- Каталог для частного ключа:
 - `/etc/pki/tls/private/`
- Директивы `ssl.conf`:
 - `SSLCertificateFile /etc/pki/tls/certs/hname.crt`
 - `SSLCertificateKeyFile /etc/pki/tls/private/hname.key`

<http://edu.softline.ru/>



Включение модуля `mod_ssl`.

Для начала настройки SSL сервера установите пакеты `mod_ssl` и `openssl`.

```
# yum install mod_ssl openssl
```

Это создаст конфигурационный файл `mod_ssl /etc/httpd/conf.d/ssl.conf`, который включен в основной конфигурационный файл Apache HTTP Server по умолчанию. Для загрузки модуля перезапустите службу `httpd`.

Использование существующих ключей и сертификатов.

Если у вас есть ранее созданные ключ и сертификат, вы можете настроить SSL сервер на использование этих файлов, вместо создания новых. Есть две ситуации, когда это не получится:

1. *Вы сменили IP адрес или имя домена.*

Сертификат выдается на определенную пару IP адреса и имя домена. Если один из этих параметров изменится, то сертификат становится недействительным.

2. *У вас есть сертификат от VeriSign и вы сменили программное обеспечение сервера.*

VeriSign – это широко используемый центр выдачи сертификатов, выдающий сертификаты для программных продуктов, IP-адресов и доменных имен. Смена программного продукта сделает сертификат недействительным.

В случае двух описанных выше сценариев вам придется запросить новый сертификат.

Если вы будете использовать существующий ключ и сертификат, переместите необходимые файлы в каталоги `/etc/pki/tls/private/` и `/etc/pki/tls/certs/` соответственно.

Вы можете сделать это при помощи следующих команд:

```
# mv key_file.key /etc/pki/tls/private/hostname.key
# mv certificate.crt /etc/pki/tls/certs/hostname.crt
```

Затем добавьте следующие строки в конфигурационный файл:

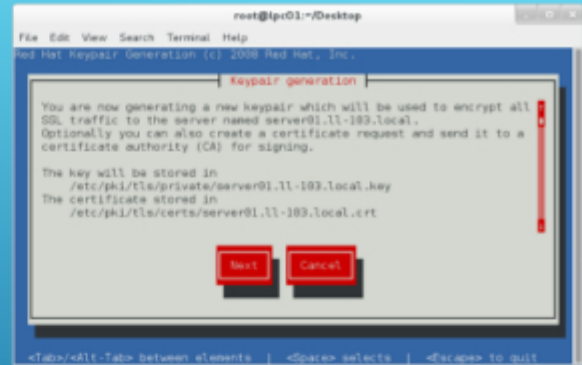
```
SSLCertificateFile /etc/pki/tls/certs/hostname.crt  
SSLCertificateKeyFile /etc/pki/tls/private/hostname.key
```

Использование ключа и сертификата от CentOS Secure Web Server:

```
# mv /etc/httpd/conf/httpsd.key /etc/pki/tls/private/penguin.example.com.key  
# mv /etc/httpd/conf/httpsd.crt /etc/pki/tls/certs/penguin.example.com.crt
```


Создание сертификатов

- Пакет: `crypto-utils`.
`# yum install crypto-utils`
- Утилита: `genkey`.
`# genkey hostname`



<http://edu.softline.ru/>

Для создания новых ключей и сертификатов необходимо установить пакет `crypto-utils`.

```
# yum install crypto-utils
```

Этот пакет содержит набор инструментов для генерации и управления SSL-сертификатами и частными ключами, и включает утилиту **genkey** Red Hat KeyPair Generation utility, которая содержит мастер создания ключей



ПРИМЕЧАНИЕ:

Замена существующего сертификата.

Если сервер уже имеет корректный сертификат, и вы заменяете его новым, укажите другой серийный номер. Это позволит уведомить браузеры клиентов об этом изменении, обновить сертификат и не получить отказ при доступе к странице. Для создания нового сертификата с указанным серийным номером, используйте следующую команду вместо **genkey**:

```
#openssl req -x509 -new -set_serial number -key hostname.key -out hostname.crt
```



ПРИМЕЧАНИЕ:

Удаление ранее созданного ключа.

Если файл ключа с указанным именем хоста уже существует в вашей системе, **genkey** откажется запускаться. В таком случае удалите существующий файл:

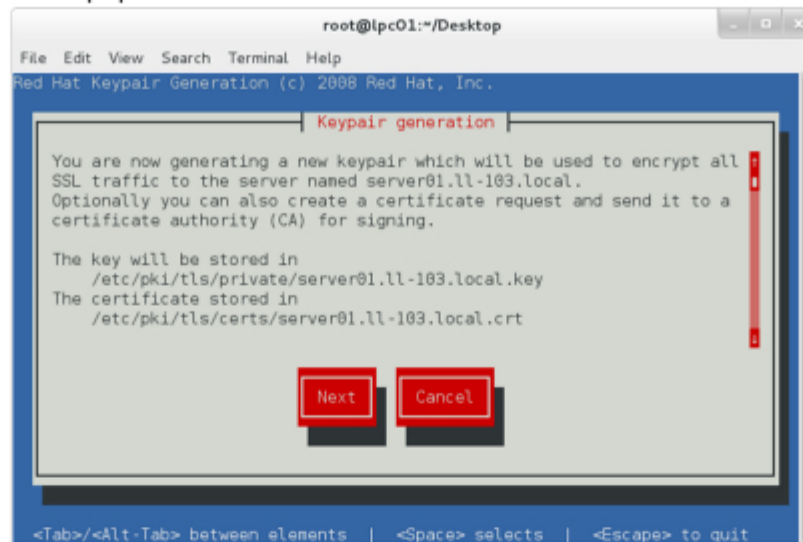
```
#rm /etc/pki/tls/private/hostname.key
```

Для запуска утилиты используйте команду **genkey** с указанием имени хоста (например, `penguin.example.com`):

```
# genkey hostname
```

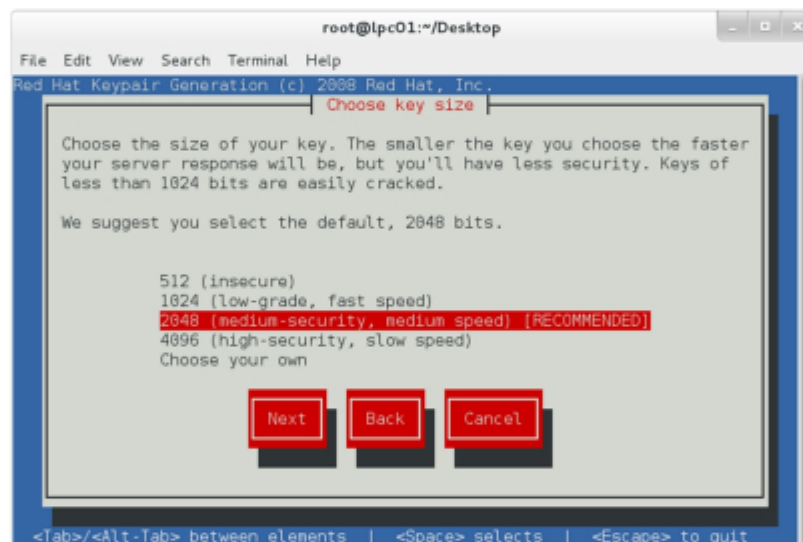
Для завершения процедуры создания ключа и сертификата используйте следующие шаги:

1. Посмотрите целевые расположения, в которых ключ и сертификат будут сохранены.

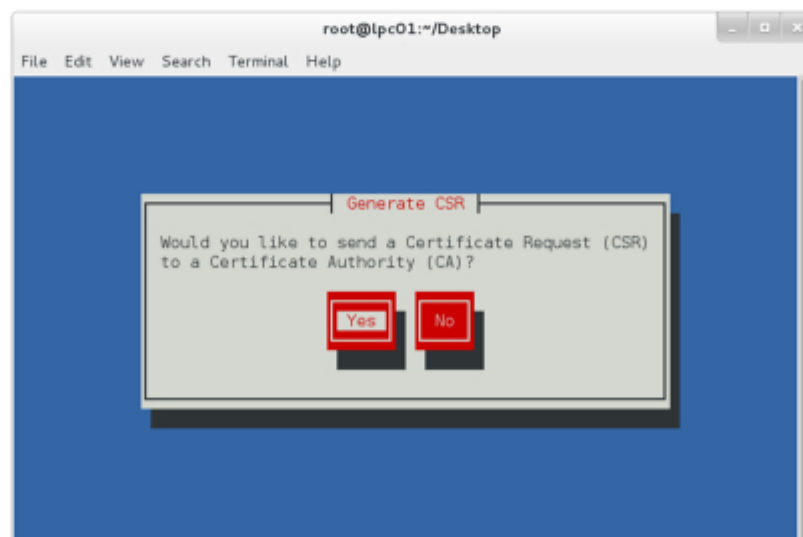


Используйте клавишу **Tab** для выбора кнопки **Next** и нажмите **Enter** для перехода к следующему шагу.

2. Используйте клавиши стрелки **Up** и **Down** для выбора необходимого размера ключа. Не забудьте, что больший размер ключа увеличивает безопасность и время отклика вашего сервера. Рекомендуется использовать 2048 bit (<http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf>).



3. Выберите необходимо ли отправить запрос в центр выдачи сертификатов.

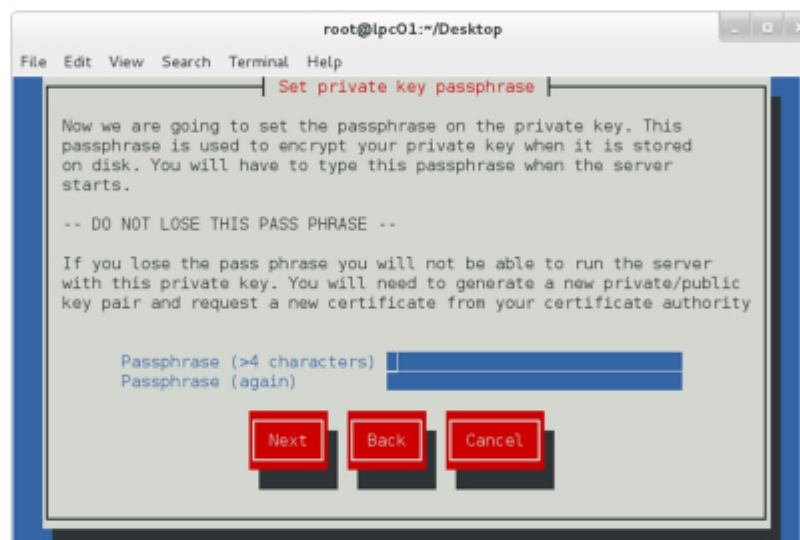


Для создания запрос используйте **Yes**, для создания самозаверяющегося сертификата используйте **No**.

4. Используйте клавишу **Пробел** для включения ([*]) или отключения ([]) шифрования частного ключа.



5. Если включили шифрования частного ключа введите парольную фразу. В целях безопасности символы не отображаются при вводе, минимальная длина парольной фразы – 5 символов.

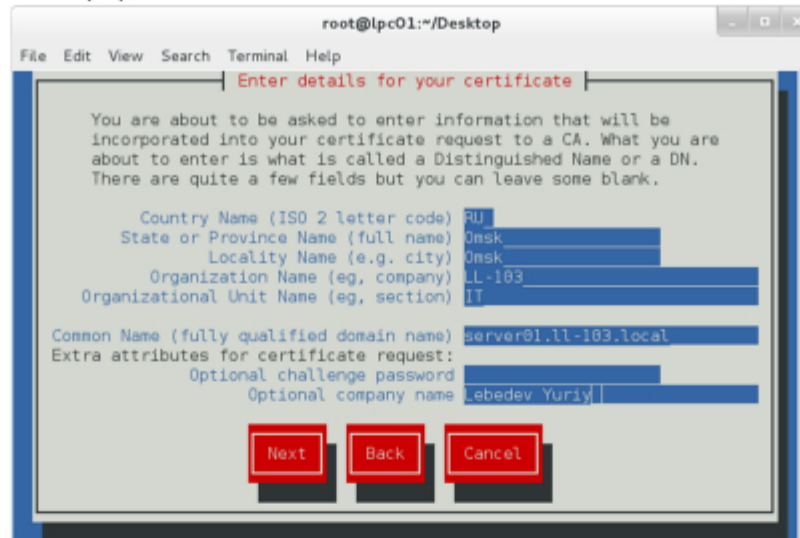


ПРИМЕЧАНИЕ:

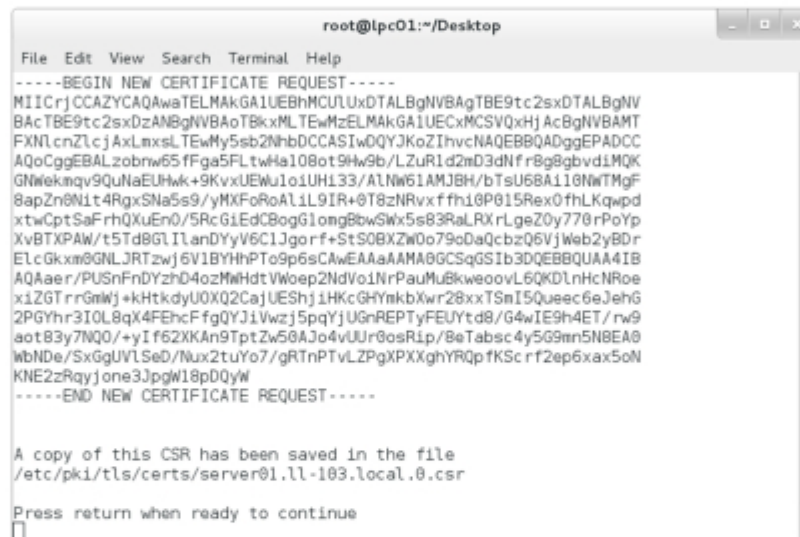
Не забудьте парольную фразу.

Ввод корректной парольной фразы требуется для запуска сервера. Если вы забудете ее, вам придется создать новые ключ и сертификат.

6. Укажите детали сертификата.



7. Если ранее вы выбрали создание запроса сертификата, вам будет предложено отправить его в центр выдачи сертификатов.



Нажмите Enter для возврата в командную строку.

После создания, добавьте расположение ключа и сертификата в конфигурационный файл `/etc/httpd/conf.d/ssl.conf`. Затем перезапустите `httpd` для применения изменений конфигурации.

```
SSLCertificateFile /etc/pki/tls/certs/hostname.crt
SSLCertificateKeyFile /etc/pki/tls/private/hostname.key
```


Разграничение прав доступа

- Web-сервер Apache позволяет ограничивать доступ к ресурсам по ip-адресам или по связке login-password.
- Доступ по умолчанию – параметр **order**:
 - **deny, allow** – разрешено всё, что не запрещено.
 - **allow, deny** – запрещено всё, что не разрешено.
- Авторизация:
 - **AuthName** – текст приглашения.
 - **AuthType** – тип авторизации.
 - **AuthUserFile** – файл с паролями.
 - **Require** – требование для предоставления доступа.
- Применяется в секциях **Directory**, **Location**.

<http://edu.softline.ru/>



Web-сервер Apache позволяет ограничивать доступ к ресурсам, базируясь на ip-адресе клиента. Правила доступа определяются параметром **order**, который определяет поведение по умолчанию и может принимать два значения:

- **allow, deny** – доступ будет предоставлен только узлам, указанным явно.
- **deny, allow** – доступ будет предоставлен всем узлам, кроме указанных.

Итак, поведение определено, теперь необходимо определить узлы, которым разрешён/запрещён доступ к ресурсам. Используйте определение правил доступа в секциях **Directory** и **Location** для конкретизации ресурсов, к которым необходимо применять правила. С помощью параметра **allow** вы определяете узлы, которым разрешён доступ, а с помощью параметра **deny** тем, кому запрещён:

```
<Directory /var/www/example.com/site>
    order allow,deny
    allow from 192.168.1.0/24
</Directory>
```

В этом примере сказано, что доступ к ресурсам, находящимся в каталоге `/var/www/example.com/site`, предоставлен только для сети `192.168.1.0/24`, всем остальным доступ будет закрыт. А в следующем примере доступ будет предоставлен всем, кроме этой же сети:

```
<Directory /var/www/example.com/site>
    order deny,allow
    deny from 192.168.1.0/24
</Directory>
```

Доступ к узлам может быть предоставлен после авторизации, это необходимо, когда ограничение по ip-адреса неприемлемо. Для включения авторизации обязательно указание текста приглашения, которым будет озаглавлено окно браузера. Также необходимо указать тип авторизации и, если необходимо, файл с учётными записями. Рассмотрим самый распространённый тип авторизации – **basic**:

```
<Directory /var/www/example.com/site>
```

```
AuthName "Restricted area"
AuthType basic
AuthUserFile /etc/httpd/conf/htpasswd
require valid-user
</Directory>
```

В данном примере указан текст приглашения и тип авторизации, файл с учётными записями пользователей и параметр **require**, определяющий условия, при которых доступ будет предоставлен. Начнём с файла `htpasswd`. Данный файл содержит зашифрованные пароли пользователей и на него должны быть установлены права доступа, позволяющие web-серверу Apache производить чтение. Больше НИКТО не должен иметь права доступа к данному файлу. Формирование базы данных пользователей производится одноимённой программой – **htpasswd**, включённой в состав Apache.

Чтобы создать файл с учётными записями пользователей, вызов команды `htpasswd` необходимо производить с указанием параметра `-c`, говорящим о том, что необходимо создать новый файл, в противном случае будет использован существующий:

```
# htpasswd -c /etc/httpd/conf/.htpasswd admin
# htpasswd /etc/httpd/conf/.htpasswd user
```

Теперь, когда файл с паролями создан, а описание авторизации произведено, осталось указать условие, необходимое для предоставления доступа к ресурсу. Это можно осуществить с помощью параметра `require`, который принимает три значения:

- `user` – предоставить доступ только определённым пользователям.
- `group` – предоставить доступ пользователям, входящим в группы.
- `valid-user` – предоставить доступ любому авторизованному пользователю.

Проксирующий сервер

- **Основные задачи прокси-сервера:**

- Обеспечение доступа из локальной сети в сеть Интернет.
- Кэширование запросов – ускорение работы, экономия трафика и времени.
- Разграничение доступа к ресурсам сети Интернет из локальной сети.
- Анонимизация доступа к ресурсам.

<http://edu.softline.ru/>



Проксирующие серверы используются для решения самых разных задач, начиная от предоставления доступа в сеть Интернет и заканчивая предоставлением анонимного доступа для совершения действий, противоречащих законодательству.

Самое частое применение – предоставление доступа в Интернет из локальной сети с возможностью гибкого ограничения доступа к тем или иным ресурсам, ограничения по времени суток и т.д. Одной из немаловажных функций прокси-сервера является кэширование, которое позволяет хранить результат предыдущего запроса и при повторении этого запроса (даже с другого узла), выдавать данные из памяти, что позволит сэкономить время и трафик.

Proxy-сервер **squid**

- **squid** – кэширующий прокси-сервер, распространяемый свободно.
- Сервер отличают:
 - Высокий уровень производительности.
 - Высокий уровень гибкости.
- Основной конфигурационный файл - `/etc/squid.conf`:
 - Читается при запуске демона **squid**.
- Порт – 3128/ТСР.



<http://edu.softline.ru/>

Проксирующий сервер **squid** один из немногих, чей конфигурационный файл можно назвать самодостаточным. Комментарии дают возможность настроить сервер даже новичку, которого может напугать объем конфигурации. На самом деле, базовая конфигурация сводится к определению нескольких параметров:

- Кэш и каталог для его хранения.
- Сетевые интерфейсы, с которых необходимо принимать запросы.
- Списки доступ, определяющие субъекты, которым разрешён доступ.

Первое, что необходимо определить – интерфейсы, на которые будут приходить запросы. Значение параметр **http_port** отвечает за данный аспект. Если указан только номер порта, это означает, что соединения будут приниматься на указанный порт всех сетевых интерфейсов.

```
http_port 3128
```

Параметр **cache_dir** используется для определения каталога, в котором будут храниться кэшируемые данные. Значение параметра состоит из 5ти частей:

- Тип кэша – в подавляющем числе случаев устанавливается **ufs**.
- Путь к каталогу – каталог с кэшируемыми данными.
- Размер кэша – указывается в мегабайтах.
- Число каталогов первого уровня.
- Число каталогов второго уровня – число каталогов определяет, как много уникальных объектов может храниться в кэше.

После определения, каталог для кэша необходимо создать, выполнив команду:

```
# squid -z
```

Списки доступа – **acl** (access list), предназначены для определения критериев, применяемых при определении доступа к проксирующему серверу. Изначально конфигурационный файл построен таким образом, что доступ к сайтам посредством сервера **squid** предоставлен только узлу 127.0.0.1. Чтобы определить дополнительные узлы/сети, необходимо добавить строки:

```
acl internal-subnet 192.168.0.0/255.255.255.0
acl internal-subnet 192.168.1.0/255.255.255.0
acl internal-subnet 192.168.2.0/255.255.255.0
acl banned-hosts   192.168.1.10-192.168.1.20
acl banned-hosts   192.168.1.25
acl partner-host    172.16.0.5
```

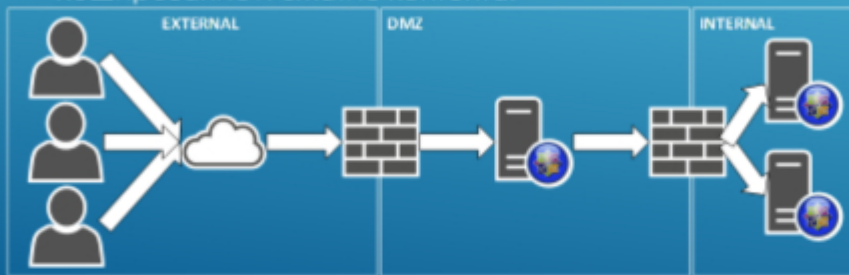
Но определение `acl` никак не влияет на предоставление доступа. Для этого предназначен параметр `http_access`, в качестве значения которого указывается имя списка доступа и указания `allow/deny`. Чтобы разрешить доступ для наших списков доступа, необходимо указать:

```
http_access    allow    internal-subnet
http_access    allow    partner-host
http_access    deny     banned-hosts
```

Обратное проксирование

- Основные задачи обратного прокси-сервера:

- Скрытие внутренних серверов и их характеристик.
- Программный фаервол (защита от DOS или DDOS).
- Разгрузка SSL (SSL Offloading).
- Балансировка нагрузки.
- Кэширование и сжатие контента.



Обратный прокси-сервер (reverse proxy) – тип прокси-сервера, который ретранслирует запросы клиентов из внешней сети на один или несколько серверов, логически расположенных во внутренней сети. При этом для клиента это выглядит так, будто запрашиваемые ресурсы находятся непосредственно на прокси-сервере. В отличие от прозрачного прокси, который перенаправляет запросы клиентов к любым серверам в Интернете и возвращает им результат, обратный прокси непосредственно взаимодействует лишь с ассоциированными с ним серверами и возвращает ответ только от них.

Использование:

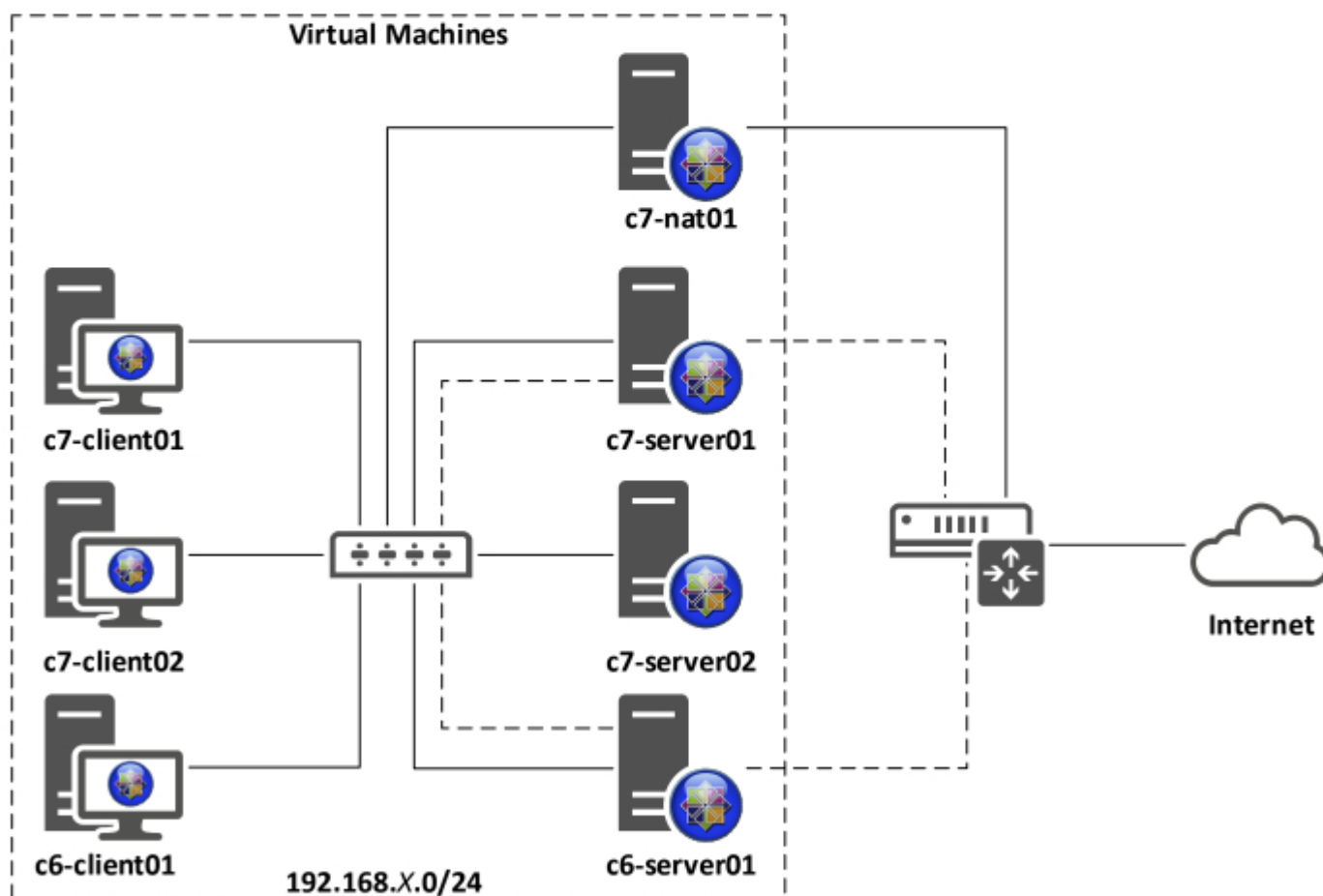
- Обратный прокси-сервер может скрывать существование опрашиваемых им серверов и их характеристики.
- Применение программного фаервола в обратном прокси-сервере может защитить от наиболее распространенных веб-атак, таких как DOS или DDOS. Без обратного прокси-сервера удаление вредоносного ПО может оказаться непростой задачей.
- Основной веб-сайт может не поддерживать подключение по SSL, однако это можно реализовать с помощью обратного прокси-сервера, который может быть оборудован аппаратным SSL-ускорителем.
- Выполнение функций балансировщика нагрузки между несколькими серверами, подменяя URL таким образом, чтобы использовался наиболее уместный сервер.
- Уменьшение нагрузки на основные серверы благодаря кэшированию статического и динамического контента. Эта возможность известна как акселерация веб-сайтов. Сервер может отсортировать свой кэш по частоте запросов к контенту, что значительно уменьшит нагрузку на основные серверы.
- Сжатие содержимого для уменьшения времени его загрузки.

Лабораторная работа.

Лабораторная работа 4

- **Упражнения:**
 - Установка и настройка Apache, PHP и MariaDB (LAMP).
 - Настройка SSL для Apache HTTP Server (Дополнительно).
 - Подготовка окружения к развертыванию squid.
 - Установка и настройка squid.
 - Проверка работы squid (Настройка клиентов).
 - Разграничение доступов в squid (Дополнительно).
- **Необходимые машины:**
 - C7-NAT01.
 - C7-SERVER01.
 - C7-CLIENT01.
 - C7-CLIENT02.
- **Учетные данные:**
 - Пользователь: `root`.
 - Пароль: `Pa$$w0rd`.

<http://edu.softline.ru/>



Упражнение 1 – Установка и настройка Apache, PHP и MariaDB (LAMP).

LAMP – акроним, обозначающий набор (комплекс) серверного программного обеспечения, широко используемый во Всемирной паутине. LAMP назван по первым буквам входящих в его состав компонентов:

- Linux – операционная система Linux.
- Apache – веб-сервер.
- MariaDB – СУБД.
- PHP – язык программирования.

1. Проверьте разрешение имени сервера `server02.11-100-X.local`:
2. Проверьте работу `server02.11-100-X.local`:
3. Войдите на `server02.11-100-X.local`, используя пароль `Pa$$w0rd`:
4. Подключите дополнительный репозиторий от Fedora Project:
`yum install epel-release`



ПРИМЕЧАНИЕ:

В репозитории CentOS 7 отсутствует `phpMyAdmin`.

5. Проверьте наличие новых репозиториях epel:
6. Установите MariaDB:
`yum install mariadb-server mariadb -y`
7. Запустите службу MariaDB:
8. Проверьте статус службы MariaDB:
9. Настройте автоматический запуск службы MariaDB:
10. Настройте безопасность MariaDB, задайте пароль: `Pa$$w0rd`, остальные параметры примите по умолчанию (отвечайте на вопросы нажатием клавиши Enter):
`mysql_secure_installation`
11. Установите Apache HTTP Server 2.4 (`httpd`):
12. Запустите службу Apache HTTP Server:
13. Проверьте статус службы Apache HTTP Server:
14. Настройте автоматический запуск службы Apache HTTP Server:
15. Откройте в файерволе порты 80 и 443:
16. Проверьте доступность сайта с помощью браузера: `http://server02.11-100-X.local`.
17. Установите PHP5:
`yum -y install php`
18. Найдите и установите несколько дополнительных пакетов-расширений для PHP:



ПРИМЕЧАНИЕ:

Обязательно установите: `php-mysql`.

`yum install php-mysql -y`

19. Перезапустите службу Apache HTTP Server:
20. Протестируйте работу PHP, создав страницу и открыв ее в браузере `http://server02.11-100-X.local/test.php`:
`vi /var/www/html/test.php`


```
<?php
phpinfo();
?>
```

21. Установит phpMyAdmin:

```
# yum install phpMyAdmin -y
```

22. Разрешите удаленный доступ к phpMyAdmin, изменив секцию <Directory> в файле:
/etc/httpd/conf.d/phpMyAdmin.conf:

```
# vi /etc/phpMyAdmin/config.inc.php
```

```
<Directory /usr/share/phpMyAdmin/>
    Options none
    AllowOverride Limit
    Require all granted
</Directory>
```

23. Измените аутентификацию phpMyAdmin с cookie на http:

```
# vi /etc/phpMyAdmin/config.inc.php
```

```
$cfg['Servers'][$i]['auth_type'] = 'http'; // Authentication
method (config, http or cookie based)?
```

24. Перезапустите службу Apache HTTP Server:

25. Проверьте работу phpMyAdmin с помощью браузера: **http://server02.11-100-X.local/phpmyadmin**, используйте учетную запись: **root** и пароль **Pa\$\$w0rd**.

Ответы к упражнению 1.

1. Проверьте разрешение имени сервера `server02.11-100-X.local`:
`nslookup server02.11-100-X.local`
2. Проверьте работу `server02.11-100-X.local`:
`ping server02.11-100-X.local`
3. Войдите на `server02.11-101-X.local`, используя пароль `Pa$$w0rd`:
`ssh root@server02.11-101-X.local`
4. Подключите дополнительный репозиторий от Fedora Project:
`yum install epel-release`

ПРИМЕЧАНИЕ:

В репозитории CentOS 7 отсутствует `phpMyAdmin`.

5. Проверьте наличие новых репозиториях epel:
`yum repolist`
6. Установите MariaDB:
`yum -y install mariadb-server mariadb`
7. Запустите службу MariaDB:
a. `systemctl start mariadb`
8. Проверьте статус службы MariaDB:
a. `systemctl status mariadb`
9. Настройте автоматический запуск службы MariaDB:
a. `systemctl enable mariadb`
10. Настройте безопасность MariaDB, задайте пароль: `Pa$$w0rd`, остальные параметры примите по умолчанию (отвечайте на вопросы нажатием клавиши Enter):
`mysql_secure_installation`
11. Установите Apache HTTP Server 2.4 (`httpd`):
`yum install httpd -y`
12. Запустите службу Apache HTTP Server:
`systemctl start httpd.service`
13. Проверьте статус службы Apache HTTP Server:
`systemctl status httpd.service`
14. Настройте автоматический запуск службы Apache HTTP Server:
`systemctl enable httpd.service`
15. Откройте в файерволе порты 80 и 443:
`firewall-cmd --permanent --zone=public --add-service=http`
`firewall-cmd --permanent --zone=public --add-service=https`
`firewall-cmd --reload`
16. Проверьте доступность сайта с помощью браузера: `http://server02.11-101-N.local`.
17. Установите PHP5:
`yum -y install php`
18. Найдите и установите несколько дополнительных пакетов-расширений для PHP:
`yum search php`
`yum install php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-snmp php-soap curl curl-devel php-mysql -y`

ПРИМЕЧАНИЕ:

Обязательно установите: `php-mysql`.

`yum install php-mysql -y`

19. Перезапустите службу Apache HTTP Server:

```
# systemctl restart httpd.service
```

20. Протестируйте работу PHP, создав страницу и открыв ее в браузере `http://server02.11-100-X.local/test.php`:

```
# vi /var/www/html/test.php
```

```
<?php
phpinfo();
?>
```

21. Установит phpMyAdmin:

```
# yum install phpMyAdmin -y
```

22. Разрешите удаленный доступ к phpMyAdmin, изменив секцию <Directory> в файле: `/etc/httpd/conf.d/phpMyAdmin.conf`:

```
# vi /etc/phpMyAdmin/config.inc.php
```

```
<Directory /usr/share/phpMyAdmin/>
    Options none
    AllowOverride Limit
    Require all granted
</Directory>
```

23. Измените аутентификацию phpMyAdmin с cookie на http:

```
# vi /etc/phpMyAdmin/config.inc.php
```

```
$cfg['Servers'][$i]['auth_type'] = 'http'; // Authentication
method (config, http or cookie based)?
```

24. Перезапустите службу Apache HTTP Server:

```
# systemctl restart httpd.service
```

25. Проверьте работу phpMyAdmin с помощью браузера: `http://server02.11-100-X.local/phpmyadmin`, используйте учетную запись: `root` и пароль `Pa$$w0rd`.

Упражнение 2 – Настройка SSL для Apache HTTP Server (Дополнительно).

Самостоятельно настройте SSL шифрование для одного из сайтов на **c7-server01**.

Упражнение 3 – Подготовка окружения к развертыванию squid.

1. Перед выполнением упражнения 3 откатите виртуальные машины в начальное состояние:

- **c7-server01**
- **c7-client01**
- **c7-client02**

2. На машине **c7-server01** настройте сетевой интерфейс **eth0**:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0

BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.X.2
NETMASK=255.255.255.0
```

3. На машине **c7-server01** настройте сетевой интерфейс **eth2**:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0

BOOTPROTO=dhcp
DEVICE=eth2
ONBOOT=yes
```

4. На машине **c7-client01** настройте сетевой интерфейс **eth0**:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0

BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.X.51
NETMASK=255.255.255.0
```

5. На машине **c7-client02** настройте сетевой интерфейс **eth0**:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0

BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.X.52
NETMASK=255.255.255.0
```

6. Проверьте доступ во внешнюю сеть на машине **c7-server01**.

```
# ping lebedevum.blogspot.com -c 4
```

7. Проверьте доступ в отсутствие внешнего доступа на машинах **c7-client01** и **c7-client02**.

```
# ping lebedevum.blogspot.com -c 4
```

Упражнение 4 – Установка и настройка squid.

1. Установите Squid (**squid**).
2. Настройте следующие параметры в **squid.conf**:

```
http_port 3128
cache_dir ufs /usr/local/squid/var/cache/squid 100 16 256
visible_hostname proxy01.ll-100-X.local
```

3. Откройте tcp порт 3128 в firewalld.

```
# firewall-cmd -zone=public -add-port=3128/tcp -permanent
# firewall-cmd -reload
```

4. Запустите службу squid и настройте ее автоматический запуск.

Ответы к упражнению 4.

1. Установите Squid (**squid**).

```
# yum install squid -y
```

2. Настройте следующие параметры в **squid.conf**:

```
# vi /etc/squid/squid.conf

# Строка 59 - проверьте.
http_port 3128
# Строка 62 - раскомментируйте.
cache_dir ufs /usr/local/squid/var/cache/squid 100 16 256
# Строка 75 - добавьте.
visible_hostname proxy01.11-100-X.local
```

3. Откройте tcp порт 3128 в firewalld.

```
# firewall-cmd -zone=public -add-port=3128/tcp -permanent
# firewall-cmd -reload
# iptables-save
```

4. Запустите службу squid и настройте ее автоматический запуск.

```
systemctl start squid
systemctl status squid
systemctl enable squid
systemctl is-enabled squid
```

Упражнение 4 – Проверка работы squid (настройка клиентов).

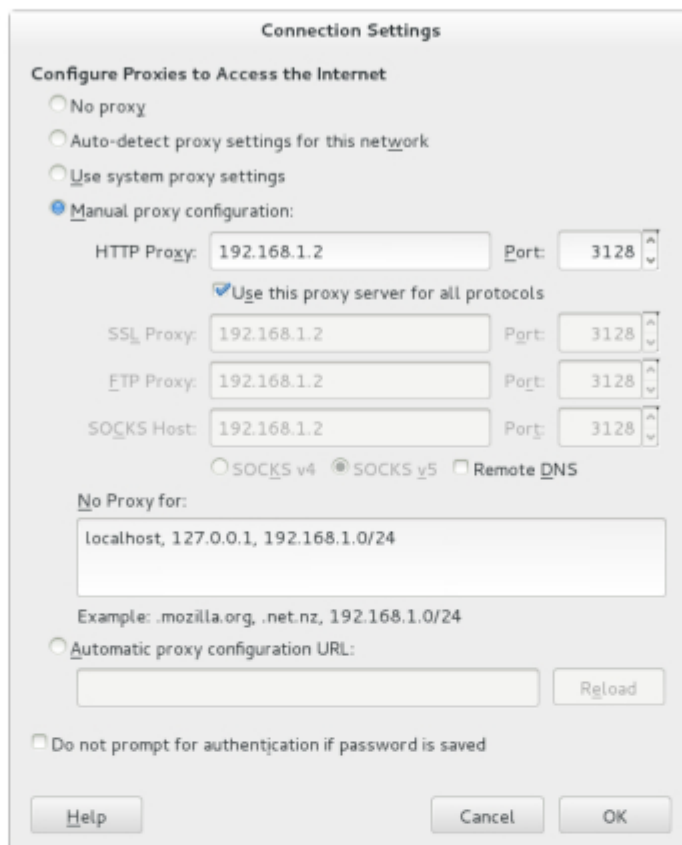
1. На машине **c7-server01** выведите журнал доступа squid (**access.log**).
2. На машине **c7-client01** настройте Mozilla Firefox на работу с Proxy-сервером: **192.168.1.2**.
3. Проверьте доступ в интернет – <http://lebedevum.blogspot.com>.
4. На машине **c7-client02** настройте Konqueror на работу с Proxy-сервером: **192.168.1.2**.
5. Проверьте доступ в интернет – <http://lebedevum.blogspot.com>.

Ответы к упражнению 5.

1. На машине **c7-server01** выведите журнал доступа squid (**access.log**).

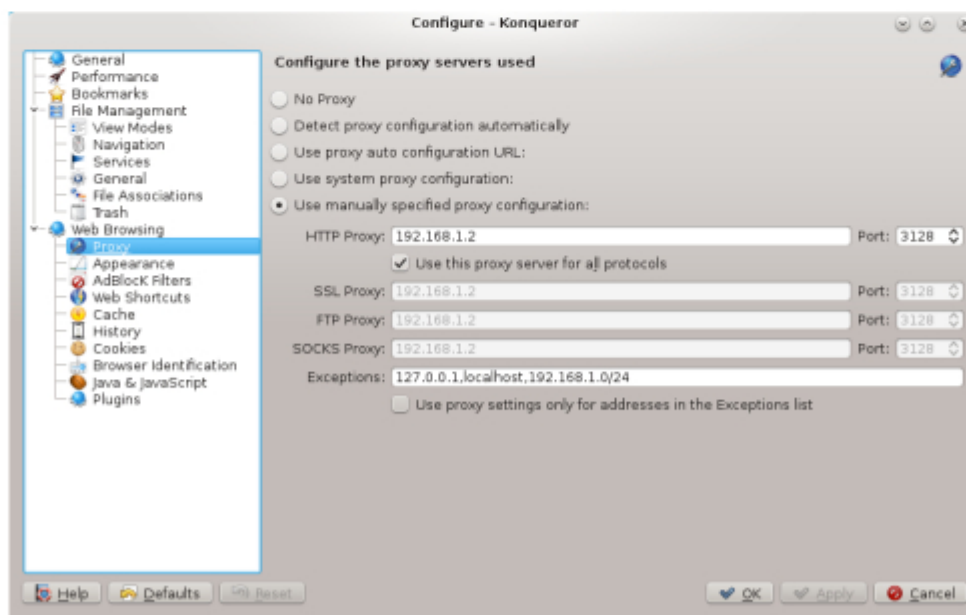
```
# tail -f /var/log/squid/access.log
```

2. На машине **c7-client01** настройте Mozilla Firefox на работу с Proxy-сервером: **192.168.1.2**.



3. Проверьте доступ в интернет – <http://lebedevum.blogspot.com>.

4. На машине **c7-client02** настройте Konqueror на работу с Proxy-сервером: **192.168.1.2**.

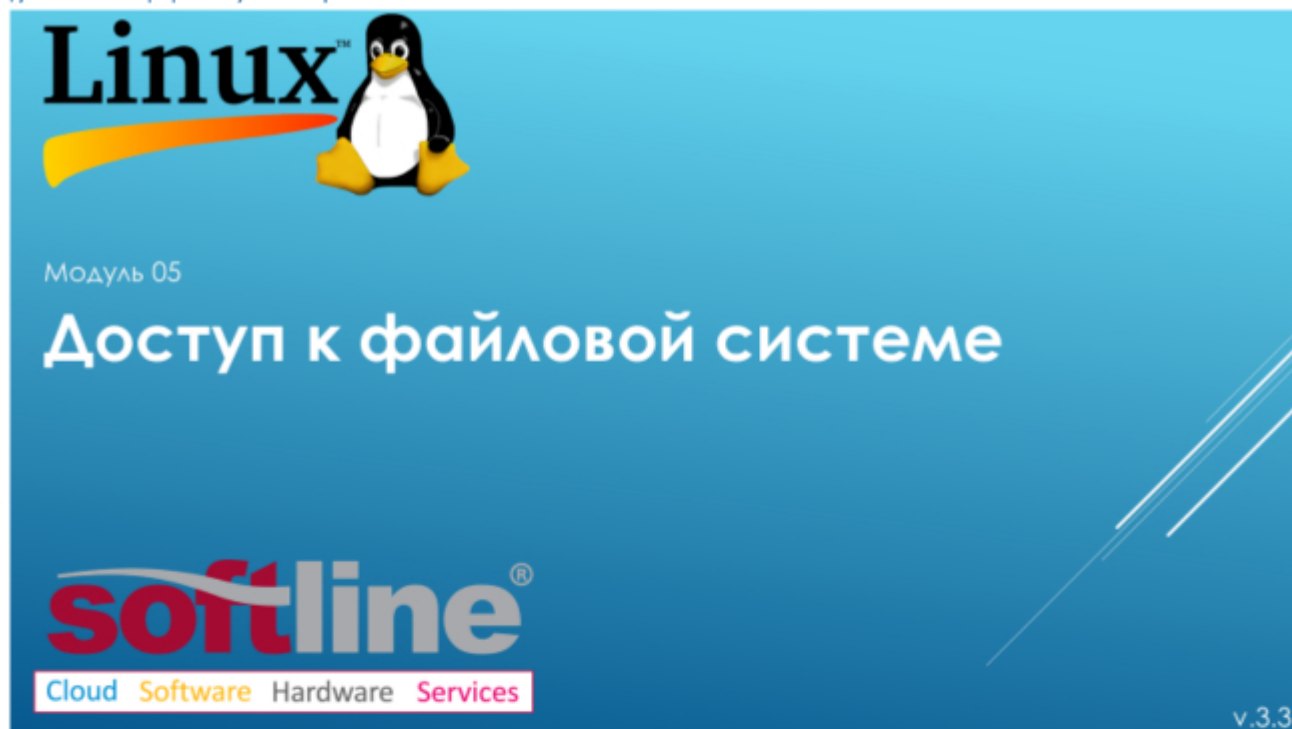


5. Проверьте доступ в интернет – <http://lebedevum.blogspot.com>.

Упражнение 6 – Разграничение доступов в squid (Дополнительно).

Самостоятельно настройте разграничение доступов на squid – полностью запретите доступ к интернету для c7-client01, а для c7-client02 оставьте доступ к сети.

Модуль 05: Доступ к файловой системе.



В модуле представлены:

- Основные концепции протокола FTP.
- FTP-сервер **vsftpd**.
- Основные концепции протокола NFS.
- NFS-сервер **nfsd**.
- NFS-клиент.
- Основные концепции протоколов SMB/CIFS.
- SMB-сервер **samba**.

Протокол FTP

- File Transfer Protocol – протокол передачи файлов.
- В обычном режиме использует встречное TCP-соединение (Active Mode):
 - Имеет расширение – Passive Mode.
- Соединение происходит без использования шифрования.
- Программная реализация:
 - Сервер:
 - *vsftpd*
 - *proftpd*
 - *Pure-FTPd*.
 - Клиент:
 - *lftp*.
 - Браузеры.
 - *FileZilla*.

<http://edu.softline.ru/>



Протокол, работающий на прикладном уровне сетевой модели OSI. Предназначен для просмотра содержимого файловой системы удалённого узла и передачи файлов. Протокол FTP является одним из старейших протоколов семейства TCP.

Основным недостатком является незащищённость соединений, то есть вся конфиденциальная информация будет передаваться по сети в открытом виде, что может привести к утечке важных данных, именно поэтому в последних программных реализациях включена поддержка SSL-шифрования.

FTP является протоколом, использующим необычную схему установления соединения между сетевыми узлами. Для работы по протоколу FTP используются два TCP-порта – 20 (ftp-data) и 21 (ftp), первый порт используется для передачи данных, а второй для передачи управляющих команд. Изначально протокол предполагал создание встречного TCP-соединения.

Создание сессии в активном режиме выглядит так – ftp-клиент производит соединение с произвольного непривилегированного порта на 21-й порт сервера. После этого клиент открывает порт, инкрементированный на единицу (N+1) и выдаёт команду серверу PORT, указывая на прослушиваемый порт. Сервер, получив данные о том, какой порт клиента готов для передачи данных, производит соединение с этим портом с 20-го порта. Проблемы возникают тогда, когда на пути входящего соединения с сервера встают сетевые фильтры, которые блокируют такой трафик, не говоря уже о сетевых узлах, находящихся за NAT. Решение было найдено быстро и эффективно.

Создание сессии в пассивном режиме выглядит наоборот – ftp-клиент сразу открывает два непривилегированных порта и соединяется с 21-м портом сервера. Но теперь, вместо команды PORT ftp-клиент посылает команду PASV, сообщая серверу, что соединение производится в пассивном режиме. Ftp-сервер приняв команду PASV производит открытие непривилегированного порта и выдаёт команду PORT ftp-клиенту. Теперь уже клиент производит второе соединение и начинается передача данных.

Существует специальный пользователь для анонимного доступа к содержимому FTP-сервера – *anonymous*. Некоторые серверы имеют каталог *uploads/incoming*, в которые допускается запись файлов анонимными пользователями.

FTP-сервер **vsftpd**

- Реализация протокола FTP, отличающаяся высоким уровнем безопасности и производительности:
 - Поддержка виртуальных учётных записей.
 - Работа с виртуальными узлами.
 - Гибкая настройка разнообразных лимитов.
 - Шифрование с помощью SSL.
 - Возможность запуска в двух режимах:
 - `standalone`.
 - `transient`.
- Основной конфигурационный файл:
 - `/etc/vsftpd/vsftpd.conf`.

<http://edu.softline.ru/>



Настройка сервера сервера **vsftpd** производится очень просто. Необходимо произвести редактирование конфигурационного файла **vsftpd.conf** для достижения необходимого результата. Файл снабжён подробнейшими комментариями.

```
anonymous_enable=YES      #разрешать анонимный вход
local_enable=YES           #разрешить вход с использованием системных учётных записей
write_enable=YES           #разрешить запись
local_umask=022            #параметр umask для создаваемых файлов и каталогов
```

Иногда, необходимо предоставлять право записи анонимным пользователям, для этого существуют следующие директивы:

```
anon_upload_enable=YES    #разрешать анонимным пользователям загрузку файлов
anon_mkdir_write_enable=YES #разрешать анонимным пользователям создавать каталоги
```

Важным параметром безопасности является **chroot_list_enable**, отвечающий за "закрытие" пользователя в его домашнем каталоге, это необходимо для того, чтобы он не мог подняться выше и, например посмотреть файл `/etc/passwd`. Этот параметр используется только совместно с параметром **chroot_list_file**, определяющем путь к текстовому файлу, в котором построчно будут указаны учётные записи. Если **chroot_list_enable** установлен в значение **YES**, то в файле указываются те, кого следует "закрывать", если **NO**, то в файле указываются те, кого НЕ надо закрывать.

```
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
# cat /etc/vsftpd/chroot_list
user
```

Протокол NFS

- Network File System – протокол сетевого доступа к файловым системам.
- Работа основана на протоколе RPC.
- Осуществляет подключение удалённых файловых систем:
 - Абстрагирована от типа файловой системы.
 - Поддерживает различные средства аутентификации.
- Программная реализация:
 - Сервер:
 - `nfsd`.
 - Клиент:
 - `nfs`.
- Последняя версия - NFSv4.

<http://edu.softline.ru/>



Протокол NFS был разработан компанией SUN в 1985 году. Основное предназначение протокола – предоставление прозрачного доступа к удалённым файловым системам, следовательно, архитектура NFS предполагает использование клиент-серверной модели, когда сервер «экспортирует» файловые системы, а клиенты могут получить к ним доступ.

Основной конфигурационный файл для экспортирования файловых систем – `/etc/exports`. В данном файле описываются все экспортируемые файловые системы или отдельные каталоги. Синтаксис, включает в себя:

path hosts (options), где:

path – путь к экспортируемому каталогу;

hosts – сетевые узлы, которым разрешено подключать каталог;

options – опции, описанные в ман-руководстве `exports`.

В качестве опций могут использоваться следующие:

rw – разрешать подключение в режиме read-write;

ro – разрешать подключение только в режиме read only;

anonuid – идентификатор пользователя, используемый для запросов от анонимного пользователя;

anongid – идентификатор группы, используемый для запросов от анонимной группы;

root_squash – все запросы, производимые от суперпользователя root (`uid=0/gid=0`) производить от анонимного `uid` и `gid`;

no_root_squashing – не изменять `uid/gid` суперпользователя root;

В качестве примера рассмотрим следующую запись:

```
/var/music          192.168.1.0.24 (ro)
/var/finance        *.example.com (rw,no_root_squashing)
```

NFS-сервер **nfsd**

- Требуется поддержка в ядре.
- Осуществляет "экспорт" файловых систем.
- Основной конфигурационный файл – `/etc/exports`.
- Экспортом каталогов управляет программа **exportfs**:
 - Вызов без параметров приведёт к выводу всех экспортированных данных.
 - **exportfs -r** – очистить таблицу ядра и внести в неё все данные из конфигурационного файла.

<http://edu.softline.ru/>



Сервер NFS использует протокол RPC и зависит от сервисов `portmap` и `rpc.mountd`, которые должны быть запущены в системе. Также требуется поддержка протокола `nfs` в ядре.

Необходимая настройка уже была произведена в конфигурационном файле `/etc/exports`, теперь мы должны применить внесённые изменения следующей командой:

```
# exportfs -r
```

Также команда **exportfs** может принимать следующие параметры:

- **v** – вывод всех ресурсов, предоставленных в доступ посредством NFS
- **a** – произвести экспорт всех ресурсов, описанных в `/etc/exports`.

NFS-клиент

- Требуется поддержка в ядре.
- Подключение ресурса, предоставленного в доступ сервером осуществляется командой `mount`:

```
# mount nfs.example.com:/var/music /mnt/nfs
```
- Определение в файле `/etc/fstab`.
- Автоматическое подключение с помощью `autofs`.

<http://edu.softline.ru/>



Как и любое устройство хранения информации, ресурсы NFS подключаются с помощью команды `mount`, отключение, соответственно, осуществляется с помощью команды `umount`.

Основной файл с инструкциями по подключению устройств хранения информации – `/etc/fstab`, который читается программой `mount`. В этом файле может быть описан и любой NFS-ресурс:

```
nfs.example.com:/var/music /mnt/nfs nfs defaults 0 0
```

Очень интересным является использование программы **autofs**. Достаточно представить ситуацию, когда в организации используется несколько десятков серверов и несколько сотен пользователей, которые хотят всегда работать в привычном пространстве и с одним и тем же профилем. Для этого создают отдельный сервер, на котором надёжно хранятся домашние каталоги пользователей. На каждом из рабочих серверов система **autofs** настраивается таким образом, что при входе в свой домашний каталог, пользователь инициирует подключение nfs-ресурса с пользовательскими данными.



ПРИМЕЧАНИЕ:

В состав Windows 8 и Windows 8.1 в редакции Enterprise входит NFS-клиент.

Протокол SMB/CIFS

- Server Message Block – протокол совместного использования файлов Microsoft Windows.
- CIFS (Common Internet File System) – протокол, основанный на SMB и являющийся его преемником.
- Работает поверх протокола NetBIOS.
- SMB/CIFS обеспечивает:
 - Управление сессиями.
 - Файловый доступ.
 - Сервис печати.
 - Сервис сообщений.

<http://edu.softline.ru/>



Протокол SMB был создан компаниями Microsoft и IBM для обеспечения совместного доступа к файлам в рамках протокола NetBIOS. NetBIOS работает на трёх уровнях модели OSI – сетевом, транспортном, сеансовом, но не поддерживает маршрутизацию, что делает его протоколом локальных сетей. SMB – протокол прикладного уровня сетевой модели OSI, осуществляющий следующие функции:

- Управление сессиями.
- Доступ к удалённой файловой системе.
- Постановка документов в очередь печати сетевого принтера.
- Передача сообщений.

На сегодняшний день основным протоколом является CIFS (Common Internet File System), созданный на основе протокола SMB.

В Linux присутствует поддержка обоих протоколов.

SMB-сервер **samba**



- **SAMBA – программная реализация протокола SMB/CIFS, обеспечивает:**
 - Авторизацию и аутентификацию пользователей.
 - Предоставление доступа к файловой системе.
 - Предоставление доступа к принтерам.
 - Разрешение имён в среде NetBIOS.
 - Построение доменной среды для MS Windows.
- **Основной конфигурационный файл - /etc/samba/smb.conf.**
- **Состоит из двух частей:**
 - **smbd** – функции AA, доступ к ресурсам.
 - **nmdb** – сервер имён WINS, сетевое окружение.



<http://edu.softline.ru/>

SAMBA – программная реализация протокола SMB, предназначенная, в основном, для обеспечения взаимодействия с серверами и рабочими станциями под управлением ОС Windows. Разрабатывается Samba Team под предводительством Andrew Tridgel.

На сегодняшний день Samba представляет собой мощное средство работы в гетерогенных сетях, выступая связующим звеном между ОС Windows и ОС Linux и предлагая следующие возможности:

- Авторизация и аутентификация пользователей – возможность использования централизованных сетевых хранилищ учётных записей. С помощью сервиса winbind возможно установление соответствий между учётными записями Active Directory и учётными записями Linux.
- Предоставление доступа к файловой системе и принтерам.
- Разрешение имён – samba может выступать в качестве WINS-сервера, поддерживает репликацию списков сетевого окружения.

Также, в ядро Linux включена поддержка файловой системы **smb/cifs**, что позволит администратору подключать удалённые каталоги в существующую файловую структуру посредством команды mount и её конфигурационного файла – **/etc/fstab**.

Работа сервера Samba распределена между двумя сервисами – **smbd** и **nmdb**.

Сервис **smbd** необходим для реализации таких сервисов, как авторизация и аутентификация, а также для предоставления доступа к локальным ресурсам (каталогам, принтерам).

Основной функцией сервиса **nmdb** является разрешение имён и реализация "сетевого окружения".

Настройка сервера **samba**



- Конфигурационный файл `/etc/samba/smb.conf` состоит из секций:
 - `[global]` – секция предназначена для определения основных параметров работы сервера.
 - `[homes]` – секция предоставляет доступ пользователей к своим домашним каталогам посредством протокола smb.
 - `[printers]` – секция предназначена для определения принтеров, которые необходимо предоставить в доступ.
- Каждый открываемый ресурс должен иметь свою секцию:
 - `[music]`
 - `[documentation]`
- Утилита `testparm` позволяет осуществить проверку файла `smb.conf` на предмет правильности синтаксиса.

<http://edu.softline.ru/>



Конфигурирование сервера Samba осуществляется посредством редактирования файла `/etc/samba/smb.conf`, разделённого на секции, каждая из которых содержит определённый набор параметров.

Основная секция `[global]` предназначена для определения параметров, влияющих на работу сервера в целом и параметров, определяющих умолчательные значения для других секций. В этой секции определяются такие параметры как:

```
[global]
workgroup = MYGROUP           # имя рабочей группы либо имя домена
server string = Samba Server  # описание сервера
security = user                # тип безопасности
wins support = yes            # взять на себя функции WINS-сервера
```

Тип безопасности может принимать следующие значения:

- **user** – данный режим безопасности обеспечивает предоставление доступа к ресурсам (даже списку) только после авторизации.
- **share** – данный режим отличается от `user` тем, что авторизоваться необходимо при попытке доступа непосредственно к определённому ресурсу, а не при входе на сервер.
- **domain** – идентичен типу `user`, но для использования необходимо ввести samba-сервер в домен Windows. После этого авторизация будет производиться с помощью серверов, указанных параметром:

```
password server = server1 server2
```

- **ads** – samba-сервер применяет авторизацию через Active Directory с использованием Kerberos:

```
realm = KERBEROS.REALM
password server = kerberos.server
```

Теперь, когда определены глобальные параметры и тип авторизации, остаётся определить ресурсы, предоставляемые в доступ. Каждый ресурс должен быть описан отдельной секцией, содержащей необходимые параметры. Рассмотрим основные:

- **comment** – комментарий.
- **path** – путь к каталогу.
- **browseable** – ресурс должен быть виден в списке доступных ресурсов при входе на сервер.
- **writable** – пользователи могут производить запись (создавать/удалять/изменять файлы).
- **write list** – список пользователей, кому разрешена запись.
- **printable** – ресурс поддерживает печать (для принтеров).
- **valid users** – список пользователей, кому разрешён доступ к ресурсу.

Рассмотрим пример описания ресурсов:

```
[homes]
    comment = Home Directories
    browseable = yes
    writable = yes
```

- Ресурс **[homes]** является служебным и предоставляет доступ к домашним каталогам пользователей.

```
[public]
    comment = Exchange
    path = /var/samba
    public = yes
    writable = yes
    write list = @users
```

- Ресурс **[public]** предоставляет публичный доступ к каталогу **/var/samba**. Включена возможность записи для членов группы **users**.

Сетевая печать



- Сервер SAMBA позволяет предоставлять доступ к принтерам:
 - Опция `printing` указывает источник информации о принтерах:
`printing = cups`
 - Опция `load printers` указывает на необходимость автоматической загрузки всех доступных принтеров:
`load printers = yes`
 - Специальная секция `[printers]` предоставляет сетевой доступ к принтерам:

```
[printers]
path = /usr/spool/samba
printable = yes
```

<http://edu.softline.ru/>



Сервер Samba позволяет предоставлять в доступ принтеры, определённые в системе **cups**. Параметр «**load printers**» включает автоматическую загрузку информации о принтерах из CUPS:

```
load printers = yes
[printers]
    comment = Printers
    path = /usr/spool/samba
    browseable = yes
    guest ok = no
    printable = yes
```

Пользователи **samba**



- Сервер **samba** поддерживает авторизацию и аутентификацию посредством:
 - Протокола **ldap**.
 - Сервиса **winbindd**.
 - Локальной базы учётных записей.
- Учётные записи **samba** хранятся в файле **/etc/samba/smbpasswd**, который управляется утилитой **smbpasswd**:
 - **-a** – добавить системного пользователя в **smbpasswd**.
 - **-e** – активировать учётную запись.
 - **-d** – деактивировать учётную запись.

<http://edu.softline.ru/>



Изначально сервер Samba настроен на использование локальной базы данных с учётными записями пользователей – файла **/etc/samba/smbpasswd**. Управление базой осуществляется программой **smbpasswd**.

Добавляемый пользователь должен быть системным пользователем операционной системы, находиться в файле **/etc/passwd**. Для добавления используется команда:

```
# smbpasswd -a username
```

Вызванная без параметров, программа позволит изменить пароль пользователя:

```
# smbpasswd username
```

Утилиты управления



- **net** – многофункциональная утилита:
 - **net lookup** – разрешение имён.
 - **net user** – управление пользователями.
 - **net group** – управление пользователями.
 - **net rpc** – запуск RPC-команд.
- **smbclient** – утилита для доступа к ресурсам.
- **nmblookup** – утилита для осуществления запросов к серверу WINS.

<http://edu.softline.ru/>



Программа **net** предназначена для управления серверами Samba/CIFS. Вызванная без аргументов, эта программа выдаст подробную справку о принимаемых аргументах. Например, чтобы получить список пользователей домена необходимо дать команду:

```
# net user -U USERNAME
```

Для управления необходимо указание протокола, используемого для передачи команд:

- ADS – протокол, использующийся в ActiveDirectory.
- RAP – протокол, использующийся в старых версиях ОС Windows.
- RPC – протокол для управления операционными системами Windows 2000/XP.

Для получения списка возможных команд, передайте команде **net** имя протокола в качестве аргумента:

```
# net rpc
....
net rpc shutdown to shutdown a remote server
net rpc service to start, stop and query services
....
```

Программа **smbclient** предоставляет доступ к ресурсам так же, как если бы вы использовали протокол ftp. Чтобы подключиться к удалённому ресурсу необходимо дать команду:

```
# smbclient //server/share -U USERNAME
```

После подключения запустится оболочка, в которой пользователь может получить справку о возможных командах, дав команду **help**. Но, не трудно заметить, что утилита требует указания имени ресурса, которое не всегда известно. Чтобы получить список ресурсов, предоставляемых сервером, дайте команду:

```
# smbclient -L //server -U USERNAME
```

Программа **nmblookup** действует подобно программе **nslookup**, но применительно к именам, используемым в Samba/CIFS-окружении.

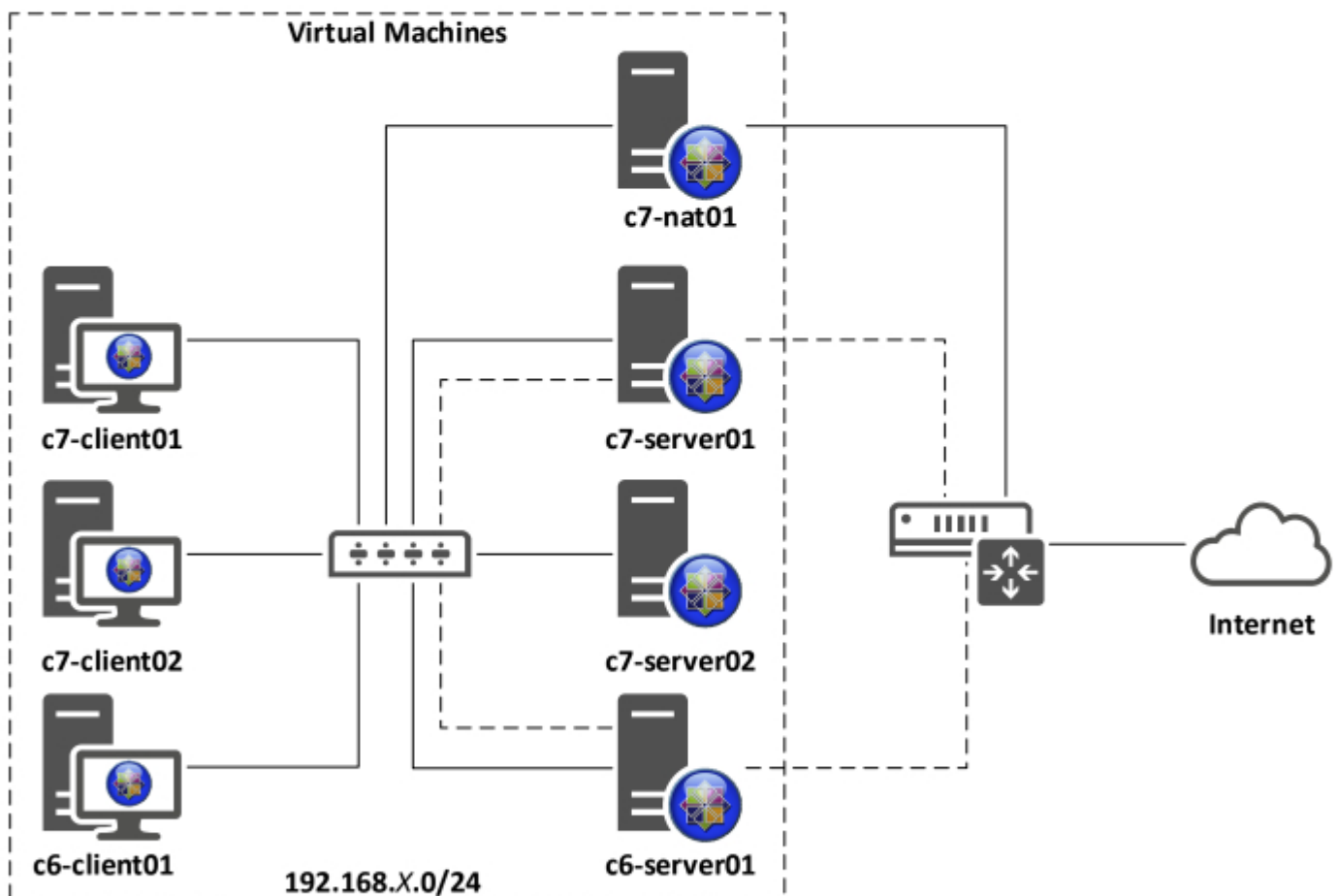
```
# nmblookup server
```


Лабораторная работа.

Лабораторная работа 5

- **Упражнения:**
 - Установка и настройка `vsftpd`.
 - Установка и настройка NFS-сервера.
 - Установка и настройка NFS-клиента.
 - Установка и настройка сервера `samba`.
 - Настройка доступа к папкам из Windows (дополнительно).
- **Необходимые машины:**
 - C7-NAT01
 - C7-SERVER01
 - C7-CLIENT01
- **Учетные данные:**
 - Пользователь: `root`.
 - Пароль: `Pa$$w0rd`.

<http://edu.softline.ru/>



Упражнение 1 – Установка и настройка vsftpd.

1. Установите пакеты **vsftpd** и **ftp**.

```
# yum install vsftpd ftp -y
```

2. Отредактируйте файл **vsftpd.conf**, внесите следующие параметры:

```
anonymous_enable=NO
ascii_upload_enable=YES
ascii_download_enable=YES
ftpd_banner=Welcome to LL-103 FTP service Demo!
use_localtime=YES
```

3. Запустите службу **vsftpd** и настройте ее автоматический запуск.

```
# systemctl start vsftpd
# systemctl status vsftpd
# systemctl enable vsftpd
# systemctl is-enabled vsftpd
```

4. Разрешите службу **ftp** и tcp-порт 21 в **firewalld**.

```
# firewall-cmd --permanent --add-port=21/tcp
# firewall-cmd --permanent --add-service=ftp
# firewall-cmd --reload
```

5. Добавьте разрешение в SELinux.

```
# setsebool -P ftp_home_dir on
```

6. Добавьте пользователя **user01**.
7. Проверьте работу FTP-сервера локально при помощи команды **ftp**.
8. Проверьте работу FTP-сервера при помощи Mozilla Firefox с машины **c7-client01**.
9. Установите на машину **c7-client01** FileZilla (**filezilla**) из репозитория EPEL.

**ПРИМЕЧАНИЕ:**

*FileZilla (**filezilla**) не входит в репозиторий по умолчанию, для ее установки используется репозиторий EPEL.*

Ответы к упражнению 1.

1. Установите пакеты **vsftpd** и **ftp**.

```
# yum install vsftpd ftp -y
```

2. Отредактируйте файл **vsftpd.conf**.

```
# vi /etc/vsftpd/vsftpd.conf
```

```
# Строка 12 – закоментируйте.
```

```
anonymous_enable=NO
```

```
# Строки 82, 83 – раскомментируйте.
```

```
ascii_upload_enable=YES
```

```
ascii_download_enable=YES
```

```
# Строка 86 – Раскомментируйте и введите приветствие.
```

```
ftpd_banner=Welcome to LL-103 FTP service Demo!
```

```
# Строка 130 – добавьте строку.
```

```
use_localtime=YES
```

3. Запустите службу **vsftpd** и настройте ее автоматический запуск.

```
# systemctl start vsftpd
```

```
# systemctl status vsftpd
```

```
# systemctl enable vsftpd
```

```
# systemctl is-enabled vsftpd
```

4. Разрешите службу **ftp** и tcp-порт 21 в **firewalld**.

```
# firewall-cmd --permanent --add-port=21/tcp
```

```
# firewall-cmd --permanent --add-service=ftp
```

```
# firewall-cmd --reload
```

```
# iptables-save
```

5. Добавьте разрешение в SELinux.

```
# setsebool -P ftp_home_dir on
```

6. Добавьте пользователя.

```
# useradd user01
```

```
# passwd user01
```

7. Проверьте работу FTP-сервера.

```
# ftp 192.168.1.101
```

```
Connected to 192.168.1.101 (192.168.1.101).
```

```
220 Welcome to CentOS Based FTP service.
```

```
Name (192.168.1.101:root): user01
```

```
331 Please specify the password.
```

```
Password:
```

```
230 Login successful.
```

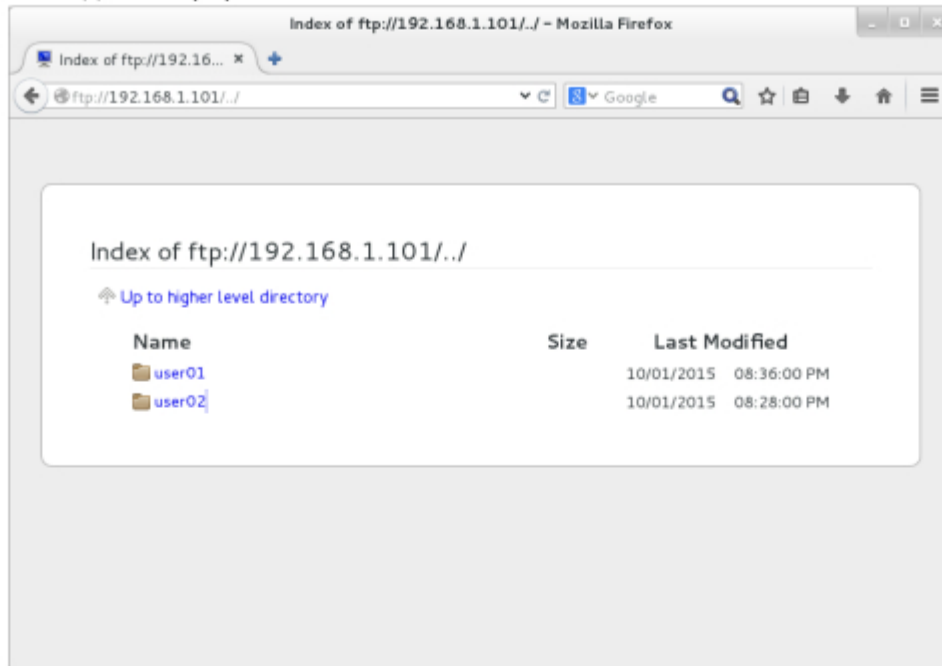
```
Remote system type is UNIX.
```

```
Using binary mode to transfer files.
```

```
ftp> quit
```

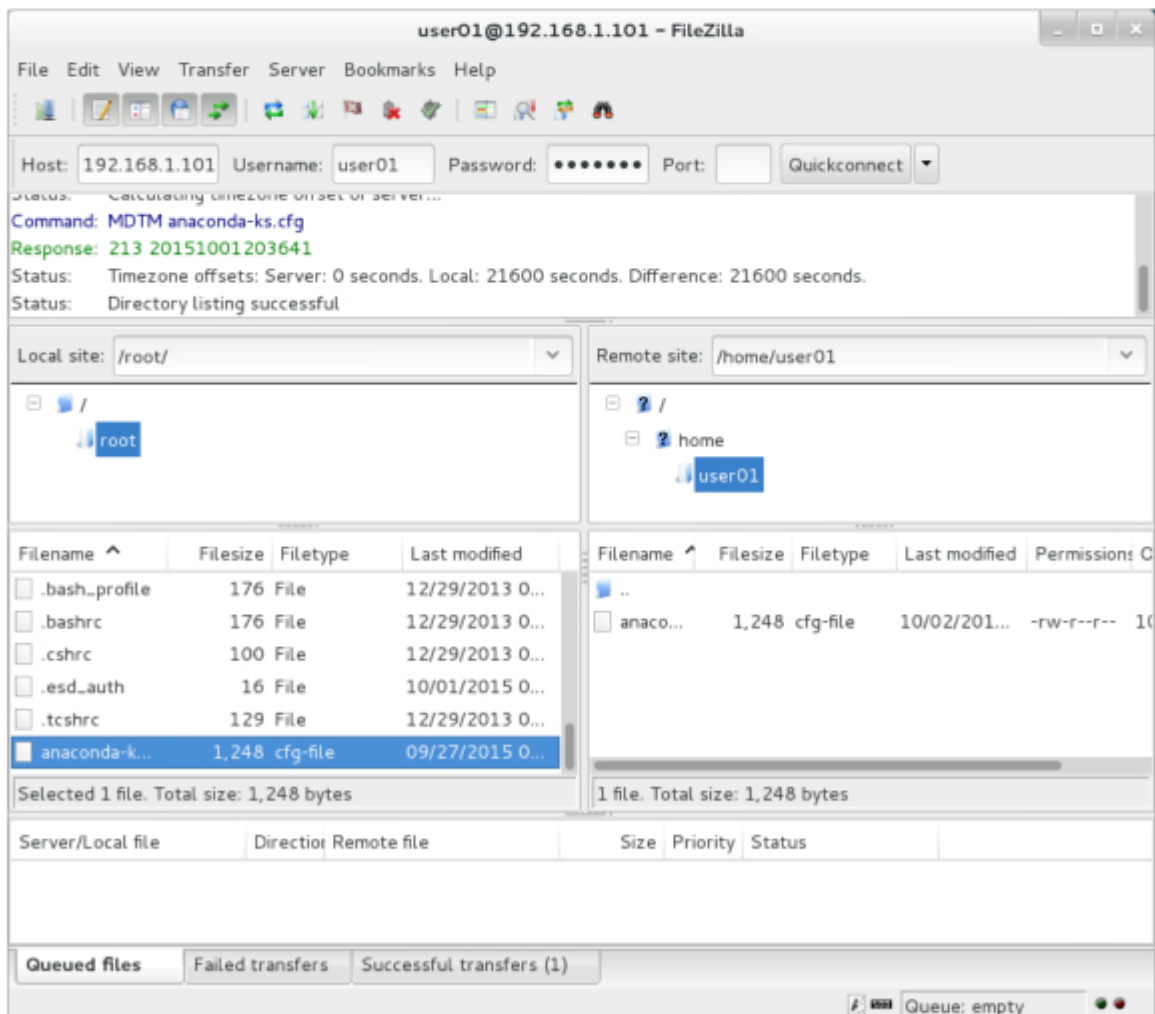
```
221 Goodbye.
```

10. Проверьте работу FTP-сервера при помощи Mozilla Firefox с машины **c7-client01**.



11. Установите на машину **c7-client01** FileZilla (**filezilla**) из репозитория EPEL.

```
# yum install epel-release -y
# yum install filezilla -y
```



Упражнение 2 – Установка и настройка NFS-сервера.

1. Установите пакет **nfs-utils**.
2. Создайте каталог для экспорта **/var/nfs**.
3. Предоставьте следующие права на каталог:
 - User **rw**
 - Group **rw**
 - Other **rw**
4. Запустите службы **rpcbind** и **nfs-server**, затем настройте их автоматический запуск.
5. Экпортируйте каталоги **/home** и **/var/nfs** для **s7-client01**, при экспорте используйте следующие опции:
 - **rw**
 - **sync**
 - **no_root_squash**
 - **no_all_squash**

**ПРИМЕЧАНИЕ:**

Для экспорта, без указания клиента используйте *****.

6. Перезапустите службу **nfs-server**.
7. Добавьте исключение в **firewalld** для службы **nfs**.

```
# firewall-cmd --permanent --zone=public --add-service=nfs
# firewall-cmd --reload
```

Ответы к упражнению 2.

8. Установите пакет **nfs-utils** на машине **c7-server01**.

```
# yum install nfs-utils
```

9. Создайте каталог для экспорта **/var/nfs**.

```
# mkdir /var/nfs
```

10. Предоставьте следующие права на каталог:

- User **rw**
- Group **rw**
- Other **rw**

```
# chmod -R 777 /var/nfs/
```

11. Запустите службы **rpcbind** и **nfs-server**, затем настройте их автоматический запуск.

```
# systemctl start rpcbind
# systemctl status rpcbind
# systemctl enable rpcbind
# systemctl is-enabled rpcbind

# systemctl start nfs-server
# systemctl status nfs-server
# systemctl enable nfs-server
# systemctl is-enabled nfs-server
```

12. Экспортируйте каталоги **/home** и **/var/nfs** для **c7-client01**, при экспорте используйте следующие опции:

- **rw**
- **sync**
- **no_root_squash**
- **no_all_squash**

```
# vi /etc/exports
/var/nfsshare 192.168.X.Y(rw,sync,no_root_squash,no_all_squash)
/home        192.168.X.Y(rw,sync,no_root_squash,no_all_squash)
```

**ПРИМЕЧАНИЕ:**

Для экспорта, без указания клиента используйте *****.

13. Перезапустите службу **nfs-server**.

```
# systemctl restart nfs-server
```

14. Добавьте исключение в **firewalld** для службы **nfs**.

```
# firewall-cmd --permanent --zone=public --add-service=nfs
# firewall-cmd --reload
# iptables-save
```

Упражнение 3 – Установка и настройка NFS-клиента.

1. На машине **c7-client01** установите пакет **nfs-utils**.
2. Создайте следующие каталог для монтирования.
 - **/mnt/nfs/home**
 - **/mnt/nfs/var/nfsshare**
3. Смонтируйте каталоги согласно схеме:
 - **192.168.0.100:/home => /mnt/nfs/home/**
 - **192.168.0.100:/var/nfs => /mnt/nfs/var/nfsshare/**
4. Проверьте монтирование.
5. Создайте файл во вновь смонтированном каталоге **/mnt/nfs/var/nfs/**
6. При помощи файла **/etc/fstab** настройте монтирование в процессе загрузки системы.
7. Перезапустите **c7-client01** и убедитесь, что каталоги смонтированы.

Ответы к упражнению 3.

1. На машине **c7-client01** установите пакет **nfs-utils**.

```
# yum install nfs-utils
```

2. Создайте следующие каталог для монтирования.

- **/mnt/nfs/home**
- **/mnt/nfs/var/nfsshare**

```
# mkdir -p /mnt/nfs/{home, var/nfs}
```

3. Смонтируйте каталоги согласно схеме:

- **192.168.0.100:/home => /mnt/nfs/home/**
- **192.168.0.100:/var/nfs => /mnt/nfs/var/nfsshare/**

```
# mount -t nfs 192.168.X.100:/home /mnt/nfs/home/  
# mount -t nfs 192.168.X.100:/var/nfs /mnt/nfs/var/nfs/
```

4. Проверьте монтирование.

```
# df -kh
```

5. Создайте файл во вновь смонтированном каталоге **/mnt/nfs/var/nfs/**

```
# ls -R / >> /mnt/nfs/var/nfs/test.txt
```

6. При помощи файла **/etc/fstab** настройте монтирование в процессе загрузки системы.

```
# vi /etc/fstab  
192.168.0.100:/home /mnt/nfs/home nfs defaults 0 0  
192.168.0.100:/var/nfsshare /mnt/nfs/var/nfsshare nfs defaults 0 0
```

7. Перезапустите **c7-client01** и убедитесь, что каталоги смонтированы.

```
# reboot  
# df -kh
```

Упражнение 4 – Установка и настройка сервера samba.

1. Установите пакеты сервера Samba (**samba***) на машине **c7-server01**.

```
# yum install samba* -y
```

2. Создайте каталог для предоставления в общий доступ: **/samba/share01**.

```
# mkdir -p /samba/share01
```

3. Назначьте на каталог права с маской (**0777**).

```
# chmod -R 0777 /samba/anonymous_share
```

4. Отредактируйте конфигурационный файл **smb.conf**.

```
# vi /etc/samba/smb.conf
```

```
#Внесите следующие изменения в секции [global].
```

```
# Добавьте строки.
```

```
unix charset = UTF-8
```

```
dos charset = CP932
```

```
# Смените группу на группу Windows по умолчанию.
```

```
workgroup = WORKGROUP
```

```
# раскомментируйте.
```

```
hosts allow = 127. 192.168.X.
```

```
# Раскомментируйте.
```

```
max protocol = SMB2
```

```
# Проверьте.
```

```
security = user
```

```
#Добавьте строку
```

```
map to guest = Bad User
```

```
#Добавьте следующую секцию в конец файла.
```

```
[share01]
```

```
path = /samba/share01
```

```
writable = yes
```

```
browsable = yes
```

```
guest ok = yes
```

```
guest only = yes
```

```
create mode = 0777
```

```
directory mode = 0777
```

5. Запустите службы **smb** и **nmb**, затем настройте их автоматический старт.

```
# systemctl start smb
```

```
# systemctl status smb
```

```
# systemctl start nmb
```

```
# systemctl status nmb
```

```
# systemctl enable smb
```

```
# systemctl enable nmb
```

6. Проверьте конфигурацию Samba

```
# testparm
```

Вывод должен иметь следующий вид:

```
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit
(16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[share01]"
Loaded services file OK.
...
[share01]
  path = /samba/share01
  read only = No
  create mask = 0777
  directory mask = 0777
  guest only = Yes
  guest ok = Yes
```

7. Настройте разрешения в firewalld.

```
# firewall-cmd --permanent --add-port=137/tcp
# firewall-cmd --permanent --add-port=138/tcp
# firewall-cmd --permanent --add-port=139/tcp
# firewall-cmd --permanent --add-port=445/tcp
# firewall-cmd --permanent --add-port=901/tcp
# firewall-cmd --reload
# iptables-save
```

8. Отключите SELinux.

```
# vi /etc/sysconfig/selinux
SELINUX=disabled
# reboot
```

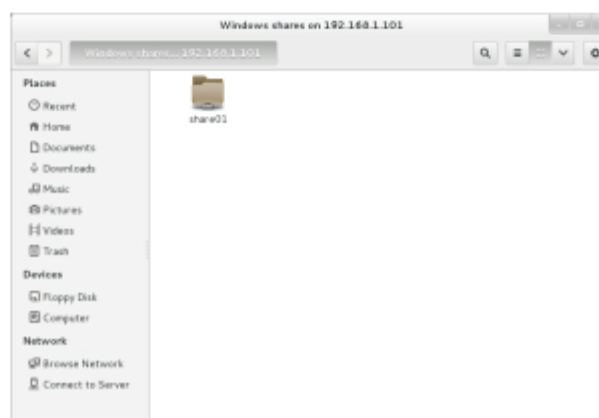


ПРИМЕЧАНИЕ:

Чтобы не перезагружать компьютер можно использовать следующие настройки SELinux:

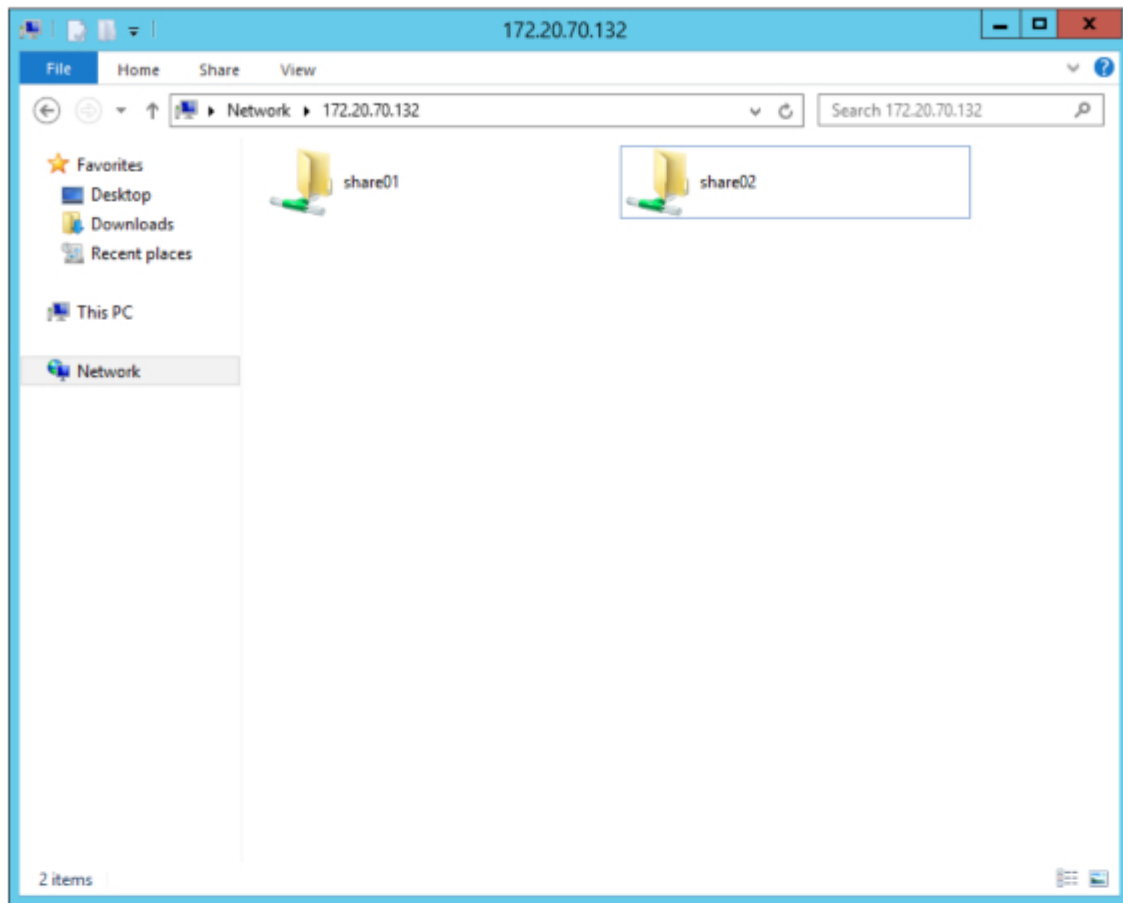
```
# setsebool -P samba_enable_home_dirs on
# chcon -t samba_share_t /samba/anonymous_share/
```

9. Проверьте доступ с машины c7-client01.

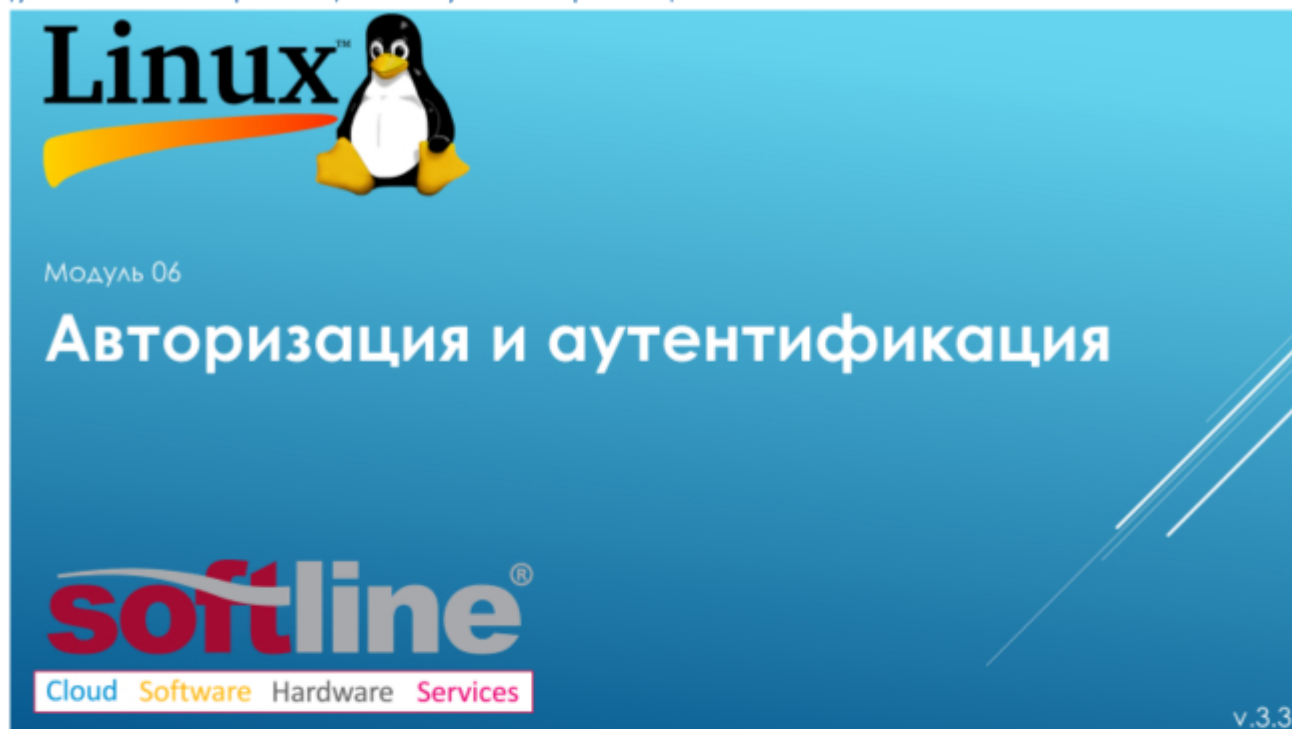


Упражнение 5 – Настройка доступа к папкам из Windows (дополнительно).

Самостоятельно настройте и проверьте доступ к SMB-серверу с хоста под управление Microsoft Windows.



Модуль 06: Авторизация и аутентификация



В модуле представлены:

- Основные концепции процедуры аутентификации.
- Система Name Service Switch.
- PAM.
- Настройка PAM.
- PAM-модули.
- Система NIS.
- Настройка NIS.
- Основные концепции протокола LDAP.
- Инструменты для работы с OpenLDAP.
- Настройка сервера OpenLDAP.
- Реализации служб каталога.

Процедура аутентификации

- Аутентификация и авторизация – разные понятия.
- Аутентификация – процедура, при которой проверяется соответствие введённых логина и пароля.
- Авторизация – проверка достаточности полномочий для выполнения определённых действий.
- Для успешной аутентификации пользователя необходимо выполнение двух условий:
- Получение информации об учётной записи пользователя.
- Аутентификация учётной записи пользователя (проверка его пароля).

<http://edu.softline.ru/>



Очень важно понимать различия между аутентификацией и авторизацией. Это можно сравнить со входом на режимный объект. Представьте, что вы попали на атомную станцию. На контрольно-пропускном пункте вы предъявляете паспорт, проверив который, охрана убеждается в том, что перед ними действительно тот, за кого вы себя выдаёте. Это была процедура аутентификации, охрана убедилась, что вы это вы. Но что же дальше? А дальше вы пытаетесь пройти на охраняемую территорию, но охранник, посмотрев в списки тех, кому проход разрешён и не найдя в них вашей фамилии, отказывает в проходе. Это и есть процедура авторизации.

Для проведения процедуры аутентификации операционная система должна иметь доступ к информации о пользователях и иметь возможность произвести сравнение введённого вами пароля с тем, что находится в базе данных учётных записей. Вспомните два файла, **/etc/passwd** и **/etc/shadow**. В первом находится информация о пользователях, во втором – пароли пользователей.

Когда, при входе в систему вы вводите имя пользователя и пароль, система транслирует введённые данные в идентификаторы пользователя и группы для дальнейшей работы.

Система Name Service Switch

- Функции, используемые при аутентификации пользователя, заложены стандартную библиотеку `libc`.
- Для указания альтернативных источников информации о пользователях необходимо вносить изменения в `libc`.
- Для унификации существуют библиотеки `/lib/libnss_service.so`:
 - `libnss_files.so`
 - `libnss_nis.so`
 - `libnss_ldap.so`
- Каждая из библиотек предназначена для осуществления запросов к различным БД.
- Управление NSS производится посредством файла `/etc/nsswitch.conf`.

<http://edu.softline.ru/>



Стандартная библиотека `libc` используется для получения информации об учётных записях пользователей, их паролях и т.д. из файлов `/etc/passwd`, `/etc/shadow`. В случае, если администратор захочет использовать любую иную БД, он вынужден вносить изменения в библиотеку `libc`. Чтобы этого избежать, функции поиска информации были вынесены в отдельные модули. И теперь администратор системы может сам указывать, какую БД необходимо использовать для аутентификации пользователей.

Таким образом, в системе Linux имеются все средства для осуществления запросов к различным базам данных, таким как: `nis`, `ldap`, текстовым файлам и т.д.

Name Service Switch – система, предназначенная для определения того, какая информация, откуда и в каком порядке должна быть получена. Основным конфигурационным файлом является `/etc/nsswitch.conf`, в котором определяются следующие параметры:

```
information: database
```

Поле **information** определяет тип информации:

passwd – информация о пользователях
shadow – информация о теневых пользователях (паролях)
groups – информация о группах

Поле `database` определяет порядок проверки баз данных. Базы данных указываются через пробел и будут проверяться в порядке следования до первого попадания, то есть, как только информация будет найдена, дальнейшая проверка прекратится. Имя сервиса определяется именем модуля `libnss`. То есть, используя имя `files`, вы даёте системе NSS указание использовать модуль `/lib/libnss_files.so`, а используя `ldap`, вы даёте указание использовать `/lib/libnss_ldap.so`. Возможные значения определяются наличием соответствующих модулей:

files – информация должна быть получена из файлов.
nis – информация должна быть получена из NIS.
ldap – информация должна быть получена из `ldap`.

Полный список возможных значений может быть получен в man-странице **nsswitch.conf**.

Для получения информации о пользователях необходимо использовать программу **getent**, которая использует систему NSS. Использование данной программы связано с тем, что администратор не будет вынужден искать информацию о том, где находятся списки пользователей, групп и т.д. Он просто использует **getent**, не заботясь о том, где на самом деле хранится информация. Для использования, **getent** необходимо передать в качестве аргумента данные о том, какую информацию необходимо вывести.

В качестве примера рассмотрим процедуру получения списка всех пользователей, известных нашей операционной системе:

```
# getent passwd
```

Для получения информации о конкретном элементе, необходимо указание его имени. Например, чтобы получить информацию о группе **wheel** необходимо дать команду:

```
# getent group wheel
```

PAM

- PAM - Pluggable Authentication Modules.
- Программы, выполняющие аутентификацию, должны иметь доступ к соответствующим базам данных пользователей.
- Для обеспечения безопасности и универсальности используется библиотека `libpam`.
- `libpam` выполняет проверку основываясь на конфигурационном файле, соответствующем вызывающей программе.
- Все модули `pam` находятся в `/lib/security`.
- Все конфигурационные файлы находятся в `/etc/pam.d`.

<http://edu.softline.ru/>



Огромное множество программ требуют выполнения авторизации и аутентификации, следовательно, должны иметь доступ к соответствующим базам данных, например, `/etc/shadow`, что приводит к появлению двух проблем:

1. Программа должна содержать в себе функции доступа к базам данных и выполнения соответствующих проверок.
2. Под угрозой находится общая безопасность системы, когда любая программа имеет доступ к паролям пользователей.

Для решения этих проблем была создана система PAM, предназначенная для унификации процедур авторизации и аутентификации без предоставления избыточных прав доступа.

Для авторизации и аутентификации, программа выполняет вызов библиотеки `libpam`, которая в свою очередь производит чтение конфигурационного файла из каталога `/etc/pam.d`, имя которого совпадает с именем вызывающего процесса, в случае, если такой файл не найден, производится вызов файла `/etc/pam.d/other`. Конфигурационный файл содержит данные о том, как должна производиться аутентификация и авторизация, с указанием соответствующих ограничений и действий, выполняемых во время проверки.

Конфигурация PAM

- Каждый конфигурационный файл PAM должен состоять из записей, содержащих 3 поля:
 - `test` – определяет тип выполняемой проверки.
 - `control` – определяет, как выполнение проверки повлияет на результат.
 - `module` – определяет используемый для проверки модуль.
- Программа, вызывая `libpam`, указывает, какую проверку необходимо выполнить (поле `test`):
 - `auth` – произвести аутентификацию пользователя.
 - `account` – произвести авторизацию пользователя.
 - `password` – контроль изменения пароля.
 - `session` – открытие, закрытие сессии, ведение журнала событий.

<http://edu.softline.ru/>



Конфигурационный файл PAM для каждого из сервисов состоит из нескольких строк. Каждая строка содержит в себе тип выполняемой проверки, требования к проверке и используемый модуль.

Существует четыре типа проверок:

- **auth** – проверка, используемая для аутентификации пользователя. Для её выполнения необходимо, чтобы вызывающая программа передала имя пользователя и его пароль.
- **account** – данная проверка не имеет отношения к аутентификации, обеспечивая применение различных ограничений
- **password** – проверка используется при изменении пароля пользователя
- **session** – проверка используется при открытии/завершении сессии для пользователя
- Каждая группа проверок будет возвращать результат, независимый от другой группы.

Результат выполнения проверки

- Для любого из типов проверок необходимо указание применения данных проверок:
 - **required** – проверка должна быть успешной для всей группы проверок. Если результат будет отрицательным остальные проверки будут выполнены.
 - **requisite** – То же самое что и **required**, но если результат будет отрицательным, то управление будет возвращено вызывающему приложению.
 - **sufficient** – проверка достаточна для признания аутентификации успешной, в случае, если до этого не провалилась проверка **required**. Неудача данной проверки не отражается на результате дальнейших проверок. Если результат положительный – прекратить дальнейшую работу.
 - **optional** – результат выполнения не важен, в любом случае необходимо продолжать работу.

<http://edu.softline.ru/>



Каждый файл системы PAM выполняется построчно. Вызывающая программа должна сообщить **libpam** о том, какую проверку необходимо выполнить. В свою очередь, **libpam** открывает необходимый конфигурационный файл и исполняет данную проверку построчно. В случае, если во втором поле описания проверки указано **required**, то в случае успеха необходимо переходить к следующей строке, если же результат проверки был отрицательным, то вызывающей программе будет возвращено значение **false** и дальнейшие проверки будут прекращены.

Рассмотрим пример конфигурационного файла:

```
auth      required    pam_env.so
auth      sufficient  pam_unix.so nullok try_first_pass
auth      requisite   pam_succeed_if.so uid >= 500 quiet
auth      required    pam_deny.so
account   required    pam_unix.so
account   sufficient  pam_succeed_if.so uid < 500 quiet
account   required    pam_permit.so
password  requisite   pam_cracklib.so try_first_pass retry=3
password  sufficient  pam_unix.so md5 shadow nullok try_first_pass use_authtok
password  required    pam_deny.so
session   optional    pam_keyinit.so revoke
session   required    pam_limits.so
session   [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session   required    pam_unix.so
```

При аутентификации, будут использованы все правила **auth** в том порядке, что они указаны в файле:

1. Обязательно выполнение модуля **pam_env.so**
2. Достаточно выполнение модуля **pam_unix.so**
3. Если результат выполнения модуля **pam_succeed_if.so** будет отрицательным, прекратить работу
4. Обязательно выполнение модуля **pam_deny.so**.

PAM-модули

- `pam_stack.so` – модуль возвращает результат, основываясь на результате выполнения другого конфигурационного файла из `/etc/pam.d`.
- `pam_deny.so` – модуль, всегда возвращающий отрицательный результат.
- `pam_permit.so` – модуль, всегда возвращающий положительный результат.
- `pam_securetty.so` – модуль, проверяющий, разрешено ли входить пользователю `root` на данную консоль.

<http://edu.softline.ru/>



Модуль `pam_stack` позволяет производить проверку, основываясь на результате выполнения проверок другого конфигурационного файла из `/etc/pam.d`. Дело в том, что создавать сервисные конфигурации `pam` трудоёмко. Поэтому во многих дистрибутивах есть специальный файл `/etc/pam.d/system-auth`, который является базисом для выполнения проверок для многих сервисов. Достаточно рассмотреть файл `/etc/pam.d/login`, который используется при входе пользователя в систему:

<code>auth</code>	<code>required</code>	<code>/lib/security/pam_securetty.so</code>
<code>auth</code>	<code>required</code>	<code>/lib/security/pam_stack.so service=system-auth</code>
<code>auth</code>	<code>required</code>	<code>/lib/security/pam_nologin.so</code>
<code>account</code>	<code>required</code>	<code>/lib/security/pam_stack.so service=system-auth</code>
<code>password</code>	<code>required</code>	<code>/lib/security/pam_stack.so service=system-auth</code>
<code>session</code>	<code>required</code>	<code>/lib/security/pam_stack.so service=system-auth</code>
<code>session</code>	<code>optional</code>	<code>/lib/security/pam_console.so</code>

Как несложно заметить, при авторизации будет вызван модуль `pam_securetty.so`, а после него модуль `pam_stack.so`, которому в качестве параметра передаётся имя сервисного `pam`-файла, который необходимо использовать.

Модуль `pam_unix` является основным модулем для авторизации и аутентификации пользователей. Если он используется в проверках типа `auth`, то осуществляет аутентификацию пользователей. В секции `account` данный модуль осуществляет проверку времени истечения пароля, не заблокирован ли пользователь. В проверках `password` модуль `pam_unix` осуществляет установку паролей. Рассмотрим некоторые параметры, передаваемые модулю:

nullok – разрешить пустые пароли
md5 – необходимо использовать хэширование с помощью `md5`
shadow – использовать теневые пароли, файл `/etc/shadow`

Модули `pam_deny` и `pam_permit` всегда возвращают ложь и истину, соответственно.

PAM-модули для секции **auth**

- **pam_nologin.so** – осуществляет проверку наличия файла **/etc/nologin**:
 - Может использоваться во время профилактических работ, когда необходимо временно изолировать систему от пользователей.
- **pam_listfile.so** – проверка осуществляется на основе текстового файла и определяемых администратором критериев:
 - Текстовый файл со списком каких-либо данных. Модуль проверяет наличие одного из атрибутов учетной записи в файле и применяет заданное действие.

<http://edu.softline.ru/>



При перезагрузке системы создаётся специальный пустой файл **/etc/nologin**. Это делается для того, чтобы в самый уязвимый момент в систему не смог войти ни один пользователь, не являющийся **root**.

Это достигается с помощью модуля **pam_nologin**, основной задачей которого является проверка наличия файла **/etc/nologin** и блокировка входа в систему пользователя с **uid** отличным от нуля.

Модуль **pam_listfile** позволяет администратору задать произвольный список, согласно выбранному критерию. Рассмотрим простой пример – файл **/etc/pam.d/vsftpd**:

```
auth required pam_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers onerr=succeed
```

Данная проверка означает, что при её прохождении будет осуществлён поиск имени пользователя в файле **/etc/vsftpd/ftpusers**. Если имя пользователя будет найдено в списке, то проверка должна завершиться отрицательно. Если же произошла ошибка при проверке, проверка должна завершиться успешно. Возможные значения параметра **item**:

- **tty.**
- **user.**
- **group.**
- **shell.**

PAM-модули для секции **account**

- `pam_access.so` – работа модуля основана на текстовом файле, в котором указано, каким пользователям и группам вход разрешен, а каким нет:
 - Конфигурация по умолчанию – `/etc/security/access.conf`.
- `pam_time.so` – проверка времени доступа:
 - Конфигурация по умолчанию – `/etc/security/time.conf`.
- `pam_succeed_if.so` – проверка будет успешна, если проверка заданного администратором критерия будет успешна.

<http://edu.softline.ru/>



Модуль `pam_access` проверяет конфигурационный файл `/etc/security/access.conf`, где определено, для каких пользователей и групп проверка будет завершена успешно, а для каких необходимо выдать отрицательный результат. В следующем примере успешное завершение будет только для учетных записей пользователей `root` и `admin`, всем остальным вход будет запрещён:

```
+ : root admin: ALL
- : ALL : ALL
```

Первое поле определяет разрешение, символом "+" отмечены те списки пользователей, для которых проверка должна завершиться успешно.

Второе поле включает в себя имя пользователя/группы. При этом, сначала указанное имя будет искаться в списке системных пользователей и если не будет найдено там, то в списке системных групп.

Третье поле определяет ip-адреса узлов, с которых разрешён вход.

Модуль `pam_time` позволяет определять время доступа, определённое в конфигурационном файле `/etc/security/time.conf`.

Модуль `pam_succeed_if`, строго говоря, не является модулем, применимым только в проверках типа `account`, но, тем не менее, здесь он используется наиболее часто и позволяет осуществлять проверку ряда условий. На основе этих проверок и будет выдан итоговый результат.

```
account      sufficient      pam_succeed_if.so uid < 500 quiet
```

В примере показано, что проверка будет завершена успешно, если идентификатор пользователя лежит в пределах от 0 до 499(включительно).

PAM-модули для секции **session**

- `pam_limits.so` – ограничение ресурсов:
 - Конфигурация по умолчанию – `/etc/security/limits.conf`.
- `pam_mkhome.so` – модуль предназначен для создания домашнего каталога пользователя при входе. Используется при сетевой аутентификации и авторизации.
- `pam_umask.so` – модуль предназначен для установки параметра `umask` при входе пользователя в систему.

<http://edu.softline.ru/>



С помощью модуля `pam_limits` и конфигурационного файла `/etc/security/limits.conf` возможна установка различных ограничений на использование системных ресурсов. В конфигурационном файле находятся подробные комментарии с перечислением возможных ограничений. Следующий пример ограничивает количество входов в систему для пользователя `admin` и ограничивает предоставляемое время процессора для пользователя `user`:

```
admin      -      maxlogins 2          # два входа в систему
user       hard cpu      5          # пять минут процессорного времени
```

Когда вы используете сетевую авторизацию, то, скорее всего, не на всех узлах будут созданы домашние каталоги, а если нет каталога, то назвать полноценной возможную работу пользователя на каждом из узлов – нельзя. Вы можете создать каталоги вручную для всех сотен пользователей на каждом из узлов. А можете использовать модуль `pam_mkhome.so`, который создаст домашний каталог при входе пользователя в систему.

Система NIS

- Простейший сервер каталогов:
 - Прост в настройке.
 - Прост в использовании.
 - Отсутствует шифрование передаваемых данных.
- Для объединения узлов используется понятие – домен.
- Для обеспечения работоспособности требуются:
 - **portmap** – для динамического выделения сетевых портов.
 - **ypserv** - NIS-сервер.
 - **yplib** - NIS-клиент.

<http://edu.softline.ru/>



Сервер NIS – это хранилище данных. В нём могут храниться учетные записи пользователей, групп. Основное преимущество NIS в его простом использовании и обслуживании. Но есть и существенный недостаток, делающий применимость NIS-серверов очень ограниченной. Дело в том, что все данные между сервером и многочисленными клиентами производится в открытом тексте без использования шифрования.

Для включения необходимо установить пакеты **portmap** и **ypserv**. Определите имя NIS-домена с помощью команды **nisdomainname**:

```
# nisdomainname nisname
```

Единственным средством обеспечения хоть какой-то безопасности является ограничение подключений, основанное на ip-адресе клиента. В файле **/var/yp/securenets** необходимо указать все сети и адреса, которым разрешено подключение:

```
255.255.255.0      192.168.1.0
```

Остаётся запустить сервисы **ypserv** и **portmap**.

Настройка NIS

- NIS заполняется на основе системных данных.
- Конфигурирование NIS сосредоточено на каталоге `/var/yp` и файле `Makefile`, в частности:
 - **NORUSH** – необходимость распространения данных на вторичные серверы.
 - **MINUID** – UID, начиная от которого необходимо использовать учетные записи пользователей.
 - **MINGID** – GID, начиная от которого необходимо использовать учетные записи групп.
 - **all** – список данных, которые необходимо поместить в NIS.
- Подключения клиентов при помощи утилиты: **authconfig**.

<http://edu.softline.ru/>



Конфигурирование NIS заключается в наполнении базы данных необходимой информацией. Редактируя файл `/var/yp/Makefile` вы можете указать, какую именно информацию вы хотите поместить в NIS. Следующий параметр говорит о том, что в NIS необходимо поместить информацию о пользователях:

```
all: passwd group
```

Чтобы сформировать БД вы можете дать любую из указанных ниже команд:

```
# /usr/lib/yp/ypinit -m
# cd /var/yp && make
```

Чтобы обеспечить возможность изменения паролей на удалённых узлах запустите сервис **yppasswd**.

Программа **authconfig** позволяет произвести конфигурирование системы для подключения к NIS или LDAP:

```
# authconfig --enablenis --nisdomain domain --nisserver 192.168.1.X --updateall
```

При этом будут изменены файлы:

- `/etc/sysconfig/network`
- `/etc/yp.conf`
- `/etc/nsswitch.conf`
- `/etc/pam.d/system-auth`

Протокол LDAP

- LDAP (Lightweight Directory Access Protocol) – это набор открытых протоколов для доступа к централизованному хранилищу информации по сети (Базируется на стандарте X.500).
- CentOS 7 (RHEL7) содержит OpenLDAP 2.4 – реализацию протоколов LDAPv2 и LDAPv3 с открытым исходным кодом.
- Для обеспечения безопасности соединения: Secure Socket Layer (SSL) или Transport Layer Security (TLS).
- Терминология LDAP:
 - Запись (entry) – одиночный объект в каталоге LDAP.
 - Атрибут (attribute) – информация связанная с записью.
- LDIF – LDAP Data Interchange Format – это текстовое представление LDAP записи.

<http://edu.softline.ru/>



LDAP (Lightweight Directory Access Protocol) – это набор открытых протоколов используемых для доступа к централизованному хранилищу информации по сети. Это базируется на стандарте **X.500** для предоставления общего доступа к каталогу, но менее комплексный и требовательный к ресурсам. В связи с этим LDAP иногда называют “X.500 Lite”.

Также, как и X.500, LDAP организует информацию в иерархию при помощи каталогов. Эти каталоги могут хранить различную информацию, такую как имена, адреса или номера телефонов и могут быть использованы в структуре похожей на Network Information Service (NIS), позволяющей кому-либо получать доступ к его учетной записи с любой машины в сети с включенным LDAP.

LDAP зачастую используется для централизованного управления пользователями и группами, аутентификации пользователей или настройки систем. Также он может служить в качестве виртуального телефонного каталога, позволяя пользователям легко получить доступ к контактной информации других пользователей. Дополнительно он может быть связан с другими LDAP серверами в мире, что в свою очередь создаст глобальный репозиторий информации. Однако, наиболее часть LDAP используется в рамках отдельной организации: университета, департамента или частной компании.

Мы будем разбирать установку и настройку OpenLDAP 2.4 – реализацию протоколов LDAPv2 и LDAPv3 с открытым исходным кодом.



ПРИМЕЧАНИЕ:

Active Directory Domain Service с Windows 2000 Server использует LDAPv3.

Введение в LDAP.

Используя клиент-серверную архитектуру, LDAP предоставляет надежное средство для создания централизованного каталога информации, доступного по сети. Когда клиент пытается изменить информацию в каталоге, сервер проверяет права пользователя на внесение данного изменения, и затем добавляет или обновляет запрошенные записи. Чтобы убедиться в безопасности соединения, могут быть использованы криптографические протоколы Secure Socket Layer (SSL) или Transport Layer Security (TLS), они позволят предотвратить перехват информации при передаче.

**ПРИМЕЧАНИЕ:**

Комплект OpenLDAP в CentOS 7 (RHEL 7) более не использует OpenSSL. Вместо этого используется решение Mozilla Network Security Services (NSS). OpenLDAP продолжает работать с существующими сертификатами, ключами и прочими настройками TLS.

LDAP сервер поддерживает несколько системных баз данных, которые позволяют администратору гибко выбирать наиболее подходящее решение для хранения информации. Из-за хорошо определенного Application Programming Interface (API) множество приложений может взаимодействовать с LDAP сервером, увеличивая и качество, и количество.

Терминология LDAP.

Следующий список терминов свойственных LDAP потребуется для дальнейшего рассмотрения вопроса.

Запись (entry) – одиночный объект в каталоге LDAP. Каждая запись идентифицируется по уникальному Distinguished Name (DN).

Атрибут (attribute) – информация, непосредственно связанная с записью. Например, если организация представлена в виде LDAP записи, атрибуты, ассоциированные с этой организацией, должны включать адрес, номер телефона и так далее.

Атрибут может иметь одиночное значение или несортированный список значений, разделенных пробелами. Часть атрибутов является опциональной, прочие являются обязательными. Обязательные атрибуты указываются при помощи определения **objectClass**, и могут быть найдены в файлах схемы, расположенных в каталоге `/etc/openldap/slapd.d/cn=config/cn=schema/`.

Утверждение атрибута и соответствующего значения также относится к Relative Distinguished Name (RDN). В отличие от DN, имена которых должны быть уникальны во всем каталоге, RDN уникальны только в рамках записи.

LDIF – LDAP Data Interchange Format – это текстовое представление LDAP записи. Имеет следующий вид:

```
[id] dn: distinguished_name
attribute_type: attribute_value...
attribute_type: attribute_value...
...
```

Опциональный id – это номер определяемый приложением, которое используется для редактирования записи. Каждая запись может содержать столько пар атрибутов (**attribute_type**) и их значений (**attribute_value**), сколько необходимо, до тех пор, пока они определены в соответствующем файле схемы. Пустая строка определяет конец записи.

Возможности OpenLDAP.

Комплект OpenLDAP предоставляет несколько важных возможностей:

Поддержка LDAPv3 – Множество изменений в протоколе LDAP со второй версии разработаны для обеспечения повышенного уровня безопасности LDAP. Среди прочих улучшений, включена поддержка следующих протоколов: Simple Authentication and Security Layer (SASL), Transport Layer Security (TLS) и Secure Socket Layer (SSL).

LDAP Over IPC – Использование коммуникаций внутри процесса (IPC) расширяют безопасность, за счет отказа от необходимости взаимодействия через сеть.

Поддержка IPv6 – OpenLDAP совместим с IPv6.

Поддержка LDIFv1 – OpenLDAP полностью совместим с LDIF версии 1.

Обновленный C API – Текущий C API расширяет возможности программистов для соединения и использования серверов каталогов LDAP.

Расширенный индивидуальный LDAP сервер – это включает в себя обновленную систему контроля доступа, пул потоков, лучшие инструменты и многое другое.

Настройка OpenLDAP сервера.

1. Установить комплект OpenLDAP.
2. Настроить конфигурацию.
3. Запустить службу **slapd**.
4. Воспользоваться утилитой **ldapadd** для добавления записей в каталог LDAP.
5. Воспользоваться утилитой **ldapsearch** для проверки доступности службы **slapd** и доступности информации.

Установка комплекта OpenLDAP.

- Комплект библиотек и инструментов OpenLDAP состоит из следующих пакетов:
 - `openldap`
 - `openldap-clients`
 - `openldap-servers`
 - `openldap-servers-sql`
 - `compat-openldap`
- Дополнительно, следующие пакеты часто используются вместе с LDAP сервером:
 - `nss-pam-ldapd`
 - `mod_ldap`

<http://edu.softline.ru/>



Комплект библиотек и инструментов OpenLDAP состоит из следующих пакетов:

openldap	– пакет, содержащий библиотеки, необходимые для запуска сервера OpenLDAP и клиентских приложений.
openldap-clients	– пакет содержащий утилиты командной строки и утилиты для настройки и запуска LDAP сервера.
openldap-servers	– пакет, содержащий и службы и утилиты для настройки и запуска LDAP сервера. Он включает изолированный демон LDAP (Standalone LDAP Daemon), slapd .
openldap-servers-sql	– пакет, содержащий модуль поддержки SQL.
compat-openldap	– пакет содержащий библиотеки совместимости OpenLDAP.

Дополнительно, следующие пакеты часто используются вместе с LDAP сервером:

nss-pam-ldapd	– пакет, содержащий nsldap , локальную службу имен LDAP, которая позволяет пользователю выполнять локальные LDAP запросы.
mod_ldap	– пакет, содержащий модули mod_authnz_ldap и mod_ldap . Модуль mod_authnz_ldap – это модуль LDAP авторизации для Apache HTTP Server. Этот модуль может аутентифицировать права пользователя в каталоге LDAP и может контролировать доступ основываясь на имени пользователя, DN, членстве в группе, произвольном атрибуте или комплексной строке фильтрации. Модуль mod_ldap содержит те же пакеты, предоставляя настраиваемый общий кэш памяти, для сокращения повторного доступа к каталогу из-за множества HTTP запросов, а также поддерживает SSL/TLS.

Для установки пакета используйте команду **yum**. Например, для выполнения базовой установки LDAP сервера, используйте следующую команду:

```
yum install openldap openldap-clients openldap-servers
```

Инструменты сервера OpenLDAP

- Содержатся в пакете **openldap-servers**.
 - **slapacl** – Проверка доступ к списку атрибутов.
 - **slapadd** – Добавление записи из LDIF-файла в каталог LDAP.
 - **slapauth** – Проверка списка идентификаторов (ID).
 - **slapcat** – Экспорт записей из каталога LDAP в LDIF-файлы.
 - **slapdn** – Проверка списка Distinguished Names (DNs).
 - **slapindex** – Повторная индексация каталога **slpad**.
 - **slappasswd** – Создание зашифрованного пользовательского пароля.
 - **slapschema** – Проверка соответствия базы данных со схемой.
 - **slaptest** – Проверка конфигурации сервера LDAP.

<http://edu.softline.ru/>



Для выполнения административных задач, пакет **openldap-servers** устанавливает следующий набор инструментов вместе со службой **slapd**:

- | | |
|-------------------|---|
| slapacl | – Позволяет проверять доступ к списку атрибутов. |
| slapadd | – Позволяет добавлять записи из LDIF-файла в каталог LDAP. |
| slapauth | – Позволяет проверить список идентификаторов (ID) на права аутентификации и авторизации. |
| slapcat | – Позволяет вынуть записи из каталога LDAP в формате по умолчанию и сохранить их в LDIF-файле. |
| slapdn | – Позволяет проверить список Distinguished Names (DNs) на основе имеющегося синтаксиса схемы. |
| slapindex | – Позволяет повторно индексировать каталог slpad в зависимости от текущего контента. Перед запуском утилиты необходимо внести изменения в опции индексирования в конфигурационном файле. |
| slappasswd | – Позволяет создать зашифрованный пользовательский пароль для дальнейшего использования с утилитой ldapmodify или в конфигурационном файле slapd . |
| slapschema | – Позволяет проверить соответствие базы данных с соответствующей схемой. |
| slaptest | – Позволяет проверить конфигурацию сервера LDAP. |



ПРИМЕЧАНИЕ:

Несмотря на то, что только **root** может запускать **slapadd**, служба **slapd** запускается от имени пользователя **ldap**. Поэтому сервер каталога не может модифицировать файлы, созданные при помощи **slapadd**. Для исправления этой ошибки используйте следующую команду:

```
# chown -R ldap:ldap /var/lib/ldap
```



ВНИМАНИЕ:

Для сохранения целостности данных остановите службу **slapd** перед использованием **slapadd**, **slapcat** или **slapindex**:

```
# systemctl stop slapd.service
```

Клиентские инструменты OpenLDAP

- Содержатся в пакете **openldap-clients**.
 - **ldapadd** – Добавление записи в каталог LDAP.
 - **ldapcompare** – Сравнение подаваемых атрибутов с записями каталога LDAP.
 - **ldapdelete** – Удаление записи из каталога LDAP.
 - **ldapexop** – Выполнение расширенных операций LDAP.
 - **ldapmodify** – Изменение записи в каталоге LDAP.
 - **ldapmodrdn** – Изменение значения RDN записи каталога LDAP.
 - **ldappasswd** – Установка или изменение пароля пользователя LDAP.
 - **ldapsearch** – Поиск записей в каталоге LDAP.
 - **ldapurl** – Сбор или разбор LDAP URL.
 - **ldapwhoami** – Выполнение операции **whoami** на сервере LDAP.

<http://edu.softline.ru/>



Пакет **openldap-clients** устанавливает следующие инструменты, которые могут быть использованы для добавления, изменения и удаления записей в каталоге LDAP:

- ldapadd** – Позволяет добавить записи в каталог LDAP, как из файла, так и при помощи стандартного ввода. Это символическая ссылка на **ldapmodify -a**.
- ldapcompare** – Позволяет сравнивать подаваемые атрибуты с записями каталога LDAP.
- ldapdelete** – Позволяет удалять записи из каталога LDAP.
- ldapexop** – Позволяет выполнять расширенные операции LDAP.
- ldapmodify** – Позволяет изменять записи в каталоге LDAP, как при помощи файла, так и при помощи стандартного ввода.
- ldapmodrdn** – Позволяет изменять значение RDN записи каталога LDAP.
- ldappasswd** – Позволяет устанавливать или изменять пароль пользователя LDAP.
- ldapsearch** – Позволяет искать записи в каталоге LDAP.
- ldapurl** – Позволяет собрать или разобрать LDAP URL.
- ldapwhoami** – Позволяет выполнить операцию **whoami** на сервере LDAP.

За исключением **ldapsearch**, эти утилиты гораздо проще использовать вместе с соответствующим файлом, содержащим необходимые изменения, нежели вводить команду для каждой записи, которую необходимо изменить в каталоге LDAP. Формат файлов для каждой утилиты описан в страницах руководства **man**.

Настройка OpenLDAP сервера

- Конфигурация OpenLDAP хранится в каталоге `/etc/openldap/`.
 - `/etc/openldap/ldap.conf` – Конфигурационный файл для клиентских приложений.
 - `/etc/openldap/slapd.d/` – Каталог с конфигурацией `slapd`.
- Глобальные опции настройки сервера LDAP:
 - `/etc/openldap/slapd.d/cn=config.ldif`
- По умолчанию сервер OpenLDAP использует Berkeley DB (BDB).
- Конфигурация BDB:
 - `/etc/openldap/slapd.d/cn=config/olcDatabase={1}bdb.ldif`
- Начиная с OpenLDAP 2.3, каталог `/etc/openldap/slapd.d/` также содержит определения LDAP (схему).

<http://edu.softline.ru/>



По умолчанию конфигурация OpenLDAP хранится в каталоге `/etc/openldap/`. Ниже представлены наиболее важный файл и каталог:

`/etc/openldap/ldap.conf` – Конфигурационный файл для клиентских приложений, которые используют библиотеки OpenLDAP. В том числе `ldapadd`, `ldapsearch`, Evolution и так далее.

`/etc/openldap/slapd.d/` – Каталог, содержащий конфигурацию `slapd`.

OpenLDAP более не читает конфигурационный файл `/etc/openldap/slapd.conf`. Вместо него используется база данных размещенная в каталоге `/etc/openldap/slapd.d/`. Если у вас есть существующий файл `slapd.conf` от предыдущей установки, вы можете его конвертировать в новый формат:

```
# slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d/
```

Конфигурация `slapd` содержится в записях LDIF организованных в иерархической структуре каталогов. И рекомендуется использовать серверные инструменты для внесения изменений в записи.



ПРИМЕЧАНИЕ:

Ошибка в LDIF файле может не дать службе `slapd` запуститься. Поэтому настоятельно рекомендуется избегать непосредственного редактирования LDIF файлов в каталоге `/etc/openldap/slapd.d/`.

Изменение глобальной конфигурации.

Глобальные опции настройки сервера LDAP сохранены в файле `/etc/openldap/slapd.d/cn=config.ldif`. Следующие директивы как правило используются:

`olcAllows`

Директива `olcAllows` позволяет указать какие возможности необходимо включить. Записывается в следующей форме:

```
olcAllows: feature...
```

Она принимает список возможностей, разделенный пробелами. Опция по умолчанию – `bind_v2`.

Доступные опции **olcAllows**:

- bind_v2** – Включает прием связанных запросов LDAP версии 2.
- bind_anon_cred** – Включает анонимную привязку, когда Distinguished Name (DN) пустой.
- bind_anon_dn** – Включает анонимную привязку, когда Distinguished Name (DN) не пустой.
- update_anon** – Включает обработку анонимных операций обновления.
- proxy_authz_anon** – Включает обработку анонимного управления прокси авторизацией.

Пример использования директивы **olcAllows**:

```
olcAllows: bind_v2 update_anon
```

olcConnMaxPending

Директива **olcConnMaxPending** позволяет указать максимальное число ожидающих запросов для анонимных сессий. Записывается в следующей форме:

```
olcConnMaxPending: число
```

Опция по умолчанию: 100.

Пример использования директивы **olcConnMaxPending**:

```
olcConnMaxPending: 100
```

olcConnMaxPendingAuth

Директива **olcConnMaxPendingAuth** позволяет указать максимальное число ожидающих запросов для аутентифицированной сессии. Записывается в следующей форме:

```
olcConnMaxPendingAuth: число
```

Опция по умолчанию 1000.

Пример использования директивы **olcConnMaxPendingAuth**:

```
olcConnMaxPendingAuth: 1000
```

olcDisallow

Директива **olcDisallow** позволяет указать какие возможности необходимо отключить. Записывает в следующей форме:

```
olcDisallow: feature...
```

Она принимает список возможностей, разделенный пробелами. Никакие возможности не отключены по умолчанию.

Доступные опции **olcDisallows**:

- bind_anon** Отключает прием анонимных запросов привязки.
- bind_simple** Отключает механизм простой аутентификации привязки.
- tls_2_anon** Отключает применение анонимных сессий, когда получает команду **START TLS**.
- tls_authc** Запрещает команду **START TLS** при аутентификации.

Пример использования директивы **olcDisallows**:

```
olcDisallow: bind_anon
```

olcIdleTimeout

Директива **olcIdleTimeout** позволяет вам указать, как много секунд ждать перед закрытием простаивающего подключения. Она имеет следующий вид:

olcIdleTimeout: *number*

Опция отключена по умолчанию, это значит, что она установлена в 0.

Пример использования директивы **olcIdleTimeout**:

olcIdleTimeout: 180

olcLogFile

Директива **olcLogFile** позволяет указать файл журнала, для записи сообщений. Она имеет следующий вид:

olcLogFile: *file_name*

Сообщения журнала по умолчанию пишутся в стандартный журнал ошибок.

Пример использования директивы **olcLogFile**

olcLogFile: /var/log/slapd.log

olcReferral

Опция **olcReferral** позволяет вам указать URL сервера для обработки запросов, в том случае когда сервер не может обработать его самостоятельно. Она имеет следующий вид:

olcReferral: *URL*

Эта опция отключен по умолчанию.

Пример использования директивы **olcReferral**:

olcReferral: ldap://root.openldap.org

olcWriteTimeout

Опция **olcWriteTimeout** позволяет указать как много секунд ждать перед закрытием соединения с невыполненным запросом записи. Она имеет следующий вид:

olcWriteTimeout: *number*

Опция отключена по умолчанию, это значит, что она установлена в 0.

Пример использования директивы **olcWriteTimeout**:

olcWriteTimeout: 180

Изменение конфигурации базы данных.

По умолчанию сервер OpenLDAP использует Berkeley DB (BDB) в качестве фоновой базы данных. Конфигурация для этой базы данных сохранена в файле **/etc/openldap/slapd.d/cn=config/olcDatabase={1}dbd.ldif**. Следующие директивы используются для управления конфигурацией базы данных:

olcReadOnly

Директива **olcReadOnly** позволяет использовать базу данных в режиме только для чтения. Она имеет следующий вид:

olcReadOnly: boolean

Она принимает значение **TRUE** (режим только для чтения включен) или **FALSE** (разрешает изменения в базе данных). По умолчанию выставлено значение **FALSE**.

Пример использования директивы **olcReadOnly**:

olcReadOnly: TRUE

olcRootDN

Директива **olcRootDN** позволяет вам указать пользователя, который будет не ограничен контролем доступа или параметрами лимита административного набора на каталоге LDAP. Она имеет следующий вид:

olcRootDN: distinguished_name

Она принимает Distinguished Name (DN). По умолчанию выставлено: **cn=Manager,dn=my-domain,dc=com**.

Пример использования директивы **olcRootDN**:

olcRootDN: cn=root,dn=example,dn=com

olcRootPW

Директива **olcRootPW** позволяет установить пароль для пользователя указанного при помощи директивы **olcRootDN**. Она имеет следующий вид.

olcRootPW: password

Она принимает как плоский текст, так и хэш. Для генерации хэша используйте следующую строку:

```
$ slappaswd
New password:
Re-enter new password:
{SSHA}WczWsyPEnMchFflGRTweq2q7XJcvmSxD
```

Пример использования директивы **olcRootPW**

olcRootPW: {SSHA}WczWsyPEnMchFflGRTweq2q7XJcvmSxD

olcSuffix

Директива **olcSuffix** позволяет указать домен для которого предоставлять информацию. Она имеет следующий вид.

olcSuffix: domain_name

Она принимает полное квалифицированное доменное имя (Fully Qualified Domain Name, FQDN) Опция по умолчанию: **dc=my-domain,dc=com**.

Пример использования директивы **olcSuffix**

olcSuffix: dc=example,dc=com

Расширение схемы.

Начиная с OpenLDAP 2.3, каталог **/etc/openldap/slapd.d/** также содержит определения LDAP, которые ранее располагались в **/etc/openldap/schema/**. Это делает возможным расширение схемы, которую использует OpenLDAP, для поддержки дополнительных типов атрибутов и классов объектов, при помощи примеров на базе файлов схемы по умолчанию.



ПРИМЕЧАНИЕ:

За дополнительной информацией обратитесь к:
<http://www.openldap.org/doc/admin24/schema.html>

Управление службой сервера OpenLDAP

- Запуск службы.

```
# systemctl start slapd.service
# systemctl enable slapd.service
```
- Остановка службы.

```
# systemctl stop slapd.service
# systemctl disable slapd.service
```
- Перезапуск службы.

```
# systemctl restart slapd.service
```
- Проверка состояния службы.

```
# systemctl status slapd.service
# systemctl is-active slapd.service
# systemctl is-enabled slapd.service
```

<http://edu.softline.ru/>



Запуск службы.

Для запуска службы **slapd** в текущей сессии, введите следующую команду с правами root:

```
# systemctl start slapd.service
```

Для настройки автоматического запуска в процессе загрузки, используйте следующую команду с правами root:

```
# systemctl enable slapd.service
```

Остановка службы.

Для остановки службы **slapd** в текущей сессии, введите следующую команду с правами root:

```
# systemctl stop slapd.service
```

Для отключения автоматического запуска в процессе загрузки, используйте следующую команду с правами root:

```
# systemctl disable slapd.service
```

Перезапуск службы.

Для немедленного перезапуска службы **slapd**, введите следующую команду с правами root:

```
# systemctl restart slapd.service
```

Служба будет остановлена и немедленно запущена, используйте эту команду для перезагрузки конфигурационного файла.

Проверка состояния службы.

Для проверки запущенного состояния службы **slapd**, используйте следующую команду:

```
systemctl is-active slapd.service
active
```

Аутентификации через OpenLDAP

- Необходимые пакеты:
 - `openldap`
 - `openldap-clients`
 - `nss-pam-ldapd`
- Утилита:
 - `authconfig`

<http://edu.softline.ru/>

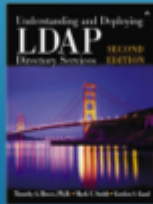
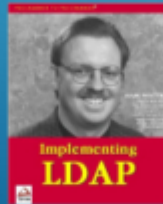
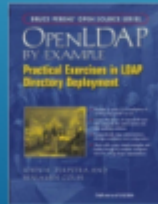


Для настройки системы на аутентификацию с использованием OpenLDAP, необходимо убедиться, что пакеты установлены и на LDAP сервер и на клиентскую машину. Для установки необходимых пакетов на клиентскую машину используйте:

```
# yum install openldap openldap-clients nss-pam-ldapd
```


Дополнительные ресурсы

- OpenLDAP Software Administrator's Guide
 - <http://www.openldap.org/doc/admin24/>
 - `/usr/share/doc/openldap-servers-version/guide.html`
- Описание установленных файлов схемы
 - `/usr/share/doc/openldap-servers-version/README.schema`
- Страницы руководства `man`.
- Веб-сайты.
- Литература.



<http://edu.softline.ru/>

Следующие ресурсы содержат дополнительную информацию про Lightweight Directory Access Protocol. Перед настройкой LDAP в производственной среде, настоятельно рекомендуется познакомиться с ними и в частности с **OpenLDAP Software Administrator's Guide**.

Установленная документация.

Следующая документация устанавливается вместе с пакетом `openldap-servers`:

`/usr/share/doc/openldap-servers-version/guide.html`

Копия **OpenLDAP Software Administrator's Guide**.

`/usr/share/doc/openldap-servers-version/README.schema`

Файл **README** содержащий описание установленных файлов схемы.

Дополнительно, вместе с пакетами `openldap`, `openldap-servers` и `openldap-clients` устанавливается набор страниц руководства `man`:

Клиентские приложения:

`man ldapadd` – Описывает как добавлять записи в каталог LDAP.

`man ldapdelete` – Описывает как удалять записи из каталога LDAP.

`man ldapmodify` – Описывает как изменять записи в каталоге LDAP.

`man ldapsearch` – Описывает как искать записи в каталоге LDAP.

`man ldappasswd` – Описывает как изменять пароль пользователю LDAP.

`man ldapcompare` – Описывает как использовать утилиту `ldapcompare`.

`man ldapwhoami` – Описывает как использовать утилиту `ldapwhoami`.

`man ldapmodrdn` – Описывает как изменять RDN записей.

Серверные приложения:

man slapd – Описывает опции командной строки сервера LDAP.

Административные приложения:

man slapadd – Описывает опции командной строки, используемые для добавления записей в базу данных **slapd**.

man slapcat – Описывает опции командной строки, используемые для создания LDIF-файла из базы данных **slapd**.

man slapindex – Описывает опции командной строки, используемые для восстановления индексов основываясь на содержимом базы данных **slapd**.

man slappasswd – Описывает опции командной строки, используемые для генерации пользовательских паролей для каталогов LDAP.

Конфигурационные файлы:

man ldap.conf – Описывает формат и опции доступные в конфигурационном файле клиента.

man slapd-config – Описывает формат и доступные опции в конфигурации каталога LDAP.

Полезные веб-сайты:

<http://www.openldap.org/doc/admin24/> - Текущая версия **OpenLDAP Software Administrator's Guide**.

<http://www.kingsmountain.com/ldapRoadmap.shtml> - **Jeff Hodges' LDAP Roadmap & FAQ** содержит ссылки на несколько полезных ресурсов и новости, связанные с протоколом LDAP.

<http://www.ldapman.org/articles/> - Коллекция статей которая содержит хорошее введение в LDAP, включая методы дизайна дерева каталогов и настройки структуры каталогов.

<http://www.padl.com/> - Веб-сайт разработчиков нескольких полезных инструментов LDAP.

Дополнительная литература:

OpenLDAP by Example by John Terpstra and Benjamin Coles; Prentice Hall. – Коллекция практических упражнения по развертыванию OpenLDAP.

Implementing LDAP by Mark Wilcox; Wrox Press, Inc. – Книга, рассказывающая про LDAP, как с позиции системного администратора, так и с позиции разработчика программного обеспечения.

Understanding and Deploying LDAP Directory Services by Tim Howes et al.; Macmillan Technical Publishing. – Книга, описывающая принципы дизайна LDAP для развертывания в производственном окружении.

Реализации служб каталога

- **389 Directory Server (389-DS)**. LDAP сервер, разрабатываемый RedHat в рамках проекта Fedora.
 - <http://www.port389.org/>
- **FreeIPA** - идентификация, политики и аудит (IPA), спонсируется RedHat.
 - <http://www.freeipa.org>

<http://edu.softline.ru/>



389 Directory Server (389-DS) – это LDAP-сервер корпоративного уровня с открытым исходным кодом для Linux, он разрабатывается сообществом Red Hat. Он защищен для использования в производственной среде, полностью функционален поддерживает репликацию в режиме мульти-мастер и достаточно широко применяется для крупных LDAP развертываний в мире. 389 Directory Server доступен для бесплатной загрузки и может быть настроен для пилотного проекта меньше чем за час. 389-DS может обрабатывать тысячи запросов от множества пользователей одновременно и эффективно.

FreeIPA – Проект с открытым исходным кодом, спонсируемый RedHat – для предоставления идентификации, политик и аудита (IPA) в первую очередь предназначен для сетей с Linux и Unix машинами.

FreeIPA Состоит из:

- **389-DS** – Реализация LDAP
- **MIT** – Аутентификация Kerberos 5 и Single Sign-On (SSO).
- **Apache HTTP Server** и **Python** – Веб-интерфейс.
- **DogTag** – Интегрированный центр выдачи сертификатов (CA).
- **BIND** – Интегрированный DNS.
- **SAMBA** – Интеграция с Microsoft Active Directory.

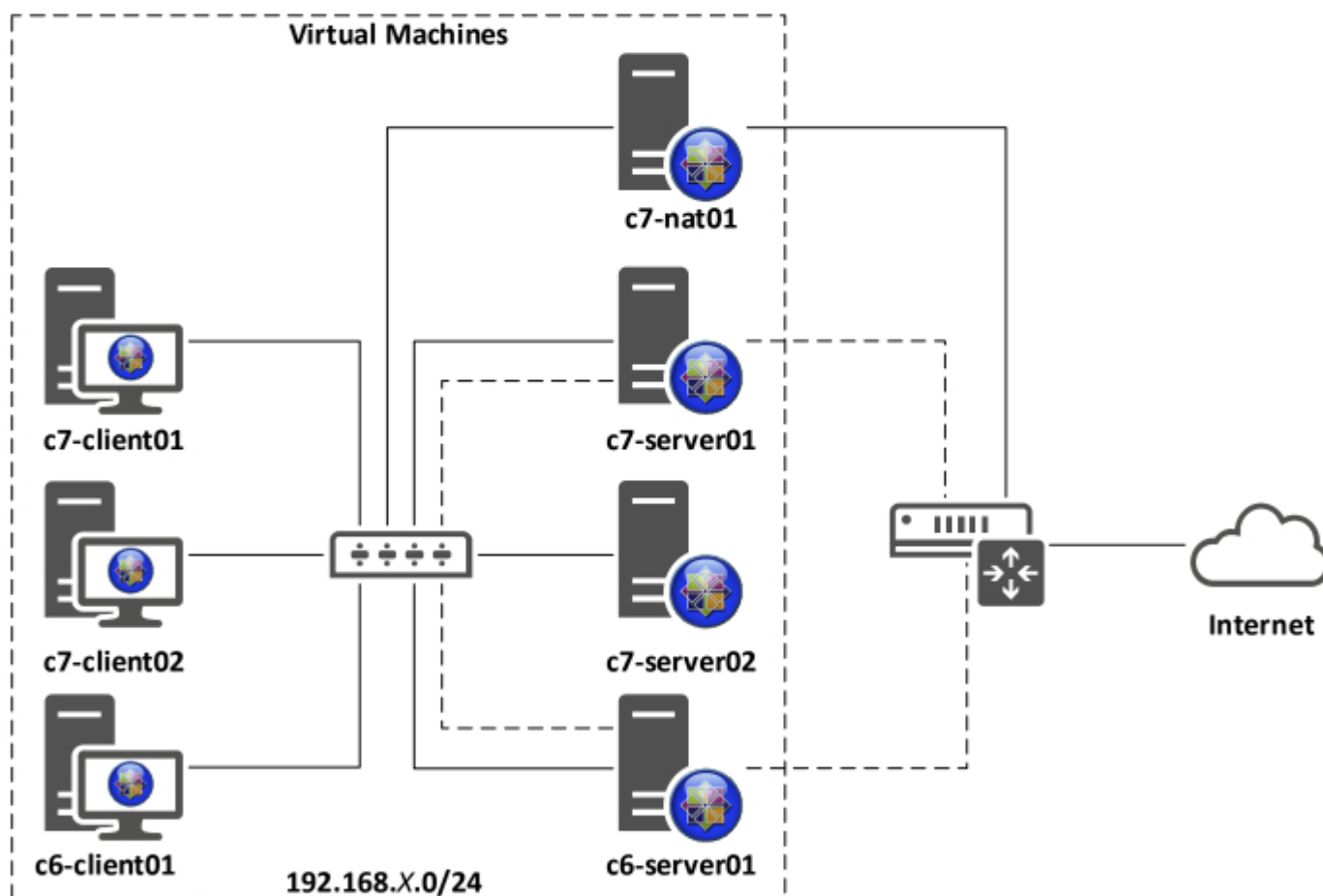
Лабораторная работа.

Лабораторная работа 6

- **Упражнения:**
 - Установка и настройка сервера OpenLDAP.
 - Установка и настройка 389 Directory Server (389-DS).
- **Необходимые машины:**
 - c7-nat01
 - c7-server01
 - c7-server02
 - c7-client01
- **Учетные данные:**
 - Пользователь: `root`.
 - Пароль: `Pa$$w0rd`.



<http://edu.softline.ru/>



Упражнение 1 – Установка и настройка сервера OpenLDAP.

**ПРИМЕЧАНИЕ:**

Примеры всех конфигурационных файлов можете взять по ссылке у инструктора.
<http://ССЫЛКА/files/LDAP-files/>

1. Установите пакеты **openldap**, **openldap-clients**, **openldap-servers** на машине **c7-server01**.

```
# yum install openldap openldap-clients openldap-servers -y
```

2. Настройте SELinux:

- **allow_ypbind=0**
- **authlogin_nsswitch_use_ldap=0**

```
# setsebool -P allow_ypbind=0 authlogin_nsswitch_use_ldap=0
```

**ПРИМЕЧАНИЕ:**

В качестве альтернативы, вы можете отключить SELinux.

3. Запустите службу **slapd** и настройте ее автоматический запуск.

```
# systemctl enable slapd.service
# systemctl start slapd.service
```

4. Скопируйте пример базы данных в **/var/lib/ldap**.

```
# cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

5. Смените владельца и группу на **ldap**.

```
# chown -R ldap:ldap /var/lib/ldap
```

6. Сгенерируйте пароль и настройте корневого администратора OpenLDAP.

**ВНИМАНИЕ:**

Чтобы избежать сложностей с кодировкой специальных символов используйте простые пароли, например: **123**.

```
# slappasswd
# vi ldaprootpasswd.ldif
dn: olcDatabase={0}config,cn=config
changetype: modify
add: olcRootPW
olcRootPW: {SSHA}PASSWORD
# ldapadd -H ldapi:/// -f ldaprootpasswd.ldif
```

7. Добавьте стандартные определения LDAP из каталога **/etc/openldap/schema/**.

```
# ldapadd -H ldapi:/// -f /etc/openldap/schema/cosine.ldif
# ldapadd -H ldapi:/// -f /etc/openldap/schema/nis.ldif
# ldapadd -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif
```

8. Добавьте определение вашего домена в базу LDAP.

```
# vi ldapdomain.ldif
dn: olcDatabase={1}monitor,cn=config
changetype: modify
replace: olcAccess
olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth"
  read by dn.base="cn=Manager,dc=11-100-X,dc=local" read by * none

dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=11-100-X,dc=local

dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=Manager,dc=11-100-X,dc=local

dn: olcDatabase={2}hdb,cn=config
changetype: modify
add: olcRootPW
olcRootPW: {SSHA}PASSWORD

dn: olcDatabase={2}hdb,cn=config
changetype: modify
add: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange by
  dn="cn=Manager,dc=11-100-X,dc=local" write by anonymous auth by self
write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by dn="cn=Manager,dc=11-100-X,dc=local" write by *
read

# ldapmodify -H ldap:// -f ldapdomain.ldif
```

9. Настройте структуру вашего домена.

```
# vi baseldapdomain.ldif
dn: dc=11-100-X,dc=local
objectClass: top
objectClass: dcObject
objectclass: organization
o: 11-100-X.local
dc: 11-100-X

dn: cn=Manager,dc=11-100-X,dc=local
objectClass: organizationalRole
cn: Manager
description: Directory Manager

dn: ou=Users,dc=11-100-X,dc=local
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=11-100-X,dc=local
objectClass: organizationalUnit
ou: Group

# ldapadd -x -D cn=Manager, dc=11-100-X,dc=local -W -f baseldapdomain.ldif
```

10. Создайте пользователя и группу LDAP.

```
# adduser ldapuser
# vi ldapgroup.ldif
dn: cn=Manager,ou=Group,dc=11-100-X,dc=local
objectClass: top
objectClass: posixGroup
```



```

        gidNumber: 1004
# ldapadd -x -W -D "cn=Manager,dc=ll-100-X,dc=local" -f ldapgroup.ldif
# vi ldapuser.ldif
    dn: uid=ldapuser,ou=People, dc=ll-100-X,dc=local
    objectClass: top
    objectClass: account
    objectClass: posixAccount
    objectClass: shadowAccount
    cn: ldapuser
    uid: ldapuser
    uidNumber: 1004
    gidNumber: 1004
    homeDirectory: /home/ldapuser
    userPassword: {SSHA}PASSWORD
    loginShell: /bin/bash
    gecos: ldapuser
    shadowLastChange: 0
    shadowMax: 0
    shadowWarning: 0
# ldapadd -x -D cn=Manager, dc=ll-100-X,dc=local -W -f ldapuser.ldif

```

11. Разрешите службу **ldap** в **firewalld**.

```
# firewall-cmd --add-service=ldap
```

12. Настройте машину **c7-server02** на использование OpenLDAP.

```

# authconfig --enableldap \
--enableldapauth --ldapserver=rhel7.mydomain.com \
--ldapbasedn="dc=ll-100-X,dc=local" --enablemkhomedir --update

```

13. Войдите в систему от имени пользователя **ldapuser**.



ВНИМАНИЕ:

По окончании лабораторной работы откатите машины в начальное состояние при помощи диспетчера Hyper-V (Hyper-V Manager).

Упражнение 2 – Установка и настройка 389 Directory Server (389-DS).

Выполняйте данное упражнение на машине **c7-server02**.

Задание 1: Подготовка.

1. Задайте полное доменное имя (FQDN) в файле **/etc/hosts**.

```
# vi /etc/hosts
```

Добавьте в конец файла следующую строку:

```
192.168.1.102    c7-server02.ll-100-1.local c7-server02
```

2. Разрешите в **firewalld** следующие TCP-порты: **389, 636, 9830**.

```
# firewall-cmd --permanent --add-port=389/tcp
# firewall-cmd --permanent --add-port=636/tcp
# firewall-cmd --permanent --add-port=9830/tcp
# firewall-cmd --reload
```

3. Добавьте репозиторий EPEL.

```
# yum install epel-release
```

4. Добавьте репозиторий REMI.

```
# wget http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
# rpm -Uvh remi-release-7.rpm
```

5. Проверьте, что новые репозитории добавились в **yum**.

```
# yum repolist
```

6. Настройте следующие параметры для **sysctl**:

Параметр	Значение
net.ipv4.tcp_keepalive_time	300
net.ipv4.ip_local_port_range	1024 65000
fs.file-max	64000

Отредактируйте файл **/etc/sysctl.conf**

```
# vi /etc/sysctl.conf
```

Добавив в него следующие строки:

```
net.ipv4.tcp_keepalive_time = 300
net.ipv4.ip_local_port_range = 1024 65000
fs.file-max = 64000
```

7. Настройте лимиты для модуля **pam_limits**:

Вставьте в файл **/etc/security/limits.conf**.

```
# vi /etc/security/limits.conf
```

Следующие лимиты:

```
*                soft    nofile    8192
*                hard    nofile    8192
```

8. Добавьте в конец файла `/etc/profile` следующую строку:

```
ulimit -n 8192
```

9. Добавьте в конец файла `/etc/pam.d/login` следующую строку:

```
session    required    /lib/security/pam_limits.so
```

10. Создайте пользователя `ldapadmin` и задайте ему пароль (Например: `Pa$$w0rd`)

```
# useradd ldapadmin
# passwd ldapadmin
```

Задание 2: Установка 389 Directory Server.

1. Установите пакеты `389-ds-base` и `openldap-clients`

```
# yum install 389-ds-base openldap-clients
```

2. Загрузите и установите дополнительные пакеты:

- `idm-console-framework-1.1.14-1.el7.noarch.rpm`
- `389-adminutil-1.1.22-1.el7.x86_64.rpm`
- `389-admin-1.1.42-1.el7.x86_64.rpm`
- `389-admin-console-1.1.10-1.el7.noarch.rpm`
- `389-console-1.1.9-1.el7.noarch.rpm`
- `389-ds-console-1.2.12-1.el7.noarch.rpm`

Ссылку на пакеты получите у инструктора.

```
# yum localinstall idm* 389*
```



ПРИМЕЧАНИЕ:

Пакеты также доступны по следующей ссылке:

ftp://rpmfind.net/linux/epel/testing/7/x86_64/3/

Задание 3: Типичная настройка 389 Directory Server.

1. Воспользуйтесь мастером для типовой настройки `389-DS`.

```
# setup-ds-admin.pl
```

Примеры ответов на вопросы мастера для типовой настройки:

Вопрос	Ввод	Клавиша
Would you like to continue with set up? [yes]:		<Enter>
Would you like to continue? [no]:	yes	<Enter>
Choose a setup type [2]:		<Enter>
Computer name [c7-server02.ll-100-1.local]:		<Enter>
System User [nobody]:	ldapadmin	<Enter>
System Group [nobody]:	ldapadmin	<Enter>
Do you want to register this software with an existing configuration directory server? [no]:		<Enter>
administrator ID [admin]:		<Enter>
Password:	Pa\$\$w0rd	<Enter>
Password (confirm):	Pa\$\$w0rd	<Enter>
Administration Domain [ll-100-1.local]:		<Enter>

Directory server network port [389]:		<Enter>
Directory server identifier [c7-server02]:		<Enter>
Suffix [dc=ll-100-1, dc=local]:		<Enter>
Directory Manager DN [cn=Directory Manager]:		<Enter>
Password:	Pa\$\$w0rd	<Enter>
Password (confirm):	Pa\$\$w0rd	<Enter>
Administration port [9830]:		<Enter>

Задание 4: Настройка служб 389 Directory Server.

1. Настройте автоматический запуск для `dirsrv.target` и `dirsrv-admin`.

```
# systemctl enable dirsrv.target
# systemctl enable dirsrv-admin
```

2. Запустите: `dirsrv.target` и `dirsrv-admin`.

```
# systemctl start dirsrv.target
# systemctl start dirsrv-admin
```

Задание 5: Проверка работы 389 Directory Server.

1. Протестируйте работу LDAP-сервера при помощи утилиты `ldapsearch`.

```
# ldapsearch -x -b "dc=ll-100-1,dc=local"
```

Пример вывода:

```
# extended LDIF
#
# LDAPv3
# base <dc=ll-100-1,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# ll-100-1.local
dn: dc=ll-100-1,dc=local
objectClass: top
objectClass: domain
dc: ll-100-1

# Directory Administrators, ll-100-1.local
dn: cn=Directory Administrators,dc=ll-100-1,dc=local
objectClass: top
objectClass: groupofuniquenames
cn: Directory Administrators
uniqueMember: cn=Directory Manager

# Groups, ll-100-1.local
dn: ou=Groups,dc=ll-100-1,dc=local
objectClass: top
objectClass: organizationalunit
ou: Groups

# People, ll-100-1.local
dn: ou=People,dc=ll-100-1,dc=local
objectClass: top
objectClass: organizationalunit
ou: People

# Special Users, ll-100-1.local
dn: ou=Special Users,dc=ll-100-1,dc=local
objectClass: top
objectClass: organizationalUnit
ou: Special Users
description: Special Administrative Accounts
```

```
# Accounting Managers, Groups, ll-100-1.local
dn: cn=Accounting Managers,ou=Groups,dc=ll-100-1,dc=local
objectClass: top
objectClass: groupOfUniqueNames
cn: Accounting Managers
ou: groups
description: People who can manage accounting entries
uniqueMember: cn=Directory Manager

# HR Managers, Groups, ll-100-1.local
dn: cn=HR Managers,ou=Groups,dc=ll-100-1,dc=local
objectClass: top
objectClass: groupOfUniqueNames
cn: HR Managers
ou: groups
description: People who can manage HR entries
uniqueMember: cn=Directory Manager

# QA Managers, Groups, ll-100-1.local
dn: cn=QA Managers,ou=Groups,dc=ll-100-1,dc=local
objectClass: top
objectClass: groupOfUniqueNames
cn: QA Managers
ou: groups
description: People who can manage QA entries
uniqueMember: cn=Directory Manager

# PD Managers, Groups, ll-100-1.local
dn: cn=PD Managers,ou=Groups,dc=ll-100-1,dc=local
objectClass: top
objectClass: groupOfUniqueNames
cn: PD Managers
ou: groups
description: People who can manage engineer entries
uniqueMember: cn=Directory Manager

# search result
search: 2
result: 0 Success

# numResponses: 10
# numEntries: 9
```

Задание 6: Настройка графических консолей 389 Directory Server.

1. Установите графическую оболочку **Xfce**.

```
# yum groupinstall Xfce -y
```

2. Перезагрузите сервер.

```
# reboot
```

3. Проверьте работу консоли с машины **c7-client01**.

```
# ssh -X root@192.168.1.102 /usr/bin/389-console -a
http://192.168.1.102:9830
```



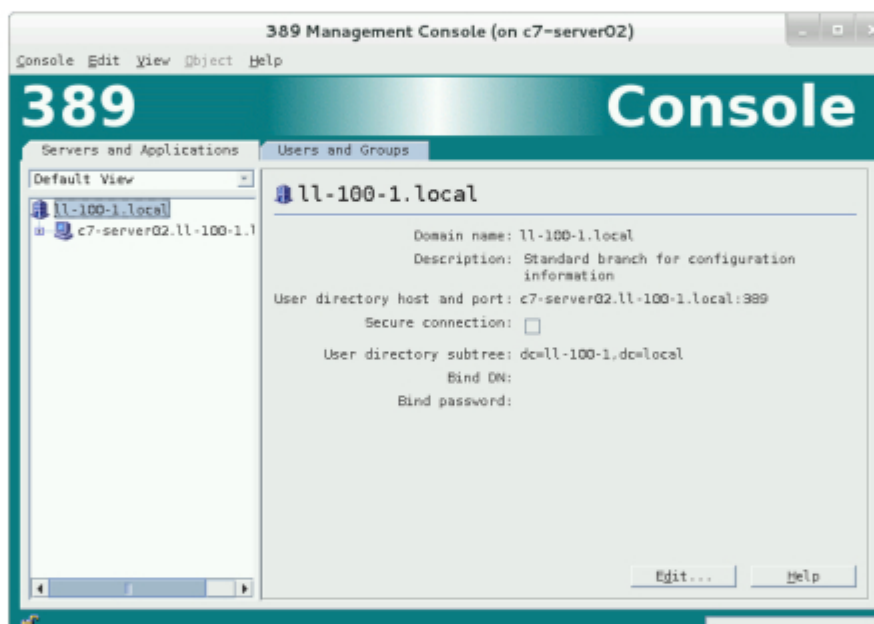
ПРИМЕЧАНИЕ:

Дальнейшие задания в консолях можно выполнять как локально на **c7-server02**, так и удаленно с **c7-client01**.

4. Войдите в 389 management Console при помощи учетной записи **admin** и пароля **Pa\$\$w0rd**, если вы использовали этот пароль на этапе настройки.

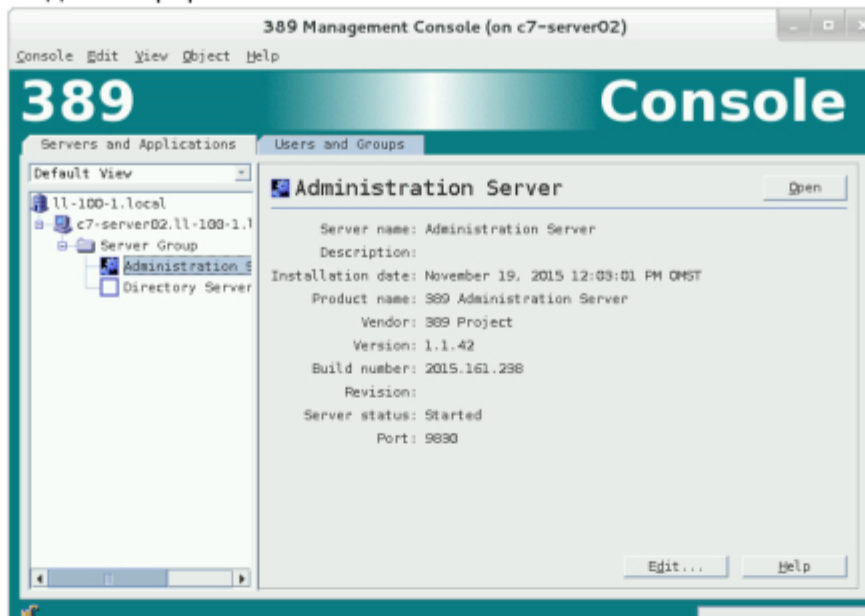


Если до этого момента все было выполнено правильно вы увидите следующую консоль:



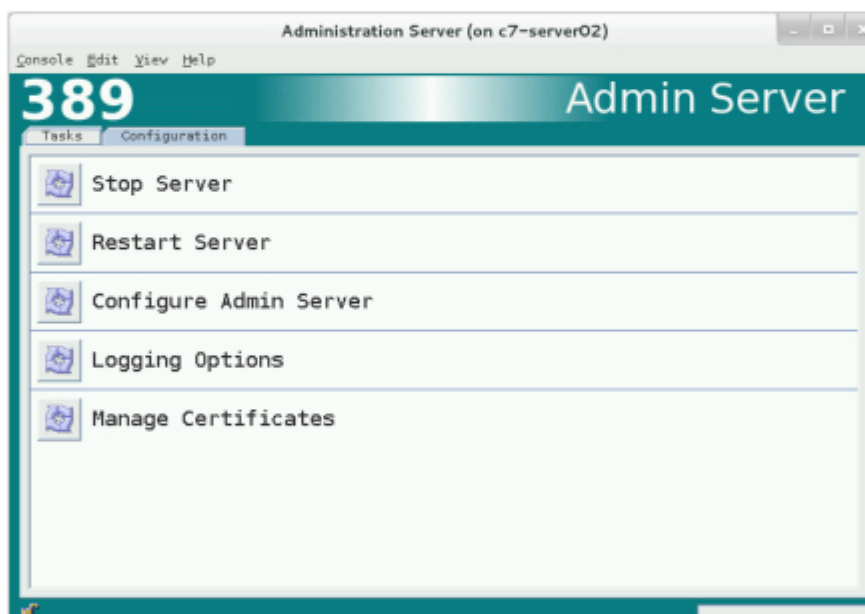
Задание 7: Знакомство с Administration Server (Сервером управления).

1. Перейдите к разделу Administration Server (Server Group → Administration Server) и нажмите Open.

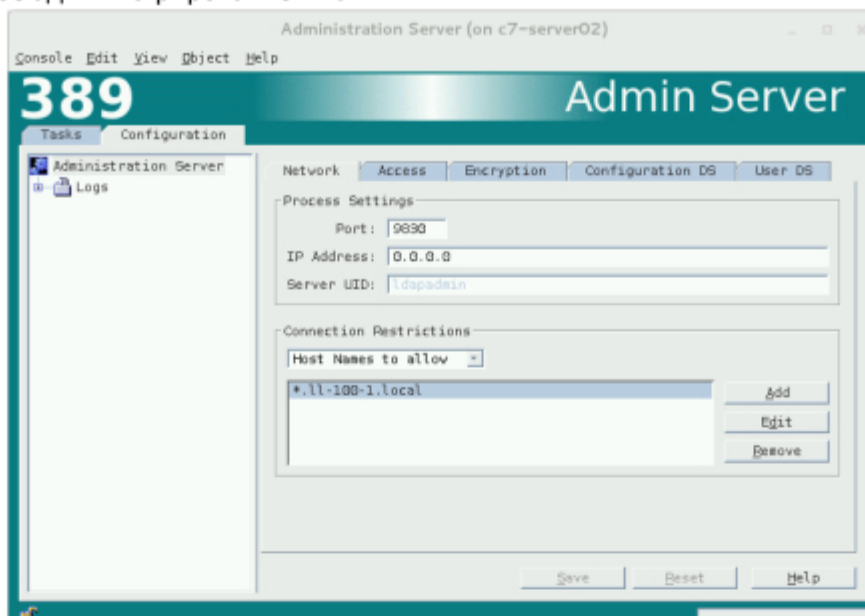


2. Здесь вы найдете две вкладки: Tasks (Задачи) и Configuration (Настройки).

На вкладке Tasks (Задачи) вы можете останавливать, перезапускать и настраивать сервер.

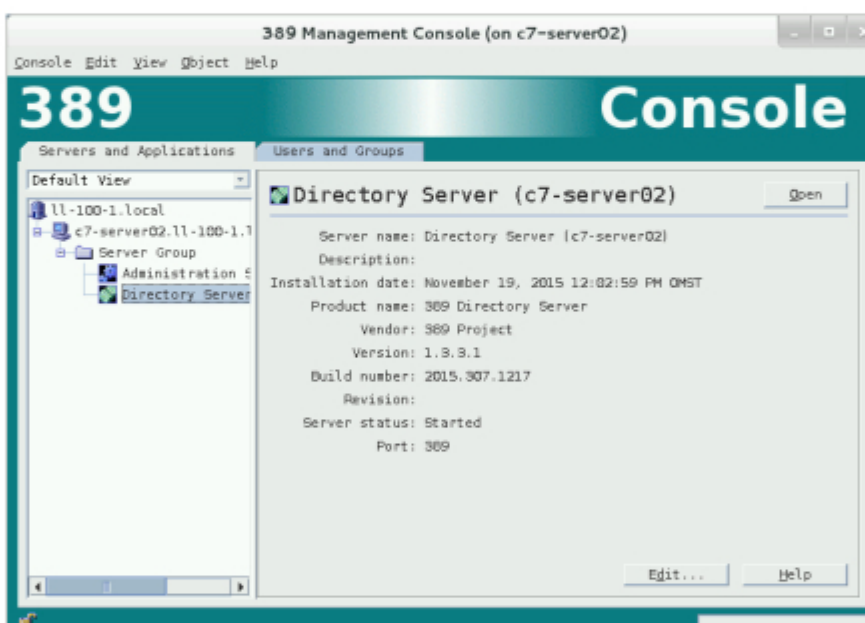


3. На вкладке Configuration (Настройка) вы можете настраивать IP-адрес, пароль администратора LDAP, каталог пользователя по умолчанию. Также вы можете определить какие имена хостов и IP-адреса могут получить доступ к LDAP-серверу.

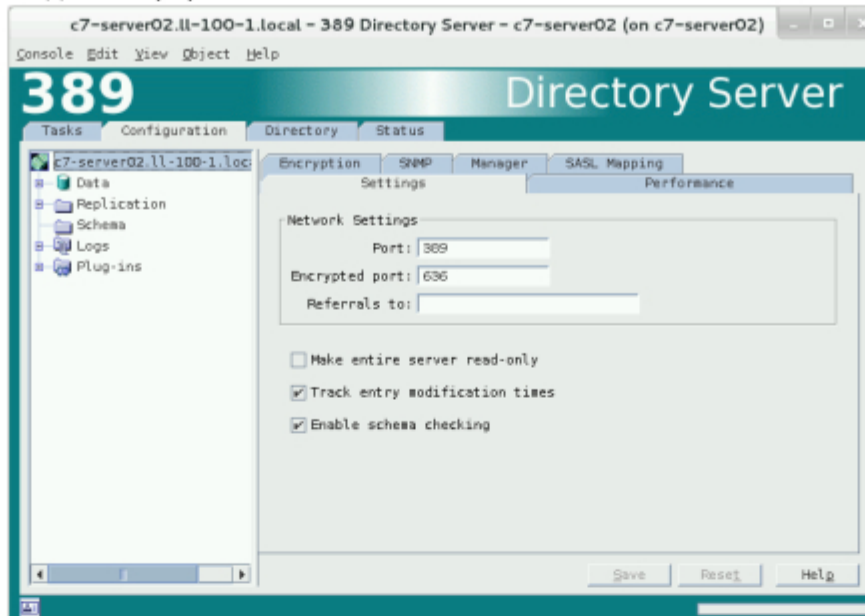


Задание 8: Знакомство с Directory Server (Сервером каталога).

1. Перейдите к разделу Directory Server (Server Group → Directory Server) и нажмите Open.



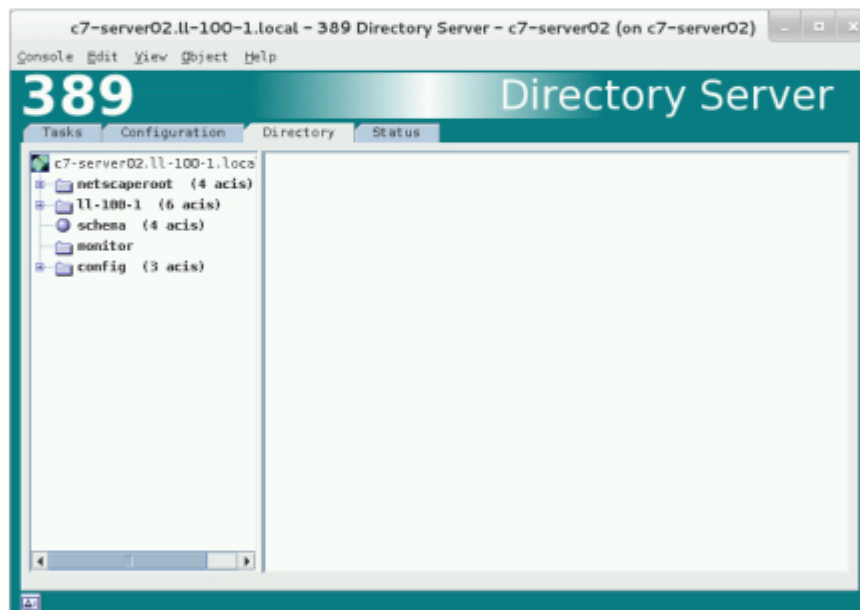
2. В данной секции вы можете выполнять любые задачи настройки вашего LDAP-сервера. Вы можете менять порт по умолчанию, создавать пользователей, группы, подразделения и т.д.



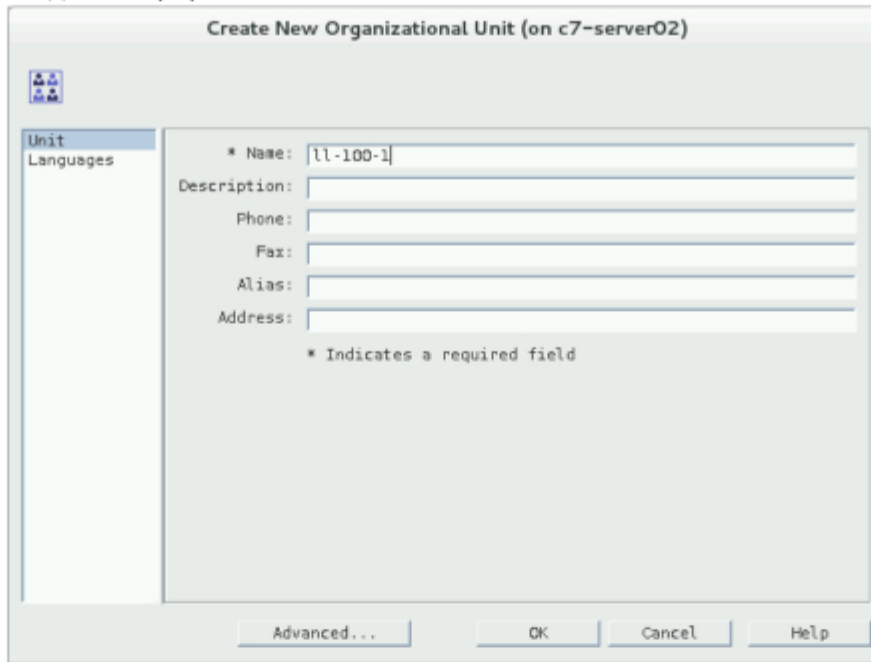
В данной секции очень много настроек обязательно познакомьтесь со всеми доступными вкладками.

Задание 9: Создание Organization units (Подразделений), Groups (Групп) и Users (Пользователей).

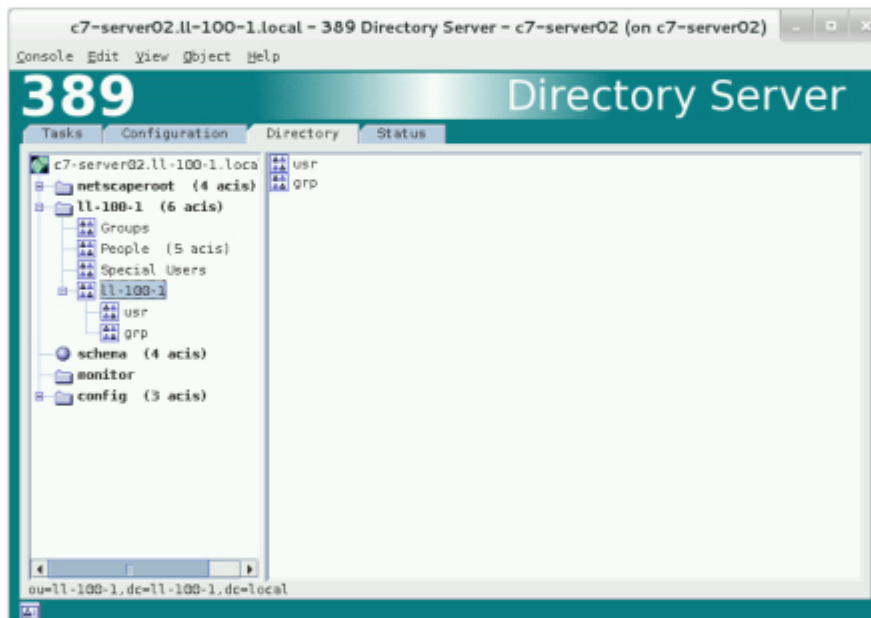
1. Перейдите на вкладку Directory (Каталог).



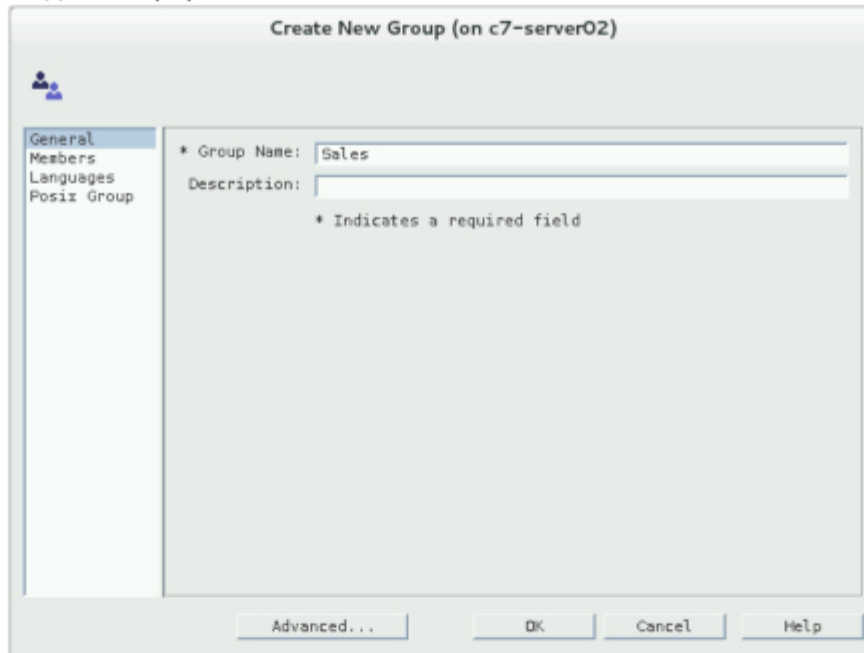
2. Нажмите правой кнопкой на имени вашего домена (Например: ll-100-1) выберите **New -> Organization Unit**.
3. В поле **Name** введите ll-100-1 и нажмите **Ok**.



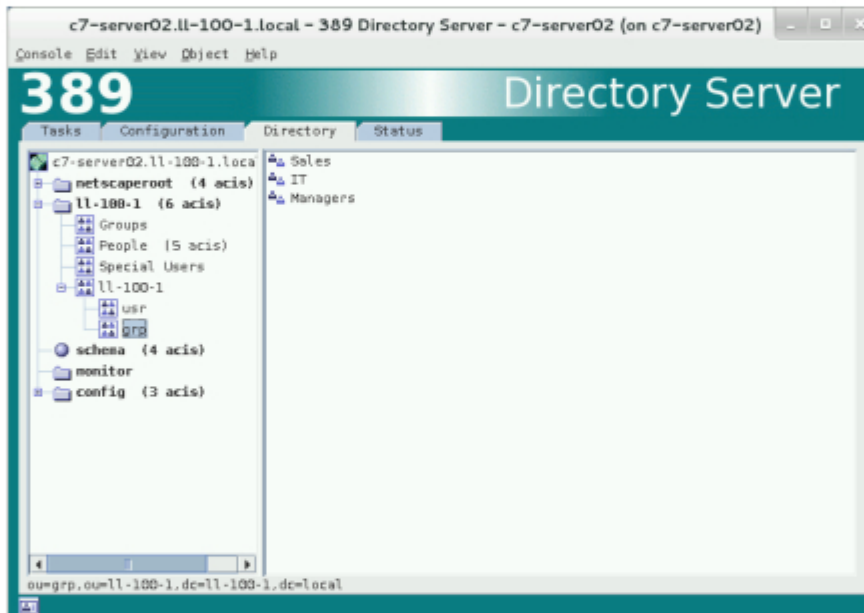
4. По аналогии в подразделении **ll-100-1** создайте два подразделения: **usr** и **grp**.



5. В подразделении **grp** создайте группы Sales, IT и Managers.



Чтобы получилось следующим образом:



6. Создайте пользователей в подразделении **usr**, согласно данным в таблице:

First Name:	Petr	Ivan	Semen
Last Name:	Petrov	Ivanov	Sidorov
Cpmmon Name:	Petr Petrov	Ivan Ivanov	Semen Sidorov
User ID:	PPetrov	IIvanov	SSidorov
Password:	Pa\$\$w0rd	Pa\$\$w0rd	Pa\$\$w0rd
E-Mail:	PPetrov@ll-100-1.local	IIvanov@ll-100-1.local	SSidorov@ll-100-1.local

7. Самостоятельно разместите пользователей по группам.

Задание 10: Проверка работы 389 Directory Server.

1. Проверьте что вновь созданные объекты могут быть обнаружены при помощи поиска:

2. Проверьте, что изменения были внесены в каталог LDAP-сервера при помощи утилиты **ldapsearch**.

```
# ldapsearch -x -b "dc=ll-100-1,dc=local"
```

Пример вывода:

```
# extended LDIF
#
# LDAPv3
# base <dc=ll-100-1,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# ll-100-1.local
dn: dc=ll-100-1,dc=local
objectClass: top
objectClass: domain
```



```

dc: 11-100-1

# Directory Administrators, 11-100-1.local
dn: cn=Directory Administrators,dc=11-100-1,dc=local
objectClass: top
objectClass: groupofuniquenames
cn: Directory Administrators
uniqueMember: cn=Directory Manager

# Groups, 11-100-1.local
dn: ou=Groups,dc=11-100-1,dc=local
objectClass: top
objectClass: organizationalunit
ou: Groups

# People, 11-100-1.local
dn: ou=People,dc=11-100-1,dc=local
objectClass: top
objectClass: organizationalunit
ou: People

# Special Users, 11-100-1.local
dn: ou=Special Users,dc=11-100-1,dc=local
objectClass: top
objectClass: organizationalUnit
ou: Special Users
description: Special Administrative Accounts

# Accounting Managers, Groups, 11-100-1.local
dn: cn=Accounting Managers,ou=Groups,dc=11-100-1,dc=local
objectClass: top
objectClass: groupOfUniqueNames
cn: Accounting Managers
ou: groups
description: People who can manage accounting entries
uniqueMember: cn=Directory Manager

# HR Managers, Groups, 11-100-1.local
dn: cn=HR Managers,ou=Groups,dc=11-100-1,dc=local
objectClass: top
objectClass: groupOfUniqueNames
cn: HR Managers
ou: groups
description: People who can manage HR entries
uniqueMember: cn=Directory Manager

# QA Managers, Groups, 11-100-1.local
dn: cn=QA Managers,ou=Groups,dc=11-100-1,dc=local
objectClass: top
objectClass: groupOfUniqueNames
cn: QA Managers
ou: groups
description: People who can manage QA entries
uniqueMember: cn=Directory Manager

# PD Managers, Groups, 11-100-1.local
dn: cn=PD Managers,ou=Groups,dc=11-100-1,dc=local
objectClass: top
objectClass: groupOfUniqueNames
cn: PD Managers
ou: groups
description: People who can manage engineer entries
uniqueMember: cn=Directory Manager

# 11-100-1, 11-100-1.local
dn: ou=11-100-1,dc=11-100-1,dc=local
ou: 11-100-1
objectClass: top
objectClass: organizationalunit

# usr, 11-100-1, 11-100-1.local
dn: ou=usr,ou=11-100-1,dc=11-100-1,dc=local
ou: usr
objectClass: top
objectClass: organizationalunit

# grp, 11-100-1, 11-100-1.local
dn: ou=grp,ou=11-100-1,dc=11-100-1,dc=local
ou: grp
objectClass: top
objectClass: organizationalunit

```

```
# Sales, grp, ll-100-1, ll-100-1.local
dn: cn=Sales,ou=grp,ou=ll-100-1,dc=ll-100-1,dc=local
objectClass: top
objectClass: groupofuniquenames
cn: Sales
uniqueMember: uid=PPetrov,ou=usr,ou=ll-100-1,dc=ll-100-1,dc=local

# IT, grp, ll-100-1, ll-100-1.local
dn: cn=IT,ou=grp,ou=ll-100-1,dc=ll-100-1,dc=local
objectClass: top
objectClass: groupofuniquenames
cn: IT
uniqueMember: uid=IIvanov,ou=usr,ou=ll-100-1,dc=ll-100-1,dc=local

# Managers, grp, ll-100-1, ll-100-1.local
dn: cn=Managers,ou=grp,ou=ll-100-1,dc=ll-100-1,dc=local
objectClass: top
objectClass: groupofuniquenames
cn: Managers
uniqueMember: uid=SSidorov,ou=usr,ou=ll-100-1,dc=ll-100-1,dc=local

# PPetrov, usr, ll-100-1, ll-100-1.local
dn: uid=PPetrov,ou=usr,ou=ll-100-1,dc=ll-100-1,dc=local
mail: PPetrov@ll-100-1.local
uid: PPetrov
givenName: Petr
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: Petrov
cn: Petr Petrov

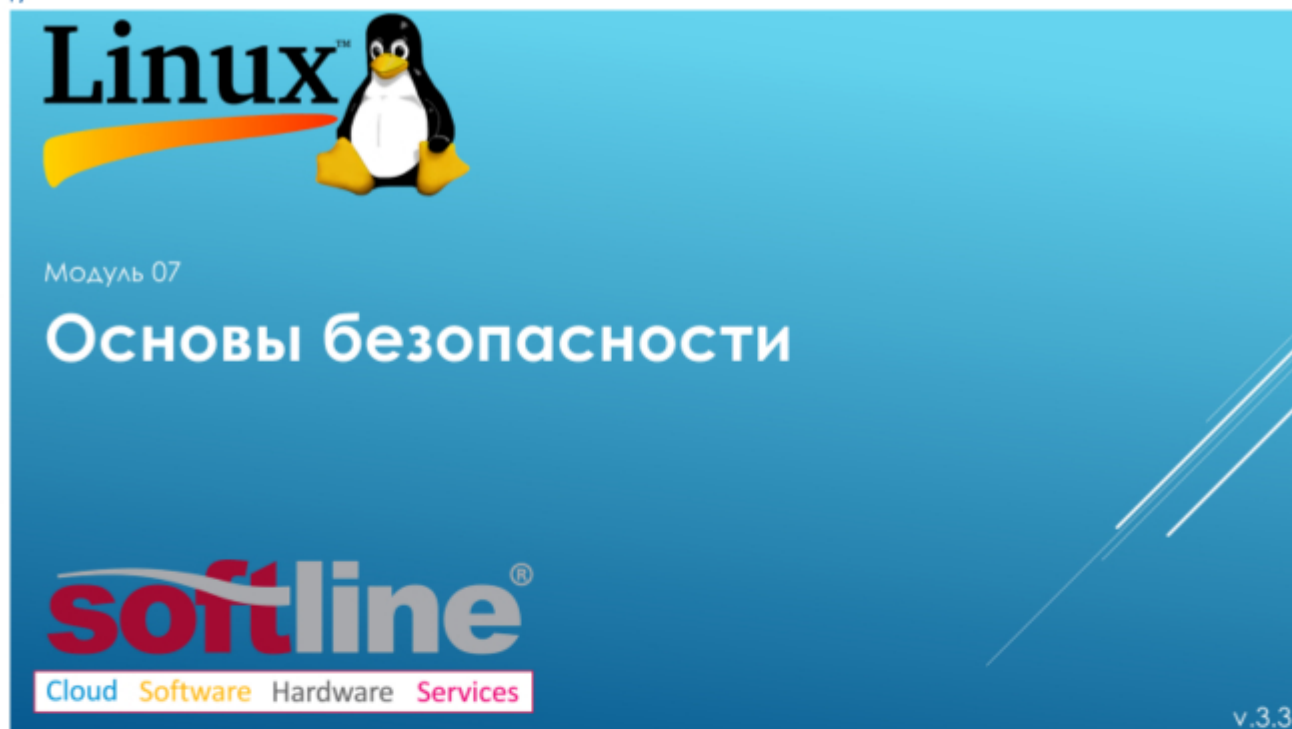
# IIvanov, usr, ll-100-1, ll-100-1.local
dn: uid=IIvanov,ou=usr,ou=ll-100-1,dc=ll-100-1,dc=local
mail: IIvanov@ll-100-1.local
uid: IIvanov
givenName: Ivan
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: Ivanov
cn: Ivan Ivanov

# SSidorov, usr, ll-100-1, ll-100-1.local
dn: uid=SSidorov,ou=usr,ou=ll-100-1,dc=ll-100-1,dc=local
mail: SSidorov@ll-100-1.local
uid: SSidorov
givenName: Semen
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: Sidorov
cn: Semen Sidorov

# search result
search: 2
result: 0 Success

# numResponses: 19
# numEntries: 18
```

Модуль 07: Основы безопасности.



В модуле представлены:

- Основные правила безопасности.
- Шаги к безопасности.
- Сетевое обслуживание.
- Противодействие.
- Атака типа MITM.
- SELinux.
- Контекст SELinux.
- Режимы работы SELinux.
- Изменение контекста.
- Резервное копирование.
- Bacula.

Безопасность

- **Безопасность информационных систем – комплекс мер, призванных минимизировать возможность несанкционированного доступа к данным:**
 - Сетевая безопасность – обеспечение защиты данных от доступа извне посредством сети.
 - Локальная безопасность – обеспечение защиты в рамках локальной работы с ОС.
 - Физическая безопасность – обеспечение доступа к системам хранения и обработки информации.
- **Ни одна система не может иметь абсолютной защиты.**

<http://edu.softline.ru/>



В современном мире крайне возросла роль информации, а следовательно и роль средств, обеспечивающих её безопасность. Ни для кого не секрет, что вычислительные системы используются не только в финансовых учреждениях, но и в таких местах, как больницы, аэропорты и других жизненно-важных структурах.

Потеря информации, либо получение к ней доступа третьими лицами грозит не только простоем в работе, но и крахом бизнеса, смертью людей. Исходя из этого, обеспечение информационной безопасности можно назвать основной деятельностью администратора вычислительных систем. Все внедрения должны производиться исходя из определения всех возможных рисков и всегда нужно помнить, что более сложная система является более уязвимой. Задача администратора найти тонкую грань между необходимым функционалом вычислительной системы и надёжным ограничением доступа к информации, обрабатываемой данной системой.

Правила безопасности

- Обеспечение безопасности можно описать простыми правилами:
 - На сервере должно быть запущено только то, что нужно.
 - Минимум программ должно быть запущено от имени суперпользователя root.
 - Должен проводиться периодический аудит:
 - Паролей.
 - Программного обеспечения.
 - Пользователи должны быть обучены и максимально ограничены в полномочиях.
 - Должна быть выработана чёткая политика безопасности, следовать которой должны все без исключения.

<http://edu.softline.ru/>



При разработке любой вычислительной системы всегда точно прописывайте все требования к сервисам, которые планируется использовать. С максимальной точностью определите достаточные полномочия пользователей к данным сервисам. Если данные шаги будут выполнены без должной аккуратности, то вы рискуете получить брешь в системе безопасности, о которой узнаете в последнюю очередь, когда пользователь, которого вы не учли, совершенно случайно узнает, что у него есть доступ к финансовой отчетности и совершенно случайно возьмёт копии, скажем, при своём увольнении.

Возьмите за практику, а лучше, пропишите в политике информационной безопасности необходимость периодической проверки паролей пользователей и программ на наличие уязвимостей. Станьте взломщиком раньше, чем придёт настоящий. Но не забывайте, прежде чем ругать пользователей за "слабые" пароли, вы должны обеспечить все условия, при которых пользователь будет осведомлён о возможных рисках. Также вы должны настроить систему таким образом, что пользователь не сможет установить плохой пароль.

Всегда следуйте правилу – "если программа может быть запущена не от пользователя root, значит она должна быть запущена не от него". В случае, когда в программе будет обнаружена уязвимость и взломщик окажется расторопнее, то всё, что он получит – права пользователя, от которого была запущена программа. А если она была запущена от root?

Помните, что взломы не всегда носят адресный характер. Чаще всего взломы производятся в автоматическом режиме, когда используются известные уязвимости, а взломанные системы участвуют в различных ботнетах, рассылая спам или используя в качестве площадки для новых взломов. Думать, что ваш сервер слишком мал и не интересен для взломщиков, по крайней мере наивно.

Шаги к безопасности

- Установите пароль на загрузчик.
- Отключите все ненужные программы.
- Оставьте открытыми сетевые порты только там, где это необходимо.
- Разрешите вход в систему только тем, кому это необходимо.
- Установите и настройте программы для автоматизации процесса обеспечения безопасности:
 - Logwatch.
 - Denyhosts.
 - Tripwire.
- Произведите проверку контролируемых серверов снаружи локальной сети.

<http://edu.softline.ru/>




Первое, что нужно сделать, ещё на этапе установки ОС определить пароль для управления вторичным загрузчиком. Пароль не позволит злоумышленнику вмешаться в процесс загрузки операционной системы. Для установки пароля внесите соответствующую запись в файл `/boot/grub/grub.conf`:

```
password --md5 $1$EfXOX$4V7RQanmwQxRwXqL/Vepv/
```

Для того, чтобы сгенерировать пароль используйте утилиту `grub-md5-crypt`.

Помните, чем проще система, тем она надёжнее. Первое и самое мощное средство упрощения системы – отключение всего не нужного. Определите, какие программы должны быть запущены в системе и отключите все ненужные, особое внимание уделите сетевым сервисам.

Вызовите программу `ps`, чтобы посмотреть, какие программы запущены в системе. Те, что являются лишними должны быть исключены из загрузки. В системах, построенных на базе RedHat вы можете использовать программу `chkconfig` для управления загрузкой программ:

```
# chkconfig postfix off
```

```
# chkconfig named on
```

В идеальной ситуации на почтовом сервере должен быть запущен только демон SMTP и POP3/IMAP, на сервере баз данных только база данных, например, Mysql/Postgresql и т.д.

Сетевое обслуживание

- Изначально дистрибутив включает установку множества сетевых сервисов.
- Каждый сетевой сервис является потенциальной брешью.
- Администратор должен чётко определить необходимые сервисы, какие сетевые интерфейсы и сети должен обслуживать сервис и т.д.
- Программа **netstat** выведет информацию о:
 - Текущих соединениях и их состоянии.
 - Открытых портах.
- **nmap** – мощное средство сетевого аудита:
 - Используйте для проверки внешних серверов.

<http://edu.softline.ru/>



Большое множество сетевых атак направлены на поиск уязвимых сетевых сервисов. Ещё чаще серверы взламываются через брешы в тех сервисах, которые на самом деле не используются владельцами сервера. Так зачем злоумышленникам давать лишние шансы?

Используйте программу **netstat** для исследования. Утилита поможет узнать вам, какие сервисы и на каких портах принимают соединения, какие соединения уже установлены и в каком состоянии находятся. Программа **netstat**, вызванная без аргументов, выведет информацию обо всех текущих соединениях. Чтобы сделать более точную и удобочитаемую выборку используйте параметры:

- **-a** – вывести все ожидаемые и установленные соединения
- **-t** – вывести информацию о соединениях по протоколу TCP
- **-u** – вывести информацию о соединениях по протоколу UDP
- **-p** – вывести идентификатор процесса, отвечающего за соединение
- **-n** – не преобразовывать имена
- **-r** – вывести таблицу маршрутизации

Для проверки удалённых узлов используйте программу **nmap**, являющейся мощным средством сетевого аудита. Используйте её для получения информации о работающих сервисах, активных узлах сети и т.д. Основные параметры программы **nmap**:

- **sS** – TCP Syn scan. На каждый порт будет послан пакет с флагом SYN, если сервис на том конце активен, то он отправит пакет Syn/Ack.
- **sP** – Ping Scan. Основное средство поиска активных узлов в сети.
- **P0** – не пинговать узел перед сканированием.
- **sU** – UDP-сканирование

Пример:

```
# nmap -sS 10.10.10.1
Starting Nmap 4.20 ( http://insecure.org ) at 2008-06-04 19:09 YEKST
Interesting ports on 10.10.10.1:
Not shown: 1694 closed ports
PORT      STATE SERVICE
```

```
22/tcp open  ssh
53/tcp open  domain
631/tcp open  ipp
MAC Address: 00:50:22:83:ED:AE (Zonet Technology)
Nmap finished: 1 IP address (1 host up) scanned in 2.751 seconds
```

Противодействие

- Существует два основных вида реагирования на опасности.
- **Мониторинг и обеспечение обороны узла:**
 - Наблюдение за сетевой активностью.
 - Изучение журнальных файлов.
 - Блокировка доступа для угрожающих узлов.
- **Мониторинг и активное противодействие:**
 - Наблюдение за сетевой активностью.
 - Изучение журнальных файлов и активное противодействие.
 - Блокировка доступа и попытка обезвредить злоумышленника на дальних подступах.

<http://edu.softline.ru/>



Для обеспечения максимальной безопасности необходимо выполнение периодических операций:

- Мониторинг журнальных файлов.
- Обновление программы, установленных в системе.
- Комплексный аудит серверов.
- Аудит паролей учётных записей пользователей.

Атака типа MITM

- MITM – Man In The Middle, она же ARP-spoofing.
- Коварная атака, позволяющая атакующему быть посредником между сетевыми узлами, пропуская через себя весь трафик.
- Основана на незащищённости протокола ARP:
 - Производится подмена адреса в ARP-таблице.
- Методы противодействия:
 - `arpwatch` – отслеживает изменения в arp-таблице с последующим уведомлением администратора.
 - Статическая ARP-таблица.

<http://edu.softline.ru/>



Сетевое взаимодействие происходит на уровне Ethernet-пакетов, а адресация выполняет по MAC-адресам, которые "вшиваются" в сетевое оборудование. Когда узел с адресом 1.2.3.4 хочет передать пакет узлу с адресом 1.2.3.5, то он сначала посылает в сеть arp-запрос с вопросом, "У кого адрес 1.2.3.5, скажи свой MAC-адрес". И узел, у которого есть запрашиваемый адрес, отправляет ответ, "У меня адрес 1.2.3.5, вот тебе мой MAC-адрес". Данная связка IP-адрес/MAC-адрес заносится в ARP-таблицу и будет использоваться для дальнейшей работы.

Существует такой набор ARP пакетов, которые называются самопроизвольными и призваны предотвращать конфликты IP-адресов и так далее. Именно такими пакетами и пользуется злоумышленник. В то время как два узла общаются между собой, злоумышленник отправляет им самопроизвольные ответы:

- Узлу 1.2.3.4 отсылается ответ, что адрес 1.2.3.5 теперь имеет другой MAC-адрес (злоумышленника).
- Узлу 1.2.3.5 отсылается ответ, что адрес 1.2.3.4 теперь имеет другой MAC-адрес (злоумышленника).

Теперь оба узла будут отправлять свои пакеты злоумышленнику. Если бы он их просто принимал, то соединение быстро бы оказалось разорванным. Именно поэтому узел злоумышленника производит необходимую маршрутизацию данных пакетов, а следовательно, для обоих узлов остаётся незамеченным.

Программа `Arpwatch` осуществляет мониторинг arp-таблицы и незамедлительно сообщает об изменениях, в ней происходящих. Но, к сожалению, это средство фиксирования проблемы, но не её предупреждения и решения.

Идеальным средством против атак данного типа является статическая Arp-таблица. Существенным минусом такого решения является то, что отключив динамическое обновление arp-таблицы мы рискуем потерять связь с теми, кто не занесён в таблицу статически. Но, тем не менее, лучше всего формировать статическую таблицу в момент пиковой активности, когда почти все компьютеры находятся в сети. Чтобы сформировать её, дайте команду:

```
# arp -an | grep -v incom | awk '{print $2" "$4}' | tr -d '()' > /etc/ethers
```

Когда таблица сформирована, остаётся загрузить её и отключить динамическое обновление таблицы ARP:

```
# arp -f /etc/ethers  
# ifconfig eth0 -arp
```

SELinux

- **Security Enhanced Linux:**
 - Реализация системы принудительного контроля доступа (MAC):
 - Права доступа определяются политикой.
 - До сих пор использовалась система дискреционного контроля доступа (DAC):
 - Права доступа определяются пользователем.
- Политика MAC применяется после DAC.
- Разработка американского агентства национальной безопасности.

<http://edu.softline.ru/>



До сих пор в Linux применялась только дискреционная система контроля за доступом. Это означало, что пользователь мог устанавливать права доступа исходя из своих предпочтений, что могло послужить потенциальной брешью в системе безопасности.

Сегодня в Linux интегрирована система принудительного контроля за доступом SELinux, которая определяет права доступа, основываясь на политике безопасности. Именно политика определяет, кто и какие файлы может видеть, читать, копировать, удалять.

Контекст SELinux

- Каждый процесс или файл имеют контекст SELinux:

`identity:role:domain/type`

- identity** – сущность, имеет значение, аналогичное UID, но таковым не является, определяет возможность использования ролей.
- role** – определяет, какие домены могут быть использованы.
- domain** – каждый процесс выполняется в определённом домене SELinux.
- type** – типы применяются к файлам и у каждого файла существует определённый тип.

<http://edu.softline.ru/>



Каждый процесс выполняется в определённом контексте SELinux, каждый объект (файл, каталог и т.д.) также имеет свой контекст SELinux.

Чтобы посмотреть контекст безопасности файла, необходимо выполнить команду:

```
# ls -Z /etc/passwd
-rw-r--r-- root root system_u:object_r:etc_t /etc/passwd
```

По умолчанию, все вновь создаваемые файлы будут наследовать контекст безопасности от родительского каталога.

Важно помнить, что к процессам применимо понятие домена, в котором этот процесс выполняется, а к файлам применяется понятие типа. Фактически домен – набор правил, определяющих, к каким типам имеет доступ процесс, выполняющийся в данном домене. Чтобы узнать контекст безопасности процесса необходимо дать команду:

```
# ps -Z | grep bash
user_u:system_r:unconfined_t 3657 pts/2 00:00:00 bash
```

Если в примере с файлом `/etc/passwd` мы увидели, что он принадлежит типу `etc_t`, то в данном примере можно увидеть, что процесс `bash` принадлежит домену `unconfined_t`.

Режимы работы SELinux

- Существует три основных режима работы SELinux:
 - `disabled` – отключен.
 - `enforcing` – включен.
 - `permissive` – включен, с целью наблюдения.
- `sestatus` – вывести текущий статус SELinux.
- `setenforce` – изменить режим работы SELinux.

<http://edu.softline.ru/>



Команда `sestatus` выведет информацию о текущем статусе SELinux:

```
# sestatus
SELinux status:                enabled
SELinuxfs mount:              /selinux
Current mode:                  permissive
Mode from config file:         permissive
Policy version:                21
Policy from config file:       targeted
```

Чтобы налету изменить режим работы SELinux, используйте команду `setenforce` и соответствующий идентификатор. Следующая команда переведёт SELinux в режим `Permissive`:

```
# setenforce 0
```

Чтобы вернуться в режим `Enforcing`, дайте команду:

```
# setenforce 1
```

Изменение контекста

- `chcon` – изменяет контекст объекта:
`chcon -t system_u:object_r:etc_t /etc/testfile`
- Скопировать контекст с одного файла на другой можно командой:
`chcon -reference=/etc/passwd /etc/testfile`
- `restorecon` – восстановить контекст:
`restorecon /etc/passwd`

<http://edu.softline.ru/>



Резервное копирование

- Резервная копия – залог отказоустойчивой работы:
 - Резервирование должно производиться периодически.
 - Копии должны храниться в безопасном месте:
 - Сейф.
 - Банковская ячейка.
 - Обязательно должны производиться периодические проверки резервных копий и восстановления.
 - Автоматизация процедуры резервирования.

<http://edu.softline.ru/>



Пожалуй, первое, что должен организовать системный администратор – систему резервного копирования. Надёжная система должна отвечать следующим требованиям:

- Отказоустойчивость
- Обеспечение целостности архивируемых данных
- Высокий уровень утилизации свободного пространства
- Малое время простоя при восстановлении
- "Гибкое" восстановление данных
- Централизованное управление и возможность работы по сети

Многие администраторы создают собственные сценарии резервного копирования. Но не всегда такие методы являются эффективными и надёжными. Поэтому, если данные, хранящиеся на серверах в вашей сети, являются ценными, то рассмотрите вариант использования таких программ как Bacula или Amanda.

Bacula

- Bacula – программный комплекс для построения централизованной системы резервирования информации:
 - **Director Daemon** – сервис, управляющий всеми заданиями.
 - **File Daemon** – сервис, выполняющий задания "на месте".
 - **Storage Daemon** – сервис, выполняющий запись на устройство хранения.
 - **Console** – интерактивное средство работы с director daemon.

<http://edu.softline.ru/>



Основное достоинство Bacula в его модульности. Каждый модуль выполняет свою функцию, не вмешиваясь в работу других:

Director daemon – основное средство управления всеми членами системы резервного копирования. Должен быть установлен на центральном сервере. Чтобы начать управлять director daemon, администратор должен подключиться к нему с помощью Bacula Console. Также существует программа Bacula Tray-Monitor, предназначенная для мониторинга процесса резервирования и своевременного реагирования на возникающие проблемы. Именно Director daemon является центральным звеном системы, раздавая задания членам и контролируя ход выполнения резервирования.

File daemon – это чернорабочий, который находится непосредственно на том сервере, файлы которого требуется сохранять. Реализация File Daemon существует не только под ОС Linux, но и под ОС Windows, что делает Bacula незаменимым помощником администратора даже в гетерогенных средах. Director Daemon связывается с File Daemon и даёт ему задание, которое будет незамедлительно выполнено.

Storage daemon – данный сервис выполняет одну из важнейших функций – непосредственную запись на устройство хранения. Storage Daemon позволяет производить запись не только на жесткие диски, но и на ленточные накопители, строго следя за остающимся свободным местом и предупреждая администратора о необходимости вставить новую кассету, параллельно снабдив инструкциями о том, как её нужно промаркировать.



ПРИМЕЧАНИЕ:

Полную документацию вы можете получить на официальном сайте проекта:
<http://www.bacula.org/>

Лабораторная работа.

Лабораторная работа 7

Упражнения:

1. Установка и настройка централизованного ведения журналов при помощи rsyslog и LogAnalyzer.
2. Резервное копирование и аварийное восстановление системы при помощи ReaR.

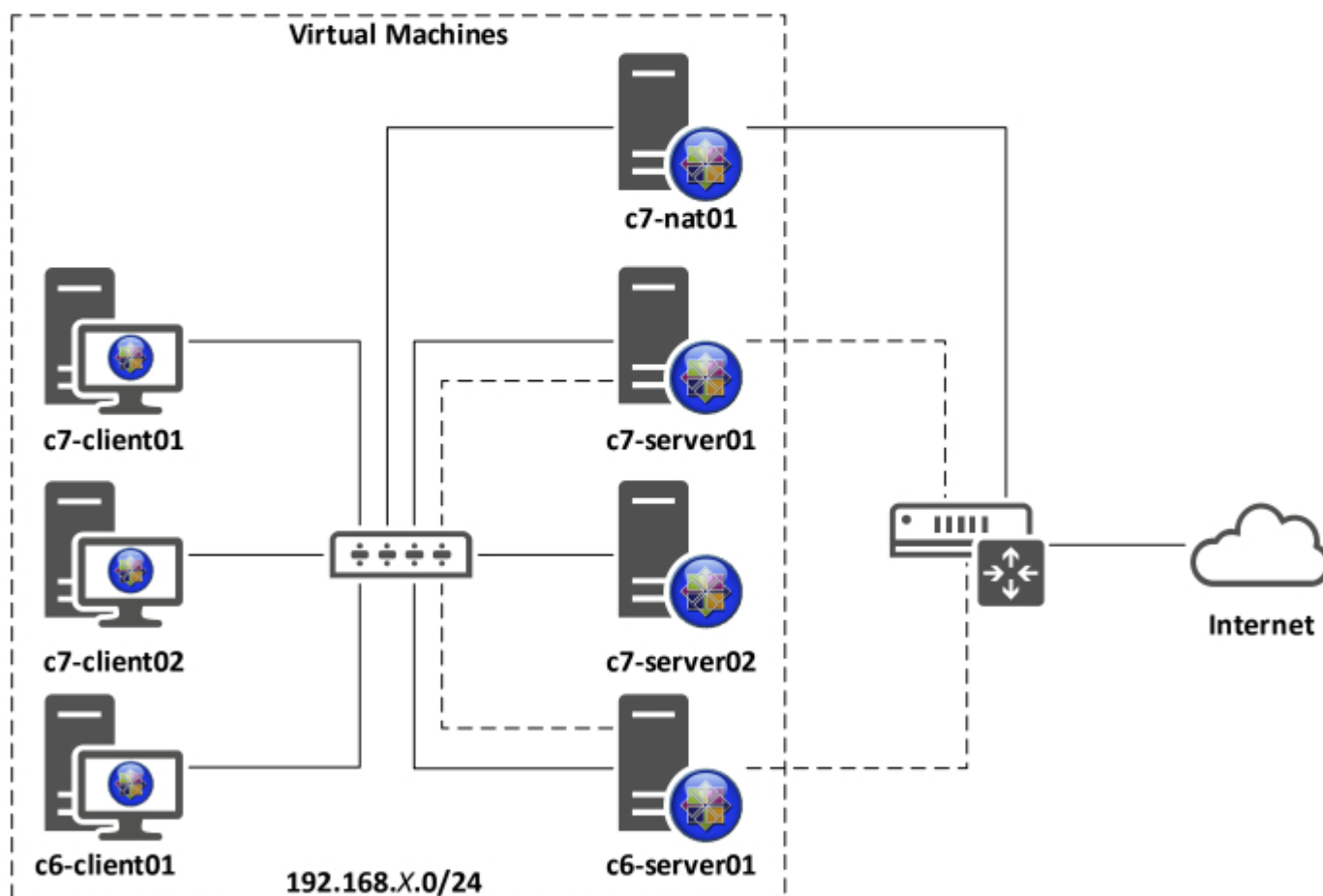
Необходимые машины:

- C7-NAT01.
- C7-SERVER01.
- C7-SERVER02.
- C7-CLIENT01.

Учетные данные:

- Пользователь: root.
- Пароль: Pa\$\$w0rd.

<http://edu.softline.ru/>





Rsyslog – это утилита с открытым исходным кодом, используемая в UNIX и UNIX-подобных операционных системах для перенаправления сообщений журналов по сети. Она использует базовый протокол syslog, расширяя его при помощи фильтра на основе содержимого, богатых возможностей фильтрации, гибкости настройки и использования протокола TCP для транспорта. Для упрощения процедуры просмотра, фильтрации и построения отчетов мы будем использовать **LogAnalyzer**, а для хранения информации **MariaDB**.

Упражнение – 1: Установка и настройка централизованного ведения журналов при помощи rsyslog и LogAnalyzer.

Выполняйте данное упражнение на машине **c7-server02**.

Задание 1: Предварительная настройка сервера:

1. Установите Apache HTTP Server.

```
# yum install httpd -y
```

2. Запустите службу **httpd** и настройте ее автоматический запуск.

```
# systemctl start httpd
# systemctl enable httpd
```

3. Разрешите TCP-порт 80 в **firewalld**.

```
# firewall-cmd --permanent --add-service=http
# firewall-cmd --reload
```

4. Установите **MariaDB** (**mariadb-server** и **mariadb**).

```
# yum install mariadb-server mariadb -y
```

5. Запустите службу **MariaDB** и настройте ее автоматический запуск.

```
# systemctl start mariadb
# systemctl enable mariadb
```

6. Настройте пароль для MySQL root.

```
# mysql_secure_installation
```

Примеры ответов на вопросы мастера для типовой настройки:

Вопрос	Ввод	Клавиша
Enter current password for root (enter for none):		<Enter>
Set root password? [Y/n]	Y	<Enter>
New password:	Pa\$\$w0rd	<Enter>
Remove anonymous users? [Y/n]	Y	<Enter>
Disallow root login remotely? [Y/n]	Y	<Enter>
Remove test database and access to it? [Y/n]	Y	<Enter>
Reload privilege tables now? [Y/n]	Y	<Enter>

7. Установите PHP (**php**, **php-mysql**, **php-gd** и **php-pear**).

```
# yum install php php-mysql php-gd php-pear -y
```

8. Перезапустите сервер Apache HTTP Server.

```
# systemctl restart httpd
```

9. Переведите SELinux в режим **permissive**.

```
# vi /etc/selinux/config
```

В строке 6 укажите значение **permissive**.

```
SELINUX=permissive
```

10. Перезагрузите машину **c7-server02**.

```
# systemctl reboot
```

11. После перезагрузки проверьте, что режим работы SELinux переключился на Permissive.

```
# getenforce
```

Задание 2: Установка и настройка rsyslog.

1. Установите **rsyslog** и все связанные пакеты (**rsyslog***)

```
# yum install rsyslog* -y
```

2. Запустите службу **rsyslog** и настройте ее автоматический запуск.

```
# systemctl restart rsyslog
```

```
# systemctl is-enabled rsyslog
```

3. Настройте параметры базы данных для **rsyslog**, изменив файл **/usr/share/doc/rsyslog-версия/mysql-createDB.sql**.

```
# vi /usr/share/doc/rsyslog-7.4.7/mysql-createDB.sql
```

Измените имя базы данных на **rsdb**.

В строке 1 и 2 замените **Syslog** на **rsdb**

```
CREATE DATABASE rsdb;
USE rsdb;
```

4. Создайте базу данных для **rsyslog**, воспользовавшись паролем, определенным ранее (Например: **Pa\$\$w0rd**).

```
# mysql -u root -p < /usr/share/doc/rsyslog-7.4.7/mysql-createDB.sql
```

5. Подключитесь к MariaDB и выполните дополнительную настройку, воспользовавшись паролем, определенным ранее (Например: **Pa\$\$w0rd**).

```
# mysql -u root -p
```

Проверьте, что база данных **rsdb** создана.

```
> show databases;
```

Создайте пользователя **rsdbuser** с паролем **Pa\$\$w0rd**.

```
> GRANT ALL ON rsdb.* TO rsdbuser@localhost IDENTIFIED BY 'centos';
> flush privileges;
> quit
```

6. Настройте интеграцию `journald` с `rsyslog`:

Для этого необходимо отредактировать конфигурационный файл `/etc/rsyslog.conf`, добавив строку «`$OmitLocalLoggin off`» между строками 9 «`$ModLoad imuxsock`» и 10 «`$ModLoad imjournal`».

```
$ModLoad imuxsock # provides support ...
$OmitLocalLoggin off
$ModLoad imjournal # provides access ...
```

Убедитесь, что файл `/etc/rsyslog.d/listen.conf` содержит следующую строку:

```
# cat /etc/rsyslog.d/listen.conf
$SystemLogSocketName /run/systemd/journal/syslog
```

7. Отредактируйте конфигурационный файл `rsyslog`.

```
# vi /etc/rsyslog.conf
```

Раскомментируйте строки 16,17 и 20,21:

```
# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514
```

В строку 22 и ниже добавьте загрузку модуля `ommysql`, параметры подключения к базе данных и разрешение удаленного соединения через 514/TCP и 514/UDP для 127.0.0.1 и 192.168.X.0/24.

```
$ModLoad ommysql
$ModLoad ommysql
*. * :ommysql:127.0.0.1,rsdb,rsdbuser,centos
$AllowedSender UDP, 127.0.0.1, 192.168.0.0/24
$AllowedSender TCP, 127.0.0.1, 192.168.0.0/24
```

12. Разрешите TCP-порт и UDP-порт 514 в `firewalld`.

```
# firewall-cmd --add-port=514/tcp --permanent
# firewall-cmd --add-port=514/udp --permanent
# firewall-cmd --reload
```

13. Перезагрузите машины `c7-server01`Задание 3: Установка и настройка LogAnalyzer.1. Загрузите `loganalyzer`, по ссылке, взятой у преподавателя

```
# wget http://172.20.70.251/files/loganalyzer/loganalyzer-3.6.5.tar.gz
```

ПРИМЕЧАНИЕ:

Также LogAnalyzer доступен по ссылке: <http://loganalyzer.adiscon.com/downloads/>

2. Распакуйте архив.

```
# tar zxvf loganalyzer-3.6.5.tar.gz
```

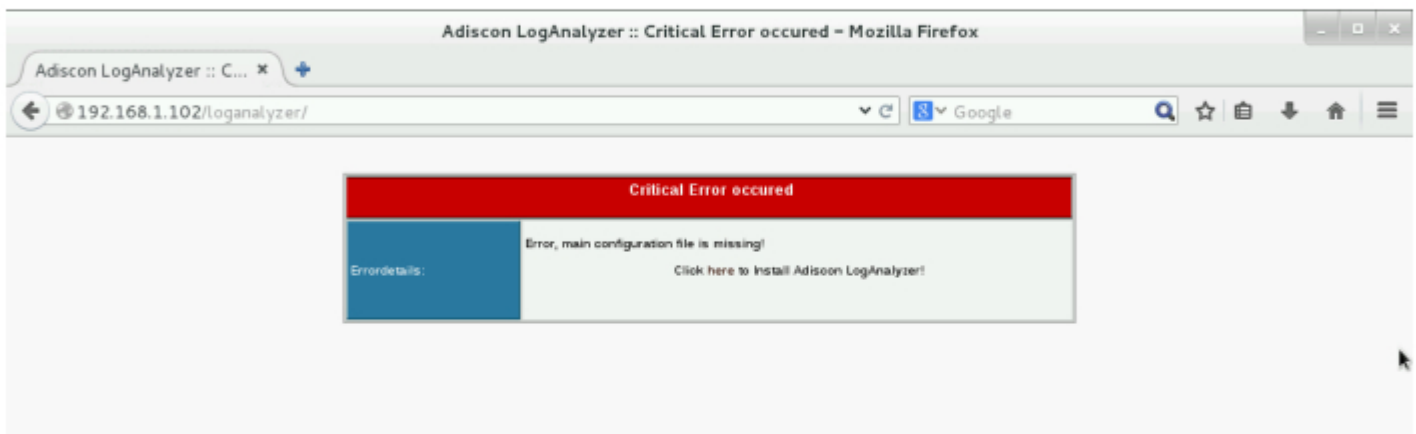
3. Переместите каталог `loganalyzer-3.6.5/src/` в `/var/www/html/loganalyzer` и содержимое каталога `loganalyzer-3.6.5/contrib/` в `/var/www/html/loganalyzer/`.

```
# mv loganalyzer-3.6.5/src/ /var/www/html/loganalyzer
# mv loganalyzer-3.6.5/contrib/* /var/www/html/loganalyzer/
```

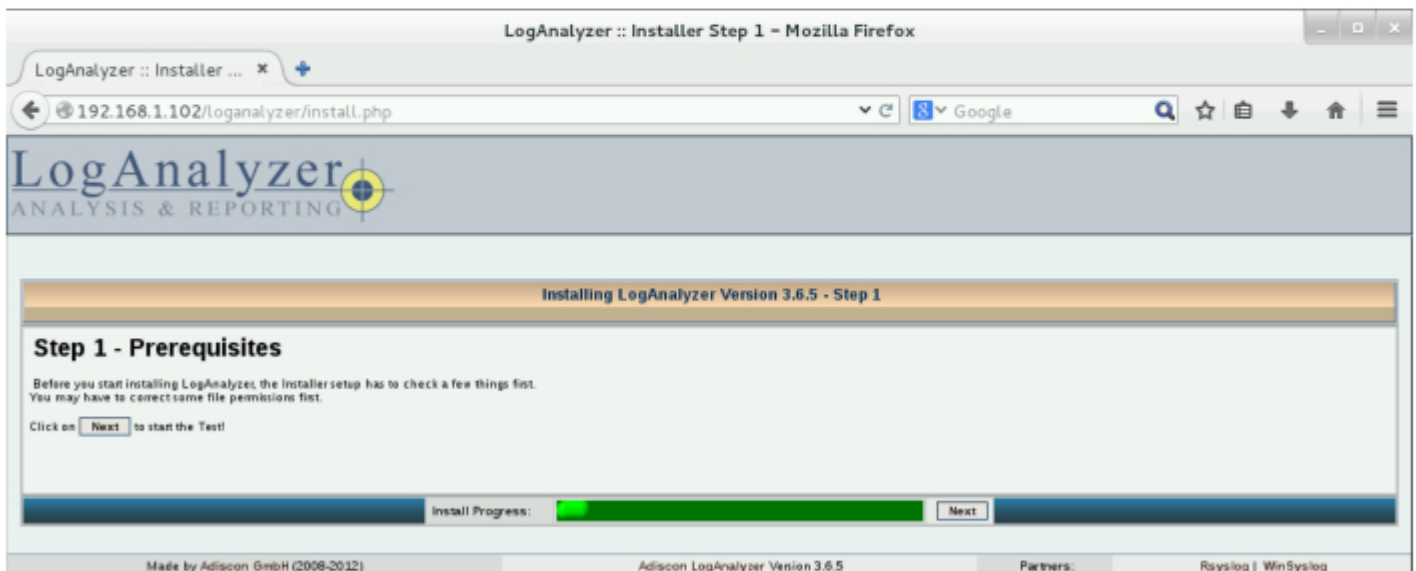
4. Установите право на исполнение и выполните конфигурационный скрипт `configure.sh`.

```
# cd /var/www/html/loganalyzer/
# chmod +x configure.sh secure.sh
# ./configure.sh
```

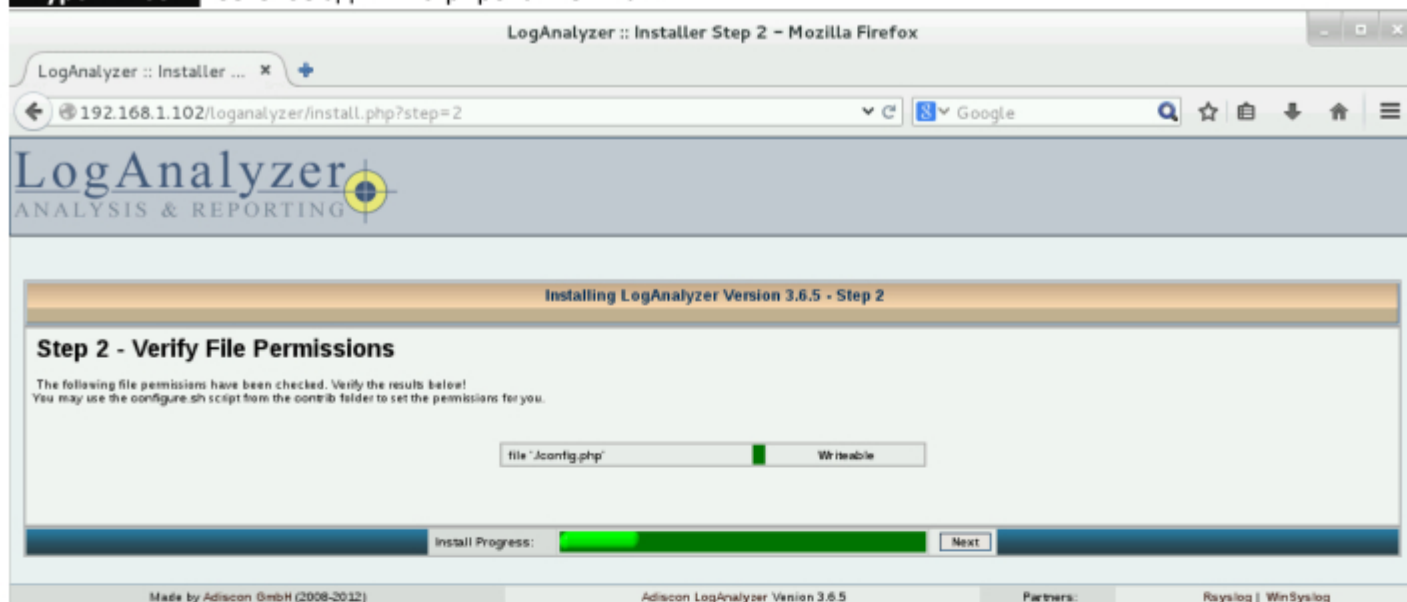
5. Проверьте доступ к LogAnalyzer при помощи браузера (`http://192.168.0.102/loganalyzer`)



6. Получив ошибку в которой указано, что отсутствует основной конфигурационный файл нажмите на ссылку **here** (здесь).
7. В шаге **Step 1 – Prerequisites** (Шаг 1 – Предварительные требования) нажмите **Next** (Далее).

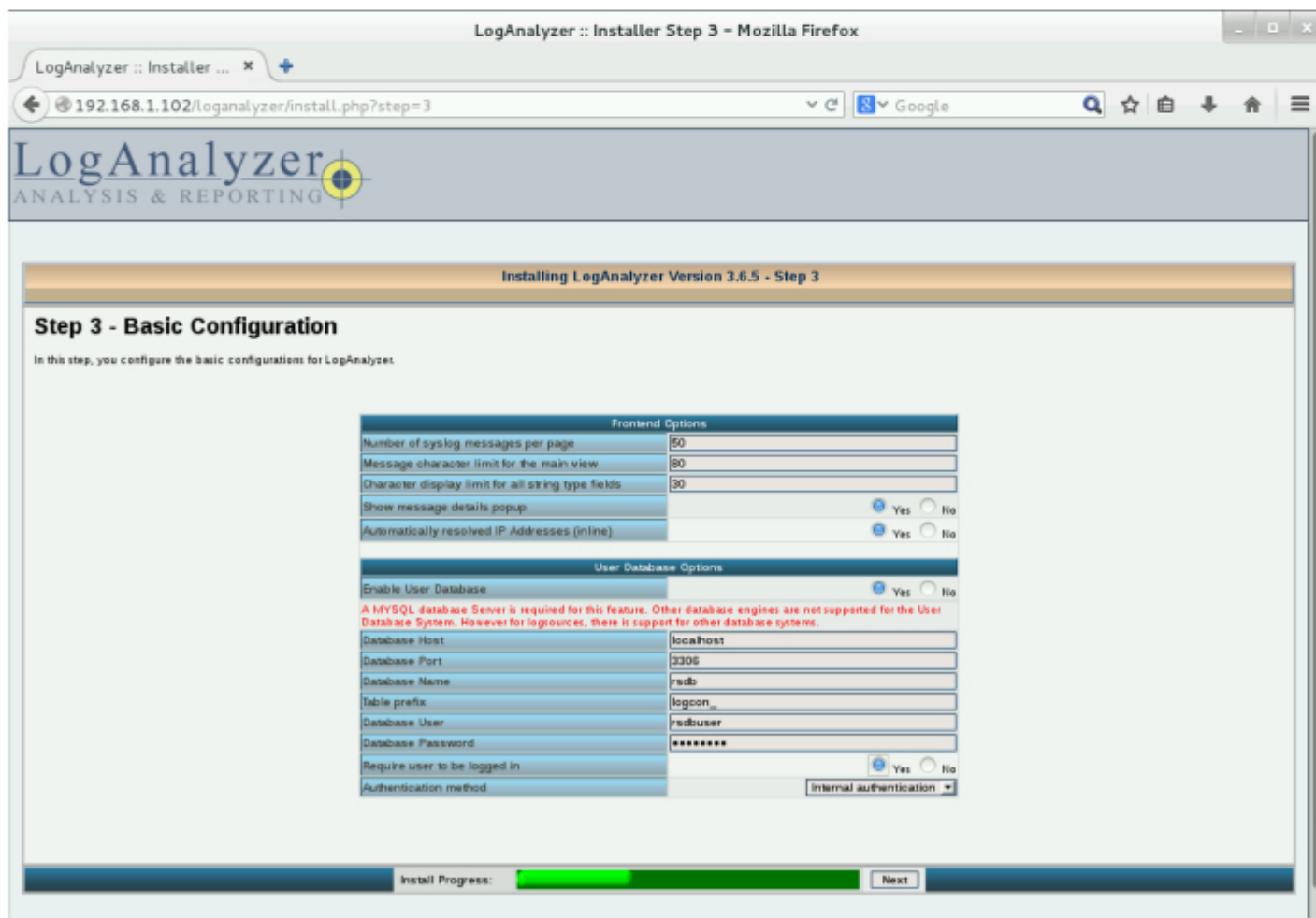


8. В шаге **Step 2 – Verify File Permissions** (Шаг 2 – Проверка прав файла) нажмите **Next** (Далее).



9. В шаге **Step 3 – Basic Configuration** (Шаг 3 – Базовая настройка) выполните настройку опций пользовательской базы данных, согласно таблице и нажмите **Next** (Далее).

Опция	Значение
Enable User Database	Yes
Database Name	rsdb
Database User	rsdbuser
Database Password	centos
Require user to be logged in	Yes



10. В шаге **Step 4 – Create Tables** (Шаг 4 – Создание таблиц) нажмите **Next** (Далее).



11. В шаге **Step 5 – Check SQL Results** (Шаг 5 – Проверка результатов SQL) нажмите **Next** (Далее).



12. В шаге **Step 6 – Creating the Main Useraccount** (Шаг 6 – Создание основной учетной записи пользователя) создайте учетную запись, согласно данным в таблице и нажмите **Next** (Далее).

Опция	Значение
Username	Student
Password	Pa\$\$w0rd

LogAnalyzer :: Installer Step 6 - Mozilla Firefox

LogAnalyzer :: Installer ...

192.168.1.102/loganalyzer/install.php?step=6

LogAnalyzer
ANALYSIS & REPORTING

Installing LogAnalyzer Version 3.6.5 - Step 6

Step 6 - Creating the Main Useraccount

You are now about to create the initial LogAnalyzer User Account.
This will be the first administrative user, which will be needed to login into LogAnalyzer and access the Admin Center!

Create User Account	
Username	student
Password	*****
Repeat Password	*****

Install Progress:

Made by Adiscon GmbH (2008-2012) Adiscon LogAnalyzer Version 3.6.5 Partners: Rsyslog | WinSyslog

13. В шаге **Step 7 – Create the first source for syslog messages** (Шаг 7 – Создание первого источника для сообщений syslog) выполните настройку опций пользовательской базы данных, согласно таблице и нажмите **Next** (Далее).

Опция	Значение
Source Type	MYSQL Native
Select View	Syslog Fields
Database Name	rsdb
Database User	rsdbuser
Database Password	centos
Enable Raw Counting	Yes

LogAnalyzer :: Installer Step 7 - Mozilla Firefox

LogAnalyzer :: Installer ...

192.168.1.102/loganalyzer/install.php?step=7

LogAnalyzer
ANALYSIS & REPORTING

Installing LogAnalyzer Version 3.6.5 - Step 7

Successfully created User 'student'.

Step 7 - Create the first source for syslog messages

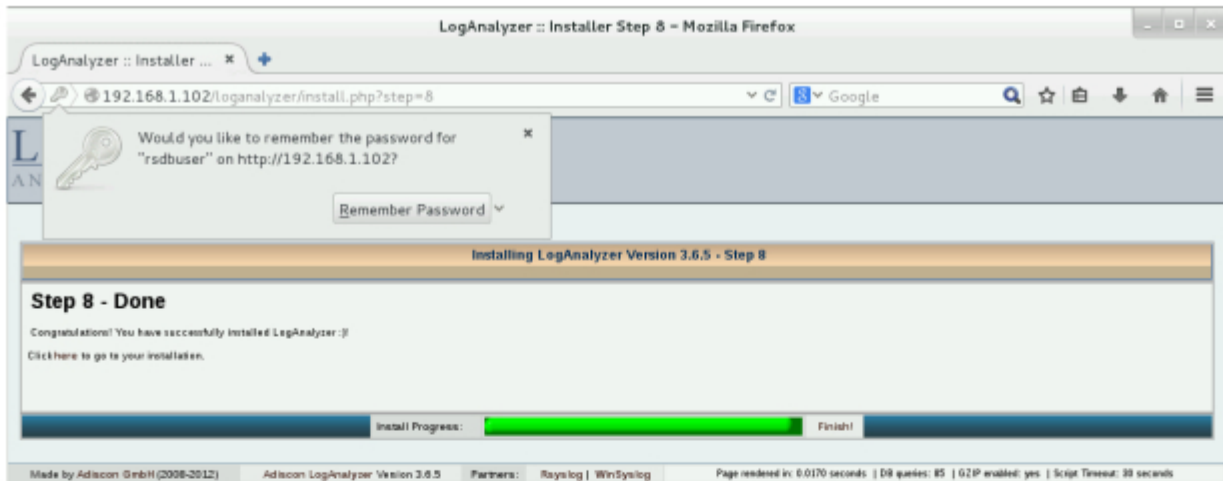
First Syslog Source	
Name of the Source	My Syslog Source
Source Type	MYSQL Native
Select View	Syslog Fields

Database Type Options	
Table type	MonitorWare
Database Host	localhost
Database Name	rsdb
Database Tablename	syslogevents
Database User	rsdbuser
Database Password	*****
Enable Raw Counting	☒ Yes ☐ No

Install Progress:

Made by Adiscon GmbH (2008-2012) Adiscon LogAnalyzer Version 3.6.5 Partners: Rsyslog | WinSyslog

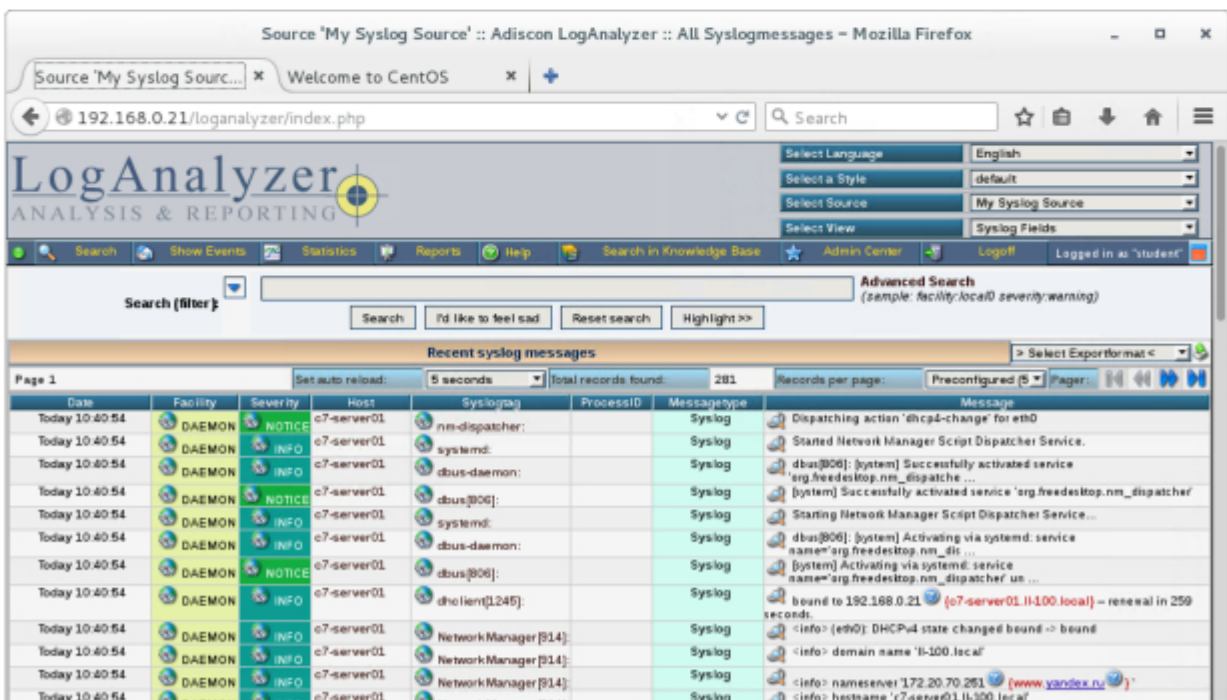
14. В шаге **Step 8 – Done** (Шаг 8 – Выполнено) нажмите **Finish** (Завершить).



15. На стартовой странице **LogAnalyzer** введите логин и пароль определенные ранее (Например: **student** с паролем **Pa\$\$w0rd**).



16. Отобразится экран основной консоли и выведет журналы **c7-server01**.



**ПРИМЕЧАНИЕ:**

Если основное окно LogAnalyzer не отображает информацию попробуйте перезапустить **c7-server01**.

Задание 4: Настройка клиентов rsyslog.

1. Подключитесь к машине c6-server01 и убедитесь, что **rsyslog** установлен.

```
# yum install rsyslog -y
Loaded plugins: fastestmirror
Setting up Install Process
Loading mirror speeds from cached hostfile
 * base: mirror.corbina.net
 * extras: mirror.nsu.ru
 * updates: mirror.nsu.ru
Package rsyslog-5.8.10-10.el6_6.x86_64 already installed and latest version
Nothing to do
```

2. Добавьте новое правило «*. * @192.168.0.21», пересылающее все записи на c7-server01 в 36-ую строку файла **/etc/rsyslog.conf**.

```
#### RULES ####
```

```
*. * @192.168.0.21
# Log all kernel messages to the console.
```

3. Перезапустите службу rsyslog и настройте автоматический запуск при старте системы:

```
# service rsyslog start
# chkconfig rsyslog on
```

4. Перейдите к экрану основной консоли LogAnalyzer и убедитесь, что сообщения от **c6-server01** стали поступать в журнал.

The screenshot shows the LogAnalyzer web interface in a Mozilla Firefox browser. The address bar shows the URL: 192.168.0.21/loganalyzer/index.php?filter=datelastx%3A2&search=Search. The interface includes a search bar with the filter 'datelastx:2' and a search button. Below the search bar, there is a table titled 'Recent syslog messages' showing logs from c6-server01. The table has columns for Date, Facility, Severity, Host, Syslogmsg, ProcessID, MessageType, and Message. The messages include DHCP-related events and system logs.

Date	Facility	Severity	Host	Syslogmsg	ProcessID	MessageType	Message
Today 11:04:51	DAEMON	INFO	c6-server01	dhclient(1147)		Syslog	bound to 192.168.0.23 (c6-server01.ll-100.local) - renewal in 241 seconds.
Today 11:04:49	DAEMON	INFO	c6-server01	dhclient(1147)		Syslog	DHCPACK from 192.168.0.1 (c7-rs01.ll-100.local) (xid=0x47b51c6d)
Today 11:04:49	DAEMON	INFO	c6-server01	dhclient(1147)		Syslog	DHCPREQUEST on eth0 to 192.168.0.1 (c7-rs01.ll-100.local) port 67 (xid=0x47b51c6d)
Today 11:04:05	DAEMON	NOTICE	c7-server01	nm-dispatcher		Syslog	Dispatching action 'dhcp4-change' for eth0
Today 11:04:05	DAEMON	INFO	c7-server01	systemd		Syslog	Started Network Manager Script Dispatcher Service.

5. Перейдите на машину c7-server02 и установите **rsyslog** и все связанные пакеты (**rsyslog***)

```
# yum install rsyslog* -y
```

6. Запустите службу **rsyslog** и настройте ее автоматический запуск.

```
# systemctl restart rsyslog
```

```
# systemctl enable rsyslog
```

7. Настройте интеграцию `journald` с `rsyslog`:

Для этого необходимо отредактировать конфигурационный файл `/etc/rsyslog.conf`, добавив строку «`$OmitLocalLoggin off`» между строками 9 «`$ModLoad imuxsock`» и 10 «`$ModLoad imjournal`».

```
$ModLoad imuxsock # provides support ...
$OmitLocalLoggin off
$ModLoad imjournal # provides access ...
```

Убедитесь, что файл `/etc/rsyslog.d/listen.conf` содержит следующую строку:

```
# cat /etc/rsyslog.d/listen.conf
$SystemLogSocketName /run/systemd/journal/syslog
```

8. Добавьте новое правило «`*.* @192.168.0.21`», пересылающее все записи на `c7-server01` в 36-ую строку файла `/etc/rsyslog.conf`.

```
#### RULES ####

*.* @192.168.0.21
# Log all kernel messages to the console.
```

9. Перезапустите службу `rsyslog` и настройте автоматический запуск при старте системы:

```
# systemctl restart rsyslog
# systemctl enable rsyslog
```

10. Перезагрузите машину `c7-server02`.

11. Перейдите к экрану основной консоли LogAnalyzer и убедитесь, что сообщения от `c7-server02` стали поступать в журнал.

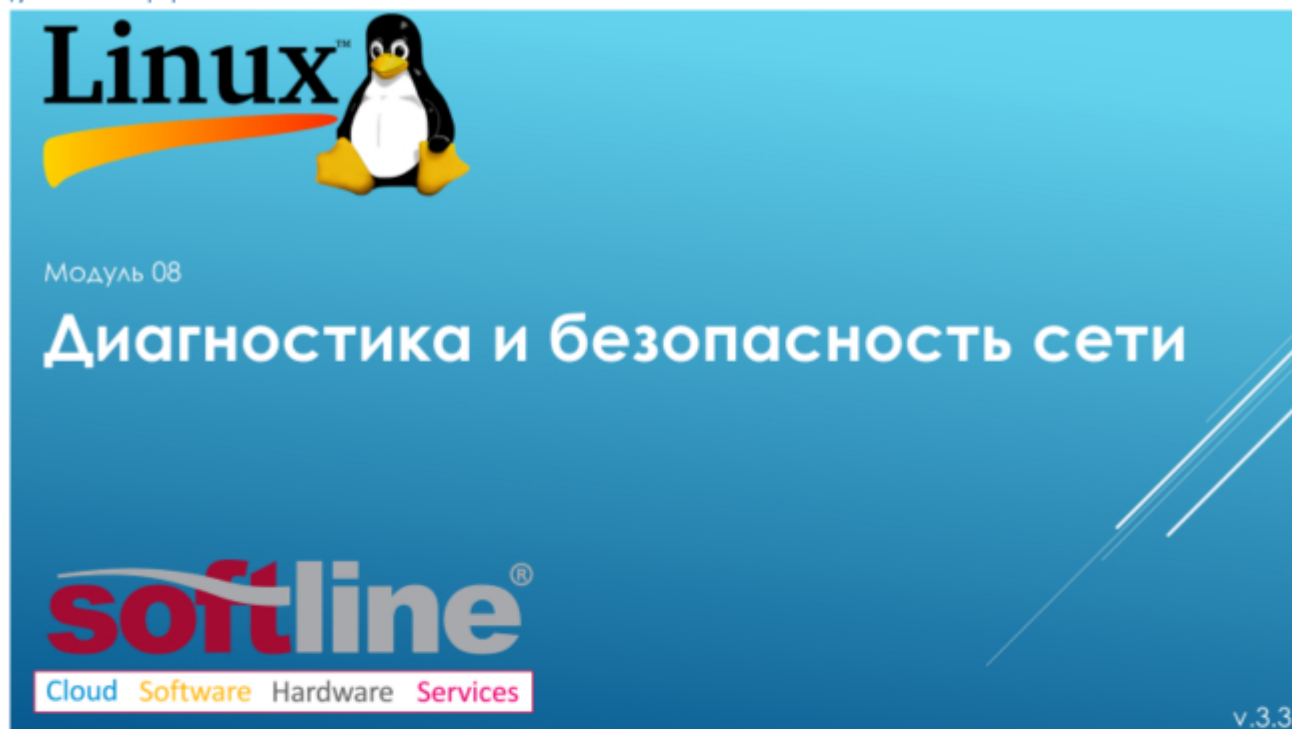
The screenshot shows the LogAnalyzer web interface in a Mozilla Firefox browser. The address bar shows the URL `192.168.0.21/loganalyzer/index.php?filter=datelastx%3A2&search=Search`. The interface includes a search bar with the filter `datelastx:2` and a table of recent syslog messages.

Date	Facility	Severity	Host	Syslogtag	ProcessID	Message type	Message
Today 11:22:07	DAEMON	INFO	c7-server02	systemd:		Syslog	Time has been changed
Today 11:22:04	DAEMON	WARNING	c7-server02	chronyd[763]:		Syslog	System clock wrong by 1.920051 seconds, adjustment started
Today 11:22:04	DAEMON	INFO	c7-server02	chronyd[763]:		Syslog	Selected source 78.140.251.2
Today 11:22:04	SECURITY	NOTICE	c7-server02	polkitd[1047]:		Syslog	Unregistered Authentication Agent for unix-process:1504:10063 (system bus name ...)
Today 11:22:04	DAEMON	INFO	c7-server02	systemd:		Syslog	Started LSB: Bring up/down networking.
Today 11:22:04	DAEMON	INFO	c7-server02	chronyd[763]:		Syslog	Source 95.104.192.10 (95-104-192-10 netroad.ru) online
Today 11:22:04	DAEMON	INFO	c7-server02	chronyd[763]:		Syslog	Source 213.141.135.201 (gateway.vyiffe.ru) online
Today 11:22:04	DAEMON	INFO	c7-server02	chronyd[763]:		Syslog	Source 45.254.216.8 (ntp1.lvlan.net) online
Today 11:22:04	DAEMON	INFO	c7-server02	chronyd[763]:		Syslog	Source 78.140.251.2 online
Today 11:22:04	DAEMON	INFO	c7-server02	network:		Syslog	[OK]
Today 11:22:04	DAEMON	INFO	c7-server02	network:		Syslog	Bringing up interface eth0: Connection successfully activated (D-Bus active pa ...)
Today 11:22:04	DAEMON	NOTICE	c7-server02	nm-dispatcher:		Syslog	Dispatching action 'up' for eth0

Упражнение 2 – Резервное копирование и аварийное восстановление системы при помощи ReaR.

Возьмите ссылку на лабораторную работу у инструктора.

Модуль 08: Диагностика и безопасность сети



В модуле представлены:

- Всегда под угрозой.
- Маршрутизация.
- Сетевой фильтр **iptables**.
- Цепочки **iptables**.
- Состояния пакетов.
- Команды **iptables**.
- Критерии **iptables**.
- Действия **iptables**.
- Протокол-ориентированные критерии.
- Дополнительные параметры.
- Модули **iptables**.
- Сетевая трансляция.
- Средства диагностики.

Всегда под угрозой

- **Любой сетевой узел всегда находится под пристальным вниманием:**
 - Сканеры открытых портов.
 - Сканеры уязвимостей.
 - Перебор паролей к открытым службам.
- **Необходимость противодействия:**
 - Выработка политики безопасности.
 - Использование программных (аппаратных) комплексов для защиты сетевых узлов.
 - Постоянный автоматизированный мониторинг.

<http://edu.softline.ru/>



Сетевой фильтр **iptables**

- Проект был основан в 1998 году Расти Расселом (Rusty Russel).
- Является продолжением проекта **ipchains**.
- Требуется поддержка в ядре.
- **iptables** – утилита для управления сетевым брандмауэром **netfilter**:
 - Создание и управление правилами фильтрации.
 - Правила загружаются в пространство ядра.
 - Обеспечение сетевой трансляции NAT.
 - Возможность расширения функционала с помощью модулей.
 - Возможность "отслеживания" пакетов.

<http://edu.softline.ru/>



Проект **iptables** стартовал в 1998 году и пришёл на смену таким проектам как **ipfwadm** (linux 2.0) и **ipchains** (linux 2.2). На сегодняшний день развитием **netfilter** занимается одноимённая команда (<http://netfilter.org>). Основным новшеством, появившемся в **iptables** стали 4 таблицы, предназначенные для фильтрации, модификации и трансляции трафика.

iptables – это система управления пакетным фильтром **netfilter**, несмотря на многочисленные случаи отождествления этих понятий. Брандмауэр **netfilter** работает в пространстве ядра, что делает его очень производительным.

Проект **netfilter** предполагает прохождение сетевого трафика через цепочки, состоящие из наборов правил. Каждое правило, в свою очередь, состоит из различных критериев и действий, применяемых к сетевым пакетам. По мере прохождения пакета через цепочку, к нему "примеряются" все критерии и при совпадении условий, выполняется определённое действие. Действия могут быть самыми разнообразными, начиная от блокирования пакета и заканчивая подсчётом для последующей обработки с целью сбора статистических данных. По своей идеологии **iptables**, строго говоря, не является "чистым" фильтром, скорее, это комплекс средств для обеспечения управления трафиком.

Ещё одним достоинством системы **netfilter** является возможность создания собственных цепочек и перенаправление на них сетевого трафика.

Цепочки **iptables**

- **iptables** состоит из 5 цепочек:
 - **PREROUTING** – первоначальная обработка всех входящих пакетов.
 - **INPUT** – входящие пакеты, адресованные данному узлу.
 - **FORWARD** – входящие транзитные пакеты, адресованные другим узлам.
 - **OUTPUT** – исходящие пакеты от данного узла.
 - **POSTROUTING** – окончательная обработка всех исходящих пакетов.

<http://edu.softline.ru/>



Сетевой фильтр netfilter состоит из таблиц и цепочек. Каждая таблица имеет определённый набор цепочек.

Таблица **mangle** предназначена для внесения изменений в заголовки пакетов.

Таблица **nat** предназначена для преобразования сетевых адресов, Network Address Translation

Таблица **filter** используется для фильтрации пакетов.

Рассмотрим порядок прохождения таблиц и цепочек транзитными пакетами:

(!)

Транзитным называется пакет, пунктом назначения или источника которого не является узел, на который этот пакет прибыл. Каждый такой пакет попадает сначала на входной сетевой интерфейс после чего перемещается в цепочки **PREROUTING** таблиц **mangle** и **nat**. На этом этапе системой ещё не принято решение о дальнейшей маршрутизации пакета. После принятия решения о маршрутизации, когда становится известно, что пакет является транзитным, он попадает в цепочки **FORWARD** таблиц **mangle** и **filter**. На данном этапе возможно внесение изменений в заголовок пакета и его фильтрация. В конце концов пакет попадает в цепочки **POSTROUTING** таблиц **mangle** и **nat**.

Рассмотрим этапы, которые проходит пакет, отправленный локальным приложением.

Рассмотрим этапы, которые проходит пакет, отправленный локальному приложению.

Состояния пакетов

- Существует 4 основных состояния сетевых пакетов:
 - **NEW** – первый пакет для заданного соединения, как правило, в случае TCP-соединения – SYN-пакет.
 - **RELATED** – соединение, относящееся к уже установленному соединению с типом **ESTABLISHED**. Пример - FTP-соединение.
 - **ESTABLISHED** – установленное соединение, означает, что оба узла успешно обменялись инициализационными пакетами.
 - **INVALID** – пакеты, состояние которых определить невозможно.

<http://edu.softline.ru/>



iptables имеет в своём составе механизм определения состояний "**state machine**", основной задачей которого является трассировка соединений с помощью трассировщика **conntrack**. Трассировщик отслеживает все соединения и ведёт их учёт в таблице, доступной в файле **/proc/net/ip_conntrack**.

Существует 4 основных состояния пакетов:

- **NEW** – состояние, указывающее на то, что это первый пакет для соединения. Например SYN-пакет является первым в соединении и получит состояние **NEW**.
- **RELATED** – данное состояние говорит о том, что пакет относится к другому соединению типа **ESTABLISHED**. Например FTP-соединения создают новые соединения в рамках существующего, эти пакеты получают состояние **RELATED**.
- **ESTABLISHED** – пакет не является первым в соединении.
- **INVALID** – пакет не может быть идентифицирован.

Команды **iptables**

- Первое, что нужно указать программе **iptables** – команда:
 - **-A** – добавить правило в конец цепочки:

```
# iptables -A INPUT
```
 - **-D** – удаление правила из цепочки:
 - По набору критериев.
 - По порядковому номеру.
 - **-R** – заменить правило в цепочке.
 - **-I** – вставить правило на определённое место в цепочке.
 - **-L** – вывести список правил в данной цепочке:

```
# iptables -L INPUT
```
 - **-F** – сброс всех правил из указанной цепочки.
 - **-P** – определение политики по умолчанию.

<http://edu.softline.ru/>



Каждое правило **iptables** состоит из команды, имени цепочки и имени таблицы:

```
iptables COMMAND -t TABLE
```

Пакет проходит проверку в цепочке по всем правилам по очереди, следовательно, очень важно определить верный порядок следования цепочек. Так как проверка идёт до первого попадания. Следовательно, если вы запрещаете прохождение пакета после того, как разрешили это, то до запрещающих правил пакет даже не дойдёт. Очень важно аккуратно использовать команды **iptables**:

- **-A** – добавит ваше правило в самый конец цепочки. Данная команда используется чаще всего.

```
iptables -A INPUT
```

- **-I** – вставит правило на определённую позицию, следовательно, необходимо указать номер позиции.

```
iptables -I INPUT 15
```

- **-D** – удаляет правило. Чтобы команда **iptables** могла идентифицировать удаляемое правило, необходимо указать либо его порядковый номер, либо все критерии.

```
iptables -D 13
```

- **-R** – заменит правило новым. Также требуется указание порядкового номера заменяемого правила.
- **-L** – произведёт вывод всех правил указанной цепочки. Желательно использовать вместе в параметром **--numeric** для предотвращения преобразования имён. Также очень рекомендуется использование параметра **--line-numbers** для вывода порядковых номеров правил в цепочках.

В случае, если таблица не указана, будет использована таблица **filter**.

Команды являются лишь средством управления правилами. Но как же создать само правило? Для этого нам понадобятся критерии и действия.

Критерии **iptables**

- Правило состоит из критериев:
 - Протокол (**-p**) – определяет сетевой протокол.
 - Интерфейс (**-i/-o**) – определяет входящий и исходящий интерфейсы.
 - Адрес источника (**-s**) – определяет ip-адрес источника.
 - Порт источника (**--sport**) – определяет порт источника.
 - Адрес назначения (**-d**) – определяет ip-адрес назначения.
 - Порт назначения (**--dport**) – определяет порт назначения.

<http://edu.softline.ru/>



Каждое правило должно четко идентифицировать необходимый пакет, в этом вам помогут критерии **iptables**. Создадим правило, которое будет применять к пакетам, соответствующим критериям:

- Входящий пакет на интерфейсе **eth0**
- К пакету будут применять фильтрующие действия, то есть будет использована таблица **filter**
- Адрес источника должен принадлежать диапазону **192.168.1.0/24**
- Порт назначения – **23 (telnet)**

Очень важно полностью определить правило перед тем как его реализовывать в **iptables**, что мы и сделали. Теперь необходимо перевести критерии с русского языка на язык **iptables**:

- Если мы хотим применить фильтрующие действия к входящему пакету, то мы должны использовать таблицу **filter** и цепочку **INPUT**:

```
iptables -A INPUT -t filter
```

- Пакет должен быть получен на интерфейсе **eth0**:

```
iptables -A INPUT -t filter -i eth0
```

- Адрес источника пакета должен принадлежать диапазону **192.168.1.0/24**:

```
iptables -A INPUT -t filter -i eth0 -s 192.168.1.0/24
```

- Порт назначения должен быть **23**. Если вы в критериях указываете сетевой порт, то обязательным является указание протокола:

```
iptables -A INPUT -t filter -p tcp -i eth0 -s 192.168.1.0/24 --dport 23
```

Действия iptables

- Действия определяют дальнейшую судьбу пакета:
 - **ACCEPT** – пропустить пакет.
 - **DROP** – запретить пакет.
 - **REJECT** – запретить пакет и отправить icmp-уведомление.
 - **LOG** – производить журналирование пакетов и событий (не прерывает цепочку).
 - **DNAT** – произвести DNAT-трансляцию.
 - **SNAT** – произвести SNAT-трансляцию.
- **-j** – указать действие.

<http://edu.softline.ru/>



Перед определением правил необходимо применить политику по умолчанию. Если пакет не будет отвечать ни одному критерию в цепочке, то к нему будет применено правило, определённое по умолчанию:

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Модифицируем созданное ранее правило и разрешим прохождение пакетов на 23 порт до telnet-сервера:

```
iptables -A INPUT -t filter -p tcp -i eth0 -s 192.168.1.0/24 --dport 23 -j ACCEPT
```

Но вы не должны забывать о том, что любое соединение носит двунаправленный характер и данное правило не будет иметь смысла без ещё одного:

```
iptables -A OUTPUT -t filter -p tcp -o eth0 -d 192.168.1.0/24 --sport 23 -j ACCEPT
```

Чтобы включить сетевую трансляцию необходимо использовать таблицу nat и действие SNAT, применимое в данной таблице.

```
iptables -A POSTROUTING -t nat -s 192.168.2.0/24 -d any/.0 -j SNAT --to-source 1.2.3.4
```

Разберём правило подробнее:

- Так как сетевая трансляция подразумевает подмену адреса источника из приватной сети в публичный адрес маршрутизатора, то применять трансляцию нужно перед самым выходом пакета из маршрутизатора. Поэтому мы используем цепочку **POSTROUTING**.
- Для сетевой трансляции предназначена таблица nat
- Необходимо указать критерии, по которым iptables будет определять, какие пакеты необходимо транслировать, а какие нет. Мы указываем, что пакеты, адресом источника которых является адрес, входящий в диапазон **192.168.2.0/24**. Также указываем, что адресом назначения должен быть адрес, входящий в диапазон **any/0**, что означает **0.0.0.0/0**.

- Действие **SNAT** включает изменение адреса источника на адрес, указанный в параметре `--to-source`, который, как правило, является адресом внешнего интерфейса маршрутизатора

Критерии протоколов

- Критерии протокола TCP:
 - `--tcp-flags` – проверка флагов, установленных в заголовке TCP-пакета.
 - `--syn` – инициализация соединения.
- Критерии протокола ICMP:
 - `--icmp-type` – определение типа icmp-сообщения.

<http://edu.softline.ru/>



В iptables существует возможность задания критериев, характерных для определённого протокола, при использовании таких критериев необходимо обязательно указывать и протокол. Это можно сделать двумя способами:

```
iptables -A INPUT -p tcp
iptables -A FORWARD -p udp
```

или

```
iptables -A INPUT -p 6
iptables -A FORWARD -p 17
```

Во втором случае были использованы идентификаторы протоколов которые можно узнать из файла `/etc/protocols`.

Для протокола tcp существует возможность указания определённых флагов, например:

```
iptables -A INPUT -p tcp --tcp-flags SYN,ACK SYN
```

Обратите внимание на то, что у параметра `--tcp-flags` два аргумента. Первый означает список проверяемых флагов, а второй говорит о том, какие флаги должны быть установлены. Соответственно, под данное правило попадут все пакеты, у которых флаг **SYN** установлен, а флаг **ACK** сброшен.

Параметр `--icmp-type` позволяет определять типы icmp-сообщений. чтобы разрешить только ответы на ping необходимо создать правила:

```
iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
iptables -A INPUT -p icmp -j DROP
```

Чтобы узнать все типы ICMP-сообщений необходимо дать команду:

```
iptables --protocol icmp --help
```

Дополнительные параметры

- Для отрицания используйте параметр "!".
- В качестве порта можно указывать диапазон – 0:1023.
- В качестве порта допускается указание его имени из файла /etc/services:

```
# iptables -A INPUT -p tcp --dport ssh
```
- Для указания маски сети используйте VLSN и CIDR форматы:
 - 192.168.1.0/255.255.255.0
 - 192.168.1.0/24

<http://edu.softline.ru/>



Если вы хотите запретить все входящие пакеты на порты с 0 по 1023, то можете дать команду:

```
iptables -A INPUT -p tcp --dport 0:1023 -j DROP
```

А если наоборот, хотите запретить входящий трафик на все порты кроме 0:1023, то дайте команду:

```
iptables -A INPUT -p tcp ! --dport 0:1023 -j DROP
```


Модули **iptables**

- Функционал **iptables** может быть расширен модулями, вот некоторые из них:
 - **multiport** – возможность указания списка портов.
 - **owner** – возможность использования в качестве критерия UID пользователя.
 - **state** – проверка состояния пакета.
- Параметр **"-m"** используется для загрузки модулей:

```
# iptables -A INPUT -p tcp -m multiport --dport 20,21,22 -j ACCEPT
```

<http://edu.softline.ru/>



Модули позволяют существенно расширить функционал **iptables**. Модуль **owner** работает только в цепочке **OUTPUT** и позволяет в качестве критерия использовать **UID** пользователя:

```
iptables -A OUTPUT -m owner --uid-owner 500 -j DROP
```

Модуль **state** позволяет отслеживать состояние пакетов:

```
iptables -A INPUT -m state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state NEW -j DROP
```

Первым правилом разрешено прохождение всех пакетов, которые относятся к уже установленным соединениям. Второе же правило запрещает установку любых соединений.

Firewalld

- Firewalld – динамически управляемый межсетевой экран.
- Введен в качестве межсетевого экрана по умолчанию в CentOS 7 (RHEL 7).



<http://edu.softline.ru/>



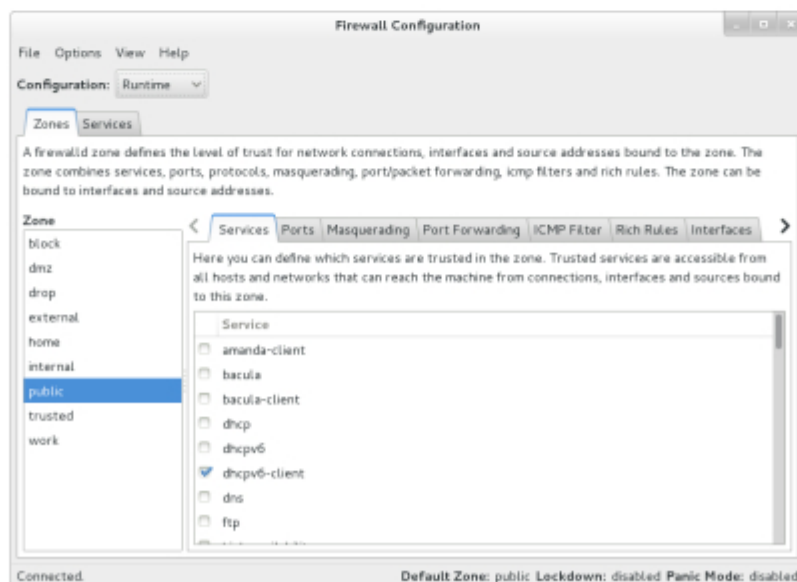
Введение в firewalld.

Демон динамического межсетевого экрана (Firewall) **firewalld** предоставляет динамически управляемый межсетевой экран с поддержкой для сетевых зон (Zones) для назначения уровня доверия сетям, их ассоциированным подключениям и интерфейсам. Он поддерживает параметры межсетевого экрана для IPv4 и IPv6. Он поддерживает изернет-мосты (Ethernet bridges) и имеет разделение запущенных (runtime) и перманентных (permanent) опций настройки. У него также есть интерфейсы для служб и приложений для добавления правил межсетевого экрана напрямую.

Понимание firewalld.

Инструмент настройки с графическим интерфейсом, **firewall-config**, используется для настройки **firewalld**, который, в свою очередь, использует инструмент **iptables** для взаимодействия с **Netfilter** в ядре, который реализует фильтрацию пакетов.

Для использования **firewall-config**, потребуется ввод пароля администратора.



Инструмент **firewall-config** имеет ниспадающее меню выбора, называемое **Configuration**. Оно позволяет выбирать между режимами **Runtime** (Запущенный) и **Permanent** (Постоянный).

**ПРИМЕЧАНИЕ:**

Если вы выбираете режим *Permanent*, дополнительная строка иконок появляется в нижнем левом углу. Эти иконки доступны только в постоянном режиме конфигурации, так как параметры службы не могут быть изменены в запущенном режиме.

Служба межсетевого экрана, предоставляемая **firewalld** – динамическая, а не статическая. Таким образом можно вносить изменения в конфигурацию в любое время, и они будут незамедлительно применены, без необходимости сохранять или применять изменения. Никаких разрывов связи текущих сетевых подключений не происходит, также, как и никакая часть межсетевого экрана не перезапускается.

firewall-cmd – клиент командной строки, также предоставляется. Он может быть использован для создания перманентных и не перманентных изменений, как написано в **man firewall-cmd(1)**.

**ПРИМЕЧАНИЕ:**

Команда **firewall-cmd** может быть выполнена пользователем *root* и административным пользователем, а также членом группы *wheel*. В последнем случае команда будет разрешена через механизм **polkit**.

Конфигурация **firewalld** сохраняется в различных XML-файлах в **/usr/lib/firewalld/** и **/etc/firewalld/**. Это позволяет достичь максимальной гибкости, так как файлы могут быть отредактированы, записаны в резервную копию, использованы в качестве шаблонов для других установок и так далее.

Другие приложения могут взаимодействовать с **firewalld** при помощи D-bus.

Сравнение **firewalld с **systemd-config-firewall** и **iptables**.**

Основные различия между службами **firewalld** и **iptables** заключаются в следующем:

- Служба **iptables** хранит конфигурацию в **/etc/sysconfig/iptables**, а **firewalld** сохраняет ее в различных XML-файлах в **/usr/lib/firewalld/** и **/etc/firewalld/**.

**ПРИМЕЧАНИЕ:**

Файл **/etc/sysconfig/iptables** не существует по умолчанию, тогда как **firewalld** установлен по умолчанию в CentOS 7 (RHEL 7).

- Каждое изменение для службы **iptables** обозначает очистку всех старых правил и прочтение всех новых правил из файла **/etc/sysconfig/iptables**, в то время как **firewalld** не пересоздает все правила, а только применяет изменения. Также имеется возможность изменять параметры в процессе работы без обрыва соединений.

Обе службы используют инструмент **iptables** для общения с фильтром пакетов ядра.

Firewall Stack представлен на схеме ниже.

Работа с FirewallD - Общие положения об использовании.

Для того чтобы, например, разрешить или запретить функции брандмауэра в зонах, вы можете использовать либо графическое средство конфигурирования **firewall-config**, либо консольную утилиту **firewall-cmd**.

Консольное приложение **firewall-cmd** поддерживает все функции межсетевого экрана. В режимах статуса и запроса утилита не выдаёт ответных сообщений, но команда возвращает состояние.

Получение статуса FirewallD:

```
# firewall-cmd --state
```

Эта команда возвращает статус FirewallD, но видимого ответного сообщения нет. Чтобы получить состояние в визуальном различимом виде, следует поступить так:

```
# firewall-cmd --state && echo "Running" || echo "Not running"
```

Перезагрузка брандмауэра без потери информации о состоянии:

```
# firewall-cmd --reload
```

Если вместо этого вы используете ключ **--complete-reload**, информация о состоянии будет потеряна. Эту опцию следует использовать только в случае серьёзных проблем с межсетевым экраном, например, если существуют такие неполадки с информацией о состоянии, что не может быть установлено ни одно соединение, но все правила межсетевого экрана верны.

Получение списка всех поддерживаемых зон:

```
# firewall-cmd --get-zones
```

Данная команда выводит список, разделённый пробелами.

Получение списка всех поддерживаемых служб:

```
# firewall-cmd --get-services
```

Данная команда выводит список, разделённый пробелами.

Получение списка всех поддерживаемых типов ICMP:

```
# firewall-cmd --get-icmptypes
```

Данная команда выводит список, разделённый пробелами.

Вывод всех зон с включённым функционалом:

```
# firewall-cmd --list-all-zones
```

Формат вывода таков:

```
<zone>

  interfaces: <interfacel> ..
  services: <servicel> ..
  ports: <portl> ..
  forward-ports: <forward portl> ..
```

```
icmp-blocks: <icmp type1> ..
```

...

Отображение зоны *<zone>* с включённым функционалом (если зона не задана, задействуется зона, указанная по умолчанию):

```
# firewall-cmd [--zone=<zone>] --list-all
```

Получение зоны, указанной по умолчанию для сетевых соединений:

```
# firewall-cmd --get-default-zone
```

Задание зоны по умолчанию:

```
# firewall-cmd --set-default-zone=<zone>
```

Все интерфейсы, относящиеся к зоне, указанной по умолчанию, будут отнесены к новой зоне по умолчанию, которая будет определять ограничения для последующих вновь иницилируемых сетевых соединений. Активные соединения не подвержены действию этой команды.

Получение активных зон:

```
# firewall-cmd --get-active-zones
```

Данная команда выводит интерфейсы, указанные в качестве части зоны, в следующей форме:

```
<zone1>: <interface1> <interface2> ..
```

```
<zone2>: <interface3> ..
```

Получение зон, относящихся к интерфейсу:

```
# firewall-cmd --get-zone-of-interface=<interface>
```

Данная команда выводит имя зоны, если интерфейс является частью зоны.

Добавление интерфейса в зону:

```
# firewall-cmd [--zone=<zone>] --add-interface=<interface>
```

Следует добавить интерфейс в зону, если он не был в зоне до этого. Если опции зоны не заданы, будет задействована зона, указанная по умолчанию. Параметры интерфейсов применяются после их перезагрузки.

Изменение зоны, которой принадлежит интерфейс:

```
# firewall-cmd [--zone=<zone>] --change-interface=<interface>
```

Данная команда схожа с опцией `--add-interface`, но она соотносит интерфейс новой зоне, даже если до этого он принадлежал другой зоне.

Удаление интерфейса из зоны:

```
# firewall-cmd [--zone=<zone>] --remove-interface=<interface>
```

Сделать запрос, относится ли интерфейс к зоне:

```
# firewall-cmd [--zone=<zone>] --query-interface=<interface>
```

Осуществляет возврат, если интерфейс в зоне. Видимого ответного сообщения нет.

Вывод включённых в зоне служб:

```
# firewall-cmd [ --zone=<zone> ] --list-services
```

Включение "режима паники", блокирующего весь сетевой трафик в случае крайней необходимости:

```
# firewall-cmd --panic-on
```

Отключение "режима паники":

```
# firewall-cmd --panic-off
```

Сделать запрос на предмет "режима паники":

```
# firewall-cmd --query-panic
```

Эта команда возвращает статус "режима паники" (0 если включено (on) и 1 в обратном случае), а также выводит на консоль yes/no.



ПРИМЕЧАНИЕ:

Имена опций были изменены в версии 0.3.0. В предыдущих версиях использовались **--enable-panic** и **--disable-panic**.

Сетевая трансляция

- Сетевая трансляция подразумевает транслирование одного ip-адреса в другой.
- Сетевая трансляция доступна в таблице `nat`.
- **SNAT** – **S**ource **N**etwork **A**ddress **T**ranslation чаще всего используется для сокрытия адреса источника путём его замены:
 - Используется в цепочке `POSTROUTING`.
- **DNAT** - **D**estination **N**etwork **A**ddress **T**ranslation:
 - Используется в цепочке `PREROUTING`.

<http://edu.softline.ru/>



Самое частое применение сетевая трансляция находит в корпоративных сетях, использующих зарезервированные не маршрутизируемые диапазоны ip-адресов. Используя такие адреса, узлы лишаются возможности посылать пакеты в сеть Интернет, так как их пакеты никогда не будут доставлены адресату. Задача сетевой трансляции подменить адрес другим – публичным. Чаще всего в качестве адреса замены используется адрес внешнего сетевого интерфейса маршрутизатора.

На маршрутизатор ложится задача по трассировке устанавливаемых соединений. При прохождении транзитного пакета, у которого в качестве адреса источника стоит адрес из зарезервированного диапазона, этот адрес изменяется на адрес маршрутизатора. Следовательно, ответный пакет в качестве адреса назначения будет иметь ip-адрес маршрутизатора, который, в свою очередь, должен определить, кому на самом деле адресован пакет, поменять адрес назначения и передать пакет настоящему адресату.

Рассмотрим пример реализации **Source NAT**:

```
iptables -A POSTROUTING -t nat -j SNAT --to-source 10.20.30.40
```

Destination NAT работает по тому же принципу, но подменяет адрес назначения, что удобно для "пробрасывания" порта. Представим ситуацию, когда web-сервер находится не на маршрутизаторе, а на узле `192.168.1.2` в локальной сети. Браузер, при соединении с web-сервером, использует доступный адрес – `10.20.30.40`, то есть адрес маршрутизатора. Но когда пакет придёт по этому адресу, маршрутизатор его откинёт, так как web-сервер на нём не запущен, а следовательно принять пакет некому. Как же решить проблему? В этом нам поможет **DNAT**:

```
iptables -A PREROUTING -t nat -p tcp -d 10.20.30.40 --dport 80 -j DNAT \
--to-destination 192.168.1.2
```

И **SNAT** и **DNAT** позволяют переопределять не только ip-адреса источника и назначения, но и порты. В случае, если порт не указывается, то используется тот, который и был указан в исходном пакете. Следующее правило продемонстрирует возможность переопределения порта. При соединении на порт 80, соединение будет переброшено на порт 8080 узла `192.168.1.2`:

```
iptables -A PREROUTING -t nat -p tcp -d 10.20.30.40 --dport 80 -j DNAT \
```


`--to-destination 192.168.1.2:8080`

Пример скрипта, задающего правила брандмауэра:

```
#!/bin/bash
fw="/usr/sbin/iptables"           # определяем переменную fw для удобства
ext_ip="10.20.30.40"              # внешний адрес маршрутизатора
int_ip="192.168.1.1"              # внутренний адрес маршрутизатора
admin_ip="1.2.3.4"                # ip-адрес администратора
int_net="192.168.1.0/24"
$fw -F                            # обнуление всех цепочек таблицы filter
$fw -t nat -F                     # обнуление цепочек таблицы nat
$fw -P INPUT DROP                 # политика по умолчанию
$fw -P OUTPUT DROP
$fw -P FORWARD DROP
$fw -A INPUT -s $admin_ip -j ACCEPT # разрешать все пакеты с узла администратора
$fw -A INPUT -p tcp --syn --dport 22 -j ACCEPT
                                     # разрешать ssh-
$fw -A INPUT -p tcp -m state --state ESTABLISHED,RELATED --dport 22 -j ACCEPT
                                     # соединения
$fw -A INPUT -p tcp -j DROP
$fw -A INPUT -p icmp -j ACCEPT     # разрешить все icmp и udp-пакеты
$fw -A INPUT -p udp -j ACCEPT
$fw -A OUTPUT -d $admin_ip -j ACCEPT
$fw -A OUTPUT -p tcp -m state --state ESTABLISHED,RELATED --sport 22 -j ACCEPT
$fw -A OUTPUT -p icmp -j ACCEPT
$fw -A OUTPUT -p udp -j ACCEPT
$fw -A INPUT -i lo -j ACCEPT
$fw -A OUTPUT -o lo -j ACCEPT
$fw -t nat -A POSTROUTING -o eth0 -d ! $int_net -j SNAT --to-source $ext_ip
```

Средства диагностики

- Часто требуется детальное изучение сетевой активности, пакетов и т.д.
- **tcpdump** – утилита для захвата и анализа сетевого трафика:
 - **-i** – какой интерфейс необходимо "слушать".
 - **-c** – выйти после захвата определённого числа пакетов.
 - **-X** – выводить содержимое пакета в ASCII.
- **ethereal** – ещё одна утилита захвата и анализа трафика:
 - Имеет графическую надстройку.
- **iptraf** – интерактивная утилита мониторинга сетевой активности.

<http://edu.softline.ru/>



Программа **tcpdump** является мощным средством захвата и анализа трафика. Если её вызвать без аргументов, то **tcpdump** начнёт захват всех пакетов на первом сетевом интерфейсе.

Для более точного захвата существует ряд примитивов, с помощью которых администратор может задать критерии. Вызов **tcpdump** с фильтром может выглядеть так:

```
tcpdump -i [INTERFACE] [TYPE] [DIRECTION] [PROTOCOL]
```

- **INTERFACE** – сетевой интерфейс, на котором необходимо ловить трафик
- **TYPE** – **host, net, port**. С помощью типа вы можете указать основной критерий, трафик для какого узла, сети или порта необходимо отлавливать, например:

```
# tcpdump host 192.168.1.2
```

```
# tcpdump net 192.168.1
```

- **DIRECTION** – **src, dst**. Критерий, указывающий на направление трафика, например:

```
# tcpdump src 192.168.1.2
```

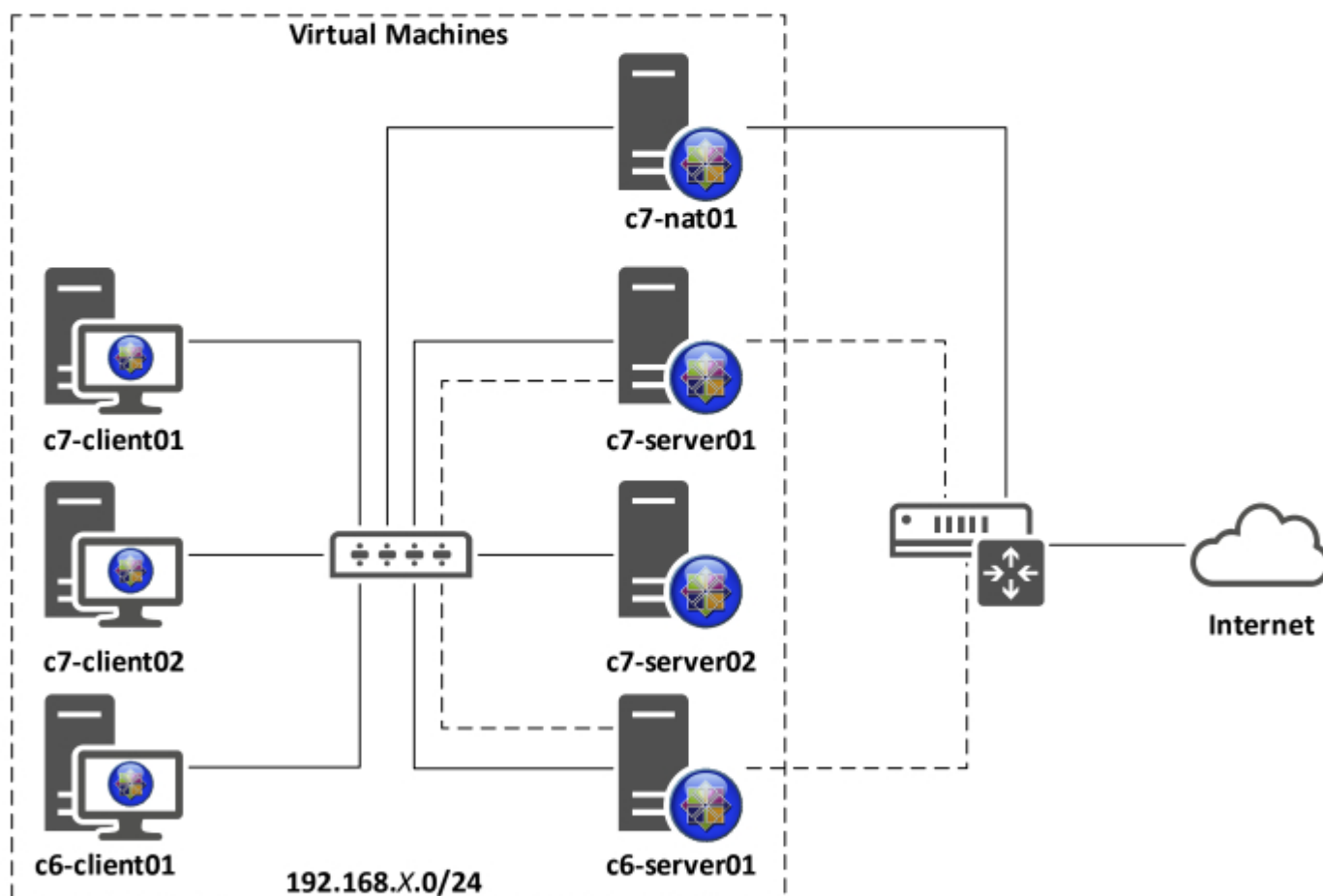
```
# tcpdump src 192.168.1.1 and dst 192.168.1.4
```

- **PROTOCOL** – критерий, определяющий протокол.

Лабораторная работа 8

- **Упражнения:**
 - Защита узла средствами `iptables`.
 - Ограничение пропускной способности.
- **Необходимые машины:**
 - `c7-nat01`.
 - `c6-server01`.
 - `c7-client01` или `c7-client02`.
- **Учетные данные:**
 - Пользователь: `root`.
 - Пароль: `Pa$$w0rd`.

<http://edu.softline.ru/>



Упражнение 1 – Защита узла средствами iptables.

Задания упражнения выполните на машине **c6-server01**.

Вам необходимо защитить свой узел с помощью пакетной фильтрации и брандмауэра **iptables**. Лабораторная работа производится совместно с коллегами по учебному классу.

1. Остановите брандмауэр **iptables**, если он был запущен ранее.

```
# service iptables stop
```

2. Проверьте, запущен ли сервис **httpd** и попробуйте соединиться с web-сервером при помощи браузера по адресу `http://localhost`. Должна открыться страница.

3. С помощью утилиты **nmap** осуществите сканирование узла соседа. Он, в свою очередь, должен просканировать ваш узел:

```
# nmap -sS 192.168.1.X
```

4. Определите политику по умолчанию для цепочки **FORWARD** и снова попросите соседа осуществить сканирование. Остались ли порты открытыми? Почему?:

```
# iptables -P FORWARD DROP
```

```
# nmap -sS 192.168.1.X
```

5. Определите политику по умолчанию для цепочки **OUTPUT** и **INPUT**:

```
# iptables -P INPUT DROP
```

```
# iptables -P OUTPUT DROP
```

6. Снова произведите сканирование. Откройте web-браузер и пройдите по адресу `http://localhost`. Открылась ли страница? Почему?

7. Переопределите политику для цепочки **INPUT**. Попросите соседа соединиться с вашим web-сервером. Открылась ли страница? Почему? Используйте программу **netstat** для определения проблемы.

```
# iptables -P INPUT ACCEPT
```

```
# netstat -na | grep :80
```

8. Разрешите соединение с web-сервером:

```
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

9. Создайте правило, при котором один из узлов вашей сети (по выбору) не сможет подключаться к web-серверу, в то время как другие смогут.

10. Создайте правило, которое все соединения из сети 192.168.1.0/24 на порт 8080 перенаправляет бы на порт 80. Попросите соседа с помощью web-браузера пройти по ссылке `http://192.168.1.X:8080`, соединение должно быть перенаправлено.

Ответы к упражнению 1.

4. Соединения с вашим ip-адресом не являются транзитными и не попадают в цепочку FORWARD, следовательно, запрет транзитного трафика никак не отражается на соединениях с вашим сервером.

6. Запретив весь входящий и исходящий трафик, вы запретили и соединения на 80й порт, даже для локальных соединений. Чтобы разрешить локальный трафик, задайте правила:

```
# iptables -A INPUT -i lo -j ACCEPT
```

```
# iptables -A OUTPUT -o lo -j ACCEPT
```

7. Используя программу netstat, вы должны увидеть, что состояние соединения с 80м портом находится в состоянии SYN_RCV, что означает, что был получен пакет с флагом SYN. Так как вы установили политику для цепочки INPUT в значение ACCEPT, то данный пакет был получен. Но для установки соединения требуется отправка ответного пакета с флагом ACK. Но политика для цепочки OUTPUT находится в состоянии DROP, поэтому соединение установлено не будет.

8. Чтобы запретить доступ к web-серверу для всех адресов, дайте правило:

```
# iptables -I 1 INPUT --p tcp --dport 80 -j DROP
```

Теперь разрешите доступ для определённых адресов:

```
# iptables -I 1 INPUT --p tcp -s 192.168.1.X --dport 80 -j ACCEPT
```

10. Для перенаправления трафика используйте действие REDIRECT

```
# iptables -t nat -A PREROUTING -p tcp -s 192.168.1.0/24 --dport 8080 -j REDIRECT --to-ports 80
```

Упражнение 2 – Ограничение пропускной способности.

Задания упражнения выполните на машине **c6-server01**.

Попробуем ограничить пропускную способность.

Наша задача, пропускать не более 5 запросов типа icmp-echo через интерфейс eth0.

1. Чтобы осуществить задуманное, сначала обнулим таблицы iptables.

iptables -F

2. Для ограничения пропускной способности воспользуемся модулем limit. Этот модуль работает с двумя параметрами --limit и --limit-burst:

- limit это скорость, с которой счетчик burst limit "откручивается назад"
- limit-burst это максимальное значение счетчика пакетов, при котором срабатывает ограничение.

Установим ограничение в 3 пакета в минуту:

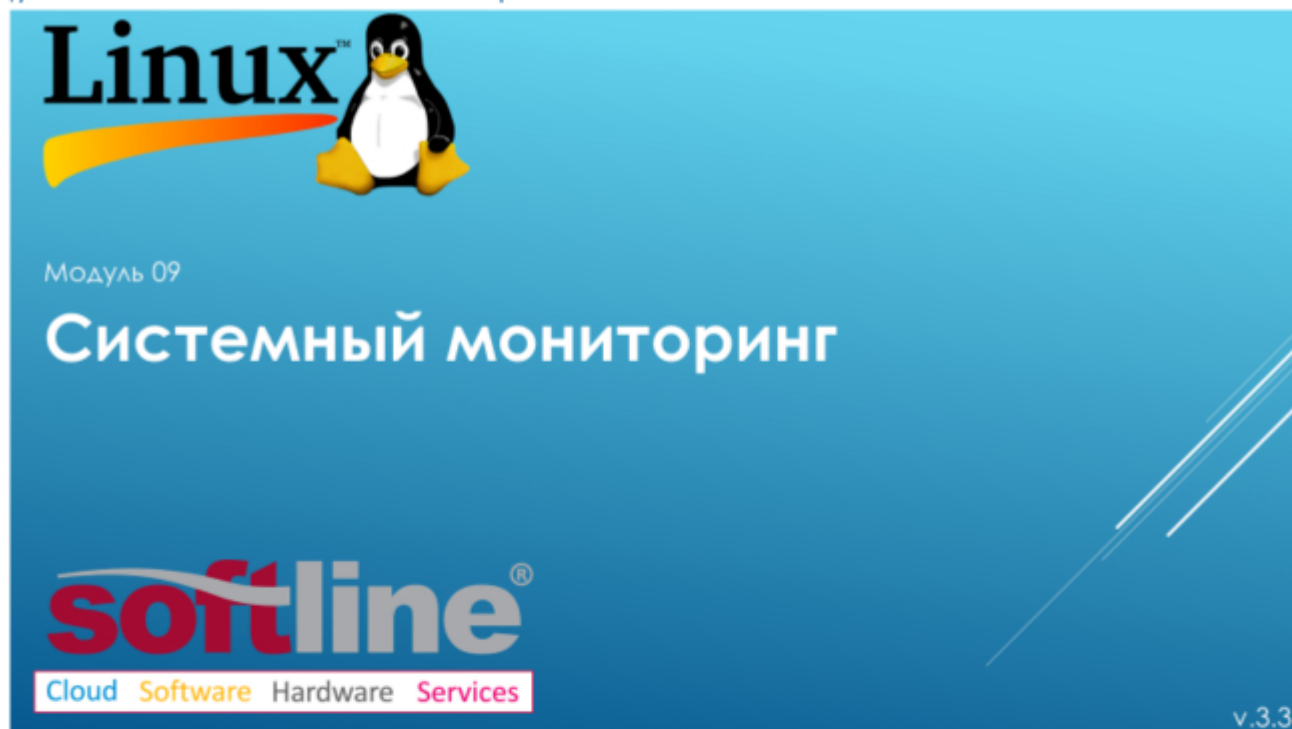
iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 3/m --limit-burst 3 -j ACCEPT

3. Этого правила недостаточно, так как при наполнении буфера с пакетами, задаваемого параметром limit-burst, пакеты не будут отброшены, а пойдут дальше по цепочке, вплоть до применения политики по умолчанию. Зададим блокирующее правило

iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

4. Попросите коллегу "попинговать" ваш узел. Проверьте, срабатывает ли ограничение.

Модуль 09: Системный мониторинг.



В модуле представлены:

- Мониторинг.
- Анализ файловой системы.
- Права доступа к файлам.
- Дополнительные атрибуты **ext***.
- **logwatch**.
- **sar**.
- **mpstat** и **iostat**.
- **Aide**.
-

Мониторинг

- Не существует неуязвимых систем.
- **Время – главный фактор в противодействии потенциальным проблемам:**
 - Заранее обнаруженная уязвимость предотвратит взлом.
 - Заранее обнаруженные проблемы в работе оборудования предотвратят потерю данных и простой в работе.
 - Заранее обнаруженный спад производительности предотвратит потерю рабочего времени компании.
- **Задача администратора – прогнозирование проблем.**

<http://edu.softline.ru/>



Как ни странно, но каждую проблему можно спрогнозировать и чем точнее будет прогноз, тем более болезненно пройдет решение возникающей проблемы.

Бреши в различных программах – явление нормальное, несмотря на все усилия программистов, они всё же остаются людьми, которые могут ошибаться. Задача администратора устранить брешь до того, как ей воспользуется злоумышленник. Поиск уязвимостей – сложная задача, а иногда – невыполнимая. Но ваша задача не столько в том, чтобы обнаружить уязвимость, сколько в устранении уже найденных уязвимостей. Существует огромное множество баз данных уязвимостей и такое же множество программ, осуществляющих проверку по данным БД.

Любой взлом, любая проблема не возникают внезапно. У администратора есть все средства для своевременного реагирования на надвигающуюся угрозу. Самое главное из них – журнальные файлы.

Анализ файловой системы

- Как правило, данные хранятся в файлах, а файлы в файловой системе.
- Повреждение файловой системы может привести к потере данных.
- Внезапно закончившееся свободное место может привести к прекращению работы важнейших сервисов:
 - **du** – покажет занятое файлами место на диске.
 - **df** – выводит сводную таблицу о свободном месте на подключенных разделах.
 - **aide** – произведёт проверку целостности файлов, что позволит своевременно определить повреждение данных.

<http://edu.softline.ru/>



Всем знакома ситуация, когда в ночь перед защитой диплома ломается жёсткий диск, на котором и находилась электронная версия диплома. А теперь представьте, если это не диплом, а необходимые данные для искусственного поддержания жизни человека в больнице или компьютеры диспетчерской аэропорта?

Из-за чего же могут быть повреждены данные или диск, на котором они хранятся? Многие не придают значения ресурсу жесткого диска, а ведь это немаловажный фактор в борьбе за работоспособность.

SMART (Self-Monitoring, Analysis and Reporting Technology) – технология во многих жестких дисках. Задача SMART производить мониторинг жизнедеятельности устройства. Чтобы обратиться к SMART используйте программу **smartctl**. Например, если вы хотите получить общую информацию о жестком диске, дайте команду:

```
# smartctl -d ata -i /dev/sda
=== START OF INFORMATION SECTION ===
Device Model:          ST9120821AS
Serial Number:         5PL2EKMY
Firmware Version:      7.24
User Capacity:         120,034,123,776 bytes
Device is:              Not in smartctl database [for details use: -P showall]
ATA Version is:        7
ATA Standard is:        Exact ATA specification draft version not indicated
Local Time is:          Fri Jun 6 20:51:07 2008 YEKST
SMART support is:       Available - device has SMART capability.
SMART support is:       Enabled
```

Производитель изначально закладывает информацию о ресурсе жесткого диска и данные, с помощью которых можно спрогнозировать оставшийся ресурс. Для того, чтобы увидеть эту информацию необход имо дать команду:

```
# smartctl -d ata -A /dev/sda
=== START OF READ SMART DATA SECTION ===
SMART Attributes Data Structure revision number: 10
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED      WHEN_FAILED      RAW_VALUE
 1 Raw_Read_Error_Rate      0x000f   100   253    006  Pre-fail     Always         -                0
 3 Spin_Up_Time              0x0002   095    094    000    Old_age     Always         -                0
 4 Start_Stop_Count          0x0033   100    100    020  Pre-fail     Always         -               1019
```

5	Reallocated_Sector_Ct	0x0033	100	100	036	Pre-fail	Always	-	0
7	Seek_Error_Rate	0x000f	076	060	030	Pre-fail	Always	-	47590095
9	Power_On_Hours	0x0032	098	098	000	Old_age	Always	-	2192
10	Spin_Retry_Count	0x0013	100	100	034	Pre-fail	Always	-	0
12	Power_Cycle_Count	0x0033	099	099	020	Pre-fail	Always	-	1038
187	Unknown_Attribute	0x0032	100	100	000	Old_age	Always	-	0
189	Unknown_Attribute	0x003a	096	096	000	Old_age	Always	-	4
190	Unknown_Attribute	0x0022	065	051	045	Old_age	Always	-	588644387
192	Power-Off_Retract_Count	0x0032	100	100	000	Old_age	Always	-	241
193	Load_Cycle_Count	0x0032	029	029	000	Old_age	Always	-	142683
194	Temperature_Celsius	0x0022	035	049	000	Old_age	Always	-	35 (Lifetime Min/Max 0/10)
195	Hardware_ECC_Recovered	0x001a	065	048	000	Old_age	Always	-	163130310
197	Current_Pending_Sector	0x0012	100	100	000	Old_age	Always	-	0
198	Offline_Uncorrectable	0x0010	100	100	000	Old_age	Offline	-	0
199	UDMA_CRC_Error_Count	0x003e	200	200	000	Old_age	Always	-	0
200	Multi_Zone_Error_Rate	0x0000	100	253	000	Old_age	Offline	-	0
202	TA_Increase_Count	0x0032	100	253	000	Old_age	Always	-	0

В выводе мы можем увидеть множество различных параметров, таких как число включений/выключений, время наработки, температуру жесткого диска и т.д.

Поле **VALUE** содержит нормализованное значение, которое сравнивается со значением поля **THRESH**, если значение окажется меньше либо равно, то выдаётся сообщение о неполадке в поле **WHEN_FAILED**.

Поле **TYPE** содержит два типа. **Old_age** означает некритичные параметры, а **pre_fail** – критичные. В случае, когда значение поля **VALUE** меньше либо равно значения в поле **THRESH**, существует высокая вероятность того, что диск перестанет работать в ближайшие 24 часа.

Вы можете провести общее тестирование:

```
# smartctl -d ata -H /dev/sda
=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
```

Данную команду вы можете поместить в планировщик **cron** и в случае, если значение будет отлично от **PASSED**, присылать уведомление на e-mail. Но есть вариант лучше и надёжнее – демон **smartd**. Его основной конфигурационный файл **/etc/smartd.conf**, где вы должны определить имя проверяемого устройства и то, какие атрибуты должны учитываться при проверке:

```
/dev/sda -a -I 194 -m hddfail@example.com
```

Данное правило говорит о том, что необходимо проверять диск **/dev/sda**, учитывая все параметры, кроме параметра 194 (температура). Остаётся только запустить демон **smartd**:

```
# service smartd start
```

Хорошо, теперь мы будем уведомлены о предстоящих аппаратных сбоях жестких дисков, но есть ещё один враг и он не менее опасен – свободное место. Рассмотрим простейший скрипт для уведомления о заканчивающемся свободном пространстве:

```
#!/bin/bash

LIMIT=80      #допустимый процент использованного пространства
USED=`df | grep /$ | awk {' print $4 ' } | sed 's/%//`
if [ $USED -ge $LIMIT ]; then
    echo "Limit has been reached" | mail -s "Size limit" admin@example.com
fi
```


Права доступа к файлам

- Важно периодически выполнять проверки с целью поиска файлов с установленными битами **SUID** и **SGID**. Такая проверка позволит обнаружить взлом на ранней стадии.
- Для поиска используйте программу `find`.
- Бит **SUID** позволяет запускать бинарный файл от имени владельца файла.
- Бит **SGID** позволяет запускать бинарный файл от имени группы-владельца файла.

<http://edu.softline.ru/>



Существенную опасность представляют программы с установленными битами **SUID** или **SGID**. Для обнаружения таких файлов необходимо использовать программу `find`:

```
# find / -type f -perm /6000
```

Для систем с повышенными требованиями к безопасности возможно подключение файловой системы с опцией `nosuid`, которая говорит о том, что бит **SUID** должен быть принудительно отключен у всех файлов, находящихся в данной файловой системе. Очень важно использовать данный параметр при подключении сменных носителей:

```
# mount /dev/fd0 /mnt/floppy -o nosuid
```

Дополнительные атрибуты **ext***

- Файловая система Ext2/3/4 поддерживает ряд специальных атрибутов файлов:
 - **lsattr** – посмотреть атрибуты файла.
 - **chattr** – изменить атрибуты файла.
- Существует возможность расширения списка владельцев файла:
 - **getfacl** – посмотреть права доступа для всех пользователей.
 - **setfacl** – установить права доступа для дополнительных пользователей.

<http://edu.softline.ru/>



У каждого файла в файловой системе Ext2/3/4 существует ряд дополнительных атрибутов:

- **A** – при обращении к файлу, время последнего доступа изменяться не будет
- **a** – файл может быть открыт только для дозаписи
- **c** – включить автоматическую компрессию
- **D** – атрибут устанавливается на каталог и означает, что при внесении изменений в каталог их нужно применять в синхронном режиме, сразу записывая на жесткий диск
- **i** – атрибут означает, что файл не может быть изменён ни при каких условиях
- **s** – обнулять информацию на диске, при удалении файла
- **u** – сохранить информацию на диске, при удалении файла для возможности восстановления

Для установки атрибута используйте команду **chattr**:

```
# chattr +i /tmp/file
# lsattr /tmp/file
----i----- /tmp/file
```

Файловая система Ext2/3/4 предоставляет возможность предоставлять права доступа к файлам не только для владельца, группы и всех остальных с помощью списков доступа. Вы можете управлять списками доступа с помощью программы **setfacl**. Её синтаксис таков:

```
setfacl [parameters] [modification_type] acl file
```

Помимо необходимых параметров вы должны указать тип модификации, "-m" означает, что список доступа будет задан в командной строке:

```
# setfacl -m u:user:rwX /tmp/file
```

Данное правило говорит о том, что для пользователя **user** необходимо разрешить чтение, запись и исполнение данного файла.

logwatch

- Изучение журнальных файлов – важнейшая составляющая обеспечения безопасности.
- Программа `logwatch` понимает огромное множество форматов записей журнальных файлов:
 - `postfix.`
 - `vsftpd.`
 - `samba.`
- После выделения важной информации производится формирование отчёта и его отправка ответственному лицу.

<http://edu.softline.ru/>



Адекватная проверка журнальных файлов – неподсильная человеку задача. Журнальных файлов очень много и все они содержат самые разнообразные события. Задача администратора сохранить файлы в сохранности для случая, когда произойдёт событие, требующее детального изучения. В повседневной же работе лучше всего использовать специализированные программы, производящие обработку журнальных файлов и формирование удобочитаемого отчёта.

Программа `logwatch` является одним из мощнейших продуктов по обработке информации из журнальных файлов, формируемой самыми разными сервисами. Минимальное конфигурирование включает в себя определение следующих параметров файла `/etc/logwatch/conf/logwatch.conf`:

- `MailTo = root` *# адрес, на который необходимо отправлять отчёты*
- `LogDir = /var/log` *# путь к каталогу, в котором хранятся журнальные файлы*
- `Range = yesterday` *# за какое время необходимо обработать журнальные файлы*
- `Detail = Low` *# уровень детализации*

Теперь, при запуске программы, она будет формировать отчёты и отправлять их по указанному адресу.

sar

- Данные снимаются каждые 10 минут программами **sa1** и **sa2**.
- Команда **sar** выводит информацию о производительности системы:
 - **-u** – статистика использования ЦПУ.
 - **-d** – статистика активности блочных устройств.
 - **-n** – статистика активности сетевых интерфейсов.
 - **-r** – статистика использования оперативной памяти и **swap**.

<http://edu.softline.ru/>



Программы **sa1** и **sa2** производят сбор статистической информации о деятельности различных подсистем, сети, процессора, памяти и т.д. Вся эта информация сохраняется в специальных бинарных файлах и может быть прочитана с помощью программы **sar**. Данная информация позволит вам оценить производительность системы с течением времени и очень важна при оптимизации системы.

Сбор данных производится с помощью планировщика **cron**.

mpstat и iostat

- **mpstat** – выводит информацию о работе всех процессоров:
 - **mpstat 2** – снимать и выводить статистику каждые 2 секунды.
- **iostat** – выводит информацию о работе процессора и информацию об операциях ввода/вывода для устройств и разделов.
- Используйте программу **watch** для обновления информации on-line:

```
# watch iostat
```

<http://edu.softline.ru/>



Программы **mpstat** и **iostat** входят в состав пакета **systat** и предназначены для получения разнообразной информации, с помощью которой администратор может определить производительность системы и найти узкие места.

Mpstat предназначена для вывода на **stdout** информации обо всех процессорах в системе, а **iostat** позволит проследить за работой устройств хранения информации.

Aide

- Aide (Advanced Intrusion Detection Environment) — программа, выполняющая контроль за целостностью файлов:
 - Своевременное реагирование на изменение важных файлов.
- Настройка должна быть выполнена сразу при подготовке сервера к размещению в рабочую среду.
- Основной конфигурационный файл `/etc/aide.conf`.
- База с данными о файлах хранится в каталоге `/var/lib/aide/`:
 - Необходимо обеспечить максимальную защиту для данного каталога.

<http://edu.softline.ru/>



Одной из немаловажных составляющих безопасной системы являются программы проверки целостности файлов. Их основной задачей является оповещение администратора об изменениях файлов. Согласитесь, если был заменён бинарный файл `/usr/bin/passwd`, это должно очень насторожить.

Программа Aide осуществляет подсчёт контрольных сумм для всех файлов, указанных в файле `/etc/aide.conf`. Данные о файлах и их контрольных суммах находятся в специальной базе данных. При проверке сумма будет вновь подсчитана и сопоставлена той, что находится в БД, если суммы окажутся разными, то делается вывод о том, что файл был изменён.

Для формирования первичной базы данных используйте команду:

```
# aide --init
```

База данных была сформирована в файле `/var/lib/aide/aide.db.new.gz`. Его нужно переименовать:

```
# cp /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

Чтобы выполнить проверку целостности дайте команду:

```
# aide --check
```

Лабораторная работа.

Лабораторная работа 9

Упражнения:

1. Аудит файловой системы.
2. Управление списками контроля доступа (ACL).

Необходимые машины:

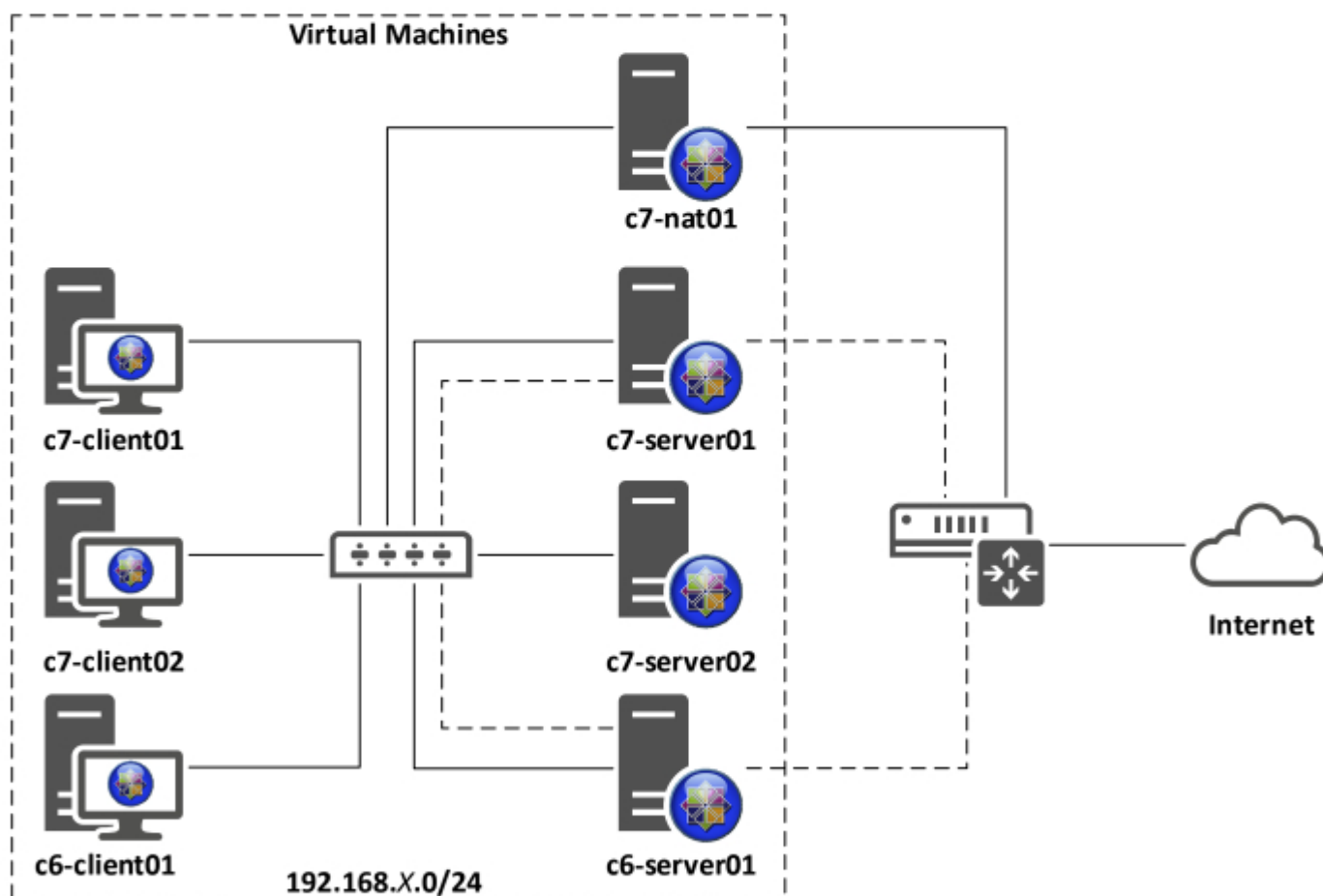
- c7-nat01.
- c7-server01.

Учетные данные:

- Пользователь: `root`.
- Пароль: `Pa$$w0rd`.



<http://edu.softline.ru/>



Упражнение 1 – Аудит файловой системы.

Вам необходимо убедиться в отсутствии бесхозных и иных "опасных" файлов. Также вам необходимо заранее обеспечить безопасность на уровне файловой системы.

1. Найдите все файлы, у которых разрешено право записи для всех пользователей:

```
# find / -type f -perm -0002 -print
```

2. Найдите все файлы, у которых нет владельцев. Это могло произойти при удалении учётной записи:

```
# find / \( -nouser -o -nogroup \) -print
```

3. Представьте ситуацию, когда вам необходимо удалить учётную запись пользователя. Составьте последовательность команд, при которой сначала будут найдены все файлы, принадлежащие учётной записи. Эти файлы должны быть перемещены в каталог **/var/backup/delete** и только потом должен быть удалён пользователь. Результатом выполнения данного задания должен стать сценарий, состоящий из необходимых команд.

4. Теперь необходимо определить безопасность на уровне файловой системы. Опция **nodev**, указанная при монтировании файловой системы, обеспечит невозможность создания файлов-устройств на данной файловой системе. Опция **noexec** и **nosuid**, будет автоматически отключать такие атрибуты файлов, как **suid** и **execute**. Такого рода опции необходимо добавлять для сменных устройств. В файле **/etc/fstab** укажите опцию **nodev** для файловой системы, подключаемой в каталог **/home**.

```
LABEL=/home          /home          ext3      nodev          1 2
```

5. Теперь попробуйте под непривилегированным пользователем скопировать любой файл-устройства из каталога **/dev** в домашний каталог:

```
$ cp /dev/sda1 ~
```


Ответы к упражнению 1.

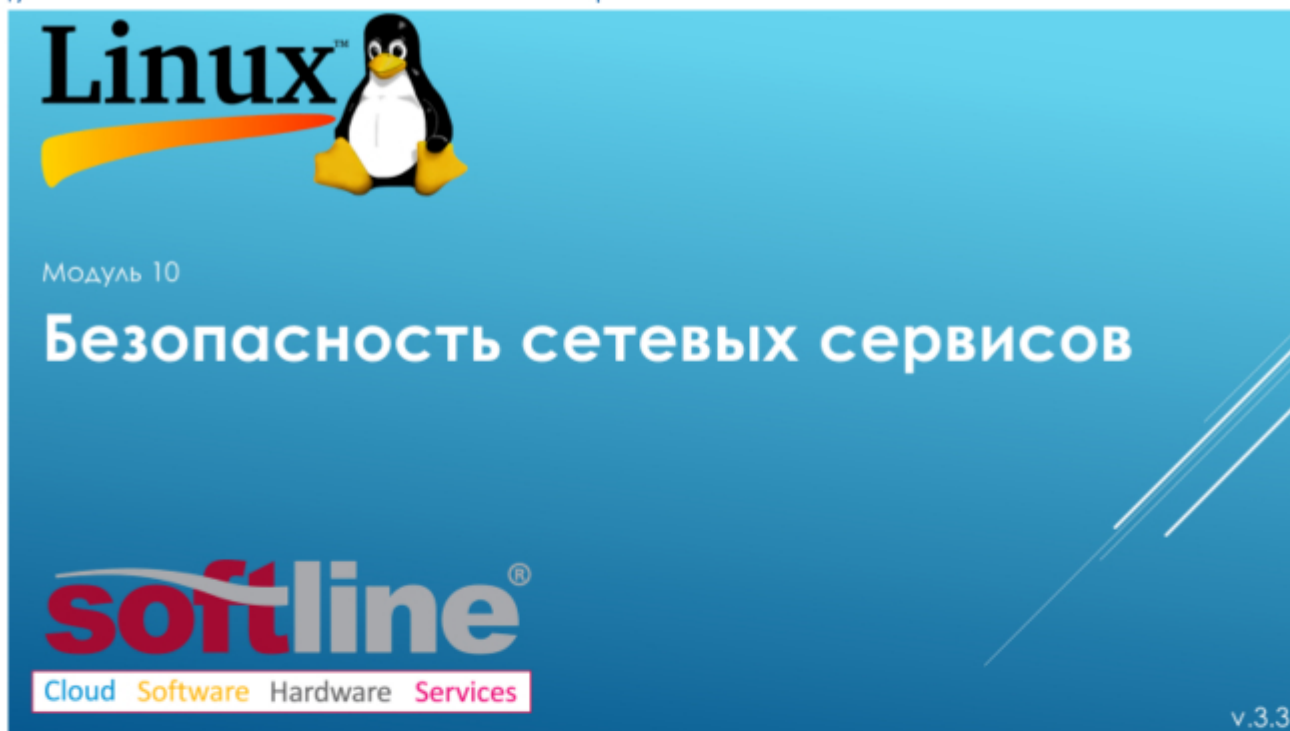
3. Рассмотрим простейший скрипт, предназначенный для выполнения задания:

```
#!/bin/bash
USERNAME="user"
DIR="/var/backup/delete"
mkdir $DIR
find / -user $USERNAME -exec mv {} $DIR \;
userdel $USERNAME
```

Упражнение 2 – Управление списками контроля доступа (ACL).

Получите ссылку на лабораторную работу у инструктора.

Модуль 10: Безопасность сетевых сервисов.



В модуле представлены:

- Безопасность сервиса.
- Конфигурирование **libwrap**.
- Программа **denyhosts**.
- Шифрование.
- Цифровые сертификаты.

Безопасность сервиса

- Каждый сетевой сервис должен обеспечивать необходимый функционал, всё лишнее должно быть отключено.
- Два основных средства обеспечения сетевой безопасности:
 - Использование средств безопасности, предоставляемых программой, реализующей сервис:
 - `postfix`.
 - `httpd` и другие.
 - Использование внешних средств обеспечения безопасности:
 - `iptables`.
 - Библиотека `libwrap`.

<http://edu.softline.ru/>



Многие приложения реализуют собственные механизмы обеспечения безопасности, включающие в себя разграничение прав доступа, сетевой фильтрации и т.д. Но зачастую требуется простое централизованное средство обеспечения сетевой доступности.

Библиотека `libwrap` позволяет осуществлять проверку, основываясь на адресе удалённого сетевого узла. Существует иное название — "`tcp wrappers`". Когда сетевой сервер принимает соединение, используя `libwrap`, он осуществляет проверку, разрешено ли обращаться данному сетевому узлу.

Узнать, поддерживает ли программа библиотеку `libwrap`, необходимо посмотреть, была ли она включена на этапе компиляции: безопасность.

```
# ldd /usr/sbin/vsftpd
.....
libssl.so.6 => /lib/libssl.so.6 (0x0012e000)
libwrap.so.0 => /lib/libwrap.so.0 (0x00173000)
libnsl.so.1 => /lib/libnsl.so.1 (0x0017b000)
.....
```

Основные конфигурационные файлы `/etc/hosts.allow` и `/etc/hosts.deny`.

Конфигурирование libwrap

- При подключении к сервису, защищённому libwrap, производится проверка в следующем порядке:
 - Файл `/etc/hosts.allow`. Если подключение разрешено, проверка прекращается, доступ предоставляется;
 - Файл `/etc/hosts.deny`. Если подключение запрещено, проверка прекращается, в доступе отказывается;
 - Если ни в одном из файлов не было необходимого правила, то доступ предоставляется.
- Синтаксис:


```
service: address
```
- Пример:


```
sshd: 192.168.1.2
```

<http://edu.softline.ru/>



Библиотека `libwrap.so` осуществляет проверку "до первого попадания". Проверка начинается с файла `/etc/hosts.allow`, если в этом файле будет найдена соответствующая запись, то соединение будет разрешено. Если разрешающая запись не была найдена, то будет выполнен поиск в файле `/etc/hosts.deny`. Если в этом файле будет найдена запись, то соединение будет запрещено. Иначе — соединение разрешается.

Каждая запись состоит из двух частей:

- Имя сервиса — имя исполняемого файла (**sshd**, **httpd**, **mysqld**). Администратор может указать несколько сервисов через запятую. Также возможно указание сервиса и интерфейса, для которого необходимо осуществлять проверку.

```
sshd, vsftpd: 1.2.3.4
```

```
httpd: 1.2.3.4
```

- Адрес — описание сетевых узлов, для которых применима данная запись. Здесь возможно указание ip-адресов, доменных имён

```
sshd, vsftpd: 1.2.3.4
```

```
sshd, vsftpd: 1.2.3. # для сети 1.2.3.0/24
```

```
sshd, vsftpd: 1.2.3.4/255.255.255.128 # для сети 1.2.3.1/25
```

```
sshd, vsftpd: .example.com # для всех имён *.example.com
```

В любой части записи возможно использование служебных слов **ALL**, **EXCEPT**:

- **ALL** — соответствует всем возможным вариантам.
- **EXCEPT** — позволяет задавать исключения.

- **LOCAL** — все доменные имена без точки в имени (например localhost).
- **UNKNOWN** — все неопознанные доменные имена (нет записи PTR).
- **KNOWN** — все опознаваемые доменные имена.
- **PARANOID** — все адреса, не имеющие прямого и обратного преобразования (записи A и PTR).

`sshd, vsftpd: 1.2.3. EXCEPT 1.2.3.4 # вся сеть 1.2.3.0/24 кроме 1.2.3.4.`

Программа denyhosts

- Блокирует доступ к сервисам с помощью `tcp_wrappers`.
- Осуществляет мониторинг журнального файла `/var/log/secure` и выявляет попытки подбора пароля к серверу SSH.
- Помещает соответствующие записи в `/etc/hosts.deny`.
- Конфигурационный файл – `/etc/denyhosts.conf`:
 - `BLOCK_SERVICE = sshd` – блокируемый сервис.
 - `DENY_THRESHOLD_INVALID = 3` – количество попыток ввести логин, после которого адрес должен быть заблокирован.
 - `DENY_THRESHOLD_VALID = 3` – количество попыток ввести пароль, после которого адрес должен быть заблокирован.
 - `DENY_THRESHOLD_ROOT = 5` – количество попыток ввести пароль суперпользователя `root`.

<http://edu.softline.ru/>



Программа `denyhosts` является удобным средством блокировки узлов, с которых производятся попытки подбора пароля через сервер SSH.

Администратору необходимо указать в конфигурационном файле путь к журнальному файлу:

```
SECURE_LOG = /var/log/secure
```

Каждая неверная попытка войти в систему регистрируется в журнальном файле, `denyhosts`, анализируя его, определяет количество неудачных попыток в течение времени для каждого узла и блокирует их, внося соответствующую запись в файл `/etc/hosts.deny`:

```
# DenyHosts: Tue May 20 20:16:45 2008 | sshd: 202.51.172.146  
sshd: 202.51.172.146  
  
# DenyHosts: Wed May 21 01:25:14 2008 | sshd: 200.198.73.194  
sshd: 200.198.73.194
```

Очень важно аккуратно определять параметры, отвечающие за допустимое количество попыток ввода неверного пароля. Если вы попытаетесь удалённо войти на сервер и случайно введёте неверный пароль, то следующий параметр заблокирует вас:

```
DENY_THRESHOLD_ROOT = 5
```

`Denyhosts` предлагает механизм авторазблокирования. Следующий параметр говорит о том, что разблокироваться должны те адреса, которые были заблокированы более чем указанный срок:

```
PURGE_DENY = 4w
```

Шифрование

- Шифрование используется там, где важна конфиденциальность данных:
 - web-сервер;
 - почтовый сервер (SMTP/POP3/IMAP4).
- Составляющие шифрования:
 - Случайные числа.
 - Однонаправленные хеши.
 - Симметричное шифрование.
 - Асимметричное шифрование.
 - Цифровые сертификаты.
- OpenSSL – программная реализация различных методов шифрования.

<http://edu.softline.ru/>



Какими бы ни были настройки сетевых сервисов, они никогда не гарантируют защиту от "подслушивания", например, при сетевой атаке типа MITM.

Многие криптоалгоритмы основаны на случайных числах. Случайные числа сложно сгенерировать программно, так как алгоритм генерации случайных чисел уже не может гарантировать случайность этих самых чисел. Поэтому в качестве источника "случайности" используются системные прерывания, сигналы, поступающие от клавиатуры, мыши.

В Linux существует два специальных устройства:

- **/dev/random** — данное устройство выдаёт энтропию, которая накапливается по мере работы пользователя — нажатий клавиш клавиатуры, использования мышки. Если энтропия закончилась, необходимо её пополнить, например, подвигать мышкой.
- **/dev/urandom** — делает то же самое, что и **/dev/random**, но когда энтропия заканчивается, то устройство приступает к искусственному генерированию случайных последовательностей.

Администратор может получить случайную последовательность с помощью библиотеки **openssl**:

```
# openssl rand
```

Однонаправленное хэширование подразумевает под собой создание отпечатка (fingerprint) на основе передаваемых данных. Чтобы убедиться в том, что в данные не были внесены изменения, нужно всего лишь пересчитать fingerprint и сравнить его с предыдущим. Отпечаток не может быть преобразован обратно в данные, являясь однонаправленным.

Алгоритмы однонаправленного хэширования: **md5, sha, sha1** и другие. Чтобы гарантировать целостность файла при его передаче по сети, необходимо получить его отпечаток и сохранить в файле:

```
# md5sum /etc/passwd > ./md5sum.txt
```

После передачи файла, принимающей стороне необходимо передать файл **md5sum.txt** и выполнить команду:

```
# md5sum -c ./md5sum.txt  
  
/etc/passwd: OK
```

Симметричное шифрование основано на ключе, который используется и для шифрования, и для дешифрования. Ключ создаётся на основе пароля. Данный метод шифрования используется и для генерирования однонаправленных хэшей, например, паролей. Основным представителем, использующим данную функцию, является программа **passwd**.

Представители симметричного шифрования: **DES**, **3DES**, **Blowfish** и многие другие

Асимметричное шифрование — наиболее надёжный способ передачи данных, основан на двух ключах — приватном (**private**) и публичном (**public**). Один ключ шифрует, другой дешифрует. Важно помнить о том, что приватный ключ должен быть защищён максимально надёжно и ни в коем случае не может попасть в руки злоумышленника.

Существует две версии асимметричного шифрования.

- Асимметричное шифрование 1 — подразумевает, что получатель генерирует пару приватный/публичный ключей и распространяет публичный ключ, надёжно сохраняя приватный. Отправитель, используя публичный ключ получателя, шифрует данные и передаёт их получателю. Получатель дешифрует данные с помощью приватного ключа.
- Асимметричное шифрование 2 — подразумевает, что отправитель генерирует пару приватный/публичный ключей, распространяет публичный ключ, надёжно сохраняя приватный. Отправитель шифрует данные с помощью приватного ключа и отправляет их получателю. Получатель дешифрует данные, используя публичный ключ отправителя.

Цифровые сертификаты

- Цифровой сертификат – структура данных, связывающая значения открытого ключа с идентифицирующей информацией о каком-либо субъекте.
- Сертификаты выдаются доверенной третьей стороной, которая может гарантировать, что данный ключ принадлежит именно этому субъекту:
 - Удостоверяющий центр формирует пару ключей и приватный передаёт подписчику.
 - Подписчик формирует цифровую подпись ключа и отправляет удостоверяющему центру.
 - Удостоверяющий центр передаёт подписчику сертификат, однозначно идентифицирующий подписчика.

<http://edu.softline.ru/>



Цифровые сертификаты необходимы для удостоверения в том, что компания, производящая зашифрованную передачу данных, является той, за которую себя выдаёт. В повседневной жизни это требуется для публичных организаций, работающих с конфиденциальными данными. В условиях небольшой компании достаточно использовать *самоподписанные* сертификаты.

Все сертификаты выдаются CA (Certification Authority), по сути являющимся доверенной компанией. CA формирует цифровой сертификат, а это — открытый ключ+идентификатор владельца+дополнительные атрибуты. Сертификаты формируются по стандарту X.509.

Самоподписанный сертификат, это сертификат, подписанный не CA, являющимся доверенным, а своим CA, про который никто в сети не знает, а следовательно, в качестве доверенного он использоваться не может. Но нужно понимать чёткую разницу, отсутствие доверия не означает отсутствие шифрования. Шифрование будет производиться в том же объёме, что и в случае с сертификатами, подписанными доверенными CA. Самоподписанным сертификат называется потому что в качестве подписчика и CA выступает одно и то же лицо.

Рассмотрим процедуру создания CA. Для этого будет использоваться openssl.

Первое, что необходимо сделать — сгенерировать приватный ключ:

```
# openssl genrsa -des3 -out ./ca.key
```

Далее необходимо сформировать запрос. Во время формирования вы должны будете ответить на ряд вопросов. Самый важный из них — Common Name. В этом поле вы должны указать имя хоста, для которого этот сертификат формируется. Если вы планируете использовать его для сайта по адресу zoneX.example.com, то и в поле Common Name вы должны указать это доменное имя.

```
# openssl req -new -key ./ca.key -out ./ca.csr
```

```
Country Name (2 letter code) [GB]:RU
```

```
State or Province Name (full name) [Berkshire]:Ural
```

```
Locality Name (eg, city) [Newbury]:Ekaterinburg
```

```
Organization Name (eg, company) [My Company Ltd]:Company
```



```
Organizational Unit Name (eg, section) []:  
Common Name (eg, your name or your server's hostname) []:example.com  
Email Address []:admin@example.com
```

Следующим шагом должна стать отправка запросу центру сертификации, но поскольку мы создаём самоподписанный сертификат, то и подписывать его будем сами:

```
# openssl x509 -req -in ca.csr -signkey ca.key -out ca.crt -days 365
```

Корневой сертификат готов и мы можем приступить к генерированию дочерних:

```
# openssl genrsa -out pop3.key 2048
```

```
# openssl req -new -key pop3.key -out pop3.csr
```

```
# openssl x509 -req -in pop3.csr -out pop3.crt -CA ca.crt -CAkey ca.key -  
CAcreateserial
```

```
Country Name (2 letter code) [GB]:RU  
State or Province Name (full name) [Berkshire]:Ural  
Locality Name (eg, city) [Newbury]:Ekaterinburg  
Organization Name (eg, company) [My Company Ltd]:Company  
Organizational Unit Name (eg, section) []:  
Common Name (eg, your name or your server's hostname) []:pop3.example.com  
Email Address []:
```

Лабораторная работа.

Лабораторная работа 10

Упражнения:

1. Использование библиотеки `libwrap.so`.

Необходимые машины:

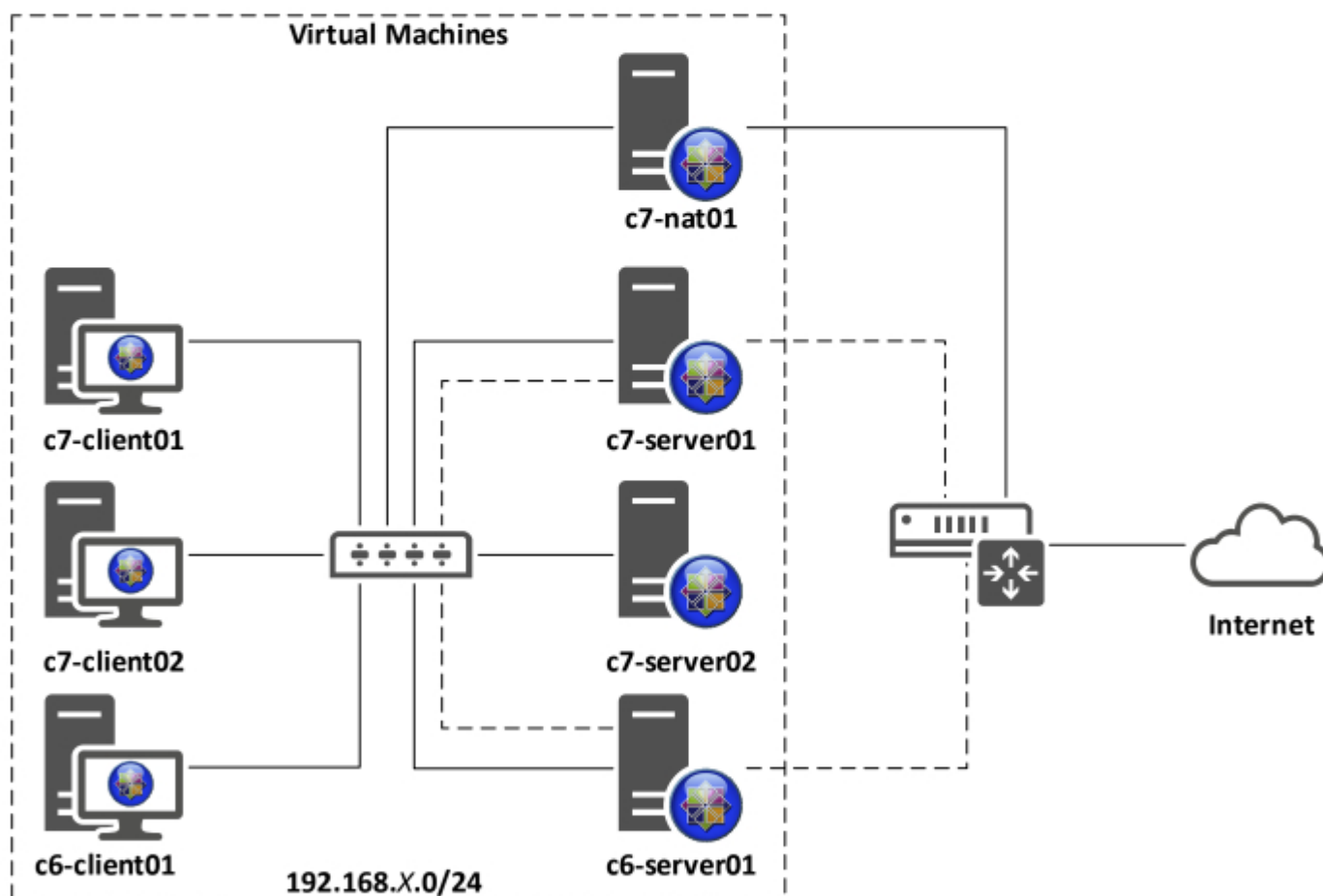
- `c7-nat01`.
- `c7-server01`.
- `c7-client01`.

Учетные данные:

- Пользователь: `root`.
- Пароль: `Pa$$w0rd`.



<http://edu.softline.ru/>



Упражнение 1 – Использование библиотеки libwrap.so.

Перед началом упражнения установите Apache HTTP Server на машине c7-server01.

Ваша задача обеспечить безопасность различных сервисов с помощью библиотеки **libwrap**.

1. Проверьте, поддерживает ли vsftpd библиотеку **libwrap**:

```
# ldd `which httpd` | grep libwrap
```

2. Теперь, когда вы убедились, что **libwrap** поддерживается, перейдём к ограничению доступа. Запретим HTTP-соединения для всех узлов сети 192.168.0.0/24. Для этого в файле **hosts.deny** создайте запись:

```
httpd: 192.168.0.0/24
```

3. Выполните попытку соединения с HTTP-сервером.
4. Теперь разрешим соединение для любого, выбранного вами узла:

```
vsftpd: 192.168.0.0/24 EXCEPT 192.168.0.51
```

5. Выполните соединение с HTTP-сервером.
6. А теперь попробуйте внести изменение в файл **/etc/hosts.allow** и сказать, что получилось:

```
httpd: 192.168.0.0/24
```

