**Лабораторная работа 1.**

**Изучение базовых команд Linux.**

**Основные теоретические сведения**

**Цель:** Первичное знакомство с командным интерпретатором. Изучение базовых команд операционной системы Linux.

**Теоретическая часть:**

Среди всех элементов операционной системы Linux самым важным, является командная строка (Терминал). Оболочка во многом определяет богатые возможности и гибкость операционной системы Linux. С помощью командной строки можно выполнять действия, которые были бы немыслимы при работе с графическим пользовательским интерфейсом. Независимо от того, KDE или GNOME, оказывается, что многие действия гораздо быстрее и эффективнее выполнить, пользуясь только командной строкой. Освоение Linux стоит начинать с изучения средств командной оболочки.

**Файлы и ничего кроме файлов**

Все, с чем Вы встретитесь в операционной системе Linux, - это файлы. Абсолютно все! Очевидно, что текстовый документ - это файл. Изображения, аудиоданные в формате МР3 и видеофрагменты - это несомненно файлы. Каталоги - это тоже файлы, содержащие информацию о других файлах. Дисковые устройства - это большие файлы. Сетевые соединения тоже файлы. Даже исполняемый процесс - это файл. С точки зрения операционной системы Linux файл представляет собой поток битов или байтов. Система не интересуется тем, что означает каждый байт. Это забота конкретных программ, выполняющихся в операционной системе Linux. Для операционной системы Linux и документ, и сетевое соединение всего лишь файлы. Как обрабатывать текстовый документ, знает редактор, а сетевое приложение умеет работать с сетевым соединением.

В отличие от Windows и МасOS в операционной системе Linux имена файлов чувствительны к регистру символов. В частности, Вы можете встретить в одном каталоге все три файла которые приведены ниже в качестве примера:

* Sit.txt
* sIt.txt
* SIT.txt

С точки зрения файловой операционной системы Linux - это различные имена файлов. Если вы попытаетесь создать файлы с этими же именами в Windows или МасOS, то вероятнее всего попытка увенчается провалом, и система предложит Вам выбрать другое имя для файла.

Чувствительность к регистру символов также означает, что при вводе команд они должны в точности совпадать с именами файлов, поддерживающих их. Так, например, удаляя файл с помощью команды rm, нельзя вводить RM, Rm или rM. Надо также следить за написанием имен, задаваемых в качестве параметров. Если вы захотите удалить файл «SIT.txt», а укажете имя Sit.txt, вы лишитесь совсем не того файла, с которым предполагали расстаться.

**Предупреждение**

Список специальных символов которые не рекомендуется использовать в названиях файлов.

/ - Нельзя использовать ни при каких обстоятельствах

\ - Должен быть предварен таким же символом. Применять не рекомендуется

- - Нельзя использовать в начале имени файла или каталога

[] - Каждый из этих символов должен быть предварен обратной косой чертой. Применять не рекомендуется

{} - Каждый из этих символов должен быть предварен обратной косой чертой. Применять не рекомендуется

\* - Должен быть предварен обратной косой чертой. Применять не рекомендуется

? - Должен быть предварен обратной косой чертой. Применять не рекомендуется

' - Должен быть предварен обратной косой чертой. Применять не рекомендуется

" - Должен быть предварен обратной косой чертой. Применять не рекомендуется

**Групповые операции:**

Предположим, что в одном из каталогов на вашем компьютере содержатся сто файлов с изображениями и два текстовых файла. Ваша задача удалить все файлы с изображениями за исключением двух текстовых файлов. Удалять файлы по одному - это утомительное занятие. В операционных системах Linux для автоматизации данного процесса можно применять символы групповых операций. Групповые операции задаются посредством звездочки (\*), знака вопроса (?) и квадратных скобок ( [ ] ).

**Пример использования групповых операций:**

Групповая операция с применение » \* » - отмечает любое (в том числе нулевое) количество любых символов.

rm sit1\*.\* Удаляться файлы : sit1.txt, sit1.jpg, sit11.jpg, sit123123.txt

rm sit\*.jpg Удаляться файлы : sit1.jpg, sit11.jpg

rm \*txt Удаляться файлы : sit1.txt, sit123123.txt

rm sit\* Удаляться файлы : sit1.txt, sit1.jpg, sit11.jpg, sit123123.txt

rm \* Удалятся все файлы в каталоге

Групповая операция с применение » ? «. Символ » ? » - соответствует одному произвольному символу.

rm sit1?.jpg Удалится файл : sit11.jpg, но не sit1.txt, sit1.jpg, sit123123.txt

rm sit?.jpg Удалится файл : sit1.jpg, но не sit1.txt, sit11.jpg, sit123123.txt

rm sit?.\* Удаляться файлы : sit1.txt, sit1.jpg, но не sit11.jpg, sit123123.txt

Групповая операция с применение » [] «. Квадратные скобки позволяют задавать один символ из набора или символ, принадлежащий определенному диапазону.

rm sit[0-1].txt Удалится файл : sit1.txt, но не sit1.jpg, sit11.jpg, sit123123.txt

rm sit1[0-2].jpg Удалится файл : sit11.jpg, но не sit1.txt, sit1.jpg, sit123123.txt

**Консольные команды:**

* $ pwd - определить текущий каталог.
* $ cd [имя каталога] — осуществить переход в заданный каталог.
* $ ls [имя каталога] - просмотреть список файлов и подкаталогов.
* $ mkdir [имя каталога] — создать каталог с заданным именем.
* $ cp <имя файла 1> <имя файла 2> - скопировать файл «имя файла 1» в файл «имя файла 2», например: cp first.txt copy1.txt.
* $ mv <имя файла 1> <имя файла 2> - переименовать файл «имя файла 1» в файл «имя файла 2», например: mv first.txt orig.txt.
* $ ln «имя файла» «имя ссылки» - создать жёсткую ссылку «имя ссылки» на файл «имя файла». Пример: ln orig.txt copy2.txt.
* $ ln -s «имя файла» «имя ссылки» - создать символическую ссылку «имя ссылки» на файл «имя файла». Пример: ln -s orig.txt copy2.txt.
* $ rm <имя файла> - удалить файл.
* $ touch <имя файла> - создание файла.
* $ man <название команды> - получение справочной документации о выбранной команде.

**Задания к лабораторной работе**

* Откройте терминал.
* Ознакомьтесь с возможностями команды pwd c помощью команды man:
* Определите текущий каталог, в котором вы находитесь командой pwd:
* Ознакомьтесь с возможностями команды cd c помощью команды man:
* Перейдите в корневой каталог командой cd
* Ознакомьтесь с возможностями команды ls c помощью команды man:
* Просмотрите содержимое корневого каталога командой ls:
* Сделайте копию экрана для использования в отчете по лабораторной работе .
* Вернитесь в домашний каталог, используя команду cd без параметров:
* Ознакомьтесь с возможностями команды mkdir c помощью команды man:
* Создайте каталог «test», используя команду mkdir:
* Перейдите в каталог «test», используя команду cd:
* Просмотрите содержимое каталога, используя команду ls:
* Создайте каталог «test2», используя команду mkdir:
* Ознакомьтесь с возможностями команды touch c помощью команды man:
* Создайте файл «text» в каталоге «test2» используя команду touch:
* Ознакомьтесь с возможностями команды mv c помощью команды man:
* Переименуйте файл «text» в «textSIT» используя команду mv
* Ознакомьтесь с возможностями команды cp c помощью команды man:
* Скопируйте файл «textSIT» в каталог «test2» под именем «copy.txt», используя команду cp:
* Ознакомьтесь с возможностями команды ln c помощью команды man:
* Создайте жесткую ссылку «link» на файл «copy.txt» используя команду ln:
* Создайте символическую ссылку «simlink» на файл «copy.txt» используя команду ln:
* Просмотрите результаты в текущем каталоге при помощи команды ls с аргументами la:
* Сделайте копию экрана для использования в отчете по лабораторной работе .
* Удалите созданные вами файлы и ссылки в лабораторной работе используя команду rm
* Сделайте копию экрана для использования в отчете по лабораторной работе .

**Вопросы к лабораторной работе**

1. Чем отличается вывод команд ls -F и ls -la?
2. С помощью какой команды и как можно переместить файл в другой каталог?
3. Какие действия вы совершаете, нажимая на кнопки «стрелка вверх» и «стрелка вниз»?
4. Куда вы переходите, выполнив команду cd без параметров?
5. Как посмотреть, какие еще параметры можно задать команде ls?
6. Что такое «жесткая ссылка»?
7. Что такое «символическая ссылка»?
8. Как осуществить просмотр подкаталогов и их содержимого ?
9. Как осуществить вывод содержимого каталога с запятыми в качестве разделителя?
10. Как осуществить просмотр скрытых файлов в домашнем каталоге?
11. Как осуществить создание нового каталога и необходимых подкаталогов рекурсивно?
12. Как осуществить рекурсивное копирование всех файлов из одного каталога в другой?
13. Как осуществить рекурсивное копирование всех файлов и подкаталогов из одного каталога в другой?
14. Как рекурсивно удалить все файлы и подкаталоги в определенном каталоге?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

**Лабораторная работа №2.**

**Разграничение прав доступа**

**Основные теоретические сведения**

**Цель:** Изучение механизмов управления доступа к ресурсам, прав доступа. Постижение понятия пользователя и группы. Приобретение практических навыков управления пользователями при помощи консольных утилит. Приобретение навыков работы с правами пользователей и правами на файлы, каталоги при помощи консольных утилит.

**Теоретическая часть**

У каждого объекта (файла) есть уникальное имя, по которому к нему можно обращаться, и конечный набор операций, которые процессы могут выполнять в отношении этого объекта. Файлу свойственны операции read, write и execute.

Совершенно очевидно, что нужен способ запрещения процессам доступа к тем объектам, к которым у них нет прав доступа. Более того, этот механизм должен также предоставлять возможность при необходимости ограничивать процессы поднабором разрешенных операций. Например, процессу A может быть дано право проводить чтение данных из файла F, но не разрешено вести запись в этот файл.

**Права доступа** означают разрешение на выполнение той или иной операции (чтение, записи, исполнения).

Когда пользователь входит в систему, его оболочка получает UID и GID (UID – идентификатор пользователя, GID - идентификатор группы), которые содержатся в его записи в файле паролей, и они наследуются всеми его дочерними процессами. Представляя любую комбинацию (UID, GID), можно составить полный список всех объектов (файлов, включая устройства ввода-вывода, которые представлены в виде специальных файлов и т.д.), к которым процесс может обратиться с указанием возможного типа доступа (чтение, запись, исполнение).

Два процесса с одинаковой комбинацией (UID, GID) будут иметь абсолютно одинаковый доступ к одинаковому набору объектов. Процессы с различающимися значениями (UID, GID) будут иметь доступ к разным наборам файлов, хотя, может быть, и со значительным перекрытием этих наборов.

**SUID (Set User ID)**

Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В операционных системах Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ на запись к важным системным файлам, например /etc/passwd, который принадлежит суперпользователю root. Если на исполняемый файл установлен бит suid, то при выполнении эта программа автоматически меняет «эффективный userID» на идентификатор того пользователя, который является владельцем этого файла. То есть, не зависимо от того - кто запускает эту программу, она при выполнении имеет права хозяина этого файла.

**SGID (Set Group ID)**

Аналогичен SUID, но относиться к группе. При этом, если для каталога установлен бит SGID, то создаваемые в нем объекты будут получать группу владельца каталога, а не пользователя.

**Практические примеры**

**Узнать права на файл/директорию**

sit@ubuntu:~$ ls -l /bin/ls

-rwxr-xr-x 1 root root 129280 Feb 18 2016 /bin/ls

Права доступа состоят из трех троек символов. Первая тройка представляет права владельца файла, вторая представляет права группы файла и третья права всех остальных пользователей.

В нашем случае это :

* «rwx» - Права владельца файла
* «r-x» - Права группы файла
* «r-x» - Права всех остальных на файл.

Символ «r» означает, что чтение (просмотр данных содержащихся в файле) разрешено, «w» означает запись (изменение, а также удаление данных) разрешено и «x» означает исполнение (запуск программы разрешен).

Таким образом, если в целом посмотреть на права мы увидим, что кому угодно разрешено читать содержимое и исполнять этот файл, но только владельцу (root) разрешено как либо модифицировать этот файл. Иными словами, нормальным пользователям разрешено копировать содержимое этого файла, то только root может изменять или удалять его.

**Определение текущего пользователя и групп в которых он состоит**

Перед тем, как изменять владельца или группу которой принадлежит файл, необходимо уметь определять текущего пользователя и группу к которой он принадлежит. Чтобы узнать под каким пользователем вы работаете, наберите whoami:

sit@ubuntu:~$ whoami

sit

Для определения в каких группах состоит пользователь sit, необходимо воспользоваться командой groups:

sit@ubuntu:~$ groups

sit adm cdrom sudo dip plugdev lxd lpadmin sambashare

Из этого примера видно, что пользователь sit состоит в группах sit, adm, cdrom, sudo, dip, plugdev, lxd, lpadmin, sambashare. Если вы хотите посмотреть, в каких группах состоит другой пользователь, то передайте его имя в качестве аргумента.

sit@ubuntu:~$ groups root

root : root

**Изменение пользователя и группы владельца**

Чтобы изменить владельца или группу файла (или другого объекта) используется команды chown или chgrp соответственно. Сначала нужно передать имя группы или владельца, а потом список файлов.

chown sit /home/sit/itmo.txt

chgrp sit /home/sit/itmo.txt

Можно также изменять пользователя и группу одновременно используя команду chown в другой форме:

chown sit:sit /home/sit/itmo.txt

**Предупреждение**

Вы не можете использовать команду chown без прав суперпользователя, но chgrp может быть использована всеми, чтобы изменить группу-владельца файла на ту группу, к которой они принадлежат.

**Знакомство с chmod**

chown и chgrp используются для изменения владельца и группы объекта файловой системы, но кроме них существует и другая программа, называемая chmod, которая используется для изменения прав доступа на чтение, запись и исполнение, которые мы видим в выводе команды ls -l. Команда chmod использует два и более аргументов: метод, описывающий как именно необходимо изменить права доступа с последующим именем файла или списком файлов, к которым необходимо применить эти изменения:

chmod +x /home/sit/itmo.sh

В примере выше в качестве метода указано +x. Как можно догадаться, метод +x указывает chmod, что файл необходимо сделать исполняемым для пользователя, группы и для всех остальных. Если мы решим отнять все права на исполнение файла, то сделаем вот так:

chmod +x /home/sit/itmo.sh

**Разделение между пользователем, группой и всеми остальными**

Часто бывает удобно изменить только один или два набора прав доступа за раз. Чтобы сделать это, просто необходимо использовать специальный символ для обозначения набора прав доступа, который необходимо изменить, со знаком «+»» или «—» перед ним. Символ «u» для пользователя, «g» для группы и «o» для остальных пользователей.

chmod go-w /home/sit/itmo.sh

Данный пример удаляет право на запись для группы и всех остальных пользователей, но оставляет права владельца нетронутыми.

**Числовые режимы**

Существует еще один достаточно распространенный способ указания прав: использование четырехзначных восьмеричных чисел. Этот синтаксис, называется числовым синтаксисом прав доступа, где каждая цифра представляет тройку разрешений. Например, в 0777, 777 устанавливают флаги для владельца, группы, и остальных пользователей. Ниже таблица показывающая как транслируются права доступа на числовые значения.

Режим Число

rwx 7

rw- 6

r-x 5

r-- 4

-wx 3

-w- 2

--x 1

--- 0

**umask**

Когда процесс создает новый файл, он указывает, какие права доступа нужно задать для данного файла. Зачастую запрашиваются права 0666 (чтение и запись всеми), что дает больше разрешений, чем необходимо в большинстве случаев. К счастью, каждый раз, когда в Linux создается новый файл, система обращается к параметру, называемому umask. Система использует значение umask чтобы понизить изначально задаваемые разрешения на что-то более разумное и безопасное. Вы можете просмотреть текущие настройки umask набрав umask в командной строке:

sit@ubuntu:~$ umask

0002

В Linux-системах значением по умолчанию для umask является 0022, что позволяет другим читать ваши новые файлы (если они могут до них добраться), но не изменять их. Чтобы автоматически обеспечивать больший уровень защищенности для создаваемых файлов, можно изменить настройки umask:

sit@ubuntu:~$ umask 0077

Такое значение umask приведет к тому, что группа и прочие не будут иметь совершенно никаких прав доступа для всех, вновь созданных файлов.

В отличие от «обычного» назначения прав доступа к файлу, umask задает какие права доступа **должны быть отключены**. Снова посмотрим на таблицу соответствия значений чисел и методов:

Режим Число

rwx 7

rw- 6

r-x 5

r-- 4

-wx 3

-w- 2

--x 1

--- 0

Воспользовавшись этой таблицей мы видим, что последние три знака в 0077 обозначают —rwxrwx. umask показывает системе, какие права доступа отключить. Совместив первое и второе становится видно, что все права для группы и остальных пользователей будут отключены, в то время как права владельца останутся нетронутыми.

**Изменение suid и sgid**

Способ установки и удаления битов suid и sgid чрезвычайно прост. Чтобы задать бит suid:

chmod u+s /home/sit/itmo.sh

Чтобы задать бит sgid:

chmod g+s /home/sit/itmo/

**Определение первого знака прав доступа**

Он используется для задания битов sticky, suid и sgid совместно с правами доступа:

suid sgid sticky режим

on on on 7

on on off 6

on off on 5

on off off 4

off on on 3

off on off 2

off off on 1

off off off 0

Ниже приведен пример того, как использовать четырех значный режим для установки прав доступа на директорию.

sit@ubuntu:~$ chmod 4775 /home/sit/itmo

sit@ubuntu:~$ ls -l /home/sit/itmo

-rwsrwxr-x 1 sit sit 0 Sep 9 12:42 /home/sit/itmo

**Консольные команды:**

* id <печать идентификатора пользователя>
* chgrp <изменить группу файла>
* chown <изменить владельца и группу файлов>
* chmod <изменить права доступа к файлу>
* usermod <изменение параметров учетной записи пользователя>
* useradd <создание нового пользователя>
* userdel <удаление пользователя>
* whoami <определение текущего пользователя>
* umask <определение или установление маски прав доступа для вновь создаваемых файлов>
* sudo su <получение прав суперпользователя>
* groups <определение к каким группам принадлежит пользователь>

**Задания к лабораторной работе**

* Откройте два терминала (в серверных Linux для переключения между терминалами (tty) обычно используется сочетание клавиш Alt+F[1-5]). В одном из них получите права суперпользователя используя команду sudo su:
* Изучите как создать пользователя с домашним каталогом с помощью команды useradd из справочной документации man
* Используя useradd создайте пользователя «sit2» c домашним каталогом «sit2».
* Установите пароль для нового пользователя «sit2» с помощью команды passwd sit2
* Выйдите из суперпользователя командой exit
* Войдите под первым терминалом в пользователя «sit», во втором в пользователя «sit2».
* Посмотрите какой идентификатор получил пользователь «sit» и пользователь «sit2» используя команду id
* Посмотрите права доступа на домашний каталог пользователей «sit» и «sit2», используя команду ls
* Создайте файл под пользователем «sit2» с маской 0077 используя umask
* Попробуйте прочитать его содержимое под пользователем «sit» используя команду cat
* Измените права доступа на файл так, чтобы пользователь «sit» мог записывать в файл, но не читать его.
* Запишите текстовую информацию в файл из под пользователя «sit» используя консольный текстовый редактор vi или nano
* Проверьте права на файл, и прочитайте его содержимое из под пользователя «sit2»
* Создайте каталог из под пользователя «sit2»
* Установите права записи для группы пользователей на данный каталог
* Добавьте пользователя «sit» в группу «sit2» с помощью команды usermod
* Проверьте в какие группы входит пользователь «sit»
* Создайте несколько файлов в каталоге, который был создан пользователем «sit2» из под пользователя «sit».
* Ознакомьтесь как удалить пользователя вместе с содержимым его домашнего каталога из справочной документации
* Удалите пользователя «sit2» вместе с его домашним каталогом.

**Вопросы к лабораторной работе**

1. Какой uid у пользователя sit2? В какие группы он входит?
2. Почему попытка удалить пользователя не удалась, и что нужно сделать для его удаления?
3. Какие права доступа установлены на домашний каталог пользователя «sit» ?
4. Как рекурсивно изменить права доступа на файлы в каталоге?
5. Как можно осуществлять переключение между пользователями в рамках одного терминала?
6. Как удалить пользователя при этом сохранив его домашний каталог и данные внутри него?
7. Какое значение umask нужно установить, чтобы владелец и группа имели право на чтение, запись и исполнение, а все остальные пользователи не имели никаких прав?
8. Как рекурсивно снять все suid биты с файлов в каталоге?
9. Как разрешить программе (файлу) исполняться?
10. Что такое бит sticky? Для чего он предназначен?
11. Зачем нужны uid и gid?
12. Почему uid пользователя задается больше 1000?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

**Лабораторная работа №3.**

**Файловые подсистемы.**

**Основные теоретические сведения**

**Цель:** Получение теоретических и практических навыков работы с таблицами разделов(MBR и GPT), создания разделов и файловых систем.

**Консольные команды:**

* fdisk <параметры> - Консольная программа для управления дисками (Работает только с MBR).
* parted <параметры> - Консольная программа для управления дисками (Работает как с MBR, так и с GPT).
* dd <параметры> - Консольная программа копирования данных.
* mkfs.<тип файловой системы> <раздел диска> - Класс консольных команд создания файловых систем на разделах.
* mount -t <тип файловой системы> <раздел диска> <точка монтирования> - Консольная программа монтирования разделов жесткого диска.

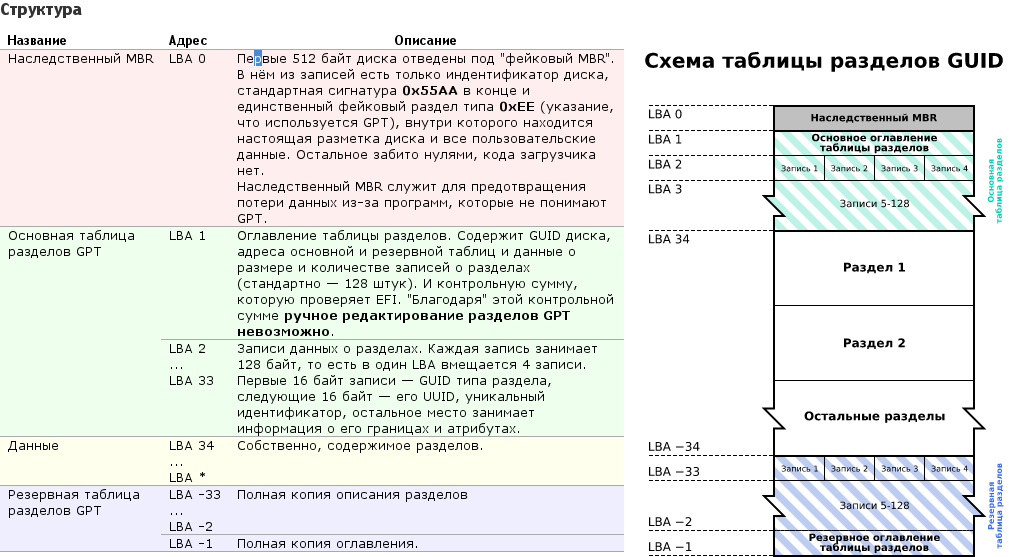
Диск делится на разделы. Как именно диск делится на разделы, определяется таблицей разделов. Таблицы разделов бывают двух типов : MBR и GPT.

**Структура MBR**

Первые 512 байт (первый сектор диска) главного устройства хранения данных занимает MBR (Master Boot Record). В состав MBR входит 446 байт кода загрузчика, четыре записи по 16 байт - это таблица разделов, 2 байта сигнатуры. Таблица разделов может состоять из первичных разделов (до 4) и логических разделов(до 128).

**Структура GPT**

GUID Partition Table, аббр. GPT — стандарт формата размещения таблиц разделов на физическом жестком диске. Он является частью расширяемого микропрограммного интерфейса (англ. Extensible Firmware Interface, EFI) — стандарта, предложенного Intel на смену BIOS. EFI использует GPT там, где BIOS использует главную загрузочную запись (англ. Master Boot Record, MBR). В GPT нет собственной программы-загрузчика, вместо этого он работает в паре с EFI. Внутри GPT используется адресация логических блоков LBA, которая абстрагирована от физики устройства (в отличие от CHS — «Цилиндр-Головка-Сектор»). Каждый логический блок занимает 512 байт. LBA 0 — первые 512 байт диска, LBA 1 — следующие, и так далее. Отрицательные значения LBA означают смещение в блоках с конца диска. Последний блок имеет смещение «-1» (LBA -1).



**Задания к лабораторной работе**

На данный момент наиболее распространенной схемой разбиения дисков является MBR. Но с развитием средств хранения данных и их объемов, возможностей MBR становится недостаточно. Это связанно с невозможностью обеспечивать доступ к разделу диска емкостью более чем 2.2 TB. На сегодняшний день уже доступны диски емкостью более 6 TB, а так же, применяются различные технологии по объединению дисков в массивы, такие как RAID и LVM. Таким образом, применение схемы разбиения дисков на основе GPT становится все более актуальным.

**Процесс загрузки**

Процесс загрузки компьютера является многоступенчатым процессом, и начинается он с инициализации системных устройств набором микропрограмм, называемых BIOS (Basic Input/Output System), которые выполняются при старте системы. После того, как BIOS успешно проверит системные устройства, идет процесс поиска загрузчика в MBR устройств хранения (CD/DVD диски, USB диск, HDD, SSD и др.) или на первом разделе устройства. После того, как загрузчик получил управление, он получает таблицу разделов и готовит к загрузке операционную систему. В семействе загрузчиков GNU/Linux яркими представителями являются GRUB и LILO. В них MBR состоит из небольшой части ассемблерного кода. Стандартный загрузчик Windows/DOS в состоянии проверить только активный раздел, считать несколько секторов с этого раздела и затем передать управление операционной системе. Он не в состоянии загрузить Linux, так как не наделен необходимым функционалом. GRand Unified Bootloader (GRUB) - это стандартный загрузчик для операционных систем семейства GNU/Linux, и всем пользователям рекомендуется по умолчанию установить его в MBR, для того чтобы иметь возможность загружать операционную систему с любого раздела, первичного или логического.

**Пример работы с MBR**

Существует специальный набор команд для работы с MBR. Так как он расположен на диске, то может быть сохранен и, в случае необходимости, восстановлен.

* dd if=/dev/sda of=/path/mbr-backup bs=512 count=1 - Для создания резервной копии MBR
* dd if=/path/mbr-backup of=/dev/sda bs=512 count=1 - Для восстановления MBR
* dd if=/dev/sda of=/path/mbr-boot-code bs=446 count=1 – Для сохранения только загрузочного кода
* dd if=/dev/sda of=/path/mbr-part-table bs=1 count=66 skip=446 - Для сохранения только таблицы разделов
* dd if=/path/mbr-backup of=/dev/sda bs=446 count=1 – Для восстановление загрузочного кода из файла mbr-backup
* dd if=/path/mbr-backup of=/dev/sda bs=1 skip=446 seek=466 count=66 - Для восстановления только таблицы разделов
* dd if=/dev/zero of=/dev/sda bs=446 count=1 - Для очистки MBR, но при этом оставить таблицу разделов

**Задания к лабораторной работе**

* Добавьте в виртуальную машину с операционной системой Linux виртуальный жесткий диск (делается это в настройках виртуальной машины).
* Запустите виртуальную машину с операционной системой Linux.
* Ознакомьтесь с командой fdisk и ее возможностями из справочной документации.
* Создайте таблицу разделов (3 первичных и 1 логический) с помощью команды fdisk на **добавленном** виртуальном диске (обычно это диск /dev/sdb).
* Запишите изменения на диск
* Проверьте факт создания разделов используя команду fdisk. (Так же, создание разделов можно проверить используя команду ls /dev/sd\*)
* Отформатируйте созданные разделы в файловую систему ext4.
* Ознакомьтесь с командами mount и umount и их возможностями из справочной документации.
* Смонтируйте созданные разделы и создайте там произвольные файлы.
* Сделайте резервную копию MBR с помощью утилиты DD.
* Сотрите таблицу разделов MBR с помощью утилиты DD.
* Восстановите MBR с помощью утилиты DD.
* Смонтируйте разделы и проверьте целостность данных.
* Отмонтируйте разделы.
* Установите gdisk <sudo apt-get install gdisk>
* Создайте таблицу разделов GPT (5 первичных разделов) с помощью gdisk.
* Отформатируйте созданные разделы в файловую систему ext3.
* Смонтируйте созданные разделы и создайте там произвольные файлы.
* Сделайте резервную копию GPT с помощью утилиты DD, предварительно определив необходимое количество байт для резервной копии.
* Сотрите GPT с помощью утилиты DD.
* Восстановите GPT с помощью утилиты DD.
* Смонтируйте разделы и проверьте целостность данных.
* Отмонтируйте разделы.
* Определите достоинства и недостатки таблиц разделов MBR и GPT.

**Вопросы к лабораторной работе**

1. Что записано в первом секторе главной загрузочной записи MBR ?
2. Функциональное назначение MBR и GPT ?
3. Структура GPT.
4. Какое максимальное количество первичных разделов можно создать при использовании таблицы разделов MBR ?
5. Какое максимальное количество первичных разделов можно создать при использовании таблицы разделов GPT ?
6. Как сохранить информацию о структуре MBR ?
7. Как создать 10 разделов c файловой системой ext3 на диске в таблице разделов MBR ?
8. Как стереть код загрузчика в MBR ?
9. Как можно смонтировать раздел диска с файловой системой в режиме только для чтения?
10. Как можно осуществить восстановление GPT разделов в случае сбоев?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

[Next](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/lr4.html)[Previous](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/lr2.html)

# Лабораторная работа №4. Обеспечение целостности и доступности данных. Raid, LVM.

## Основные теоретические сведения

**Цель:** Получение теоретических и практических навыков построения и управления RAID массивами и логическими томами

**Консольные команды:**

* mdadm <параметры> - Консольная программа управления программными RAID массивами в Linux.
* lvm <параметры> - Консольная программа управления логическими томами LVM.
* parted <параметры> - Консольная программа для управления дисками
* watch <параметры> - Консольная программа, которая позволяет следить за изменениями в выводе команды.

### RAID

RAID (Redundant Array of Independent Disks - избыточный массив независимых жестких дисков) - массив, состоящий из нескольких дисков, управляемых программным или аппаратным контроллером, связанных между собой и воспринимаемых как единое целое. В зависимости от того, какой тип массива используется, может обеспечивать различные степени быстродействия и отказоустойчивости. Служит для повышения надежности хранения данных и/или для повышения скорости чтения/записи информации.

Калифорнийский университет в Беркли предложил следующие уровни спецификации RAID, которые являются стандартом во всем мире:

* RAID 0 представлен как дисковый массив повышенной производительности, без отказоустойчивости. (Требуется минимум 2 диска)
* RAID 1 определен как зеркальный дисковый массив. (Требуется минимум 2 диска)
* RAID 2 массивы, в которых применяется код Хемминга. (Требуется минимум 7 дисков, для рационального использования)
* RAID 3 и 4 используют массив дисков с чередованием и выделенным диском четности. (Требуется минимум 4 диска)
* RAID 5 используют массив дисков с чередованием и «невыделенным диском четности». (Требуется минимум 3 диска)
* RAID 6 используют массив дисков с чередованием и двумя независимыми «четностями» блоков. (Требуется минимум 4 диска)
* RAID 10 - RAID 0, построенный из RAID 1 массивов. (Требуется минимум 4 диска, четное количество)
* RAID 50 - RAID 0, построенный из RAID 5 массивов. (Требуется минимум 6 дисков, четное количество)
* RAID 60 - RAID 0, построенный из RAID 6 массивов. (Требуется минимум 8 дисков, четное количество)

**Пример создания RAID 10**

Проверим наличие виртуальных дисков.

sit@sit:~$ sudo parted -l

Model: ATA VBOX HARDDISK (scsi)

Disk /dev/sda: 21.5GB

Sector size (logical/physical): 512B/512B

Partition Table: msdos

Disk Flags:

Number Start End Size Type File system Flags

1 1049kB 256MB 255MB primary ext2 boot

2 257MB 21.5GB 21.2GB extended

5 257MB 21.5GB 21.2GB logical lvm

Error: /dev/sdb: unrecognised disk label

Model: ATA VBOX HARDDISK (scsi)

Disk /dev/sdb: 8590MB

Sector size (logical/physical): 512B/512B

Partition Table: unknown

Disk Flags:

Error: /dev/sdc: unrecognised disk label

Model: ATA VBOX HARDDISK (scsi)

Disk /dev/sdc: 8590MB

Sector size (logical/physical): 512B/512B

Partition Table: unknown

Disk Flags:

Error: /dev/sdd: unrecognised disk label

Model: ATA VBOX HARDDISK (scsi)

Disk /dev/sdd: 8590MB

Sector size (logical/physical): 512B/512B

Partition Table: unknown

Disk Flags:

Error: /dev/sde: unrecognised disk label

Model: ATA VBOX HARDDISK (scsi)

Disk /dev/sde: 8590MB

Sector size (logical/physical): 512B/512B

Partition Table: unknown

Disk Flags:

Error: /dev/sdf: unrecognised disk label

Model: ATA VBOX HARDDISK (scsi)

Disk /dev/sdf: 8590MB

Sector size (logical/physical): 512B/512B

Partition Table: unknown

Disk Flags:

Model: Linux device-mapper (linear) (dm)

Disk /dev/mapper/sit--vg-swap\_1: 533MB

Sector size (logical/physical): 512B/512B

Partition Table: loop

Disk Flags:

Number Start End Size File system Flags

1 0.00B 533MB 533MB linux-swap(v1)

Model: Linux device-mapper (linear) (dm)

Disk /dev/mapper/sit--vg-root: 20.7GB

Sector size (logical/physical): 512B/512B

Partition Table: loop

Disk Flags:

Number Start End Size File system Flags

1 0.00B 20.7GB 20.7GB ext4

sit@sit:~$

**Задания к лабораторной работе**

Как видно из листинга, у нас присутствуют диски sda (на котором установлена операционная система Linux),sdb,sdc,sdd,sde,sdf. Теперь можно построить массив RAID 10 из дисков sdb, sdc, sdd и sde, а диск sdf пометим как диск горячей замены (применяется для горячей замены в случае отказа одного из дисков RAID массива).

**Предупреждение**

Необходимо открыть два терминала. В одном создается RAID массив, в другом осуществляется процесс наблюдения за созданием RAID массива.

Запустим процесс отслеживания состояния RAID массивов в терминале №1:

sit@sit:~$ sudo watch -n1 cat /proc/mdstat

Создадим RAID 10 в отдельном терминале №2:

sit@sit:~$ sudo mdadm -C /dev/md0 -l 10 -n 4 -x 1 /dev/sd[b-f]

[sudo] password for sit:

mdadm: Defaulting to version 1.2 metadata

mdadm: array /dev/md0 started.

sit@sit:~$

В терминале №1 наблюдаем процесс создания RAID 10:

Every 1.0s: cat /proc/mdstat Wed Sep 23 18:02:03 2015

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]

md0 : active raid10 sdf[4](S) sde[3] sdd[2] sdc[1] sdb[0]

16760832 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

[============>........] resync = 61.3% (10286144/16760832) finish=0.5min speed=201781K/sec

unused devices: <none>

Создадим раздел в 1GB c файловой системой ext4 на созданном RAID 10:

sit@sit:~$ sudo parted /dev/md0

[sudo] password for sit:

GNU Parted 3.2

Using /dev/md0

Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) mklabel

New disk label type? GPT

Warning: The existing disk label on /dev/md0 will be destroyed and all data on this disk will be lost. Do you want to continue?

Yes/No? yes

(parted) mkpart

Partition name? []?

File system type? [ext2]? ext4

Start? 0

End? 1GB

Warning: The resulting partition is not properly aligned for best performance.

Ignore/Cancel? Ignore

(parted) print

Model: Linux Software RAID Array (md)

Disk /dev/md0: 17.2GB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Disk Flags:

Number Start End Size File system Name Flags

1 17.4kB 1000MB 1000MB ext4

(parted)

Отформатируем созданный раздел в файловую систему ext4:

sit@sit:~$ sudo mkfs.ext4 /dev/md0p1

Смонтируем созданный раздел:

sudo mount -t ext4 /dev/md0p1 /mnt/

Скопируем файлы на раздел с файловой системой ext4:

sudo cp -R /var/log/\* /mnt/

Разрушим один диск и проверим целостность данных.:

Наблюдаем процесс как диск горячей замены встает на место сбойного диска

Every 1.0s: cat /proc/ Wed Sep 23 19:52:04 2015

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]

md0 : active raid10 sdf[4] sde[3] sdd[2] sdc[1] sdb[0](F)

16760832 blocks super 1.2 512K chunks 2 near-copies [4/3] [\_UUU]

[====>................] recovery = 21.8% (1832192/8380416) finish=0.4min speed=229024K/sec

unused devices: <none>

Убедимся в целостности данных на разделе:

sit@sit:~$ ls -la /mnt/

total 968

drwxr-xr-x 9 root root 4096 Sep 23 19:34 .

drwxr-xr-x 22 root root 4096 Sep 19 14:26 ..

-rw-r--r-- 1 root root 18625 Sep 23 19:34 alternatives.log

drwxr-xr-x 2 root root 4096 Sep 23 19:34 apt

-rw-r----- 1 root root 41820 Sep 23 19:34 auth.log

-rw-r--r-- 1 root root 63653 Sep 23 19:34 bootstrap.log

-rw------- 1 root root 0 Sep 23 19:34 btmp

drwxr-xr-x 2 root root 4096 Sep 23 19:34 dist-upgrade

-rw-r----- 1 root root 31 Sep 23 19:34 dmesg

-rw-r--r-- 1 root root 339677 Sep 23 19:34 dpkg.log

-rw-r--r-- 1 root root 32032 Sep 23 19:34 faillog

drwxr-xr-x 2 root root 4096 Sep 23 19:34 fsck

drwxr-xr-x 3 root root 4096 Sep 23 19:34 installer

-rw-r----- 1 root root 189514 Sep 23 19:34 kern.log

drwxr-xr-x 2 root root 4096 Sep 23 19:34 landscape

-rw-r--r-- 1 root root 292292 Sep 23 19:34 lastlog

drwx------ 8 root root 16384 Sep 23 19:32 lost+found

-rw-r----- 1 root root 173386 Sep 23 19:34 syslog

-rw-r----- 1 root root 3090 Sep 23 19:34 syslog.1

-rw-r----- 1 root root 591 Sep 23 19:34 syslog.2.gz

-rw-r----- 1 root root 30788 Sep 23 19:34 syslog.3.gz

drwxr-x--- 2 root root 4096 Sep 23 19:34 unattended-upgrades

-rw-r--r-- 1 root root 8832 Sep 23 19:34 wtmp

sit@sit:~$ sudo head -n 10 /mnt/auth.log

Sep 19 14:38:02 sit systemd-logind[506]: Watching system buttons on /dev/input/event0 (Power Button)

Sep 19 14:38:02 sit systemd-logind[506]: Watching system buttons on /dev/input/event1 (Sleep Button)

Sep 19 14:38:02 sit systemd-logind[506]: Watching system buttons on /dev/input/event5 (Video Bus)

Sep 19 14:38:02 sit systemd-logind[506]: New seat seat0.

Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event0 (Power Button)

Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event1 (Sleep Button)

Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event6 (Video Bus)

Sep 19 14:40:10 sit systemd-logind[508]: New seat seat0.

Sep 19 14:40:27 sit login[529]: pam\_unix(login:session): session opened for user sit by LOGIN(uid=0)

Sep 19 14:40:27 sit systemd-logind[508]: New session c1 of user sit.

sit@sit:~$ sudo head -n 10 /mnt/syslog

Sep 23 07:17:01 sit CRON[2263]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

Sep 23 08:17:01 sit CRON[2266]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

Sep 23 09:17:01 sit CRON[2269]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

Sep 23 10:17:01 sit CRON[2272]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

Sep 23 10:46:05 sit dhclient: DHCPREQUEST of 10.0.2.15 on eth0 to 10.0.2.2 port 67 (xid=0x6a9a8b24)

Sep 23 10:46:05 sit dhclient: DHCPACK of 10.0.2.15 from 10.0.2.2

Sep 23 10:46:05 sit dhclient: bound to 10.0.2.15 -- renewal in 42505 seconds.

Sep 23 11:17:01 sit CRON[2285]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

Sep 23 12:17:01 sit CRON[2288]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

Sep 23 13:17:01 sit CRON[2291]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

Сделаем имитацию замены извлечением и вставки нового диска.:

sit@sit:~$ sudo mdadm /dev/md0 -r /dev/sdb

mdadm: hot removed /dev/sdb from /dev/md0

sit@sit:~$ sudo mdadm /dev/md0 -a /dev/sdb

mdadm: added /dev/sdb

sit@sit:~$

Наблюдаем что диск sdb пометился как диск горячей замены.:

Every 1.0s: cat /proc/ Wed Sep 23 19:59:09 2015

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]

md0 : active raid10 sdb[5](S) sdf[4] sde[3] sdd[2] sdc[1]

16760832 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

unused devices: <none>

**Задания к лабораторной работе**

Для того чтобы остановить RAID используется параметр **–stop** команды mdadm.

Для очистки записи принадлежности к программному RAID используется параметр **–zero-superblock** команды mdadm.

### ****LVM****

LVM (Logical Volume Manager) - менеджер логических томов является уникальной системой управления дисковым пространством. Она позволяет с легкостью использовать и эффективно управлять дисковым пространством. Уменьшает общую нагруженность и сложность существующей системы. У логических томов, которые созданы через LVM, можно легко изменять размер, а названия, которые им даны, помогут в дальнейшем определить назначение тома.

* PV, Physical Volume или физический том. Чаще всего это раздел на диске или весь диск. К ним относят устройства программного и аппаратного RAID массивов (которые могут включать в себя еще несколько физических дисков). Физические тома объединяются и образуют группы томов.
* VG, Volume Group или группа томов. Это самый верхний уровень модели представления, которая используется в LVM. С одной стороны группа томов может состоять из физических томов, с другой- из логических томов и представлять собой единую структуру.
* LV, Logical Volume или логический том. Раздел в группе томов, тоже самое, что раздел диска в не-LVM системе. Является блочным устройством и, как следствие, может содержать файловую систему.
* PE, Physical Extent или физический экстент. Каждый физический том делится на блоки данных - физические экстенты. Они имеют размеры как и у логических экстентов.
* LE, Logical Extent или логический экстент. Каждый логический том также делится на блоки данных - логические экстенты. Размеры логических экстентов не меняются в рамках группы томов.

**Инициализация дисков и разделов**

Перед тем, как начать использовать диск или раздел в качестве физического тома, важно его проинициализировать. Осуществляется это с помощью команды **pvcreate**. Данная команда создаст в начале диска или раздела дескриптор группы томов.

Для диска:

sit@sit:~$ sudo pvcreate /dev/sdb

[sudo] password for sit:

Physical volume "/dev/sdb" successfully created

Для разделов:

sit@sit:~$ sudo pvcreate /dev/sdb1

[sudo] password for sit:

Physical volume "/dev/sdb1" successfully created

**Задания к лабораторной работе**

Повторяем данную операцию для всех дисков или разделов которые необходимо пометить как физические тома LVM.

**В нашем случае это - sdb, sdc , sde, sdd, sdf.**

**Предупреждение**

Если появилась ошибка инициализации диска с таблицей разделов, проверьте, что работаете с нужным диском. Убедившись в этом выполните следующие команды:

sudo dd **if**=/dev/zero of=/dev/sd\* bs=1k count=1

sudo blockdev --rereadpt /dev/sd\*

Данные команды уничтожат существующую таблицу разделов на диске sd\*. Для разделов воспользуйтесь утилитой fdisk (parted или gdisk) и установите тип раздела в 0x8e (LVM).

Просмотреть диски (разделы) которые помечены как физические тома LVM можно с помощью команды **pvdisplay**.

sit@sit:~$ sudo pvdisplay

--- Physical volume ---

PV Name /dev/sdb

VG Name storage

PV Size 8.00 GiB / not usable 4.00 MiB

Allocatable yes

PE Size 4.00 MiB

Total PE 2047

Free PE 2047

Allocated PE 0

PV UUID dt4vrH-xpIo-IOAR-4sZD-Q9cT-St7Q-dRKInS

--- Physical volume ---

PV Name /dev/sdc

VG Name storage

PV Size 8.00 GiB / not usable 4.00 MiB

Allocatable yes

PE Size 4.00 MiB

Total PE 2047

Free PE 2047

Allocated PE 0

PV UUID TD4x9x-t6dp-vrJ9-GnKk-eX1J-bU06-L17fnt

--- Physical volume ---

PV Name /dev/sdd

VG Name storage

PV Size 8.00 GiB / not usable 4.00 MiB

Allocatable yes

PE Size 4.00 MiB

Total PE 2047

Free PE 2047

Allocated PE 0

PV UUID qgJYg6-fNAu-9P2v-lBvt-u1H5-lfml-Pb186U

--- Physical volume ---

PV Name /dev/sde

VG Name storage

PV Size 8.00 GiB / not usable 4.00 MiB

Allocatable yes

PE Size 4.00 MiB

Total PE 2047

Free PE 2047

Allocated PE 0

PV UUID bKGRsE-ZNNV-XtqW-bXpn-yOI1-DMdC-8rANuv

--- Physical volume ---

PV Name /dev/sdf

VG Name storage

PV Size 8.00 GiB / not usable 4.00 MiB

Allocatable yes

PE Size 4.00 MiB

Total PE 2047

Free PE 2047

Allocated PE 0

PV UUID W6TBLw-3Yt6-ZJE2-lcOb-PMni-F95G-lxmyHW

**Создание группы томов**.

Для создания группы томов необходимо воспользоваться командой **vgcreate**. На вход программы необходимо указать имя группы и диски (разделы) которые необходимо добавить в данную группу.

sit@sit:~$ sudo vgcreate storage /dev/sd[b-f]

Volume group "storage" successfully created

Просмотреть группы томов в системе можно с помощью команды **vgdisplay**.

sit@sit:~$ sudo vgdisplay

--- Volume group ---

VG Name storage

System ID

Format lvm2

Metadata Areas 5

Metadata Sequence No 1

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 0

Open LV 0

Max PV 0

Cur PV 5

Act PV 5

VG Size 39.98 GiB

PE Size 4.00 MiB

Total PE 10235

Alloc PE / Size 0 / 0

Free PE / Size 10235 / 39.98 GiB

VG UUID Nf04a2-sQ5O-zRfO-V3jc-wpTj-KjYx-aKpeCK

**Удаление группы томов**.

Для удаления группы томов необходимо убедиться, что целевая группа томов не содержит логических томов. Далее необходимо деактивировать группу томов

sudo vgchange -an storage

После чего удалить группу томов командой

sudo vgremove storage

**Задания к лабораторной работе**

Для того, чтобы добавить ранее инициализированный физический том в существующую группу томов используется команда **vgextend**

sudo vgextend storage /dev/sd\*

Для того, чтобы удалить физический том из группы томов необходимо воспользоваться командой **vgreduce**

sudo vgreduce storage /dev/sd\*

**Создание логического тома.**

Для того, чтобы например создать логический том «sit», размером 1800Мб, необходимо выполнить команду

sudo lvcreate -L1800 -n sit storage

**Задания к лабораторной работе**

Без указания суффикса размеру раздела, по умолчанию используется множитель M «мегабайт» (в системе СИ равный 10 6 байт), что показано в примере выше. Суффиксы в верхнем регистре - KMGTPE соответствуют единицам в системе СИ с основанием 10. Например, G — гигабайт равен 10 9 байт, а суффиксы в нижнем регистре - kmgtpe соответствуют единицам в системе IEC (с основанием 2), например g — гибибайт равен 2 30 байт.

Для того, чтобы создать логический том размером 100 логических экстентов с записью по двум физическим томам и размером блока данных в 4 KB

sudo lvcreate -i2 -I4 -l100 -n sit storage

Если необходимо создать логический том, который будет полностью занимать группу томов, то сперва используйте команду vgdisplay, чтобы узнать полный размер группы томов, а после этого выполните команду **lvcreate**.

sudo vgdisplay storage | grep "Total PE"

Total PE 10230

sudo lvcreate -l 10230 storage -n sit

Эти команды создают логический том sit, полностью заполняющий группу томов. Тоже самое можно реализовать командой

lvcreate -l100%FREE storage -n sit

**Удаление логических томов.**

Перед удалением логический том должен быть размонтирован

sudo umount /dev/storage/sit

sudo lvremove /dev/storage/sit

lvremove -- do you really want to remove "/dev/storage/sit"? [y/n]: y

lvremove -- doing automatic backup of volume group "storage"

lvremove -- logical volume "/dev/storage/sit" successfully removed

**Увеличение логических томов.**

Для того, чтобы увеличить логический том, необходимо указать команде lvextend размер, до которого будет увеличен том (в экстентах или в размере)

sudo lvextend -L15G /dev/storage/sit

lvextend -- extending logical volume "/dev/storage/sit" to 15 GB

lvextend -- doing automatic backup of volume group "storage"

lvextend -- logical volume "/dev/storage/sit" successfully extended

В результате /dev/storage/sit увеличится до 15Гбайт.

**Задания к лабораторной работе**

Для изменения размера файловых систем ext2, ext3 и ext4 используйте **resize2fs**.

**Создание снапшотов LVM**

Для того, чтобы создать снапшот необходимо использовать **lvcreate -s**

sudo lvcreate -s -L10GB -n backup /dev/storage/sit

Таким образом мы создадим снапшот в 10 GB с именем backup для хранения изменений.

## Задания к лабораторной работе

### Часть 1

* Добавить пять виртуальных жестких дисков.
* Запустить Linux.
* Установить mdadm.
* Ознакомится с утилитой mdadm, ее возможностями и параметрами.
* В отдельном терминале следить за состоянием файла /proc/mdstat
* Собрать RAID 1 с помощью mdadm.
* Создать на созданном RAID файловую систему ext4.
* Смонтировать созданную файловую систему.
* Записать туда файл raid.txt с произвольным содержимым.
* Разрушить один из дисков RAID и проследить за происходящим в файле /proc/mdstat
* Проверить целостность файла raid.txt
* Остановить RAID 1.
* Очистить информацию дисков о принадлежности к программному RAID.
* Собрать RAID 0 с помощью mdadm.
* Создать на созданном RAID файловую систему ext3.
* Смонтировать созданную файловую систему.
* Записать туда файл raid.txt с произвольным содержимым.
* Разрушить один из дисков RAID и проследить за происходящим в файле /proc/mdstat
* Проверить целостность файла raid.txt
* Остановить RAID 0.
* Очистить информацию дисков о принадлежности к программному RAID.
* Собрать RAID 5 с диском горячей замены с помощью mdadm.
* Создать на созданном RAID файловую систему ext4.
* Смонтировать созданную файловую систему.
* Записать туда файл raid.txt с произвольным содержимым.
* Разрушить три диска RAID и проследить за происходящим в файле /proc/mdstat
* Проверить целостность файла raid.txt
* Остановить RAID 5.
* Очистить информацию дисков о принадлежности к программному RAID.
* Собрать RAID 10 с диском горячей замены с помощью mdadm.
* Создать на созданном RAID файловую систему ext2.
* Смонтировать созданную файловую систему.
* Записать туда файл raid.txt с произвольным содержимым.
* Разрушить два диска RAID и проследить за происходящим в файле /proc/mdstat
* Проверить целостность файла raid.txt
* Остановить RAID 10.
* Очистить информацию дисков о принадлежности к программному RAID.

### Часть 2

* Инициализировать физические диски, поверх которых будет создан LVM.
* Создать группу томов на основе четырех виртуальных жестких дисков.
* Создать логический том.
* На созданном логическом томе создать файловую систему.
* Смонтировать систему и создать файл файл LVM.txt .
* Добавить в группу томов еще один виртуальный жесткий диск.
* Определить количество добавленных экстентов.
* Расширить созданный логический том на размер добавленных экстентов.
* Увеличить размер файловой системы.
* Сделать снапшот логического тома.
* Удалить группу томов и снапшот.

## Вопросы к лабораторной работе

1. В чем достоинства и недостатки различных уровней RAID?
2. Что такое диск горячей замены RAID?
3. Как осуществить инициализацию физических дисков для использования их в качестве RAID массива?
4. Сколько минимально необходимо дисков для различных уровней RAID?
5. Сколько максимально может выйти из строя дисков в различных уровнях RAID массивов без потери данных?
6. Порядок действий для создания логического тома LVM.
7. Что такое Snapshot в LVM? Как его создать, и какое его функциональное назначение?
8. Что такое экстенты в LVM? Как создать логический том с определенным количеством экстентов?
9. Что такое логический том? Что такое физический том? В чем между ними отличие?
10. Как узнать количество экстентов в группе томов?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

# Лабораторная работа №5. Восстановление данных.

## Основные теоретические сведения

**Цель:** Получение теоритических и практических навыков программного восстановления данных.

### Восстановление данных TestDisk

**TestDisk** — свободная программа для восстановления данных, предназначенная прежде всего для восстановления потерянных разделов на носителях информации, а также для восстановления загрузочного сектора, после программных или человеческих ошибок (например, потеря MBR).

* Установка **<sudo apt-get install testdisk>**.
* Запускаем TestDisk **<sudo testdisk>**.
* Появляется окошко приветствия TestDisk, нам предлагается вести лог работы (для выполнения данной работы лог не требуется).
* Выбираем нужный диск и нажимаем **Enter**.
* Предлагается выбрать тип таблицы разделов, обычно TestDisk определяет все правильно, так что нажимаем **Enter**.
* Выбираем **Analise**.
* Выбираем **QuickSearch**.
* Нам выводят таблицу разделов. Выбираем раздел и нажимаем **P**, чтобы вывести список файлов.
* Выбираем файлы для восстановления и нажимаем **C**.
* Выбираем папку, куда будут сохранены файлы и нажимаем **C**.

### Восстановление данных PhotoRec

**PhotoRec** - это утилита, входящая в состав пакета TestDisk. Предназначена для восстановления испорченных файлов с карт памяти цифровых фотоаппаратов (CompactFlash, Secure Digital, SmartMedia, Memory Stick, Microdrive, MMC), USB flash-дисков, жестких дисков и CD/DVD. Восстанавливает файлы большинства распространенных графических форматов, включая JPEG, аудио-файлы, включая MP3, файлы документов в форматах Microsoft Office, PDF и HTML, а также архивы, включая ZIP. Может работать с файловыми системами ext2, ext3, ext4 FAT, NTFS и HFS+, причем способна восстановить графические файлы даже в том случае, когда файловая система повреждена или отформатирована.

* Установка **<sudo apt-get install testdisk>**.
* Запускаем PhotoRec **<sudo photorec>**.
* Выбираем нужный диск и нажимаем **Enter**.
* В нижнем меню можно выбрать **File Opt**, чтобы выбрать типы файлов для восстановления (по умолчанию выбраны все).
* Чтобы начать восстановление нажмите **Enter**, выбрав **Search**.
* У нас выбрана система ext4, поэтому выбираем первый вариант [ ext2/ext3 ].
* Если выбрать пункт **FREE**, то поиск будет произведен в пустом пространстве и в этом случае будут восстановлены только удаленные файлы, а если выбрать **WHOLE**, то поиск будет произведен на всем диске.
* Теперь нужно указать директорию, куда будем сохранять нужные нам файлы. Выбираем нужную папку и нажимаем **С**.
* Выбираем файлы для восстановления и нажимаем **C**.

### Восстановление данных Extundelete

**Extundelete** – утилита, позволяющая восстанавливать файлы, которые были удалены с разделов ext3/ext4.

* Установка: **<sudo apt-get install extundelete>**.
* Как только вы поняли, что удалили нужные файлы, необходимо отмонтировать раздел: **<umount /dev/<partition> >**
* Зайдите в каталог, в который будут восстанавливаться удаленные данные. Он должен быть расположен на разделе отличном от того, на котором хранились восстанавливаемые данные: **cd /<путь\_к\_каталогу\_куда\_восстанавливать\_данные>**
* Запустите **extundelete**, указав раздел, с которого будет происходить восстановление и файл, который необходимо восстановить: **sudo extundelete /dev/<partition> –restore-file /<путь к файлу>/<имя\_файла>**
* Можно так же восстанавливать содержимое каталогов: **sudo extundelete /dev/<partition> –restore-directory /<путь\_к\_директории>**

### Восстановление данных Foremost.

**Foremost** - консольная программа, позволяющая искать файлы на дисках или их образах по hex-данным, характерным заголовкам и окончаниям. Программа проверяет файлы на предмет совпадения заранее определённых hex-кодов (сигнатур), соответствующих наиболее распространённым форматам файлов. После чего экстрагирует их из диска/образа и складывает в каталог, вместе с подробным отчётом о том, чего, сколько и откуда было восстановлено. Типы файлов, которые foremost может сразу восстановить: jpg, gif, png, bmp, avi, exe, mpg, wav, riff, wmv, mov, pdf, ole, doc, zip, rar, htm, cpp. Есть возможность добавлять свои форматы (в конфигурационном файле /etc/foremost.conf), о которых программа не знает.

* Установка: **<sudo apt-get install foremost>**
* Пример использования для восстановления изображений с диска **/dev/sdb** в каталог **~/out\_dir**: **<sudo foremost -t jpg,gif,png,bmp -i /dev/sdb -o ~/out\_dir>**

## Задания к лабораторной работе

* Добавьте в виртуальную машину виртуальный жесткий диск.
* Запустите виртуальную машину с Linux.
* Запустите fdisk (gdisk или parted) и создайте таблицу разделов MBR с разделами.
* Отформатируйте созданные разделы в файловую систему ext4.
* Установите TestDisk.
* Удалите MBR (или таблицу разделов) с помощью команды DD.
* Восстановите MBR (или таблицу разделов) с помощью TestDisk.
* Смонтируйте восстановленные разделы и создайте там произвольные файлы.
* Удалите созданные файлы.
* С помощью TestDisk восстановите данные.
* Создайте произвольный каталог и запишите туда данные каталога /var/log/ .
* Удалите данные с созданного каталога.
* С помощью PhotoRec восстановите данные.
* Создайте произвольный каталог и запишите туда данные каталога /etc/ .
* С помощью Extundelete или Foremost восстановите данные.

## Вопросы к лабораторной работе

1. С помощью какой из программ, используемых в этой лабораторной работе, можно восстановить таблицу разделов?
2. Какие файловые системы поддерживает PhotoRec?
3. Какие форматы поддерживает PhotoRec?
4. Как Foremost восстанавливает файлы?
5. Можно ли восстановить данные с файловой системы NTFS, используя extundelete?
6. Все ли данные скопированные с каталога /var/log/ восстановились?
7. Все ли данные скопированные с каталога /etc/ восстановились?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

[Next](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/lr6.html)[Previous](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/lr4.html)

# Лабораторная работа №6. Шифрование данных.

## Основные теоретические сведения

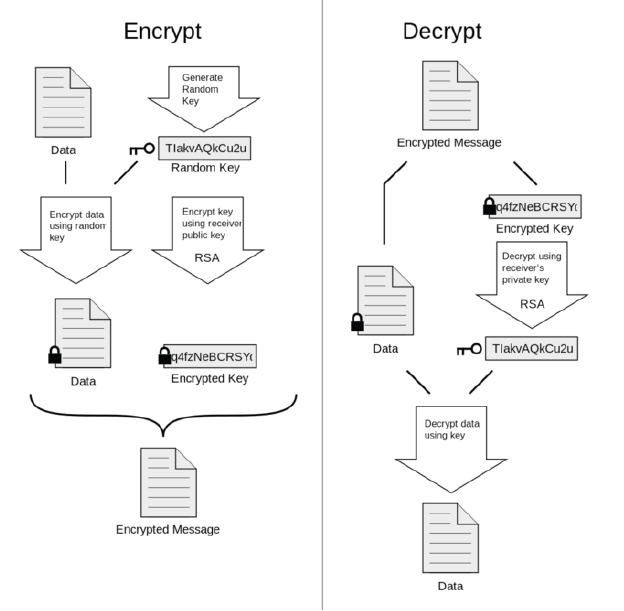
**Цель:** Получение теоретических и практических навыков работы с программными средствами шифрования данных.

### Консольные команды:

* **gpg** <параметры> - инструмент для шифрования и цифровой подписи.
* **cryptsetup** <параметры> - программа для управления шифрованными дисковыми разделами, работающая на основе модуля ядра dm-crypt.
* **truecrypt** <параметры> - программа для управления шифрованными дисковыми разделами, при помощи truecrypt.
* **fallocate** <параметры> - команда, позволяющая вручную выделять блоки для файлов.

### PGP

**PGP** (Pretty Good Privacy) — компьютерная программа, которая позволяет выполнять операции шифрования/дешифрования и цифровой подписи файлов или сообщений, а также другой информации, представленной в электронном виде, в том числе шифрование данных на запоминающих устройствах.



Процесс шифрования в PGP проходит в несколько этапов: хеширование, сжатие данных, шифрование с симметричным ключом, и, наконец, шифрованием с открытым ключом. Причём каждый этап может использовать разные алгоритмы. Так симметричное шифрование производится с использованием одного из семи симметричных алгоритмов (AES, Blowfish, 3DES, CAST5, IDEA,Twofish, Camellia) на сеансовом ключе. Сеансовый ключ в свою очередь генерируется с использованием криптографически стойкого генератора псевдослучайных чисел. Он зашифровывается открытым ключом получателя с использованием алгоритмов RSA или Elgamal (в зависимости от исходного открытого ключа получателя).

Изначально PGP разрабатывалась для защиты электронной почты на стороне клиента, но начиная с 2002 года также включает в себя шифрование жёстких дисков, директорий, файлов, сессий программ мгновенного обмена сообщениям, защиту файлов и директорий в сетевых хранилищах, пакетной передачи файлов, а в новых версиях — шифрование HTTP-запросов и ответов на стороне сервера и клиента.

### TrueCrypt

**TrueCrypt** — одна из самых известных программ для шифрования данных «на лету». Позволяет создавать виртуальный зашифрованный логический диск, хранящийся в виде особого файла - криптоконтейнера. С помощью TrueCrypt также можно полностью зашифровать раздел жёсткого диска или любого другого носителя информации, например, USB диск.

В процессе работы данная утилита создает на компьютере специальную защищенную область. Операционная система в свою очередь воспринимает эту область как файл или диск. Отличие между защищенным пространством TrueCrypt и обычным диском, в том, что на обычном диске данные обычно никак не защищены, а TrueCrypt шифрует данные «на лету», абсолютно незаметно для пользователей, и тем самым обеспечивает надежную защиту информации без специальных манипуляций с ней. Кроме того, в защищенной области TrueCrypt умеет размещать данные, которые будут не просто зашифрованы, но и скрыты от посторонних глаз.

TrueCrypt может создавать зашифрованный виртуальный диск:

* В файловом контейнере, что позволит легко работать с ним — копировать, переносить (в том числе на внешние устройства в виде файла), переименовывать или удалять;
* В виде зашифрованного раздела диска, что сделает работу более удобной и производительной, начиная с версии 5.0 появилась возможность шифровать системный раздел;
* Путём полного шифрования содержимого устройства, такого как USB диск (флоппи-диски перестали поддерживаться с версии 7.0).

В список поддерживаемых TrueCrypt алгоритмов шифрования входят AES, Twofish и Serpent.

Для того, чтобы получит доступ к зашифрованным данным применяется пароль (ключевая фраза), ключевой файл (один или несколько), а также их комбинации. В качестве ключевых файлов можно использовать любые доступные файлы на локальных, съёмных, сетевых дисках (при этом будут использоваться первые 1,048,576 байт) или генерировать свои собственные ключевые файлы.

Одна из интересных возможностей TrueCrypt — обеспечение двух уровней отрицания наличия зашифрованных данных, необходимого в случае вынужденного раскрытия пароля пользователем:

* Создание скрытого тома, что позволяет задать еще один пароль (или набор ключевых файлов) к обычному тому. Доступ к этим данным невозможно получить доступ с основным паролем, при этом скрытый том может иметь свою файловую систему, а располагается он в свободном пространстве основного тома.
* Ни один из томов TrueCrypt не может быть определен (тома TrueCrypt невозможно отличить от случайного набора данных, поэтому файл нельзя связать с TrueCrypt или с программой его создавшей, ни в какой форме и рамках).

У TrueCrypt есть графический интерфейс для Linux, но можно управлять шифрованием и из консоли.

* Создать файл ключа <truecrypt –create-keyfile /home/user/test/file> , где file - название файла-ключа. Учтите, что директория /home/user/test должна существовать.
* Создать криптоконтейнер <sudo truecrypt -k /home/user/test/file -c /dev/sda9>.
* Примонтировать <sudo mount /dev/mapper/truecrypt0 /mnt/crypto> Директория для монтирования (здесь /mnt/crypto) уже должна существовать.
* Размонтировать <truecrypt -d>.
* Чтобы снова получить доступ к информации, подключим контейнер <truecrypt -k /home/user/test/file /dev/sda9 /mnt/crypto>.

### LUCKS/dm-crypt

**LUKS (Linux Unified Key Setup)** — спецификация шифрования диска (или блочного устройства), изначально предложенная для Linux, но сейчас поддерживаемая и в ряде других операционных систем. Основана на стандартной подсистеме шифрования Linux-ядра под названием dm-crypt и следующая рекомендациям TKS1/TKS2.

Особенности:

* В качестве «контейнера» используется файл. Его размер фиксирован. Возможно изменение размера.
* «Внутри» контейнера создается файловая система, любого удобного вам формата.
* При использовании - монтируется, как обычный раздел.
* Данные сохраняются по блокам, как в обычном файле/файловой системе. То есть :
  + модификация файла внутри контейнера приводит к перезаписи блоков, занимаемых этим файлом, но не всего контейнера;
  + «потеря/порча» одного блока приводит к потере информации «того-что-было-в-этом-блоке», и не более того.
  + при синхронизации контейнера «в облако» - как правило, перезаписывается не весь файл, а «модифицированная часть», что требует малого объема трафика.

В отличие от Truecrypt:

* как правило, выше скорость обработки данных (зависит от алгоритма/размера ключа);
* проще работа с ключами;
* нет механизма «двойного дна»;
* возможны проблемы при попытке использования контейнера «из другой ОС»

## Задания к лабораторной работе

* Установить PGP, GPG <sudo apt-get install pgpgpg>
* Произвести операции шифрования и дешифрования над произвольными файлами. Для шифрования используйте команду <gpg -c>. Для дешифрования <gpg –decrypt-file> (В этом случае в директории зашифрованного файла будет создан расшифрованный. Если нужно лишь вывести на экран расшифрованное содержимое используйте <gpg –decrypt>)
* Установить TrueCrypt. Нам потребуется версия 7.1а. Скачать её можно [здесь](https://www.truecrypt71a.com/downloads/) или [здесь.](https://truecrypt.ch/downloads/)
* Создать криптоконтейнер, примонтировать его как виртуальный диск.
* Поместить в криптоконтейнер какую-то информацию.
* Отмонтировать диск и переместить криптоконтейнер.
* Повторно примонтировать криптоконтейнер как виртуальный диск. Убедиться, что криптоконтейнер может передаваться и использоваться независимо.
* Установить LUKS/dm-crypt <sudo apt-get update>, <sudo apt-get install cryptsetup>.
* Создаем файл, где будем хранить зашифрованные данные. Самый простой способ <fallocate -l 512M /root/test1>, где /root - директория хранения файла, test1 - имя файла. Так же для создания этого файла можно использовать команду dd. <dd if=/dev/zero of=/root/test2 bs=1M count=512>. Третий способ - использовать команду dd и заполнить файл случайными данными. <dd if=/dev/urandom of=/root/test3 bs=1M count=512>.
* Создать криптоконтейнер. <cryptsetup -y luksFormat /root/test1> (нужно будет согласиться переписать данные и задать пароль).
* Открыть контейнер. <cryptsetup luksOpen /root/test1 volume1>. (volume1 - имя контейнера, его мы задаем этой командой). При этом будет создан файл /dev/mapper/volume1.
* Создать в нем файловую систему <mkfs.ext4 -j /dev/mapper/volume1>.
* Создать папку для монтирования <mkdir /mnt/files>. Монтировать <mount /dev/mapper/volume1 /mnt/files>
* Теперь перенесем какие\_ниибудь файлы в криптоконтейнер. Например, скопируем папку /etc <cp -r /etc/\* /mnt/files>.
* Размонтировать <umount /mnt/files>.
* Теперь закрываем volume1. <cryptsetup luksClose volume1>. После этого наши данные зашифрованы.
* Чтобы открыть их выполним <cryptsetup luksOpen /root/test1 volume1> и <mount /dev/mapper/volume1 /mnt/files>

## Вопросы к лабораторной работе

1. Какие алгоритмы шифрования входят в комплект TrueCrypt?
2. Какие алгоритмы шифрования входят в комплект PGP, GPG?
3. Что такое «криптоконтейнер»?
4. Каковы основные достоинства и недостатки рассмотренных программных продуктов?
5. Какие алгоритмы шифрования, используемые в рассмотренных программных продуктах, наиболее надежны и почему?
6. В каких случаях рекомендуется применять шифрование данных?

[Next](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/lr7.html)[Previous](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/lr5.html)

# Лабораторная работа №7. Honeypot, Nmap.

## Основные теоретические сведения

**Цель:** Получение практических и теоретических навыков работы с honeypot, способами и методами сканирования сети.

**IP-адрес** - уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP. В сети Интернет требуется глобальная уникальность адреса; в случае работы в локальной сети требуется уникальность адреса в пределах сети.

**Задания к лабораторной работе**

**IPv4**. В 4-й версии IP-адрес представляет собой 32-битовое число. Удобной формой записи IP-адреса (IPv4) является запись в виде четырёх десятичных чисел значением от 0 до 255, разделённых точками, например, 192.168.0.3.

**IPv6**. В 6-й версии IP-адрес (IPv6) является 128-битовым. Внутри адреса разделителем является двоеточие (напр. 2001:0db8:85a3:0000:0000:8a2e:0370:7334). Ведущие нули допускается в записи опускать. Нулевые группы, идущие подряд, могут быть опущены, вместо них ставится двойное двоеточие (fe80:0:0:0:0:0:0:1 можно записать как fe80::1). Более одного такого пропуска в адресе не допускается.

**MAC-адрес** (от англ. Media Access Control — управление доступом к среде, также Hardware Address) — уникальный идентификатор, присваиваемый каждой единице активного оборудования компьютерных сетей.

При проектировании стандарта Ethernet было предусмотрено, что каждая сетевая карта (равно как и встроенный сетевой интерфейс) должна иметь уникальный шестибайтный номер (MAC-адрес), прошитый в ней при изготовлении. Этот номер используется для идентификации отправителя и получателя фрейма, и предполагается, что при появлении в сети нового компьютера (или другого устройства, способного работать в сети) сетевому администратору не придётся настраивать MAC-адрес.

**Маска сети** - битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети (при этом, в отличие от IP-адреса, маска подсети не является частью IP-пакета). Например, узел с IP-адресом 12.34.56.78 и маской подсети 255.255.255.0 находится в сети 12.34.56.0 с длиной префикса 24 бита. В случае адресации IPv6 адрес 2001:0DB8:1:0:6C1F:A78A:3CB5:1ADD с длиной префикса 32 бита (/32) находится в сети 2001:0DB8::/32.

Другой вариант определения — это определение подсети IP-адресов. Например, с помощью маски подсети можно сказать, что один диапазон IP-адресов будет в одной подсети, а другой диапазон соответственно в другой подсети.

Чтобы получить адрес сети, зная IP-адрес и маску подсети, необходимо применить к ним операцию поразрядной конъюнкции (логическое И).

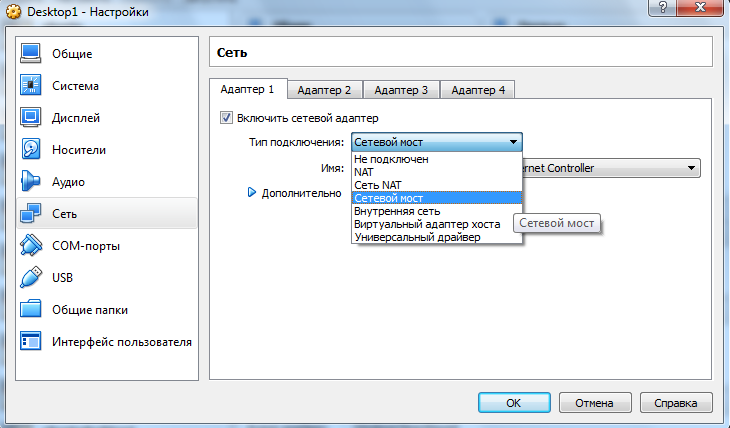
IP-адрес называют **статическим** (постоянным, неизменяемым), если он назначается пользователем в настройках устройства, либо если назначается автоматически при подключении устройства к сети и не может быть присвоен другому устройству.

IP-адрес называют **динамическим** (непостоянным, изменяемым), если он назначается автоматически при подключении устройства к сети и используется в течение ограниченного промежутка времени, указанного в сервисе назначавшего IP-адрес (DHCP).

**DHCP** - сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP, и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

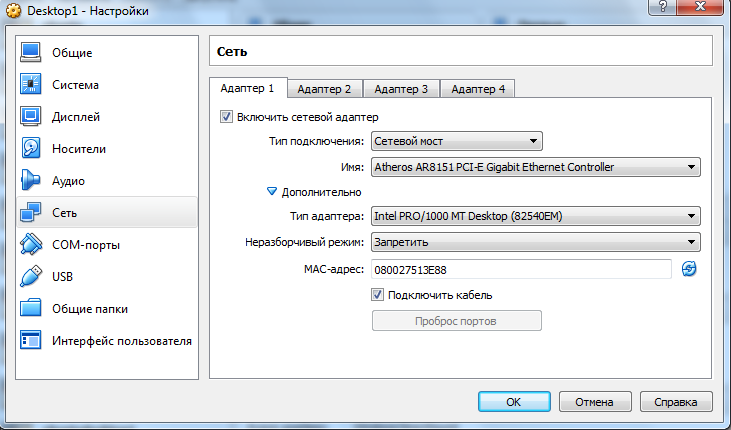
### Настройка сети в VirtualBox

Чтобы изменить настройки сети для виртуальной машины в VirtualBox, выберите машину, нажмите **Настройки** и перейдите на вкладку сеть.



Существуют следующие варианты настройки сети в VirtualBox:

* NAT
* Сеть NAT
* Сетевой мост
* Внутренняя сеть
* Виртуальный адаптер хоста
* Универсальный драйвер



**Предупреждение**

Здесь стоит обратить внимание на MAC-адрес (Может понадобится для выполнения лабораторных работ). Если вы **«клонировали»** машину, здесь нужно изменить MAC-адрес на новый.

Можно подключить к виртуальной машине еще 3 виртуальных адаптера. Настройки для них, выглядят так же, как и для «Адаптера 1».

Кроме этих настроек, есть еще настройки для сети NAT и Виртуального адаптера хоста. Чтобы получить к ним доступ, нажмите Файл/Настройки и перейдите на вкладку сеть.

**Предупреждение**

Для выполнения лабораторной работы рекомендуется выбрать тип подключения **«Сетевой мост»** на обоих машинах.

Подробнее о настройке сети в VirtualBox можно прочитать [на сайте VirtualBox](https://www.virtualbox.org/manual/ch06.html).

### Nmap

Существует несколько средств сканирования Nmap, широко применяемых в настоящее время. Рассмотрим некоторые из них.

**Ping-сканирование**

Самым распространенным и простым способом сканирования является простое ping-сканирование, которое заключается в отправке ICMP пакетов на разные хосты. Хост, который активен, ответит на эти пакеты. Форма подачи запроса:

ping 192.168.58.103

Если хост активен, то будет периодически выводиться строка вида:

64 bytes from 192.168.58.103: icmp\_seq=1 ttl=64 time=0,284mc

**TCP Connect()**

Второй доступный метод сканирования - TCP Connect. Он заключается в том, что сканирующая машина пытается установить соединение со сканируемой. Успешный результат говорит о том, что порт открыт, неудачный — о том, что он закрыт или фильтруется. Это сканирование легко обнаруживается по огромному количеству записей в log-файле неудачных попыток установления соединения и ошибок исполнения этой операции. Понятно, что средства защиты с максимальным быстродействием заблокируют адрес, вызывающий ошибки.

nmap –sT 192.168.58.103

-v: Увеличить уровень вербальности (задать дважды или более для увеличения эффекта)



**TCP-SYN**

Более совершенным методом сканирования является TCP SYN — так называемое «полуоткрытое сканирование». При вызове Nmap посылает SYN-пакет, как бы ради того, чтобы установить новое соединение. Если в ответе присутствуют флаги SYN или ACK, считается, что порт открыт. Флаг RST говорит об обратном. Если пришел ответ, говорящий о том, что порт открыт, nmap незамедлительно отправляет RST-пакет для сброса еще не установленного соединения. Сканирование осуществляется только при наличии прав суперпользователя (root).

nmap –sS 192.168.58.103



**Сканирования FIN, Xmas Tree и NULL**

Тем не менее, межсетевой экран или другие защитные средства могут ожидать приходящие SYN-пакеты. Из-за этого зачастую такой метод сканирования не дает результата. Поэтому существует еще целая группа возможных способов сканирования, альтернативных TCP SYN. Это FIN, Xmas Tree и NULL-сканирования. Большинство операционных систем по умолчанию, согласно рекомендациям, должны ответить на такие пакеты, прибывшие на закрытые порты флагом RST. Важная деталь: ни одна операционная система семейства Windows никогда не ответит RST пакетом на пришедший FIN, XmasTree или NULL пакет. Используя этот факт даже при подобных, в общем-то, не особо детальных сканированиях можно предположить, как минимум семейство операционных систем.

nmap –sF 192.168.58.103



nmap –sX 192.168.58.103



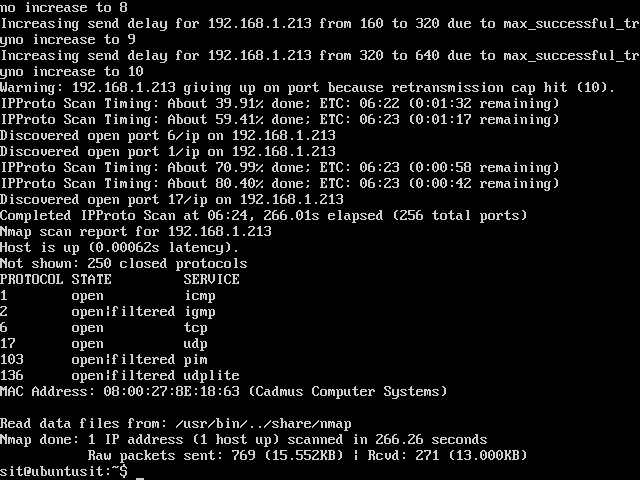
nmap –sN 192.168.58.103



**Сканирование протоколов IP**

Метод заключается в том, что хосту передаются IP пакеты без заголовков для каждого протокола сканируемого хоста. Если получено сообщение, говорящее о недоступности протокола, то этот протокол не поддерживается хостом. В противном случае — поддерживается.

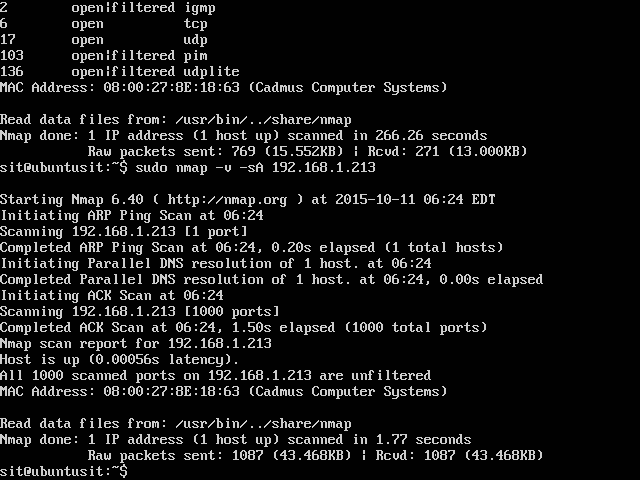
nmap –sO 192.168.58.103



**ACK-сканирование**

ACK сканирование заключается в передаче ACK пакетов на сканируемый порт. Если в ответ приходит RST пакет, порт классифицируется как не фильтруемый. Если нет ответа или пришел ответ в форме ICMP-сообщения о недоступности порта, порт считается фильтруемым. Этот метод никогда не покажет состояние порта «открыт».

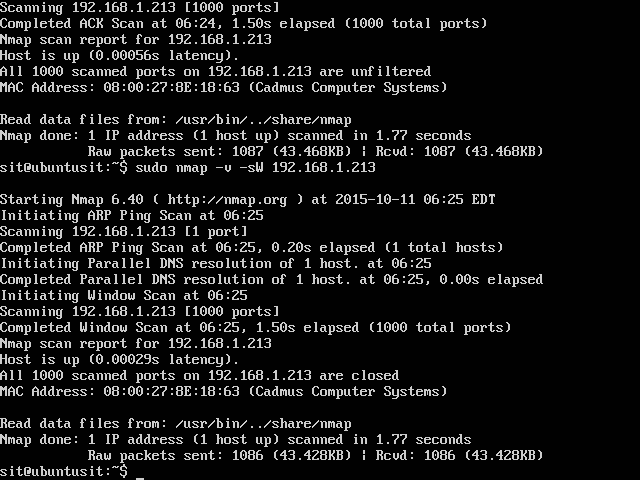
nmap –sA 192.168.58.103



**TCP Window**

TCP Window — похоже на ACK сканирование, однако по значениям поля Initial Window TCP-пакета пришедшего в ответ, можно определить открытые порты.

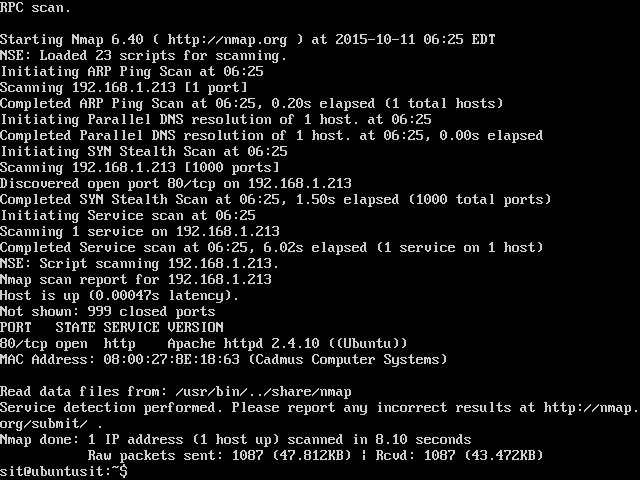
nmap –sW 192.168.58.103



**RPC-сканирование**

RPC-сканирование используется для определения программы, обслуживающей порт и её версии, и заключается в «затоплении» NULL- пакетами оболочки SunRPC открытых TCP или UDP портов хоста.

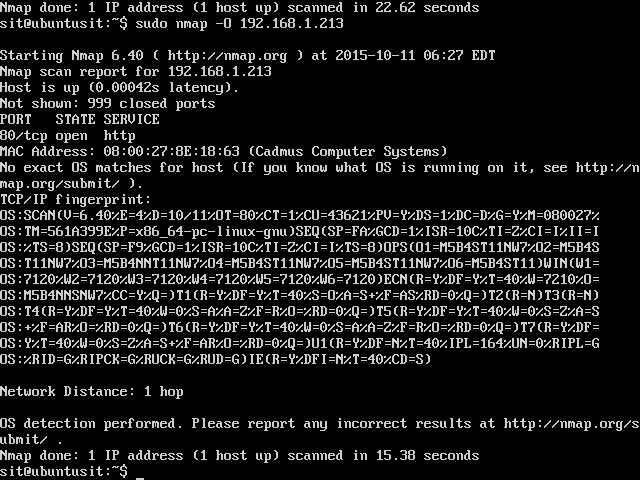
nmap –sR 192.168.58.103



**Сканирование ОС**

И, наконец, последнее — сканирование, используемое для определения ОС на сканируемом хосте.

nmap –O 192.168.58.103



На этом cкриншоте Nmap не удалось определить ОС. Так что просканируем другую систему.



Существует еще несколько методов сканирования Nmap, однако в условиях проведения данной лабораторной работы осуществить их не получится.

### Honeypot

**Honeypot («Ловушка»)** - ресурс, представляющий собой приманку для злоумышленников.

Фактически основная задача Honeypot — подвергнуться атаке или несанкционированному сканированию с целью изучения стратегии и методов сканирования и определения перечня средств, необходимых для предотвращения будущих атак. Суть работы Honeypot заключается в создании ловушек — образов систем, которые извне воспринимаются как полноценные машины с установленными на них операционными системами, а, следовательно, поддающиеся сканированию.

Использование Honeypot имеет практический и исследовательский смысл. Во-первых, если на сервере установлена хорошая система защиты, долгое время можно не замечать постоянных попыток сканирования — Honeypot укажет на их наличие. Во-вторых, серьезной проблемой специалистов по информационной безопасности является нехватка информации о методах и средствах, используемых злоумышленниками. Единственное, что позволяет получить информацию об этих средствах — испытание их действия на себе. И Honeypot является чуть ли не идеальным способом для этого, ведь в настоящее время этом точно известно: грамотно настроенный Honeypot практически невозможно распознать.

Следует отметить, что в условиях, в которых проводится лабораторная работа, придется отойти от реальной ситуации, когда хакеру неизвестно ничего о том, что из себя представляет сервер и не являются ли подключенные к нему машины всего лишь ловушками. Иными словами, нам будут известны ip-адреса сканируемых ловушек.

Для начала необходимо создать локальную сеть из двух машин. Рекомендуется использовать машины с установленными на них операционными системами Ubuntu Server. Далее необходимо разобраться непосредственно с ловушками.



На машину виртуальную машину «Hacker» необходимо установить Nmap (либо в случае с операционной системой с графической оболочкой - Zenmap) . На виртуальную машину «Server» установить и настроить Honeyd.

**Предупреждение**

Однако на некоторых машинах при сканировании может появиться проблема: ни одна из ловушек не видна ни хостовой операционной системе, ни хакеру, ни самому серверу. В этом случае на помощь приходит команда **arp**, выполняемая с терминала хостовой машины. Синтаксис:

arp -s [ip-адрес ловушки] [mac-адрес адаптера]

Mac-адрес можно легко узнать либо в настройках VirtualBox, либо с помощью команды ifconfig. То же самое необходимо сделать и с машиной хакера.

### Honeyd

**Установка**

Установка Honeyd. <sudo apt-get install honeyd>.

**Предупреждение**

Если honeyd отсутствует в репозиториях, его можно скачать ([\*.deb](http://archive.ubuntu.com/ubuntu/pool/universe/h/honeyd/honeyd_1.5c-8ubuntu1_amd64.deb) пакет) и установить. Так же, можно скачать исходные коды honeyd с [официального сайта](http://www.honeyd.org/release.php) и их скомпилировать.

Установка **Farpd**. <sudo apt-get install farpd>.

Так же для Honeyd необходимо установить следующие пакеты:

* libevent - event notification
* libdnet - packet creation
* libpcap - packet sniffing

**Настройка**

Настройка Honeyd осуществляется путем изменения конфигурационного файла **honeyd.conf**.

После установки Honeyd появится файл /etc/honeypot/honeyd.conf со стандартными настройками. Вы можете изменить настройки в этом файле на свои. Или же, создать свой конфигурационный файл с настройками и при запуске указывать его.

Пример настройки:

create default

set default default tcp action block

set default default udp action block

set default default icmp action block

create windows

set windows personality "Microsoft Windows XP Professional SP1"

set windows default tcp action reset

add windows tcp port 135 open

add windows tcp port 139 open

add windows tcp port 445 open

set windows ethernet "00:00:24:ab:8c:12"

bind 192.168.1.117 windows

Эта конфигурация с одной ловушкой. При сканировании будет выведен MAC-адрес, указанный в honeyd.conf, а процесс, запущенный на машине с Honeypot укажет, что был выведен этот адрес.

[Здесь](http://www.honeyd.org/configuration.php) можно найти конфигурационный файл с подробным описанием. Он поможет настроить собственную конфигурацию при необходимости.

После этого на сервере необходимо запустить Honeypot ( можно с записью информации о работе в log-файл, либо без неё (см. ниже)). При этом будет осуществляться сканирование ip-адресов (либо отдельных, либо интервала) с помощью средств Nmap.

honeyd -d -f [путь к файлу honeypot.conf]

honeyd -d -f [путь к файлу honeypot.conf] -l [путь к log-файлу]

**Предупреждение**

Если у вас есть ошибки в настройке конфигурационного файла «honeyd.conf», honeyd не запуститься. В терминале будет выводиться информация о работе Honeypot.

## Задания к лабораторной работе

* Настройте сеть, состоящую из двух компьютеров.
* На одну из виртуальных машин устаните web-сервер <sudo apt-get install apache2>.
* На другую установите – Nmap <sudo apt-get install nmap>.
* Определите IP адрес виртуальной машины где установлен web-сервер apache.
* Произведите сканирование web-сервера всеми описанными методами (Изучение средств сканирования Nmap).
* Установите Honeyd.
* Ознакомьтесь с информацией по настройке Honeyd и стандартным содержимым файла /etc/honeypot/honeyd.conf.
* Настройте Honeypot изменив содержание файла /etc/honeypot/honeyd.conf .
* Запустите <farpd -d> .
* Запустите honeyd. <honeyd -d -f /etc/honeypot/honeyd.conf>
* Произведите сканирование сети с honeypot.
* Измените настройки Honeypot. Усложните конфигурационный файл. Добавьте несколько ловушек, измените информацию об ОC, информацию о роутере, об открытых портах и.т.д..
* Запустите honeyd. <honeyd -d -f /etc/honeypot/honeyd.conf>
* Произведите сканирование.

## Вопросы к лабораторной работе

1. Что такое статический и динамический IP-адреса? В чём разница?
2. В чём заключается метод сканирование протоколов IP?
3. На какие пакеты большинство ОС должны ответить флагом RST?
4. Назначение, цели, описание Honeypot.
5. Какие цели может преследовать злоумышленник, взламывая сервера?
6. Какое наказание предусмотрено в РФ за взлом?
7. Как выявлять Honeypot?
8. Что такое DHCP?
9. Для чего используется RPC-сканирование?
10. Перечислите основные методы сканирования Nmap.

# Лабораторная работа №8. ЛВС, web-сервер с CMS.

## Основные теоритические сведения

**Цель:** Получить теоритических и практических навыков построения ЛВС и web-сервера на примере установки CMS.

ЛВС (Локальная вычислительная сеть) - компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт). Также существуют локальные сети, узлы которых разнесены географически на расстояния более 12 500 км (космические станции и орбитальные центры). Несмотря на такие расстояния, подобные сети всё равно относят к локальным.

### Типы подключения в Oracle VM VirtualBox

**Не подключен**

В этом режиме, VirtualBox сообщает гостю что сетевой адаптер есть, но он не подключен – так как если бы Ethernet кабель не присоединен к карте. Таким образом возможно симулировать «выдергивание» кабеля из виртуальной сетевой карты и обрыв соединения,что может быть полезно для информирования гостевой ОС об отсутствии сетевого соединения, но возможности его настройки.

**Network Address Translation (NAT)**

Режим Network Address Translation (NAT) предоставляет наиболее простой способ доступа к внешней среде из виртуальной машины. Обычно, для него не требуется никаких настроек хоста и гостевой системы. Поэтому он является сетевым режимом, настраиваемым по умолчанию.

Виртуальная машина с сетевым интерфейсом в режиме NAT подключается к сети, также как реальный компьютер подключается к Internet через маршрутизатор. «Маршрутизатором» в данном случае выступает сетевой модуль VirtualBox, который обрабатывает сетевой трафик виртуальной машины. Недостаток режима NAT, как и в случае локальной сети за маршрутизатором, в том что виртуальная машина недоступна для внешней сети (internet); вы не можете обрабатывать сетевые запросы, пока не настроите переброс портов

Можно настроить сеть NAT, выбрав этот тип подключения в настройках сети, предварительно необходимо создать сеть NAT в Файл-Настройки-Сеть.

**Сетевой мост**

В данном режиме VirtualBox использует драйвер устройства на хост системе , который обрабатывает данные проходящие через физический сетевой интерфейс. Этот драйвер обычно называют «net filter». Он позволяет перехватывать VirtualBox пакеты из физической сети и изменять данные в них, а также создавать новые программные сетевые интерфейсы. Когда гость использует такой программный интерфейс, то выглядит это так как будто бы гостевая система подключается к физической сети, хост система может посылать и принимать данные от гостевой. Это означает, что вы можете использовать физический интерфейс хоста в качестве маршрутизатора или шлюза между гостевой системой и вашей физической сетью.

**Внутренняя сеть**

Режим Внутренняя сеть похож на режим Сетевой мост в котором ВМ может связываться с внешним миром. Однако, «внешний мир» ограничен другими виртуальными машинами на том же хосте и которые подключены к той же внутренней сети.

Хотя с технической стороны, все что можно сделать используя режим внутренняя сеть , можно также сделать в режиме сетевого моста, но в режиме внутренней сети есть преимущества в безопасности. В режиме моста, весь трафик проходит через физический интерфейс хоста. Поэтому возможно подключение анализаторов пакетов (например, Wireshark) к интерфейсу хоста и сбор всего трафика проходящего через него. Если по каким либо причинам вам необходимо чтобы две и более виртуальные машины имели защищенное соединение, скрывая свои данные от хост системы и других пользователей, то режим сетевой мост не является лучшим выбором.

Внутренние сети создаются автоматически, т.е. не существует централизованной настройки . Каждая внутренняя сеть идентифицируется своим именем. При существовании более чем одной виртуальной сетевой карты с одним идентификатором внутренней сети, драйвер VirtualBox автоматически поддерживает соединение этих интерфейсов в одну сеть, выступая в роли сетевого коммутатора. VirtualBox поддерживает полностью Ethernet коммутацию, широковещательную и многоадресную рассылку сетевых пакетов и режим promiscuous mode.

**Виртуальный адаптер хоста**

Данный режим можно использовать для создания сетей из хоста и нескольких виртуальных машин, без использования физического сетевого интерфейса хоста. На хосте создается виртуальный сетевой интерфейс (подобный петлевому интерфейсу) , обеспечивающий соединения между хост системой и виртуальными машинами.

Данный режим можно рассматривать как гибрид режимов сетевого моста и внутренней сети: как и режиме моста виртуальные машины могут соединяться друг с другом и с хост системой, как будто бы они соединены через физический коммутатор. Как и в режиме внутренней сети, нет необходимости в предоставления физического сетевого интерфейса и виртуальные машины не могут общаться с внешней сетью хоста, т.к. они никак не связаны с физическим сетевым интерфейсом.

При использовании режима внутренней сети, VirtualBox создает новый программный интерфейс на хосте, который добавляется к списку существующих сетевых интерфейсов хоста. Другими словами, в режиме сетевого моста существующий физический интерфейс используется для подключения виртуальных машин, а в режиме «виртуальный адаптер хоста» создается новый «петлевой» интерфейс хоста. В режиме внутренней сети трафик между виртуальными машинами не «виден», а трафик «петлевого» интерфейса возможно перехватить.

Режим виртуального адаптера хоста удобно использовать для нескольких предварительно настроенных виртуальных систем, которые предназначены для совместного использования. Например, одна виртуальная машина представляет собой web сервер который использует вторую с сервером базы данных. Другой дополнительный, сетевой интерфейс (мост) может соединить web сервер с внешним миром для выдачи данных, но внешнему миру не будет доступа к серверу базы данных.

**Универсальный драйвер**

Редко используемый режим универсального сетевого интерфейса (Rarely used modes share the same generic network interface), позволяет выбирать пользователю драйвер, который может быть включен в VirtualBox или поставляться с пакетом расширений (extension pack).

**Задания к лабораторной работе**

Подробнее о настройке сети в VirtualBox можно прочитать [на сайте virtualbox](https://www.virtualbox.org/manual/ch06.html).

### Утилита ping

Команда PING используется для проверки состояния между двумя сетевыми соединениями. Эти соединения могут быть локальными, глобальными или подключения к интернету в целом. Команда PING посылает пакеты информации с определением IP-адреса, а затем измеряет время, которое потребуется, чтобы получить ответ от указанного компьютера или устройства.

ping [ -LRUbdfnqrvVaAB] [ -c количество] [ -i интервал] [ -l преднагрузка] [ -p шаблон] [ -s размер-пакета] [ -t ttl] [ -w ограничение-на-время-работы] [ -F идентификатор-потока] [ -I адрес] [ -M указание] [ -Q тип-обслуживания] [ -S буфер-отправки] [ -T параметр-временной-метки] [ -W время-ожидания-ответа] [ переход …] назначение

Используем команду:

ping 192.168.58.103, где 192.168.58.103 - ip адрес цели

Получим следующую информацию:

* IP адрес
* Количество отправленных байт
* Время, которое потребовалось для соединения в миллисекундах
* TTL – время жизни пакета данных в протоколе IP, предельно допустимое время его пребывания в системе.

**Предупреждение**

Необходимо нажать CTRL+C, чтобы остановить команду и показать результаты.

Чем больше круглое число соединения в миллисекундах, тем выше ожидания, что может свидетельствовать о сетевой проблеме между вашим компьютером и сервером. <ping> использует обязательные датаграммы ECHO\_REQUEST протокола ICMP для получения по этому протоколу ответов ECHO\_RESPONSE от хоста или шлюза. Датаграммы ECHO\_REQUEST состоят из заголовков IP и ICMP, структуры данных struct timeval и произвольного числа не смысловых байтов для заполнения пакета.

### CMS

Система управления содержимым (контентом) (англ. Content management system, CMS) — информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления контентом (то есть содержимым).

Основные функции CMS:

* Предоставление инструментов для создания содержимого, организация совместной работы над содержимым,
* Управление содержимым: хранение, контроль версий, соблюдение режима доступа, управление потоком документов и т. п.,
* Публикация содержимого,
* Представление информации в виде, удобном для навигации, поиска.
* В системе управления содержимым могут находиться самые различные данные: документы, фильмы, фотографии, номера телефонов, научные данные и так далее. Такая система часто используется для хранения, управления, пересмотра и публикации документации. Контроль версий является одним из основных её преимуществ, когда содержимое изменяется группой лиц.

### WordPress

WordPress — система управления содержимым сайта с открытым исходным кодом, распространяемая под GNU GPL. Написана на PHP, в качестве сервера базы данных использует MySQL. Сфера применения — от блогов до достаточно сложных новостных ресурсов и интернет-магазинов. Встроенная система «тем» и «плагинов» вместе с удачной архитектурой позволяет конструировать практически любые проекты. WordPress выпущен под лицензией GPL версии 2.

Требования:

* PHP version 5.6 or greater
* MySQL version 5.5 or greater

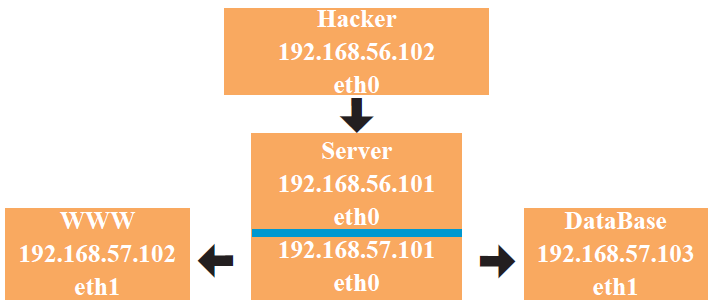
**PHP** (англ. PHP: Hypertext Preprocessor — «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools — «Инструменты для создания персональных веб-страниц»; произносится пи-эйч-пи) — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

**MySQL** — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

## Задания к лабораторной работе

### Часть 1

* Установите на 4 виртуальные машины операционную систему Ubuntu Server. Условно назовем эти машины: Hacker, Server, WWW, DataBase.



* Настроите сеть. В настройках сети (По умолчанию Файл-> настройки-> сеть, на вкладке Виртуальные сети хоста) создаем 2 адаптера ( По умолчанию 1 уже создан).
* Для каждого из них прописать разные IP-адреса 192.168.\*.\*, где \* - любое число от 0 до 255 (Если хотите сделать все как на схеме, на первом оставьте значения по умолчанию - 192.168.56.1, а на втором 192.168.57.1). На уже имеющимся адаптере можете посмотреть настройки DHCP и, по аналогии, настроить DHCP для второго адаптера. DHCP-сервер будет выдавать всем подключенным в сеть машинам IP-адреса автоматически через определенные промежутки времени. В данной лабораторной работе настройка DHCP в VirtualBox никак не отразиться на ее выполнении, а наоборот, только упростит построение сети между виртуальными машинами.
* Теперь в настройках каждой из виртуальных машин выберите вкладку сеть.
* В машине Hacker создайте 2 адаптера. В первом, чтобы вы могли использовать интернет, выберите тип подключения NAT. Во втором - виртуальный адаптер хоста (если Вы делаете все по схеме выберите тот, где ip-адрес 192.168.56.\*).
* На машине Server создайте 3 адаптера. Первый - чтобы использовать интернет. Второй - тот же виртуальный адаптер хоста, что и в машине Hacker. Третий - виртуальный адаптер хоста, но выбираете уже второй ( если вы делаете все по схеме выбираете тот, где ip-адрес 192.168.57.\*).
* На машинах WWW и DataBase создайте 2 адаптера. Первый - выход в интернет. Второй - виртуальный адаптер хоста (второй, тот где 196.168.57.\*).
* Особенностью подключения типа виртуальный адаптер хоста, является то, что компьютер, на котором запущен VirtualBox так же доступен, что может помочь во второй части лабораторной работы.
* Настройка статического IP-адреса (если используется на DHCP). Настройка сети осуществляется с помощью создания виртуального адаптера хоста. При первичном запуске всех виртуальных машин необходимо внести изменения в файл /etc/network/interfaces.
* nano /etc/network/interfaces
* В нем необходимо прописать новое соединение в форме:
* auto eth1
* iface eth1 inet static
* address 192.168.58.103
* netmask 255.255.255.0
* gateway 192.168.58.102

Поле address — ip адрес, выделяющийся машине, netmask — маска сети, используемая для разграничения адреса сети и непосредственно машины в ней, а gateway — шлюз, на который пойдут отправляемые пакеты. В данной лабораторной работе, например, в поле gateway пишется ip-адрес виртуального адаптера хоста на сервере, соответствующего локальной сети. Поэтому gateway не нужно прописывать для сервера, если этот сервер сам не подключается к какому-либо другому серверу. Запись осуществляется с правами суперпользователя. После записи необходима перезагрузка системы.

Вы можете использовать те же ip-адреса, что указаны на схеме. Для машины Server в настройках пропишите два сетевых интерфейса eth1 и eth2, из-за двух виртуальных адаптеров хоста. Названия сетевых интерфейсов начинаются с eth0, и во всех машинах, настраиваемых нами, eth0 будет автоматически использован первым виртуальным адаптером, который мы используем для выхода в интернет.

* Чтобы изменения вступили в силу необходимо перезапустить систему.
* Просмотреть сетевые интерфейсы вы можете, использовав команду ifconfig
* После настройки, используйте команду ping и проверьте локальное соединение.
* Машина Hacker – «злоумышленник», который попытается просканировать нашу сеть. Для того, чтобы он смог это сделать, необходимо установить универсальное средство сканирования – Nmap:
* sudo apt-get install nmap
* Server – сервер, атаку на который совершает Hacker. Во второй части работы будем устанавливать на него CMS WordPress.
* WWW – web-сервер, с помощью которого осуществляется доступ к базе данных DataBase. Соответственно, как и для Server, устанавливаем web-сервер. Также необходимо установить PHP.
* sudo apt-get install php5 libapache2-mod-php5 php5-mysql

Далее, необходимо создать файл /var/www/index.php, в который прописывается следующий скрипт:

<?php

$link = mysql\_connect('192.168.57.101', 'sit', 'sit');

**if** (!$link) {

**die**('Error: ' . mysql\_error());

}

**echo** 'Ok';

mysql\_close($link);

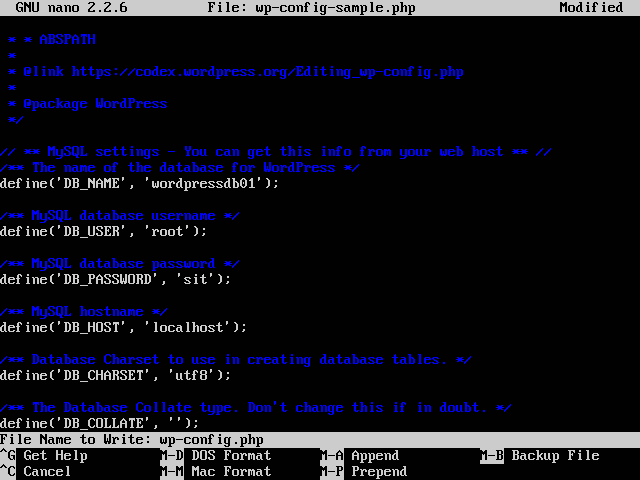
?>

Этот скрипт определяет подключение WWW к DataBase, и если ввести в адресное поле браузера ip-адрес WWW /index.php, то при успешном подключении к базе данных будет выведено «ОК».

* Настройка базы данных. На DataBase необходимо установить Mysql – универсальную систему управления базами данных.
* sudo apt-get install mysql-server
* Запускаем MySQL:
* sudo mysql -p
* Затем, в MySql необходимо добавить набор привилегий для пользователя
* (по умолчанию sit)
* ``GRANT ALL PRIVILEGES ON \*.\* TO sit@localhost IDENTIFIED``
* BY 'sit' WITH GRANT OPTION
* ``GRANT ALL PRIVILEGES ON \*.\* TO sit@"%" IDENTIFIED BY``
* 'sit' WITH GRANT OPTION
* Далее необходимо в файле /etc/mysql/my.cnf найти и закомментировать (поставить в начале строки символ #) строчку bind-address = 127.0.0.1.

### Часть 2

* Установку CMS необходимо производить на виртуальную машину Server из первой части лабораторной работы.
* Для настройки CMS WordPress, виртуальная машина Server должна быть подключена по локальной сети с машиной с графической оболочкой.
* На виртуальную машину Server необходимо установить набор програмного обеспечения LAMP (Linux Apache MySQL PHP)
* sudo apt-get update
* sudo apt-get install tasksel
* sudo tasksel
* Выберите LAMP. Чтобы отметить выделите и нажмите пробел. И нажмите Enter. Задайте пароль root ( Используйте sit)
* Установите phpmyadmin <sudo apt-get install phpmyadmin>. Выберите Apache, выберите Yes, введите пароль.
* На второй машине в браузере введите ip-адрес/phpmyadmin (Например 192.168.1.113/phpmyadmin)
* Введите имя пользователя и пароль. (Пользователь по умолчанию - root. Пароль - тот который вводили при установке phpmyadmin)
* Перейдите на вкладку Databases. Создайте Новую базу данных. В поле под надписью **Create Database** введите название базы данных (wordpressdb01) и нажмите **Create**.
* Вернитесь к виртуальной машине Server. Скачайте WordPress <wget <http://wordpress.org/latest.tar.gz>>
* Распакуйте скачанный архив в /var/www/html <sudo tar -xvpzf latest.tar.gz -C /var/www/html/>
* Теперь на второй машине введите 192.168.1.113/wordpress и нажимайте на кнопки, чтобы перейти к дальнейшим настройкам, пока не дойдете до настроек БД.
* В настройках введите имя БД - wordpressdb01, имя пользователя БД - root и пароль - sit. Нажимаете Submit.
* Вам выводят, каким должен быть быть файл wp-config.php в папке wordpress. И чтобы создать его и настроить, переходим к виртуальной машине Server.
* Перейдите в папку /var/www/html/wordpress
* Откройте в текстовом редакторе wp-config-sample.php <nano wp-config-samle.php>
* Измените настройки так, как показано на скриншоте:



* Cохраните под именем wp-cjnfig.php в той же папке. Когда будете нажимать CTRL-O и вас будут спрашивать имя сохраняемого файла, поменяйте на wp-config.php
* Возвращаемся ко второй машине и браузеру, нажимаем кнопки для дальнейшей установки.
* Заполняем настройки. Настройки потом можно будет поменять, неважно, что вы там в данный момент напишите, но запомните имя и пароль, вам их предстоит использовать сражу же после настройки, чтобы залогиниться.
* Нажимаете Install WordPress.
* Нажимаете Log In, вводите имя и пароль, которые вы только что задали.
* На этом установка CMS WordPress закончена.

## Вопросы к лабораторной работе

1. Что такое ЛВС?
2. Как проверить есть ли локальное сетевое соединение?
3. Что такое ping?
4. Какую информацию можно получить командой ping?
5. Перечислите типы подключения в Oracle VM VirtualBox.
6. Как создать локальную сеть в VirtualBox?
7. Что такое NAT?
8. Какие ограничения есть у NAT?
9. Если в настройках двух виртуальных машин выбрать Сетевой мост, будут ли они подключены по локальной сети?
10. В чем отличия между типами подключений Сетевой мост и Внутренняя сеть?
11. Что такое PHP?
12. Что такое MySQL?
13. Для чего нужна маска сети?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

# Лабораторная работа №9. Нагрузочное тестирование web-сервера.

## Основные теоретические сведения

**Цель:** С помощью систем нагрузочного тестирования определить производительность web-серверов Apache и Nginx, добиться отказа в обслуживании.

### Термины и определения

**Нагрузочное тестирование** - это автоматизированный процесс, имитирующий одновременную работу определенного количества пользователей на каком-либо общем ресурсе.

**Приложение** - тестируемое прикладное программное обеспечение.

**Виртуальный пользователь** - программный процесс, который циклически выполняет моделируемые операции.

**Итерация** - один повтор в цикле операции.

**Интенсивность выполнения операции** - частота выполнения операций в единицу времени, в тестовых скриптах задается интервалом времени между итерациями.

**Нагрузка** - совокупное количество попыток выполнить операции на общем ресурсе. Создается или пользовательской (клиентской) активностью или нагрузочными скриптами.

**Производительность** - количество выполняемых приложением операций в единицу времени.

**Масштабируемость приложения** - пропорциональный рост производительности при увеличении нагрузки.

**Профилем нагрузки** называется набор операций с заданными интенсивностями, полученными на основе статистики.

**Нагрузочной точкой** называется рассчитанное (либо заданное Заказчиком) количество виртуальных пользователей в группах, выполняющих операции с определенными интенсивностями.

**Тест производительности, бенчмарк (англ. benchmark)** — контрольная задача, необходимая для определения сравнительных характеристик производительности компьютерной системы.

Успешное прохождение ряда тестов является свидетельством стабильности системы в штатном и в разогнанном режимах.

### Цели нагрузочного тестирования

1. Оценка работоспособности и производительности приложения на этапе разработки и при передаче в эксплуатацию
2. Оценка работоспособности и производительности приложения на этапе выпуска новых релизов, патч-сетов
3. Оптимизация производительности приложения, включая оптимизацию программного кода и настройку серверов
4. Подбор соответствующей для данного приложения аппаратной и программной платформы, а также нужной конфигурации сервера

### Виды нагрузочного теста

**Нагрузочный (Load-testing)** – определяет работоспособность системы при некотором строго заданном уровне нагрузки (планируемой, рабочей).

**Устойчивости (Stress)** – используется для проверки параметров системы в экстремальных условиях и условиях сверхнагрузки, основная задача во время испытания - нарушить нормальную работу системы. Позволяет определить минимальные системные требования для работы приложения, оценить предельные возможности системы и факторы, которые ограничивают эти возможности. В рамках теста также определяется возможность системы сохранить целостность данных при возникновении внештатных аварийных ситуаций.

**Производительности (Performance)** – комплексный тест, включает в себя предыдущие два режима тестирования и предназначен для общей оценки всех показателей системы.

**Результат теста** – представляет собой максимально возможное число пользователей, которые могут единовременно получить доступ к веб-ресурсу, число запросов, которое в состоянии обработать приложение, или время ответа сервера. На основе этой информации, веб-мастер и сетевой администратор смогут заранее выявить слабые места, возникающие из-за несбалансированной работы компонентов (базы данных, маршрутизаторы, кэширующий и прокси-сервер, брандмауэры и др.), и исправить ситуацию, прежде чем система будет запущена в рабочем режиме.

### Использование Apache benchmark tool

**Apache benchmark** — одна из самых простых утилит, которая применяется для нагрузочного тестирования сайта. Идет в комплекте с веб сервером Apache, в первоначальной настройке не нуждается. Задача, которая ставится перед Apache benchmark — показать, какое количество запросов сможет выдержать веб сервер и как быстро он их обработает.

Пример нагрузки на сервер в 5000 последовательных запросов:

<ab -n 5000 <http://192.168.1.116/index.html>>

Пример нагрузки на сервер в 5000 запросов, но 500 из них будут направлены на сервер одновременно (параллельные запросы):

<ab -n 5000 -c 500 <http://192.168.1.116/index.html>>

**Задания к лабораторной работе**

Для выполнения лабораторной работы меняйте значения после -n и -c, чтобы узнать с каким количеством запросов может справиться сервер. На этих примерах выполнены HTML-запросы, для тестирования на PHP-запросы измените цель на index.php (В лабораторной работе №8 есть информация о PHP).

### Использование httperf

Еще одно консольное приложение, используемое также для создания нужного количества параллельных запросов - httperf.

Его отличие от ab в том, что httperf посылает запросы согласно своим настройкам, невзирая на то, отвечает сервер на них или уже нет. Таким образом можно определить не только какую максимальную нагрузку может выдержать сервер, но и как будет себя вести сервер в момент, когда нагрузка достигла своего пика

Пример запуска 100 запросов от 10 посетителей параллельно:

httperf --port 80 --server <domain> --uri=/ --num-conns=100 --rate=10

### Использование Siege

Установка:

sudo apt-get install siege

Количество запросов не лимитируется, но можно задавать время в течение которого выполнять тестирование. Пример: 5 пользователей, которые безостановочно загружают главную страницу в течении одной минуты.

siege -c 5 -b -t 1m ip-адрес

### Балансировщик нагрузки Nginx

Балансировка нагрузки (англ. load balancing) — метод распределения заданий между несколькими сетевыми устройствами (например, серверами) с целью оптимизации использования ресурсов, сокращения времени обслуживания запросов, горизонтального масштабирования кластера (динамическое добавление/удаление устройств), а также обеспечения отказоустойчивости (резервирования).

Установка:

sudo apt-get install nginx

Настройка:

sudo nano /etc/nginx/sites-available/default

upstream web\_backend {

server 192.168.1.113;

server 192.168.1.114;

}

server {

listen 80;

location / {

proxy\_set\_header X­Forwarded­For $proxy\_add\_x\_forwarded\_for;

proxy\_pass http://web\_backend;

}

}

После настройки перезапускаем Nginx

sudo service nginx reload

Методы балансировки нагрузки (описываются в начале секции upstream):

* ip\_hash - согласно этому методу запросы от одного и того же клиента будут всегда отправляться на один и тот же backend сервер на основе информации об ip адресе клиента. Не совместим с параметром weight.
* least\_conn - запросы будут отправляться на сервер с наименьшим количеством активных соединений.
* round-robin - режим по умолчанию. То есть если вы не задали ни один из вышеупомянутых способов балансировки - запросы будут доставляться по очереди на все сервера в равной степени.

## Задания к лабораторной работе

### Нагрузочное тестирование веб-сервера с Apache

Для тестирования используются 2 машины – одна с установленным и работающим Apache, вторая будет отсылать запросы и делать выводы о производительности web-сервера.

Тестирование на PHP-запросы:

* Определить максимальное число параллельных запросов, при котором сервер нас не будет блокировать.
* Провести тест при использовании максимального числа запросов.

Тестирование на HTML-запросы:

* Определить максимальное число параллельных запросов
* Провести тест при использовании максимального числа запросов.

Провести сравнение результатов и сформировать выводы.

### Нагрузочное тестирование веб-сервера с Nginx.

Для тестирования используется 2 виртуальные машины – одна с установленным и работающим Nginx, которой будут отсылаться запросы, другая будет отсылать эти самые запросы и делать выводы о производительности веб-сервера с Nginx.

**Задания к лабораторной работе**

<sudo apt-get nginx> - Установка Nginx

Тестирование на PHP-запросы:

* Определить максимальное число параллельных запросов, при котором сервер нас не будет блокировать.
* Провести тест при использовании максимального числа запросов.
* Сравнить с результатами, полученными при тестировании Apache

Тестирование на HTML-запросы:

* Определить максимальное число параллельных запросов.
* Провести тест при использовании максимального числа запросов.
* Сравнить с результатами, полученными при тестировании Apache

Провести сравнение результатов и сформировать выводы.

### Нагрузочное тестирование веб-серверов Apache с балансировщиком нагрузки.

Для тестирования используется 4 машины – две одинаковые с установленным и работающим Apache в качестве веб-серверов, которые соединены с третьей машиной, которая выполняет роль балансировщика нагрузки, на нем работает Nginx, четвертая машина будет отсылать эти запросы серверу и делать выводы о производительности данной связки из балансировщика нагрузки на Nginx и двумя веб-серверами Apache.

Тестирование на PHP-запросы:

* Провести тест при использовании максимального для Apache числа запросов
* Провести тест при использовании максимального для Nginx числа запросов
* Сравнить с предыдущими результатами и сформировать выводы

Тестирование на HTML-запросы:

* Провести тест при использовании максимального для Apache числа запросов
* Провести тест при использовании максимального для Nginx числа запросов
* Сравнить с предыдущими результатами и сформировать выводы

### Нагрузочное тестирование веб-серверов Nginx с балансировщиком нагрузки.

Для тестирования используется 4 виртуальные машины – две одинаковые с установленным и работающим Nginx в качестве веб- серверов, которые соединены с третьей машиной, которая выполняет роль балансировщика нагрузки, на нем работает Nginx, четвертая машина будет отсылать эти запросы серверу и делать выводы о производительности данной связки из балансировщика нагрузки на Nginx и двумя веб-серверами Nginx.

Тестирование на PHP-запросы:

* Провести тест при использовании максимального для Apache числа запросов
* Провести тест при использовании максимального для Nginx числа запросов
* Сравнить с предыдущими результатами и сформировать выводы

Тестирование на HTML-запросы:

* Провести тест при использовании максимального для Apache числа запросов
* Провести тест при использовании максимального для Nginx числа запросов
* Сравнить с предыдущими результатами и сформировать выводы

| *Итоговая таблица сравнения* | | | | | |
| --- | --- | --- | --- | --- | --- |
|  |  | **Максимальное число запросов** | **Запросы/сек** | **Время, затрачиваемое на запрос, мс** | **% успешных запросов** |
| Apache | PHP |  |  |  |  |
|  | HTML |  |  |  |  |
| LB + Apache | PHP |  |  |  |  |
|  | HTML |  |  |  |  |
| Nginx | PHP |  |  |  |  |
|  | HTML |  |  |  |  |
| LB + Nginx | PHP |  |  |  |  |
|  | HTML |  |  |  |  |

## Вопросы к лабораторной работе

1. Назначение нагрузочного тестирования?
2. Что такое нагрузка?
3. Как указать ab сделать нагрузку в 10000 запросов, 500 из которых будут направлены одновременно? Перестанет ли ваш сервер принимать входящие подключения?
4. Можно ли протестировать при помощи ab, httperf и Siege другие web-сервера? Назовите примеры.
5. Влияет ли использование скриптовых языков программирования (например, PHP) на производительность web-сервера? Объясните почему.
6. Для чего нужен балансировщик нагрузки?
7. Какие существуют методы балансировки нагрузки в nginx?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

* [Docs](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/index.html) »

* Лабораторная работа №10. DDoS (Distributed Denial of Service) - основные особенности их организации и защиты от них.

# Лабораторная работа №10. DDoS (Distributed Denial of Service) - основные особенности их организации и защиты от них.

**Цель:** Получить теоретические и практические навыки о принципах осуществления DDoS атак и защиты от них.

## Основные теоретические сведения

**DoS/DDoS**

DoS (от англ. Denial of Service — отказ в обслуживании) — хакерская атака на вычислительную систему с целью довести её до отказа, то есть создание таких условий, при которых легальные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ затруднён. Отказ «вражеской» системы может быть и шагом к овладению системой (если в нештатной ситуации ПО выдаёт какую-либо критическую информацию — например, версию, часть программного кода и т.д.). Но чаще это мера экономического давления: простой службы, приносящей доход, счета от провайдера и меры по уходу от атаки ощутимо бьют «цель» по карману. В настоящее время DoS и DDoS-атаки наиболее популярны, так как позволяют довести до отказа практически любую систему, не оставляя юридически значимых улик.

Если атака выполняется одновременно с большого числа компьютеров, говорят о DDoS-атаке (от англ. Distributed Denial of Service, распределённая атака типа «отказ в обслуживании»).

Для обнаружения распределённых сетевых атак типа «отказ в обслуживании» и защите от них необходимо классифицировать их и знать принципы их работы.

В данной работе критерием для классификации рассмотрим объект, на который нацелена атака. В таком случае, получается четыре основных класса атак, соответствующих уровням модели ISO OSI.

**Канальный уровень (L2)** — атаки направлены на исчерпание ёмкости сетевого канала. В следствие этого лишается доступ сервера к внешней сети. Для реализации используются объёмные потоки трафика. На данным момент измеряются в Гб/с. Во время этой атаки обрабатывать трафик необходимо на стороне провайдера, дата-центра. С помощью BGP Flow Spec фильтруется часть атак по сигнатурам пакета. Amplification атаки отсекаются по порту.

**Сетевой уровень (L3)** — атаки направлены на нарушение работы элементов сетевой инфраструктуры. Необходим ручной анализ сетевой инфраструктуры. Если своей автономной системы нет, то борьба с атаками данного класса ведется провайдером или дата-центром. Желательно сотрудничество с ними.

**Транспортный уровень (L4)** — атаки направлены на эксплуатацию слабых мест TCP-стека. В TCP протоколе используется таблица открытых соединений. Атаки именно на неё составляют этот класс. Необходим постоянный анализ поведения TCP-стека, TCP-клиентов, TCP-пакетов. Эвристический анализ.

**Прикладной уровень (L7)** — атаки направленны на нарушение работы Web-приложения. Атаки этого класса характеризуются большим разнообразием. Исчерпывают ресурсы сервера. Необходим поведенческий и корреляционный анализ, мониторинг ресурсов сервера. Необходима оптимальная настройка сервера под решаемые им задачи. Полностью автоматизировать борьбу с данным классом атак почти невозможно.

### Медленная атака. SlowLoris

**Уровень атаки: транспортный уровень (L4).**

**Описание и принцип работы:**

Атака Slowloris устанавливает много открытых соединений на сервере с помощью постоянной отправки незавершенных HTTP-запросов. В определенные моменты времени Slowloris отправляет следующие HTTP заголовки для каждого запроса, но не завершает соединение. Если запросы посылаются с оптимальной периодичностью, сервер начинает ожидать завершения открытых соединений. В данном случае ресурсы сервера остаются относительно свободными, но сам сервер перестаёт обслуживать новые подключения.

Дело в том, что веб-серверы Apache 1.x, Apache 2.x, dhttpd, GoAhead WebServer и Squid поддерживают ограниченное число одновременно открытых подключений. Но Slowloris не представляет угрозы для серверов IIS, lighttpd, NGINX. Они имеют эффективные механизмы распределения нагрузки и используют worker pool — «пулы рабочих потоков», которые позволяют удерживать любое количество открытых соединений при наличии свободных ресурсов.

### Медленная атака. Slow HTTP POST/GET

**Уровень атаки: транспортный уровень (L4).**

**Описание и принцип работы:**

Атака основана на уязвимости в протоколе HTTP. Slow HTTP POST атака отправляет POST заголовок с полем «Content-Length». Веб-сервер понимает, какой объём данных он должен получить. После этого с очень низкой скоростью передаётся тело POST сообщения. Это позволяет задействовать ресурсы сервера длительное время, и в последствии помешать обработке других запросов. Атака опасна для веб-серверов Microsoft IIS и Apache и NGINX со стандартными настройками в рамках протоколов HTTP, HTTPS, подключений SSL, VPN. Также атака может быть настроена для работы с SMTP и DNS-серверами.

Данный тип атаки на отказ в обслуживании, можно организовать через proxy. Трафик данной атаки схож с легитимным трафиком.

### Медленная атака. Sockstress

**Уровень атаки: транспортный уровень (L4).**

**Описание и принцип работы:**

Атака заключается в следующем. Если на веб-сервере есть объект, размер которого больше send buffer, выделенного ядром для соединения. То можно заставить ядро не принимать данные, а сервер будет пробовать отправить кусок данных, занимая стек соединений, ресурсы процессора и память. При большом количество подобных соединений TCP-стек заполниться и не будет открывать новые соединения.

**Примеры:**

Отправить в пакете размер окна равный нулю, то есть нет места для получения данных. Скрипт Sockstress отправляет такие пакеты и считает время, когда их отправлять, чтобы не загрузить persist timer.

Создать сокет с малым объёмом receive buffer на клиенте. Отправлять HTTP-запрос на объект сайта больший по размеру, чем буфер. И периодически считываем пару байт из receive buffer. Сервер будет пытаться отправлять данные и занимать ресурсы.

### Атака произвольными пакетами. HTTP-Flood

**Уровень атаки: канальный уровень (L2), прикладной уровень (L7).**

**Описание и принцип работы:**

В основе этой атаки лежит механизм отправления максимального числа HTTP запросов на 80-й порт веб-сервера. Целью атаки может быть корень сервера или ресурсоёмкий элемент. В результате данной атаки возможно прекращение предоставления услуг по HTTP, и затруднен доступ легитимных пользователей к сайту. Распознать атаку можно с помощью выявления быстрого роста количества запросов к некоторым элементам веб-сервера и логов сервера.

### Атака произвольными пакетами. UDP-Flood

**Уровень атаки: канальный уровень (L2), прикладной уровень (L7).**

**Описание и принцип работы:**

UDP flood атака основана на отправке большого количества UDP-пакетов на некоторые порты сервера. Он должен определить приложение для каждого полученного пакета, удостоверится в его неактивности и отправить в ответ ICPM-сообщение «недоступен». В итоге вырастут затрачиваемые ресурсы атакуемого сервера и полоса пропускания заполнится UDP-пакетами. В UDP протоколе нет механизма проверки отправителя пакетов, тем самым злоумышленник может подменить IP-адреса и обеспечить анонимность.

### Атака произвольными пакетами. SYN-Flood

**Уровень атаки: канальный уровень (L2), транспортный уровень (L4).**

**Описание и принцип работы:**

SYN Flood атака использует механизм рукопожатия в протоколе TCP. Работает следующим образом. Посылается пакет с флагом SYN на атакуемый сервер. Он вынужден отправить в ответ пакеты с флагами SYN+ACK. Злоумышленник игнорирует SYN+ACK пакеты сервера и не высылает в ответ пакет ACK. Либо подделывает IP-адрес SYN пакета, чтобы ответный SYN+ACK отправляется на некорректный адрес. Цель данной атаки — заполнение TCP стека множеством полуоткрытых соединений, в следствие чего сервер перестают устанавливать соединения с новыми клиентами. Запросы на соединение полученные сервером хранятся в стеке с определенным размером, который зависит от операционной системы. Они находятся в стеке, пока сервер не получит информацию об установленном соединении от клиента.

### Атака произвольными пакетами. ICMP-Flood

**Уровень атаки: канальный уровень (L2).**

**Описание и принцип работы:**

Данный тип флуда направлен на сетевое оборудование. Принцип данной атаки заключается в том, что ICMP-пакет при небольшом размере самого запроса требует от устройств значительно большего объёма работы. То есть, при отправлении сравнительно небольшого объёма ICMP запросов возникает перегрузка сетевого оборудования и значительная часть легитимных запросов теряется. Злоумышленник, меняет IP-адрес источника, отправляет ICMP Echo Request пакет к определённым компьютерам, входящим в бот-нет. Они отвечают ICMP Echo Reply пакетом, посылая его на изменённый IP-адрес. Для увеличения мощности атаки используют локальные сети (LAN) с включенной опцией направленной широковещательной рассылки (directed broadcast).

### Атака с помощью SSL

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

Secure Sockets Layer (SSL) — это протокол безопасности для защиты целостности сети связи и передачи данных. SSL может зашифровать подключение к сети на транспортном уровне. Процессы шифрования протокола SSL, расшифровки и обмена ключами потребляют огромное количество системных ресурсов. Существует два типа атака основанных на протоколе SSL.

Первый тип эксплуатирует механизм рукопожатия, который исчерпывает ресурсы атакуемого сервера. Злоумышленник посылает некорректные SSL данные на сервер, на обработку которых затрачивается большое количество вычислительной мощности.

Второй тип использует функцию повторного подтверждения соединения — SSL Renegotiation.

Установка безопасного соединения и повторное подтверждение SSL затрачивают в разы больше вычислительной мощности на сервере, чем на стороне клиента. Благодаря этому возможно осуществление атаки и истощение ресурсов атакуемого сервера. Как правило, HTTPS расшифровывается глубоко внутри организационной сети, где серверы и модули более уязвимы к вредоносному трафику. Также злоумышленники используют этот протокол для обхода механизмов безопасности. Таким образом, возможно туннелирование других атак.

### Атака почтового сервера. SMTP-Flood

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

В атаках этого типа злоумышленник пытается установить соединение с почтовым сервером и отправляет произвольные письма на сгенерированные случайным образом адреса, либо бездействует до истечения тайм-аута, удерживая соединение открытым. Каждое SMTP соединение утилизирует часть ресурсов сервера, тем самым атакующий пытается вызвать отказ в обслуживании.

Технология заключается в том, что когда мы пишем e-mail получателя несуществующий, то SMTP server оповещает нас с помощью e-mail(отправителя), что письмо не дойдёт. Мы посылаем на SMTP server очень много запросов (пишем письма) и указываем все e-mail не валидные, а отправителя(объект атаки). И к нему будут приходить уведомления, что письмо не дошло. С нескольких SMTP server будет отправляться трафик на объект атаки.

### Некорректные пакеты/фрагменты. UDP fragment flood

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

Данный тип атаки основан на отправки UDP датаграмм, которые случайным образом ссылаются на датаграммы отсутствующие в потоке. Это приводит к увеличению потребления памяти на атакуемом сервере. При атаке UDP Fragment Flood, злоумышленники посылают UDP пакеты большого размера, для истощения пропускной способности канала.

### Некорректные IP-фрагменты

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

Данный тип атак эксплуатирует уязвимости в поддержке фрагментации пакетов протокола IP. Одна из атак этого типа — это пересечение IP-фрагментов. Она реализуется с помощью уязвимости операционной системы, которая заключается в сборке фрагментированных IP- пакетов. В процессе сборки образуется цикл по принятым фрагментам. Затем из них копируется информативная часть и передаётся на IP уровень. Разработчики предусмотрели проверку на чрезмерный объем копируемой информации, но не ввели проверку на копирование фрагмента отрицательной длины. Копирование блока информации отрицательной длины равносильно копированию очень большого блока информации. Это приводит к затиранию большого участка памяти и к нарушению работы ЭВМ (электронная вычислительная машина). Существует две программы с небольшими отличиями в константах механизма, который осуществляет пересечение IP-фрагментов: newtear и teardrop. Они отправляют пакеты с заданного IP-адреса на любой порт, независимо, открыт он или закрыт. Еще один вариант данной атаки - bonk. После сборки фрагментов в пакете остаются пустые места. Это приводит к сбою ядра операционной системы и нарушению работы электронных вычислительных средств. Данные уязвимости присутствуют в старых версиях ОС(операционных систем) Windows и Linux. На сегодняшний день большинство сетевых ОС защищены от сбоев в работе, вызванных данной атакой.

### Неверные значения в заголовках пакетов

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

Данные атаки нацелены на определённые приложения и операционные системы, которые неправильно обрабатывают некорректные значения в заголовках пакетов. Пример такой атаки — Land attack. В ходе неё злоумышленник устанавливает в пакете одинаковый IP-адрес для источника и приёмника. Это приводит к зацикливанию установки соединения сервера с самим собой.

### Amplification атаки

**Уровень атаки: канальный уровень (L2).**

**Описание и принцип работы:**

В основе данных атак лежит отсутствие проверки отправителя в UDP протоколе. Ответ посылается адресату, указанному в заголовках пакета. Злоумышленник может подменить свой IP-адрес на IP-адрес атакуемого сервера в заголовках отправляемых пакетов. Также суть атаки заключается в многократном превышении объёма ответа по сравнению с запросом. Таким образом, злоумышленник может анонимно организовывать атаки с огромным объёмом трафика. Службы, работающие по UDP протоколу: DNS, NTP, SNMP, rsyslog и многие другие, могут использоваться для реализации атаки. Дело в том, что сетевые устройства с этими службами встречаются в сети повсеместно. Службы включены по умолчанию и часто некорректно настроены.

В таблице 1 представлены типы amplification атак проведённых в ходе исследования. В ней отображается по какому протоколу осуществляются атаки, коэффициент их усиления и уязвимая команда, используемая для реализации атаки.

Таблица 1 — Amplification атаки

|  |  |  |
| --- | --- | --- |
| **Протокол** | **Коэффициент усиления** | **Уязвимая команда** |
| DNS | x28-x92 | DNS server request |
| NTP | x994 | Monlist request |
| SNMPv2 | x29 | GetBulk request |
| CharGEN | x350 | Character generation request |
| BitTorrent | x4 | File search |
| RIPv1 | x131 | Malformed request |
| SSDP | x31 | SEARCH request |
| NetBIOS | x4 | Name resolution |
| Quake Network Protocol | x64 | Server info exchange |
| Steam Protocol | x5.5 | Server info exchange |

￼￼￼￼￼￼￼￼￼￼￼￼￼￼ DNS, NTP, SNMPv2 — протоколы для получения информации о доменах, синхронизации времени и сетевого управления. Часто встречаются в сети. NetBIOS и SSDP — протоколы в Windows. CharGEN — старый тестовый сервис, но его до сих пор можно встретить на различных системах. BitTorrent — протокол для обмена файлами. Quake Network Protocol и Steam Protocol — протоколы компьютерных игр.

### NTP amplification атака

**Уровень атаки: канальный уровень (L2).**

**Описание и принцип работы:**

Злоумышленник отправляет запрос monlist с IP-адресом атакуемого сервера к NTP-серверу. Ответ monlist включает в себя список 600 последних клиентов ntpd. Сущность амплификации заключается в том, что нарушитель отправляет небольшой запрос к уязвимому серверу и с него на атакуемый сервер отправляется большой поток UDP трафика. Уязвимый NTP-сервер является невольным промежуточным звеном атаки. Ntpd до версии 4.2.7p26 подвержены атаке.

### DNS Amplification атака

**Уровень атаки: канальный уровень (L2).**

**Описание и принцип работы:**

Атака основана на том, что нарушитель отправляет запрос уязвимому DNS-серверу с IP-адресом атакуемого сервера. DNS-сервер отправляет ответ, размер которого во много раз превышает запрос, жертве. Таким образом, исчерпывается канальная ёмкость атакуемого сервера.

Можно выделить ключевые моменты атаки:

1. Эффект отражения: подмена IP-адреса позволяет перенаправить ответы от всех DNS-серверов на атакуемый сервер.
2. Коэффициент усиления атаки: (amplification factor): он может принимать значения от 28 до 92. То есть на 1 байт запроса — совокупность DNS-серверов отправит 28-92 байт ответа. Это обеспечивает кратное увеличение объёма трафика.
3. Проблема «open resolver»: это неправильно настроенный или старой версии DNS-сервер. Он разрешает получать запросы из сторонних сетей, выполняя рекурсивные запросы для них, и отправлять ответы без необходимых предварительных проверок.

### HTTP flood с помощью сервисов

**Уровень атаки: прикладной уровень (L7).**

**Описание и принцип работы:**

WordPress сайт с включенным Pingback, можно использовать для проведения HTTP flood атаки на другие сайты. Они отправляют множество запросов к атакуемому сайту со случайными параметрами («?a=a» и др.), с помощью которых обходится кэширование страницы. Эта операция быстро расходует ресурсы атакуемого сервера и нарушает его работу. Злоумышленник может использовать большое количество обычных WordPress сайтов для DDoS атаки и не бояться быть обнаруженным с помощью Pingback запросу к файлу XML-RPC. Google использует FeedFetcher для кэширования любого контента в Google Spreadsheet, вставленного через формулу =image(«link»). Если в клетку таблицы вставить формулу =image(«<http://target/file.pdf>»), то Google отправит FeedFetcher скачать этот PDF файл и закэшировать для дальнейшего отображения в таблице. Но если добавлять случайный параметр к URL картинке (от «?r=1» до «?r=1000»), FeedFetcher будет скачивать её каждый раз заново. Это приведёт к исчерпанию лимита трафика атакуемого сервера. Злоумышленник может запустить массированную HTTP GET flood атаку на веб-сервер, используя браузер с одной открытой вкладкой.

## Практическая часть.

Для эффективной защиты от распределённых сетевых атак типа «отказ в обслуживании» требуется своевременное обнаружение начала атаки. Чтобы обнаружить DDoS-атаку необходимо накапливать статистические данные о трафике сервера в сети, работающем в штатном режиме. При знании среднестатистических значений характеристик сервера, можно отследить появление аномалии трафика в сети.

Для выполнения данной лабораторной работы необходимо создать две виртуальные машины или объединить в локальную сеть два компьютера. Одна машина будет жертвой, другая злоумышленником.

На машине жертве необходимо установить:

* Web-сервер (Apache, NGINX);
* Средства мониторинга и анализа ресурсов сервера и сетевого трафика (iptraf, ksysguard или другие);

**Задания к лабораторной работе**

Дополнительно ознакомьтесь со следующими ресурсами [1](https://nginx.org/ru/) , [2](https://httpd.apache.org/) , [3](http://help.ubuntu.ru/wiki/%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) , [4](http://proubuntu.com.ua/2011/11/17/system-monitoring-ubuntu.html) .

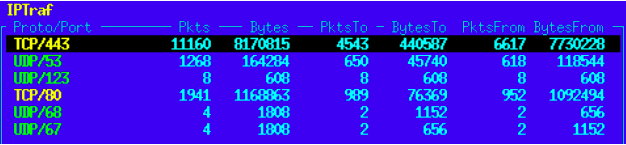
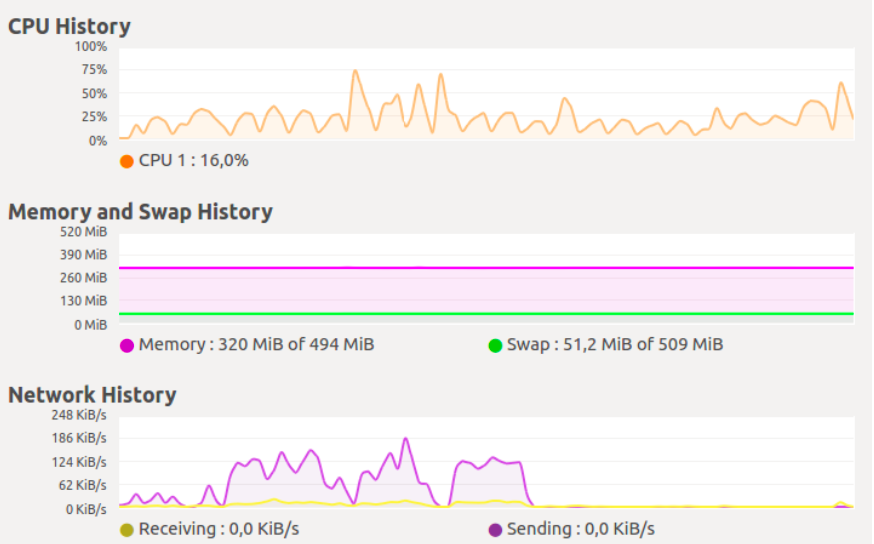
На машине злоумышленника необходимо установить:

* Программное обеспечение и скрипты для проведения DDoS атак;
* Уязвимые сервисы для усиления DDoS атак (dns, ntp, chargen);

**Задания к лабораторной работе**

Чтобы сделать сервис уязвимым, ознакомьтесь со следующими ресурсами [1](http://blogerator.ru/page/dns-amplification-ddos-otrazhenie-bind-dns-reflect-attack-chast-2) , [2](https://habrahabr.ru/post/209438/) .

На машине-жертве фиксируются системные характеристики, такие как: загрузки процессора, объем оперативной памяти и характеристики сетевого трафика, такие как: количество пакетов, открытые соединения.



**Скриншоты мониторинга и трафика**

**Задания к лабораторной работе**

Изменение указанных характеристик не обязательно обозначает атаку на сервер, но показывает отклонение от среднестатистических показателей работы в штатном режиме. Для большинства DDoS-атак можно выделить аномалии и признаки, соответствующие только им.

**В ходе данной лабораторной работы необходимо провести каждый из представленных типов DDoS атак, отследить аномалии в использовании ресурсов сервера и сетевом трафике, проверить признаки DDoS атаки и, если они подтвердились, защитить сервер от атаки с помощью настройки сервера, добавление правил для межсетевого экрана, фильтрации пакетов по сигнатурам.**

**Пошаговый алгоритм методики:**

1. Обнаружение аномалии в наблюдаемых системных характеристиках и характеристиках сетевого трафика.
2. Проверить присутствие признаков атаки в трафике.
3. Внести изменения в настройки сервера под конкретную атаку и внести правила для фильтрации.

Для каждого типа атак в сети интернет необходимо найти по несколько программ и скриптов, которые реализуют данный вид атаки.

* Для атаки HTTP flood: goldeneye, ddosim, DAVOSET, HULK, LOIC, BBHH.
* Для медленных атак: r-u-dead-yet, slowhttptest, pyloris, sockstress, torshammer. Для UDP flood: LOIC, fudp, hping3.
* Для SYN flood и атак на TCP-стек: sprut, sitekiller, mummy, hping3.
* Для ICMP flood: hping3.
* Для land атаки: fudp, hping3.
* Для атак с использованием SSL: thc-ssl-dos.
* Для amplification атак: saddam, chargen\_amp.

**Задания к лабораторной работе**

Скачать набор программного обеспечения можно по этой [ссылке](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/_static/DDoS.rar) .

**Предупреждение**

Ознакомьтесь со справочной документацией используемых Вами программ и скриптов.

## Дополнительная информация

**Медленные атаки**

**Аномалия:** Кратковременный скачок нагрузки процессора, повышение исходящего трафика и используемой оперативной памяти.

**Признак:** В TCP-стеке большое количество соединений со статусом ESTABLISHED.

**Защита:** Если установлен apache, то поставить перед ним кэширующий сервер NGINX. Установить и настроить балансировщик нагрузки. На apache можно установить mod\_security — firewall с готовыми правилами от OWASP и mod\_reqtimeout — установка таймаутов и минимальной скорости передачи данных для получения запросов.

Добавить правила межсетевого экрана:

* ограничение количества соединений с одного IP-адреса;

*# iptables -A INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 30 -j DROP*

* заблокировать IP после 10 подключений к порту 80 в течение 30 секунд;

*# iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent — set*

*# iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 30 --hitcount 10 -j DROP*

Также NGINX имеет функцию кэширования клиентского запроса перед отправкой на бэкэнд — client\_body\_buffer\_size. Бэкэнд получит запрос, только когда он полностью загрузится.

**Атака произвольными пакетами.**

**Аномалия:** Резкое увеличение нагрузки процессора, входящего и исходящего-HTTP трафика.

**Признак:** На сервер приходит много однотипных пакетов, резко выросло количество обращений к ресурсоёмкому элементу сайта.

**Защита:** Борьба с HTTP flood ведётся с помощью усовершенствования работы веб-сервера и баз данных. Также меры включают в себя использование обработки подключений методом epoll, увеличение количества соединений, отключение тайм-аута на закрытие подключений keep-alive.

worker\_processes 2; worker\_rlimit\_nofile 8000; events {

worker\_connections 4000;

use epoll; }

Установить на NGINX модуль ngx\_http\_limit\_req\_module и блокировать IP-адреса, которые стали получать ответ «Service unavailable».

http {

limit\_req\_zone $binary\_remote\_addr zone=zne:15m rate=3r/s; server {

location / {

limit\_req zone=zne burst=5;

}

**Атака произвольными пакетами. UDP-Flood**

**Аномалия:** Резкое увеличение нагрузки процессора, входящего UDP-трафика.

**Признак:** Со всех открытых UDP-портов идёт исходящий трафик.

**Защита:** Необходимо выставить ограничение на количество подключений к открытым портам и закрыть неиспользуемые порты с помощью межсетевых экранов.

Добавить правила межсетевого экрана:

* ограничить количество подключений;

*# iptables -I INPUT -p udp --dport 53 -j DROP -m iplimit --iplimit-above 1 • разрешить подключение только доверенным IP-адресам;*

*# iptables -A OUTPUT -p udp --dport 53 -d 8.8.4.4 -j ACCEPT*

* блокировать все остальные порты.

*# iptables -A OUTPUT -p udp -j DROP*

**Атака произвольными пакетами. SYN-Flood**

**Аномалия:** Кратковременный скачок используемых ресурсов процессора и многократное увеличение входящего и исходящего трафика, и их количество равно.

**Признак:** В TCP-стеке появляется большое количество полуоткрытых соединений со статусом SYN\_RECV.

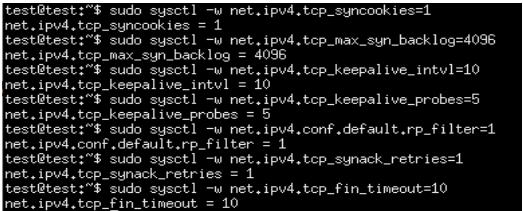
**Защита:** Современные реализации TCP протокола и некоторые межсетевые экраны имеют механизм защиты от SYN flood атак.

Принцип его действия заключается в следующем:

1. Серверу отправляется запрос на установление соединения, механизм регистрирует его в таблице.
2. После ответа сервера о подтверждении запроса на соединение механизм отправляет пакет серверу с подтверждением соединения и отправляет ответный пакет клиенту. В этот момент в механизме запускается таймер, отсчитывающий время на ответ от клиента.
3. При получении пакета с подтверждением соединения механизм считает запрос валидным, остановит таймер и отправит его на сервер.
4. Если пакет не пришёл в установленное время, механизм отправляет серверу запрос на удаление информации о соединении.

Принцип действие механизма основан на контроле времени задержки ответа клиента. При слишком коротком тайм-ауте будут обрываться легитимные пользователи, при длинном будут накапливаться соединения, что может привести к серьёзным последствиям.

Также можно использовать SYN cookie, или ограничить количество запросов на новые подключения от конкретного пользователя за конкретный период времени. На Рисунке 1 представлен пример настройки ядра операционной системы для защиты от SYN-Flood с помощью команды sysctl.



**Атака произвольными пакетами. ICMP-Flood**

**Аномалия:** Очень сильная загрузка процессора и сильное увеличение входящего и исходящего ICMP трафика, равного по значениям.

\*\* Признак:\*\* Множество пакетов передаётся по протоколу ICMP.

**Защита:**

Для противодействия атаке возможны следующие меры:

* отключить ответы на ICMP-запросы;

*#sysctl net.ipv4.icmp\_echo\_ignore\_all=1*

* понизить приоритет обработки ICMP-сообщений;
* отбросить или фильтровать ICMP-трафик межсетевым экраном;

*# iptables -A INPUT -p icmp -j DROP --icmp-type 8*

* увеличить очередь обрабатываемых подключений.

**Атака с помощью SSL**

**Аномалия:** Резкое повышение используемых системных ресурсов.

**Признак:** Появляется HTTPS трафик. Большое количество обращений к SSL серверу.

**Защита:** Можно установить правила разрыва соединения с пользователем, выполняющего функцию повторного подтверждения больше заданного количества раз в определённый период времени. Можно использовать контроллер доставки приложений (ADC), чтобы выгрузить SSL с сервера и использовать Web Application Firewall (WAF), для просмотра трафика на наличие атак. Но грамотно спланированные атаки могут превысить количество подключений к этим устройствам и нарушить их работу. Необходимо проводить поведенческий анализ. Целью этих методов является снижение объема трафика атаки пока ресурсы сервера не смогут эффективно бороться с ним. Данный тип атак не отличается поведением от легитимного трафика на сетевом уровне. Из-за этого борьба не всегда эффективна и механизмы ослабления склонны к ложным срабатываниям.

**Amplification атаки**

**Аномалия:** Огромное увеличение входящего трафика, которое может достигать до 600 Гб/с.

**Признак:** Увеличение входящих UDP-пакетов на порты, используемые уязвимыми сервисами. Например, NTP 123 порт, DNS 53 порт, СharGEN 19 порт.

**Защита:** Такие атаки фильтруются по порту источника и сигнатурам пакетов, так как для каждой amplification атаки используется определённая уязвимая команда сервиса, сигнатура которой известна.

В межсетевом экране необходимо добавить правила:

* закрыть неиспользуемые UDP порты;

*# iptables -A INPUT -p udp -j DROP*

* разрешить подключение к Х UDP порту только с IP-адреса используемого сервиса;

*#iptables -A INPUT -p udp --dport Х -d x.x.x.x -j ACCEPT*

* пропускать только пакеты, которые имеют определённую сигнатуру 0x00=0x00000000.

*# iptables -A INPUT -p udp --dport Х -m u32 --u32 «0x00=0x00000000» -j ACCEPT*

## Задания к лабораторной работе:

1. Установить необходимое ПО и настроить сервера.
2. Сделать скриншоты мониторинга ресурсов сервера и сетевого трафика в штатном режиме.
3. Провести несколько атак из списка.
4. Сделать скриншоты мониторинга для каждой проведённой атаки.
5. Написать какие аномалии вы наблюдали в используемых ресурсах сервера, характеристиках сервера, и что является признаком атаки.
6. Защитить сервер от атак и сделать скриншоты результатов.
7. Оформить отчёт по проделанной работе.

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

## Вопросы к лабораторной работе

1. Принцип работы каждой из сетевых атак типа «отказ в обслуживании» ?
2. Какие аномалии в сетевом трафике и ресурсах сервера вы заметили в каждой̆ атаке ?
3. Что является признаком конкретной атаки ?
4. В чем отличие DDoS от DoS?
5. Что такое UDP-флуд?
6. Как защититься от различных атак DoS/DDoS?
7. Что такое SYN-флуд? Укажите решение защиты от SYN-флуда с помощью iptables.
8. Как определить, что осуществляется атака SYN-флуд?
9. Как осуществить защиту от HTTP-флуда?
10. Как работают атаки типа SlowLoris?
11. Как совершить DDoS-атаку на почтовый сервер и в дальнейшем реализовать его защиту?
12. Как работают атаки типа SSL? Как определить, что осуществляется атака?
13. Перечислите Amplification атаки с коэффициентами их усиления.
14. Опишите теоретически возможность расследования DoS/DDoS атак, установление личности атакующего, определение источников атак.

* [Docs](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/index.html) »

* Лабораторная работа №11. Iptables, WEB APLICATION FIREWALL

# Лабораторная работа №11. Iptables, WEB APLICATION FIREWALL

## Основные теоретические сведения

**Цель:** Изучение межсетевых экранов. Приобретение навыков работы с Iptables и WAF.

### Межсетевой экран

Скорее всего, ранее вы уже сталкивались с таким понятием как межсетевой экран. В ядро Linux встроен свой межсетевой экран, называемый Netfilter. Управление им осуществляется с помощью утилиты Iptables.

Межсетевой экран, сетевой экран, файервол, брандмауэр — комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами. Основной задачей сетевого экрана является защита компьютерных сетей или отдельных узлов от несанкционированного доступа. Также сетевые экраны часто называют фильтрами, так как их основная задача — не пропускать (фильтровать) пакеты, не подходящие под критерии, определённые в конфигурации.

Рассмотрим принцип работы Netfilter. Когда сетевые пакеты попадают в сетевой интерфейс, они после ряда проверок ядром проходят последовательность так называемых цепочек. Пакет обязательно проходит через цепочку PREROUTING, после чего определяется, кому он, собственно, был адресован. Если пакет не адресован локальной системе (в нашем случае серверу), он попадает в цепочка FORWARD, а иначе — в цепочку INPUT, после прохождения которой отдается локальным демонам или процессам. После этого при необходимости формируется ответ, который направляется в цепочку OUTPUT. После цепочек OUTPUT или FORWARD пакет в очередной раз встречается с правилами маршрутизации и направляется в цепочку POSTROUTING. В результате прохождения пакетом цепочек фильтрации несколько раз, проверка его принадлежности определенным критериям осуществляется несколько раз. В соответствии с этими проверками к пакету применяется определенное действие:

* ACCEPT — пакет «принимается» и передается в следующую цепочку.
* DROP — удовлетворяющий условию пакет отбрасывается и не передается в другие таблицы или цепочки.
* REJECT — пакет отбрасывается, но при этом отправителю отправляется ICMP-сообщение, сообщающее об отказе.
* RETURN — пакет возвращается в предыдущую цепочку и продолжает её прохождение начиная со следующего правила
* SNAT — применить трансляцию источника в пакете. Используется только в цепочках POSTROUTING и OUTPUT таблицы nat.
* DNAT — применить трансляцию адреса назначения в пакете. Используется в цепочках PREROUTING и (очень редко) OUTPUT в таблице nat.

| *Основные команды Iptables* | | |
| --- | --- | --- |
| **Параметр** | **Описание** | **Пример** |
| –append (-A) | Позволяет добавить в указанную цепочку и таблицу заданное правило, помещаемое в КОНЕЦ списка | iptables -A FORWARD критерии -j действие |
| –delete (-D) | Позволяет удалить заданное номером или каким-либо правилом правило. В первом примере удаляются все правила с номерами 10,12 во всех цепочках, в таблицах filter. | iptables -D 10,12 iptables -t mangle -D PREROUTING критерии -j действие |
| –rename-chain (-E) | Изменить имя цепочки. | iptables -E OLD\_CHAIN NEW\_CHAIN |
| –flush (-F) | Очищает все правила текущей таблицы. Ко всем пакетам, относящимся к уже установленным соединениям, применяется терминальное действие ACCEPT — пропустить | iptables -F |
| –insert (-I) | Добавляет заданное правило в соответствии с номером. | iptables -I FORWARD 5 критерии -j действие |
| –list (-L) | Позволяет просматривать существующие правила (без явного указания таблицы - отображается таблица filter всех цепочек). | iptables -L |
| –policy (-P) | Позволяет устанавливать стандартную политику для заданной цепочки. | iptables -t mangle -P PREROUTING DROP |
| –replace (-R) | Заменяет заданное номером правило на заданное в критериях. | iptables -R POSROUTING 7 | критерии -j действие |
| –delete-chain (-X) | Удалить ВСЕ созданные вручную цепочки (оставить только стандартные INPUT, OUTPUT…) | iptables -X |
| –zero (-Z) | Обнуляет счетчики переданных данных в цепочке. | iptables -Z INPUT |
| –line-numbers | Указывать номера правил при выводе (может использоваться совместно с -L). | iptables -L –line-numbers |
| –help (-h) | Помощь | Iptables –help |
| -t таблица | Задает название таблицы, над которой необходимо совершить действие. В примере сбрасывается таблица nat во всех цепочках. | iptables -t nat -F |
| –verbose (-v) | Детальный вывод. | iptables -L -v |
|  | **Основные правила отбора пакетов** |  |
| –protocol(сокр. -p) | Определяет протокол транспортного уровня. Опции tcp, udp, icmp, all или любой другой протокол определенный в /etc/protocols | iptables -A INPUT -p tcp |
| –source(-s, –src) | IP адрес источника пакета. Может быть определен несколькими путями:Одиночный хост: host.domain.tld, или IP адрес: 10.10.10.3 Пул-адресов (подсеть): 10.10.10.3/24 или 10.10.10.3/255.255.255.0 | iptables -A INPUT -s 10.10.10.3 |
| –destination(-d) | IP адрес назначения пакета. Может быть определен несколькими путями (см. –source). | iptables -A INPUT –destination 192.168.1.0/24 |
| –in-interface (-i) | Определяет интерфейс, на который прибыл пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках INPUT,FORWARD и PREROUTING. Возможно использование знака +, тогда подразумевается использование всех интерфейсов, начинающихся на имя+ (например eth+ - все интерфейсы eth). | iptables -t nat -A PREROUTING –in-interface eth0 |
| –out-interface(-o) | Определяет интерфейс, с которого уйдет пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках OUTPUT, FORWARD и POSTROUTING. Возможно использование знака +. | iptables -t nat -A POSTROUTING –in-interface eth1 |
|  | **Неявные (необщие) параметры** |  |
| -p proto -h | Вывод справки по неявным параметрам протокола proto. | iptables -p icmp -h |
| –source-port(–sport) | Порт источник, возможно только для протоколов –protocol tcp, или –protocol udp | iptables -A INPUT –protocol tcp –source-port 25 |
| –destination-port(–dport) | Порт назначения, возможно только для протоколов –protocol tcp, или –protemocol udp | iptables -A INPUT –protocol udp –destination-port 67 |
|  | **Явные параметры** |  |
| -m state –state (устарел) он же -m conntrack –ctstate | Состояние соединения. Доступные опции: NEW (Все пакеты устанавливающие новое соединение) ESTABLISHED (Все пакеты, принадлежащие установленному соединению) RELATED (Пакеты, не принадлежащие установленному соединению, но связанные с ним. Например - FTP в активном режиме использует разные соединения для передачи данных. Эти соединения связаны.) INVALID (Пакеты, которые не могут быть по тем или иным причинам идентифицированы). | iptables -A INPUT -m state –state NEW, ESTABLISHED iptables -A INPUT -m conntrack –ctstate NEW, ESTABLISHED |
| -m mac –mac-source | Задает MAC адрес сетевого узла, передавшего пакет. MAC адрес должен указываться в форме XX:XX:XX:XX:XX:XX. | -m mac –mac-source 00:00:00:00:00:0 |
|  | **Дополнительные параметры** |  |
|  | DNAT (Destination Network Address Translation) |  |
| –to-destination | Указывает, какой IP адрес должен быть подставлен в качестве адреса места назначения. В примере во всех пакетах протокола tcp, пришедших на адрес 1.2.3.4, данный адрес будет заменен на 4.3.2.1. | iptables -t nat -A PREROUTING -p tcp -d 1.2.3.4 -j DNAT –to-destination 4.3.2.1 |
|  | **LOG** |  |
| –log-level | Используется для задания уровня журналирования (log level). В примере установлен максимальный уровень логирования для всех tcp пакетов в таблице filter цепочки FORWARD. | iptables -A FORWARD -p tcp -j LOG –log-level debug |
| –log-prefix | Задает текст (префикс), которым будут предваряться все сообщения iptables. Префикс может содержать до 29 символов, включая и пробелы. В примере отправляются в syslog все tcp пакеты в таблице filter цепочки INPUT с префиксом INRUT-filter. | iptables -A INPUT -p tcp -j LOG –log-prefix INRUT-filter |
| –log-ip-options | Позволяет заносить в системный журнал различные сведения из заголовка IP пакета. | iptables -A FORWARD -p tcp -j LOG –log-ipoptions |

в скобках – сокращенный вариант записи

Основные цепочки межсетевого экрана Netfilter:

* PREROUTING — изначальная обработка входящих пакетов
* INPUT — для входящих пакетов, адресованных непосредственно локальному компьютеру
* FORWARD — для маршрутизируемых пакетов
* OUTPUT — для пакетов, исходящих с локального компьютера
* POSTROUTING — для окончательной обработки исходящих пакетов

Таблицы межсетевого экрана Netfilter:

* raw - используется для маркировки пакетов, которые не должны обрабатываться системой определения состояний. Содержится в цепочках PREROUTING и OUTPUT.
* mangle — содержит правила модификации IP-пакетов.
* nat - предназначена для подмены адреса отправителя или получателя. Данную таблицу проходят только первые пакеты из потока - трансляция адресов или маскировка (подмена адреса отправителя или получателя) применяются ко всем последующим пакетам в потоке автоматически. Поддерживает действия DNAT, SNAT, MASQUERADE, REDIRECT. Содержится в цепочках PREROUTING, OUTPUT, и POSTROUTING.
* filter — основная таблица, используется по умолчанию если название таблицы не указано. Используется для фильтрации пакетов. Содержится в цепочках INPUT, FORWARD, и OUTPUT.

Пример создания правила для межсетевого экрана

Рассмотрим две цепочки, задающие два основных правила Iptables — PREROUTING и FORWARD.

* iptables -t nat -A PREROUTING -i eth0 -j DNAT —to-destination 192.168.57.102
* iptables -A FORWARD -d 192.168.57.102 -j ACCEPT

Первая из них определяет первоначальную обработку всех пакетов, приходящих на адаптер eth0:

* -t определяет подключаемую таблицу, в данном случае — nat — для подмены адреса отправителя или получателя
* -А — выбор цепочки
* -i — входящий интерфейс
* -j — действие с пакетами, удовлетворяющими условию — в данном случае DNAT — подмена адреса получателя
* –to-destination — выбор адреса, на который перенаправляются пакеты
* Вторая определяет проброс пакетов через сервер:
* -A — выбор цепочки
* -d — выбор адресата
* -j — выбор действия

### Web Application Firewall

WAF (Web Application Firewall) - это межсетевые экраны, работающие на прикладном уровне и осуществляющие фильтрацию трафика Web-приложений. Эти средства не требуют изменений в исходном коде Web-приложения и, как правило, защищают Web-сервисы гораздо лучше обычных межсетевых экранов и средств обнаружения вторжений.

Основные преимущества:

* Анализ поведения пользователя в используемом приложении;
* Позволяет осуществлять мониторинг HTTP трафика и проводить анализ событий в реальном режиме времени;
* Предотвращение вредоносных запросов;
* Распознавание большинства опасных угроз;
* Дополнение сетевых средств безопасности;
* Просматривать детальные отчеты об атаках и попытках взлома.

## Задания к лабораторной работе

### Часть 1

* Установите web-сервер <sudo apt-get install apache2>
* Просмотрите список текущих правил iptables таблицы filter

sudo iptables -L

* Вы увидите, что список содержит три цепочки по умолчанию (INPUT, OUTPUT и FORWARD), в каждой из которых установлена политика по умолчанию (на данный момент это ACCEPT).
* С поможью команды <sudo iptables -S> данный список можно просмотреть в другом формате, который отражает команды, необходимые для активации правил и политик.
* Чтобы сбросить текущие правила (если таковые есть), наберите:

sudo iptables -F

* Цепочка INPUT отвечает за входящий траффик.
* Чтобы внести локальный интерфейс выполните:

sudo iptables -A INPUT -i lo -j ACCEPT

* Чтобы заблокировать весь исходящий трафик, кроме портов для SSH и веб-сервера, нужно сначала разрешить подключения к этим портам. В цепочку ACCEPT добавьте два порта (порт SSH 22 и порт http 80), что разрешит трафик на эти порты.

sudo iptables -A INPUT -p tcp -m tcp –dport 22 -j ACCEPT

sudo iptables -A INPUT -p tcp -m tcp –dport 80 -j ACCEPT

* В данной работе мы не используем SSH. Так что удалим ненужное правило. Для этого:

sudo iptables -D INPUT -p tcp -m tcp –dport 22 -j ACCEPT

* Нужно добавить еще одно правило, которое позволит устанавливать исходящие соединения (т.е. использовать ping или запускать обновления программного обеспечения):

sudo iptables -I INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT

* Создав все эти правила, можно заблокировать все остальное и разрешить все исходящие соединения.

sudo iptables -P OUTPUT ACCEPT

sudo iptables -P INPUT DROP

* Просмотрите список правил

sudo iptables -L

* Добавим еще несколько правил для блокировки наиболее распространенных атак. Для начала нужно заблокировать нулевые пакеты <sudo iptables -A INPUT -p tcp –tcp-flags ALL NONE -j DROP>.
* Следующее правило отражает атаки syn-flood <sudo iptables -A INPUT -p tcp ! –syn -m state –state NEW -j DROP>. Теперь фаервол не будет принимать входящих пакетов с tcp-флагами. Нулевые пакеты, по сути, разведывательные. они используются, чтобы выяснить настройки сервера и определить его слабые места.
* Далее нужно защитить сервер от разведывательных пакетов XMAS <sudo iptables -A INPUT -p tcp –tcp-flags ALL ALL -j DROP>. Теперь сервер защищен от некоторых общих атак, которые ищут его уязвимости.
* Со второй виртуальной машиины, на которую установите nmap, проведите XMAS сканирование <sudo nmap -sX>.
* По умолчанию все не сохраненные правила действуют до следующей перезагрузки сервера; сразу же после перезагрузки не сохраненные правила будут потеряны. Самый простой способ загрузить пакет iptables-persistent <sudo apt-get install iptables-persistent>. Во время инсталляции пакет уточнит, нужно ли сохранить текущие правила для дальнейшей автоматической загрузки, если текущие правила были протестированы и соответствуют всем требованиям, их можно сохранить.

### Часть 2

* Для начала понадобится LAMP( Apache, MySQL, PHP). В лабораторной работе № 8, уже было показано, как установить его, используя tasksel.
* Установите mod\_security <sudo apt-get install libapache2-mod-security2>
* Выполните команду <sudo apachectl -M | grep –color security2>. Если на экране появился модуль по имени security2\_module (shared), значит, все прошло успешно.
* В каталоге логов Apache можно найти новый лог-файл для mod\_security. /var/log/apache2/modsec\_audit.log
* Установка ModSecurity включает в себя конфигурационный файл, который нужно переименовать: <sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf>.
* Затем перезапустите Apache <sudo service apache2 reload>.
* Стандартный конфигурационный файл настроен на DetectionOnly, то есть, фаервол только отслеживает логи, при этом ничего не блокируя. Чтобы изменить это поведение, отредактируйте файл modsecurity.conf: <sudo nano /etc/modsecurity/modsecurity.conf>
* Найдите в файле строку: «SecRuleEngine DetectionOnly». И измените ее так: «SecRuleEngine On».
* Найдите «SecResponseBodyAccess On» и замените на «SecResponseBodyAccess Off». Эта директива отвечает за буферизацию тела ответа; ее рекомендуется включать, только если требуется обнаружение и предохранение от утечки данных. Включенная директива (SecResponseBodyAccess On) не только будет использовать больше ресурсов сервера, но и увеличит размер лог-файла, следовательно, ее желательно отключить.
* По умолчанию mod\_security поставляется с базовым набором правил CRS (Core Rule Set), которые находятся в /usr/share/modsecurity-crs/
* Чтобы подгрузить эти готовые правила, нужно, чтобы веб-сервер Apache читал указанные выше каталоги. Для этого отредактируйте файл mod-security.conf:

nano /etc/apache2/mods-enabled/mod-security.conf

* Между <IfModule security2\_module> </IfModule> внесите следующие параметры:

Include "/usr/share/modsecurity-crs/\*.conf"

Include "/usr/share/modsecurity-crs/activated\_rules/\*.conf"

* Директория activated\_rules аналогична директории Apache mods-enabled. Правила доступны в каталогах: /usr/share/modsecurity-crs/base\_rules ; /usr/share/modsecurity-crs/optional\_rules ; /usr/share/modsecurity-crs/experimental\_rules
* Чтобы активировать правила, нужно создавать символические ссылки в каталоге activated\_rules. <cd /usr/share/modsecurity-crs/activated\_rules/>
* Добавьте несколько правил, например <sudo ln -s /usr/share/modsecurity-crs/base\_rules/modsecurity\_crs\_30\_http\_policy.conf> ; <sudo ln -s /usr/share/modsecurity-crs/base\_rules/modsecurity\_crs\_49\_generic\_attacks.conf>
* Чтобы новые правила вступили в исполнение, нужно перезапустить Apache <sudo service apache2 reload>

## Вопросы к лабораторной работе

1. Что такое межсетевой экран?
2. Для чего используется межсетвой экран?
3. Принцип работы Netfilter.
4. Таблицы межсетевого экрана Netfilter. Для чего они используются?
5. Что такое правила межсетевого экрана?
6. Как создавать правила для межсетевого экрана утилитой Iptables?
7. Как сохранить правила для последующей автозагрузки?
8. Что такое Web Application Firewall?
9. Как настроить правила в WAF mod\_security?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

* [Docs](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/index.html) »

* Лабораторная работа №12. Sandbox

**Лабораторная работа №12. Sandbox**

**Основные теоретические сведения**

**Цель:** Получение теоритических и практических навыков работы с песочницами и файловыми антивирусами.

Вредоносная программа (англ. malware, malicious software — «злонамеренное программное обеспечение») — любое программное обеспечение, предназначенное для получения несанкционированного доступа к вычислительным ресурсам самой ЭВМ или к информации, хранимой на ЭВМ, с целью несанкционированного использования ресурсов ЭВМ или причинения вреда (нанесения ущерба) владельцу информации, и/или владельцу ЭВМ, и/или владельцу сети ЭВМ, путем копирования, искажения, удаления или подмены информации. Многие антивирусы считают крэки, кейгены и прочие программы для взлома вредоносными программами, или потенциально опасными.

Песочница (англ. Sandbox) - это ограниченная среда в вашей системе для исполнения гостевых программ без доступа к главной операционной системе. Это закрытое для доступа извне виртуальное пространство, в котором можно работать с программным обеспечением без изменения системных файлов.

Песочница (от англ. Sandbox, схожие понятия — англ. honeypot, англ. fishbowl) - в компьютерной безопасности, механизм для безопасного исполнения программ. Песочницы часто используют для запуска не протестированного кода, непроверенного кода из неизвестных источников, а также для запуска и обнаружения вирусов.

Песочница — в компьютерной безопасности специально выделенная среда для безопасного исполнения компьютерных программ.

Песочница обычно представляет собой жёстко контролируемый набор ресурсов для исполнения гостевой программы — например, место на диске или в памяти. Доступ к сети, возможность сообщаться с главной операционной системой или считывать информацию с устройств ввода обычно либо частично эмулируют, либо сильно ограничивают. Песочницы представляют собой пример виртуализации. Повышенная безопасность исполнения кода в песочнице зачастую связана с большой нагрузкой на систему — именно поэтому некоторые виды песочниц используют только для не отлаженного или подозрительного кода.

Cuckoo Sandbox — система для автоматического исследования вредоносного ПО, эксплоитов, вредоносных скриптов, документов, архивов и ссылок. Система способна проверять документы pdf, doc, xls, rtf, скрипты Python, JS, DLL библиотеки, бинарники, jar и многое другое.

Файловый Антивирус — компонент модуля Защита компьютера, контролирующий файловую систему компьютера. Он запускается при старте операционной системы, постоянно находится в оперативной памяти компьютера и проверяет все открываемые, сохраняемые и запускаемые файлы. Каждый файл, к которому вы обратитесь, будет перехвачен Файловым Антивирусом и проверен на присутствие известных вирусов.

Clam AntiVirus — пакет антивирусного ПО, работающий во многих операционных системах, включая Unix-подобные ОС, OpenVMS, Microsoft Windows и Apple Mac OS X.

Главная цель Clam AntiVirus — интеграция с серверами электронной почты для проверки файлов, прикреплённых к сообщениям. В пакет входит масштабируемый многопоточный демон clamd, управляемый из командной строки сканер clamscan, а также модуль обновления сигнатур по Интернету freshclam.

**Задания к лабораторной работе**

**Установите Cukoo Sandbox.**

**Задания к лабораторной работе**

Если будут проблемы с установкой, воспользуйтесь документацией Cukoo sandbox <http://docs.cuckoosandbox.org/en/latest/installation/>

\* Установка зависимостей

cd /tmp

apt-get update

apt-get install git automake mongodb mingw32 dkms unzip wget python python-sqlalchemy python-bson python-pip python-dpkt python-jinja2 python-magic python-mysqldb python-gridfs python-libvirt python-bottle python-pefile python-chardet -y

apt-get install python-dev libxml2-dev libxslt1-dev libevent-dev libpcre3 libpcre3-dev zlib1g-dev libtool libpcre++-dev –y

apt-get install mariadb-server -y

pip install lxml

pip install cybox==2.0.1.4

pip install maec==4.0.1.0

pip install django

pip install py3compat

pip install pymongo

apt-get install ssdeep python-pyrex subversion libfuzzy-dev -y

svn checkout http://pyssdeep.googlecode.com/svn/trunk/ pyssdeep

cd pyssdeep

python setup.py build

python setup.py install

pip install pydeep

cd /tmp

wget https://github.com/plusvic/yara/archive/v2.1.0.tar.gz

tar xzf v2.1.0.tar.gz

cd yara-2.1.0

chmod +x build.sh

./build.sh

make install

cd yara-python

python setup.py build

python setup.py install

cd /tmp

wget http://distorm.googlecode.com/files/distorm3.zip

unzip distorm3.zip

cd distorm3

python setup.py build

python setup.py install

add-apt-repository ppa:pi-rho/security

apt-get update

apt-get install volatility

**Установка и настройка Virtualbox**:

wget -q http://download.virtualbox.org/virtualbox/debian/oracle\_vbox.asc -O- | sudo apt-key add –

sh -c 'echo "deb http://download.virtualbox.org/virtualbox/debian trusty contrib" >> /etc/apt/sources.list.d/virtualbox.list'

apt-get update

apt-get install virtualbox-4.3

cd /tmp

VBOX\_LATEST\_VERSION=$(curl http://download.virtualbox.org/virtualbox/LATEST.TXT)

wget http://download.virtualbox.org/virtualbox/${VBOX\_LATEST\_VERSION}/Oracle\_VM\_VirtualBox\_Extension\_Pack-{VBOX\_LATEST\_VERSION}.vbox-extpack

vboxmanage extpack install /tmp/Oracle\_VM\_VirtualBox\_Extension\_Pack-${VBOX\_LATEST\_VERSION}.vbox-extpack

cd /opt

wget http://dlc.sun.com.edgesuite.net/virtualbox/${VBOX\_LATEST\_VERSION}/VBoxGuestAdditions\_${VBOX\_LATEST\_VERSION}.iso

**Установка Cuckoo Sandbox**:

useradd cuckoo

usermod -a -G vboxusers cuckoo

id cuckoo

cd /opt

wget http://downloads.cuckoosandbox.org/1.1/cuckoo\_1.1.tar.gz

tar xzf cuckoo\_1.1.tar.gz

**Настройка Cuckoo Sandbox**:

cd /opt/cuckoo

./utils/community.py --signatures --force

**Настройка БД**:

mysql -u root -p

> create database cuckoo;

> grant all privileges on cuckoo.\* to cuckoo**@localhost** identified by 'cuck00pass' ;

> flush privileges;

> quit;

**Настраиваем cuckoo:**:

\* Файл /opt/cuckoo/conf/cuckoo.conf

Включаем запись дампа памяти:

memory\_dump = on

Настраиваем подключение к бд:

connection = mysql://cuckoo:cuck00pass\**@localhost**/cuckoo

Увеличиваем временные лимиты:

default = 240

critical = 1200

vm\_state = 600

\* Файл /opt/cuckoo/conf/memory.conf

Отключаем сохранение дампов памяти:

delete\_memdump = yes

\* Файл /opt/cuckoo/conf/processing.conf

Включаем анализ оперативной памяти:

memory = yes

\* nano /opt/cuckoo/conf/virtualbox.conf

Меняем режим работы Virtualbox:

mode = headless

Меняем названия виртуальной машины с cuckoo1 на WindowsXP:

machines = WindowsXP

[WindowsXP]

label = WindowsXP

\* Файл /opt/cuckoo/conf/reporting.conf

Включим импорт отчётов в MongoDB для работы веб интерфейса

[mongodb]

enabled = yes

**Задания к лабораторной работе**

* На этом настройка Cuckoo закончена, теперь приступим к Virtualbox и гостевой ОС.

**Загрузка виртуальной ОС с сайта:**

wget https://az412801.vo.msecnd.net/vhd/VMBuild\_20131127/VirtualBox/IE6\_WinXP/Linux/IE6.WinXP.For.LinuxVirtualBox.sfx

chmod +x IE6.WinXP.For.LinuxVirtualBox.sfx

./IE6.WinXP.For.LinuxVirtualBox.sfx

vboxmanage **import** **IE6**\ -\ WinXP.ova --vsys 0 --unit 10 --disk=/root/VirtualBox\ VMs/WindowsXP/WindowsXP.vmdk --memory 1024 --vmname WindowsXP

**Настраиваем сеть**:

vboxmanage hostonlyif create

vboxmanage modifyvm "WindowsXP" --nic1 hostonly --hostonlyadapter1 vboxnet0 --nicpromisc1 allow-all --hwvirtex off --vtxvpid off

**Настраиваем общие папки**:

mkdir -p /opt/cuckoo/shares/setup

mkdir -p /opt/cuckoo/shares/WindowsXP

vboxmanage sharedfolder add "WindowsXP" --name "WindowsXP" --hostpath /opt/cuckoo/shares/WindowsXP --automount

vboxmanage sharedfolder add "WindowsXP" --name setup --hostpath /opt/cuckoo/shares/setup --automount --readonly

vboxmanage modifyvm "WindowsXP" --nictrace1 on --nictracefile1 /opt/cuckoo/shares/WindowsXP/dump.pcap

**Включаем доступ по RDP, порт можете указать любой**:

vboxmanage modifyvm "WindowsXP" --vrdeport 5000 --vrde on

**Задания к лабораторной работе**

**На этом конфигурация виртуальных контейнеров полностью закончена, осталось настроить iptables, tcpdump.**:

iptables -A FORWARD -o eth0 -i vboxnet0 -s 192.168.56.0/24 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

iptables -A POSTROUTING -t nat -j MASQUERADE

sysctl -w net.ipv4.ip\_forward=1

setcap cap\_net\_raw,cap\_net\_admin=eip /usr/sbin/tcpdump

getcap /usr/sbin/tcpdump

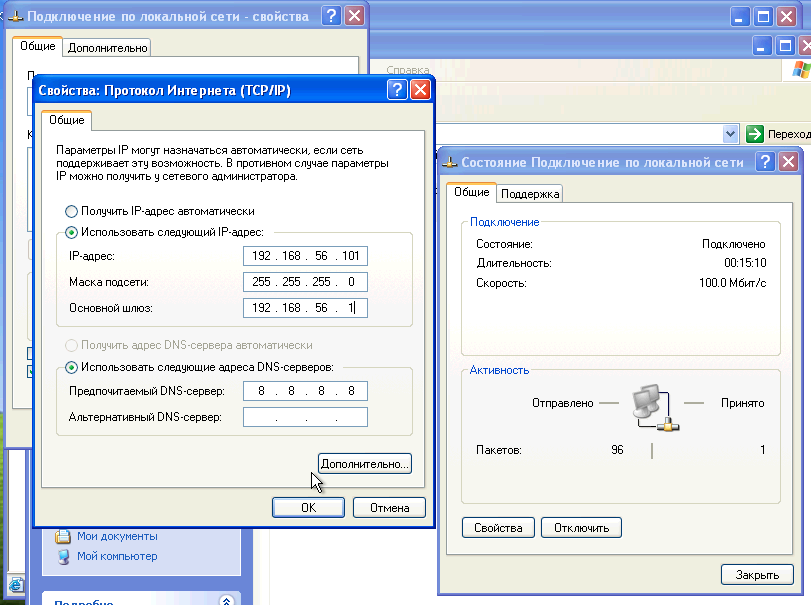
ifconfig vboxnet0 192.168.56.1

* После установки и настройки Windows переходим непосредственно на саму гостевую ОС.

**Предупреждение**

(Если возникли проблемы с установкой виртуальной машины, можете посмотреть, как это сделано в этой статье <http://gwallgofi.com/cuckoo-sandbox-part-2-installing/> )

* Следующим образом настроим подключение к сети (dns можете указать любой):

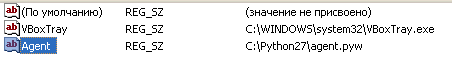


* Установим VboxTools с диска, который подключен к системе.
* Устанавливаем Pyton 2.7: <http://python.org/download/>
* Устанавливаем <http://www.activestate.com/activepython>
* Устанавливаем PIL Python модуль, для создания скриншотов: <http://www.pythonware.com/products/pil/>
* Отключаем автоматическое обновление Windows.
* Отключаем брандмауэер.
* Копируем агент из сетевой папки setup в папку C:Python27, Ставим агент на автозагрузку, для этого добавляем в ветку реестра(пуск->выполнить->regedit) HKLMSOFTWAREMicrosoftWindowsCurrentVersionRun строковый параметр

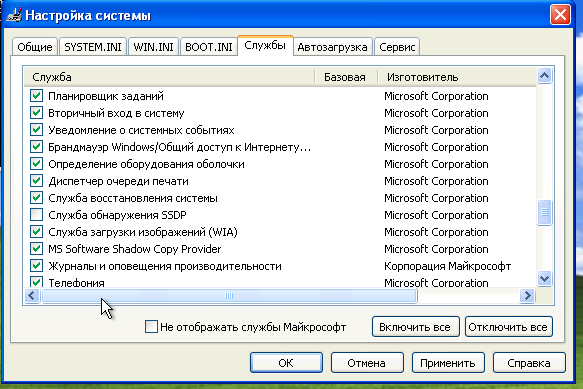
Имя:“Agent“

Тип:“REG\_SZ“

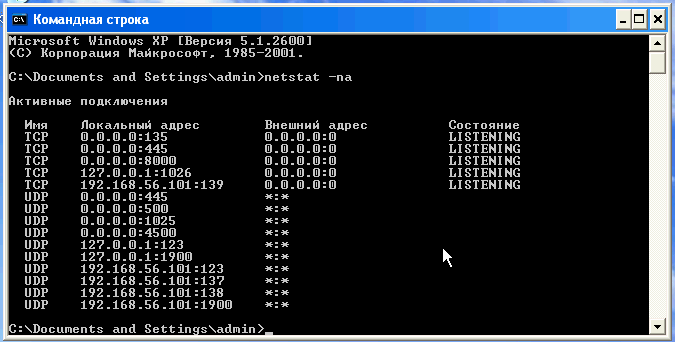
Содержание:«C:Python27agent.pyw»



* Включаем IE, в настройках ставим домашней страницей пустую вкладку, по желанию в свойствах обозревателя выключаем все защитные механизмы.
* Отключаем SSDP: пуск->выполнить->msconfig и в разделе службы отключаем «Служба обнаружения SSDP», чтобы в отчётах не фигурировали сетевые запросы этой службы.



* Перезагружаемся и в появившемся при загрузке окне выбираем «При перезагрузке не выводить это сообщение» и ОК.
* После перезагрузки гостевой ОС, пуск->выполнить->cmd и в консоли набираем netstat –na и смотрим есть ли агент на 8000-ом порту



* По желанию устанавливаем различное уязвимое ПО старых версий (браузеры, Flash player, Java, Acrobat Reader…)
* На этом установка гостевой ОС закончена.Делаем снапшот (не выключая гостевую ОС) <vboxmanage snapshot «WindowsXP» take «WindowsXPSnap01» –pause>
* И выключаем: <vboxmanage controlvm «WindowsXP» poweroff>
* Запуск Cukoo Sandbox <python cuckoo.py>
* Загрузите вредоносное ПО, для проверки работы Cukoo Sandbox. Здесь есть небольшой список ресурсов с образцами вредоносного ПО <https://zeltser.com/malware-sample-sources/> . Например, Можете загрузить с <http://malshare.com/> или с <https://malwr.com/> .
* Чтобы отправить файл в Cukoo на анализ используйте команду submit: <python submit.py /path/to/binary>
* Запустите web-интерфейс cukoo и просмотрите результаты <python web.py>

**Вопросы к лабораторной работе**

1. Что такое песочница?
2. Принцип работы песочниц.
3. Где используют песочницы?
4. Преимущества и недостатки песочниц.
5. Альтернативы использованию песочниц.
6. Что такое эмуляция?
7. Что такое эвристический анализ? В чем отличия от сигнатурного анализа?
8. Для чего нужен файловый антивирус?
9. Что такое вредоносное ПО?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

* [Docs](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/index.html) »

* Лабораторная работа №13. Антиспам (ASSP)

**Лабораторная работа №13. Антиспам (ASSP)**

**Основные теоретические сведения**

**Цель:** Изучить работу почтового сервера, получить практические навыки работы по защите от спама.

Почтовый сервер, сервер электронной почты, мейл-сервер — в системе пересылки электронной почты так обычно называют агент пересылки сообщений (англ. mail transfer agent, MTA). Это компьютерная программа, которая передаёт сообщения от одного компьютера к другому. Обычно почтовый сервер работает «за кулисами», а пользователи имеют дело с другой программой — клиентом электронной почты (англ. mail user agent, MUA).

Спам (англ. spam) — рассылка коммерческой и иной рекламы или подобных коммерческих видов сообщений лицам, не выражавшим желания их получать. Также, название распространяемых материалов. Распространителей спама называют спамерами.

В общепринятом значении термин «спам» в русском языке впервые стал употребляться применительно к рассылке электронных писем. Незапрошенные сообщения в системах мгновенного обмена сообщениями (например, ICQ) носят название SPIM (англ.)русск. (англ. Spam over IM).

Доля спама в мировом почтовом трафике составляет от 60% (2006) до 80% (2011). Самый большой поток спама распространяется через электронную почту. В настоящее время доля вирусов и спама в общем трафике электронной почты составляет по разным оценкам от 70 до 95 процентов

Anti-Spam SMTP Proxy (ASSP) - это ПО с открытым исходным текстом, платформонезависимый SMTP прокси сервер, в котором реализованы белые списки и фильтрация на основании теоремы Байеса.

Антиспамовый сервер собирает письма со спамом и нормальную почту, потом на основании вероятности встречи слов из анализируемого письма в каждой из коллекций (спам или не-спам) сервер делает вывод о том, является ли письмо спамом или нет.

Другие возможности ASSP:

* настройка через веб-интерфейс в браузере,
* автоматическое ведение белого списка,
* наличие необрабатываемых адресов и доменов,
* адреса для сбора спама,
* поддержка дополнительных регулярных выражений для идентификации спама и не-спама,
* обнаружение спама, кодированого MIME,
* автоматическое ведение баз спама и нормальной почты,
* защита от пересылки почты третьми лицами,
* простейший контроль вирусов,
* наличие почтового интерфейса для управления и пополнения коллекций спама и нормальной почты,
* проверка отправителя по RBL и SPF.

**Задания к лабораторной работе**

* На две виртуальные машины установить почтовый сервер Zimbra Collaboration Server. ([На странице загрузки](https://www.zimbra.com/downloads/zimbra-collaboration-open-source), можете посмотреть, какие системы поддерживаются)
* Установите зависимости <sudo apt-get install libgmp10 libperl5.18 unzip pax sysstat sqlite3 dnsmasq wget libaio1>
* Откройте <nano /etc/hostname>. Измените имя хоста на «mail.sit.local»
* Узнайте ip-адрес <ifconfig>.
* Откройте <sudo nano /etc/hosts>. Добавьте строку «192.168.1.113 mail.sit.local mail» , где 192.168.1.113 ваш ip-адрес.
* Откройте <sudo nano /etc/dnsmasq.conf> И добавьте туда:

server=192.168.1.113

domain=sit.local

mx-host=sit.local, mail.sit.local, 5

mx-host=mail.sit.local, mail.sit.local, 5

listen-address=127.0.0.1

* Перезагрузите виртуальную машину <sudo reboot>
* Скачайте Zimbra для своей системы <https://www.zimbra.com/downloads/zimbra-collaboration-open-source>
* Извлеките архив и перейдите в паку, которую извлекли из архива.
* Запустите установку. <sudo ./install.sh>
* Согласитесь с лицензионным соглашением. Y.
* Устанавливайте все пакеты ( Выбираете Y), кроме zimbra-dnscache (Выбирайте N, так как уже используем dnsmask).
* Когда появится меню введите **6** и нажмите Enter.
* Введите **4** и нажмите Enter, введите пароль администратора, минимум 6 символов.
* Введите **r** и нажмите Enter, для возврата в главное меню, затем введите **a** и нажмите Enter, чтобы принять изменения.
* На запросы «Save configuration data to a file» и «The system will be modified - continue?» введите y.
* Ждите пока Zimbra не установится
* Чтобы проверить работу, можете ввести <su - zimbra> <zmcontrol status>.
* Подключитесь к Zimbra в браузере <https://192.168.1.113/> или к странице администратора <https://192.168.1.113:7071/>
* На одну из виртуальных машин, для защиты от спама, установите ASSP (Anti-Spam SMTP Proxy Server). <http://sourceforge.net/projects/assp/> . Скачайте и разархивируйте ASSP.

sudo apt-get install build-essential pmtools libterm-readline-perl-perl libterm-readline-gnu-perl libyaml-perl libtext-glob-perl libnumber-compare-perl libio-compress-perl libemail-mime-perl libemail-send-perl libemail-valid-perl libfile-readbackwards-perl libwww-perl libmime-types-perl libmail-dkim-perl libmail-spf-perl libmail-srs-perl libnet-cidr-lite-perl libnet-dns-perl libnet-ldap-perl libnet-smtp-server-perl libthreads-perl libthread-queue-any-perl libtie-dbi-perl libschedule-cron-perl libio-socket-ssl-perl libdbd-anydata-perl libdbd-csv-perl libdbd-ldap-perl libdbd-mock-perl libdbd-odbc-perl libdbd-mysql-perl libfile-find-rule-perl libfile-slurp-perl libfile-which-perl libfile-chmod-perl liblinux-usermod-perl libcrypt-rc4-perl libtext-pdf-perl libsmart-comments-perl libcam-pdf-perl libpdf-api2-perl imagemagick perlmagick poppler-utils xpdf libauthen-sasl-perl libnet-snmp-perl libsnmp-base libsnmp-dev libsnmp-perl snmp libsnmp-\*-perl libsnmpkit-dev libregexp-optimizer-perl libnet-smtp-tls-perl liblingua-stem-snowball-perl liblingua-identify-perl unzip libberkeleydb-perl

sudo apt-get install tesseract-ocr tesseract-ocr-\*

sudo apt-get install libmodule-signature-perl libtest-pod-perl libtest-pod-coverage-perl libarchive-zip-perl

sudo apt-get install libssl-dev

**sudo cpan**

[…]

Would you like to configure as much as possible automatically? [yes]

[…]

Would you like me to automatically choose some CPAN mirror

sites for you? (This means connecting to the Internet) [yes]

cpan> install Test::Perl::Critic

cpan> install CPAN

cpan> reload cpan

cpan> force install Mail::SPF::Query

cpan> install Net::IP::Match::Regexp Net::SenderBase Net::Syslog Thread::State Sys::MemInfo Crypt::CBC Crypt::OpenSSL::AES DBD::Log DBD::MVS\_FTPSQL DBD::Multiplex DBD::Ovrimos DBD::PgPP DBD::Sprite DBD::Template DBD::mysqlPP DBIx::AnyDBD LEOCHARRE::DEBUG LEOCHARRE::CLI PDF::Burst Image::OCR::Tesseract PDF::GetImages PDF::OCR PDF::OCR2 Mail::DKIM::Verifier Convert::Scalar Unicode::GCString Sys::CpuAffinity

cpan> exit

sudo apt-get install clamav clamav-daemon

sudo freshclam

sudo /etc/init.d/clamav-daemon start

sudo apt-get install libfile-scan-perl

sudo cpan

cpan[1]> test File::Scan::ClamAV

cpan[1]> look File::Scan::ClamAV

/.cpan/build/File-Scan-ClamAV-1.91-Ik8fWD# make install

/.cpan/build/File-Scan-ClamAV-1.91-Ik8fWD# exit

cpan[1]> exit

**Скачайте ASSP и настройте его**:

http://sourceforge.net/projects/assp/

unzip ASSP\_2.3.3\_13137\_install.zip

sudo mkdir -p /usr/share/assp

sudo mv -f assp/\* /usr/share/assp

rm -rf assp ASSP\_2.3.3\_13137\_install.zip Install.txt MacOSX-launchd.txt quickstart.txt Win32-quickstart-guide.txt

sudo chown -R nobody:nogroup /usr/share/assp

sudo chmod 755 /usr/share/assp/assp.pl

sudo nano /etc/init.d/assp

Contents:

===

#!/bin/sh -e

# Start or stop ASSP

#

# original version by Ivo Schaap <ivo@lineau.nl> had issues on Debian4. Modified by atramos.

#

### BEGIN INIT INFO

# Provides: ASSP (Anti-Spam SMTP Proxy)

# Required-Start: $syslog, $local\_fs

# Required-Stop: $syslog, $local\_fs

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: Start ASSP

# Description: Enable service provided by daemon.

### END INIT INFO

PATH=/bin:/usr/bin:/sbin:/usr/sbin

case “$1” in

start)

echo -n “Starting the Anti-Spam SMTP Proxy”

cd /usr/share/assp

perl assp.pl 2>&1 > /dev/null &

;;

stop)

echo -n “Stopping the Anti-Spam SMTP Proxy”

kill -9 ps ax | grep "perl assp.pl" | grep -v grep | awk '{ print $1 }'

;;

restart)

$0 stop || true

$0 start

;;

``\*)``

echo “Usage: /etc/init.d/assp {start|stop|restart}”

exit 1

;;

esac

exit 0

===

* sudo chmod 755 /etc/init.d/assp
* sudo /usr/share/assp/assp.pl
* Нажмите Ctrl+C
* sudo update-rc.d assp defaults
* sudo /etc/init.d/assp start
* Переходим на [http://antispam\_host:55555](http://antispam_host:55555/)
* login:root
* pass:nospam4me
* Через машину без ASSP, отправляйте спам на машину с ASSP. Скриптов для генерации спама в интернете полно, но нужен такой, где можно использовать свой почтовый сервер (Например, можете использовать Social Engineer Toolkit). Посмотрите, как блокируется спам и как система обучается.

**Вопросы к лабораторной работе**

1. Что такое почтовый сервер?
2. Принцип работы почтового сервера.
3. Что такое спам?
4. Что такое ASSP?
5. Как работает ASSP?
6. Можно ли полностью защитить почтовый сервер от спама?
7. Какие еще угрозы почтовому серверу существуют?
8. Какие методы защиты почтового сервера существуют?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

* [Docs](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/index.html) »

* Лабораторная работа №14. NIPS/NIDS: Snort

**Лабораторная работа №14. NIPS/NIDS: Snort**

**Основные теоретические сведения**

**Цель:** Получить сведения о том, как осуществляется защита с помощью систем обнаружения и предотвращения вторжений. Научиться использовать SNORT.

Система обнаружения вторжений (IDS) — программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими в основном через Интернет.

Сетевая система обнаружения вторжений (англ. network intrusion detection system, NIDS) — система обнаружения вторжений, которая отслеживает такие виды вредоносной деятельности, как DoS атаки, сканирование портов или даже попытки проникновения в сеть.

В пассивной IDS при обнаружении нарушения безопасности, информация о нарушении записывается в лог приложения, а также сигналы опасности отправляются на консоль и/или администратору системы по определенному каналу связи. В активной системе, также известной как Система Предотвращения Вторжений (IPS — Intrusion Prevention system (англ.)), IDS ведет ответные действия на нарушение, сбрасывая соединение или перенастраивая межсетевой экран для блокирования трафика от злоумышленника. Ответные действия могут проводиться автоматически либо по команде оператора.

Обнаружение проникновения позволяет организациям защищать свои системы от угроз, которые связаны с возрастанием сетевой активности и важностью информационных систем. При понимании уровня и природы современных угроз сетевой безопасности, вопрос не в том, следует ли использовать системы обнаружения проникновений, а в том, какие возможности и особенности систем обнаружения проникновений следует использовать.

Snort — свободная сетевая система предотвращения вторжений (IPS) и обнаружения вторжений (IDS) с открытым исходным кодом, способная выполнять регистрацию пакетов и в реальном времени осуществлять анализ трафика в IP-сетях.

Выполняет протоколирование, анализ, поиск по содержимому, а также широко используется для активного блокирования или пассивного обнаружения целого ряда нападений и зондирований, таких как попытки атак на переполнение буфера, скрытое сканирование портов, атаки на веб-приложения, SMB-зондирование и попытки определения операционной системы. Программное обеспечение в основном используется для предотвращения проникновения, блокирования атак, если они имеют место.

Snort использует правила, написанные простым , но в то же время гибким и достаточно мощным языком. Существует ряд общих принципов написания, запомнить которые достаточно просто.

Большая часть правил Snort умещается в 1 строку. Это следствие того, что до версии 1.8 нельзя было использовать многострочные записи. В более поздних версиях правила можно растягивать на несколько строк, вставляя в конец строки символ “” (без кавычек).

Правила Snort состоят из двух частей: заголовка правила и параметров правила. Заголовок содержит описание действия, протокол передачи данных, IP-адреса, сетевые маски и порты источника и назначения. Параметры правила хранят предупреждающее сообщение, а также информацию о том, какую часть обнаруженного пакета нужно обработать в случае срабатывания правила.

**Задания к лабораторной работе**

* Узнайте свой ip адрес командой ifconfig
* Установите SNORT <sudo apt-get install snort>
* При установке будет нужно указать защищаемую сеть. ВВедите *.*.\*.0/24 (Где *.*.\* - первые три числа вашего ip-адреса, например эот будет 192.168.1.0/24, если вы используете VirtualBox и у вас в настройках сети стоит сетевой мост)
* Запустите SNORT <sudo service snort start>
* Настройка правил
* Перейдите в каталог /etc/snort/rules < cd /etc/snort/rules)
* Создайте файл с правилами <nano test.rules>

alert tcp any any -> any any (content:»<https://www.google.ru/>» ; msg:»Someone open Google website» ; sid: 12312313;)

* Перейдите в каталог /etc/snort <cd /etc/snort)
* Теперь нужно изменить содержимое конфигурационного файла Snort < sudo nano snort.conf>
* Найдите строчки с правилами (они начинаются с include $RULE\_PATH, это в части Step 7) и добавьте файл с нашими правилами

include $RULE\_PATH/test.tules

* В файле snort.conf так же укажите домашнюю сеть. В Step 1 измените строчку «ipvar HOME\_NET any» , на

ipvar HOME\_NET 192.168.1.0/24

* Запустите snort <sudo snort -A console -i eth0 -c snort.conf>
* Зайдите на <https://www.google.ru/> и проверьте в терминале, как работает правило.
* Теперь нам понадобиться еще одна виртуальная машина, на ней должен быть установлен nmap.
* Со второй ВМ используйте ping, посмотрите, как реагирует SNORT
* Используйте различные методы сканирования nmap( используйте -sS, -sT, -sN, -sU, -sX, -sF и посмотрите, как реагирует SNORT;
* В файл test.rules добавьте правило обнаружения сканирования nmap -sN (NULL Scan)

alert tcp any any -> any any (msg:»NULL Scan»; flags: 0; sid:322222;)

* Запустите snort <sudo snort -A console -i eth0 -c snort.conf>
* Со второй виртуальной машины произведите NULL сканирование <sudo nmap -sN>, проверьте, как работает правило.
* Можно загрузить обновленные правила SNORT, для этого:
* Зарегистрируйтесь на сайте <https://www.snort.org/> и скачайте последнюю версию правил
* Разархивируйте скачанный архив и скопируйте каталоги rules, so\_rules и preproc\_rules в /etc/snort :

sudo cp -R ./rules/ /etc/snort/

sudo cp -R ./so\_rules/ /etc/snort/

sudo cp -R ./preproc\_rules/ /etc/snort/

**Вопросы к лабораторной работе**

1. Что такое IDS?
2. Что такое сетевая система обнаружения вторжений?
3. Чем отличаются пассивные и активные IDS?
4. Что такое SNORT?
5. Какие задачи выполняет SNORT?
6. Как работают правила SNORT?
7. Как писать правила для SNORT?
8. Зачем писать собственные правила SNORT?
9. Зачем загружать обновление правил SNORT?
10. Как в SNORT создавать логи?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**

* [Docs](https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/index.html) »

* Лабораторная работа №15. SIEM

**Лабораторная работа №15. SIEM**

**Основные теоретические сведения**

**Цель:** Получение теоритических и практических навыков работы с SIEM

SIEM (Security information and event management) – объединение двух терминов, обозначающих область применения ПО: SIM - Security information management - управление информационной безопасностью и SEM - Security event management - управление событиями безопасности. Технология SIEM обеспечивает анализ в реальном времени событий (тревог) безопасности, исходящих от сетевых устройств и приложений. SIEM представлено приложениями, приборами или услугами, и используется также для журналирования данных и генерации отчетов в целях совместимости (с прочими бизнес-данными).

Перед системой SIEM ставятся следующие задачи.

* Агрегация данных: управление журналами данных; данные собираются из различных источников сетевые устройства и сервисы, датчики систем безопасности, серверы, базы данных, приложения; обеспечивается консолидация данных с целью критических событий.
* Корреляция: поиск общих атрибутов, связывание событий в значимые кластеры. Технология обеспечивает применение различных технических приемов для интеграции данных из различных источников для превращения исходных данных в значащую информацию. Корреляция является типичной функцией подмножества Security Event Management.
* Оповещение: автоматизированный анализ коррелирующих событий и генерация оповещений (тревог) о текущих проблемах. Оповещение может выводиться на «приборную» панель самого приложения, так и быть направлено в прочие сторонние каналы: e-mail, GSM-шлюз итп.
* Средства отображения (информационные панели): отображение диаграмм помогающих идентифицировать паттерны отличные от стандартного поведения.
* Совместимость (трансформируемость): применение приложений для автоматизации сбора данных, формированию отчетности для адаптации агрегируемых данных к существующим процессам управления информационной безопасностью и аудита.
* Хранение данных: применение долговременного хранилища данных в историческом порядке для корреляции данных по времени и для обеспечения трансформируемости. Долговременное хранение данных критично для проведения компьютерно-технических экспертиз, поскольку расследование сетевого инцидента вряд ли будет проводиться в сам момент нарушения.
* Экспертный анализ: возможность поиска по множеству журналов на различных узлах; может выполняться в рамках программно-технической экспертизы.

SIEM способна выявлять:

* сетевые атаки во внутреннем и внешнем периметрах;
* вирусные эпидемии или отдельные вирусные заражения, неудаленные вирусы, бэкдоры и трояны;
* попытки несанкционированного доступа к конфиденциальной информации;
* фрод и мошенничество;
* ошибки и сбои в работе информационных систем;
* уязвимости;
* ошибки конфигураций в средствах защиты и информационных системах.

Splunk Enterprise - платформа для операционной аналитики. Способна осуществлять мониторинг и анализ всех действий, от посещений веб-сайтов и транзакций до сетевых операций и зарегистрированных вызовов.

Splunk – это мощный инструмент операционной аналитики, отслеживающий логи любых систем и собирающий их в единую базу.

Особенности системы:

* Сбор данных из удалённых источников
* Корреляция сложных событий, охватывающих множество разнородных источников данных в среде.
* Масштабирование для сбора и индексации сотен терабайтов данных в день
* Возможность комбинирования данных из традиционных реляционных БД и Hadoop для последующего анализа.
* Ролевая модель доступа к данным.
* Возможность создавать собственные приложения. Можно создавать панели (dashboard’ы), из которых формировать свое собственное Splunk-приложение. У Splunk есть магазин приложений (хотя большинство из них бесплатны), где есть море уже готовых конфигураций для анализа популярных систем, например, UNIX syslog, логи Apache, Microsoft Exchange и т.д.

**Задания к лабораторной работе**

* Загрузите Splunk Enterprise с <http://www.splunk.com/ru_ru/download/splunk-enterprise.html> Выберите вышу систему, после чего нужно будет зарегистрироваться.
* После загрузки дистрибутива, его необходимо установить. Установка deb пакета выполняется командой <dpkg -i splunk\_package\_name.deb>. О других типах установки можно прочитать [здесь](http://docs.splunk.com/Documentation/Splunk/4.3.2/Installation/InstallonLinux)
* Для запуска Splunk выполните </opt/splunk/bin/splunk start>
* Запустите web-интерфейс, при запуске splunk будет указано, как полдключиться к нему (Что-то похожее на [https://sit-VirtualBox:8000](https://sit-virtualbox:8000/) ), чтобы начать использовать систему.
* Учётные данные по умолчанию – admin – changeme. При первом входе Вам будет предложено их изменить.
* В левой части окна будут перечислены приложения, установленные в Splunk и доступные для работы. Приложение это своего рода среда или интерфейс, в котором пользователь работает с событиями, которые собирает Splunk. По умолчанию доступно приложение Search and Reporting. У Splunk есть несколько основных типов расширения функциональности – приложения (Apps) и дополнения (Add-on).
* В центральной части экрана будет пустое окно, на котором предполагается размещение главного дашборда. В правой верхней части расположено меню для управления системой Splunk, в том числе всеми источниками данных.
* Подключим источник событий. Добавим журнал событий Linux, для мониторинга. В правой верхней части экрана выбирайте меню Settings и переходите в Data Inputs
* Перейдите в меню Settings – Data Input - Files & directories. Тип Files & directories позволяет получать события из локальных файлов и директорий.
* Нажмите на кнопку «New», введите путь к файлу auth.log (var/log/.auth.log) и выберите continuously monitor.
* Нажмите «Next». Выберите тип данных (sourcetype – operating system) из списка, а именно «linux\_audit». В открывшемся окне можно ничего не менять. Если всё прошло успешно, то после нажатия на «Start searching» вы увидите перечень событий из журнала аудита.
* Перейдите в меню Settings – Data Input - Files & directories. Добавьте домашнюю директорию, в ней создайте и удалите несколько файлов, Просмотрите журнал событий в Splunk.
* Добавьте еще несколько файлов, директорий и логов, через меню Settings – Data Input - Files & directories.
* Перейдите в приложение «Search and Reporting». Вы попадете на вкладку Search.
* Найдите события, которые относятся к файлу var/log/.auth.log , для этого введите «source=var/log/.auth.log». Здесь так же можно выбрать записи который относятся к Sourcetype (sourcetupe=operating system) – это имя типа данных, куда предполагается относить все данные определённого типа, или Host (host=splunk) – это идентификатор источника, от которого приходят события в какой-либо sourcetype (обычно доменное имя или ip-адрес). Можно фильтровать данные, введя в строку поиска определенные параметры, вы получите записи, только с этими параметрами. Можно делать составные запросы. Один запрос может состоять из множества подзапросов разделенных между собой pipe (|), и справа налево каждый следующий запрос оперирует данными полученными в результате выполнения предыдущего.
* Сбор логов – это далеко не всё, что необходимо для безопасности. Для SIEM нужно, чтобы система не только собирала логи, но и находила события, связанные с нарушениями безопасности. При слежении за логами, можно автоматически обнаруживать любые угрозы безопасности. Splunk можно использовать вместе с IDS.
* В лабораторной работе №14, вы уже познакомились IDS Snort. Так что, установите и настройте Snort, так же как в лабораторной работе №14. Запустите Snort с ведением логов <sudo snort -A console -i eth0 -c snort.conf -l /var/log/snort/>. Произведите различные типы сканирования nmap, и проверку правил Snort. И добавьте логи Snort в Splunk. Вы так же можете загрузить приложение Snort для Splunk <https://splunkbase.splunk.com/app/340/> . Вместо Snort можно так же использовать OSSEC, для OSSEC тоже есть приложение в Splunk.

**Вопросы к лабораторной работе**

1. Что такое SIEM?
2. Для чего используют SIEM?
3. В чем отличие SIEM от IDS?
4. Где используют SIEM?
5. В чем преимущества SIEM?
6. В чем недостатки SIEM?
7. Какие существуют альтернативы использованию SIEM?

**Составьте отчет о выполнении лабораторной работы.**

**Включите в него копии экрана и ответы на вопросы лабораторной работы.**