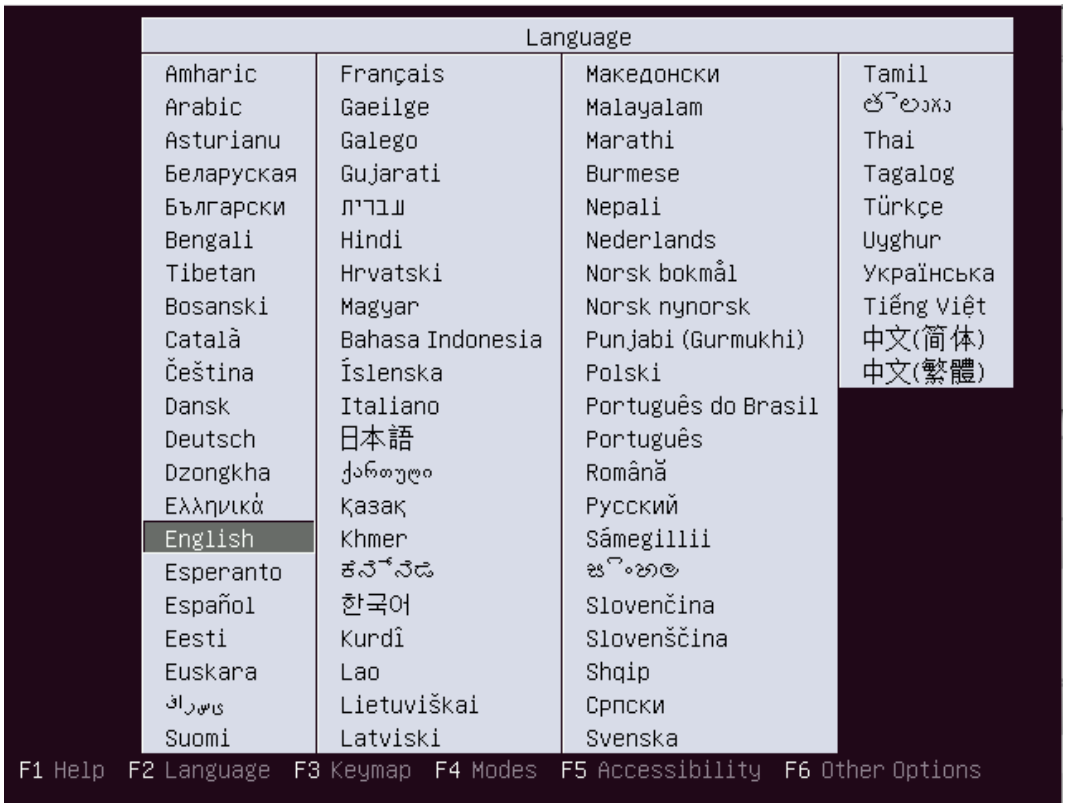
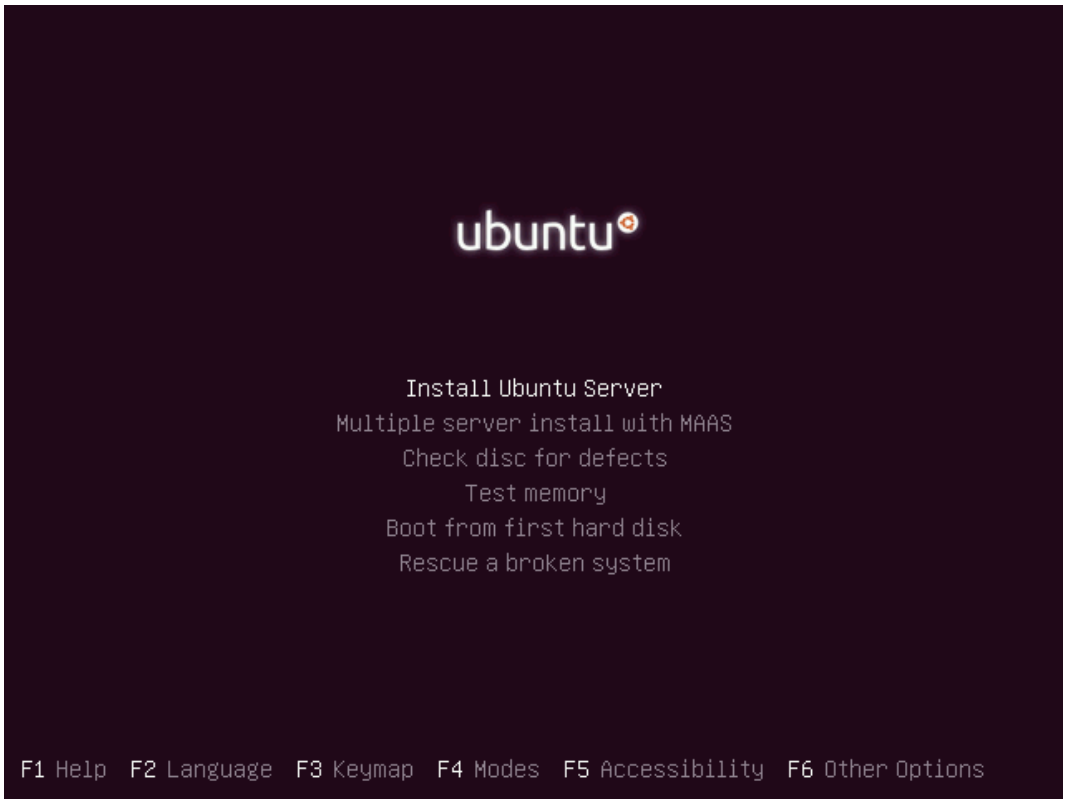
Подготовка к лабораторным работам. Установка Ubuntu 16.04 Server (либо 14.04)

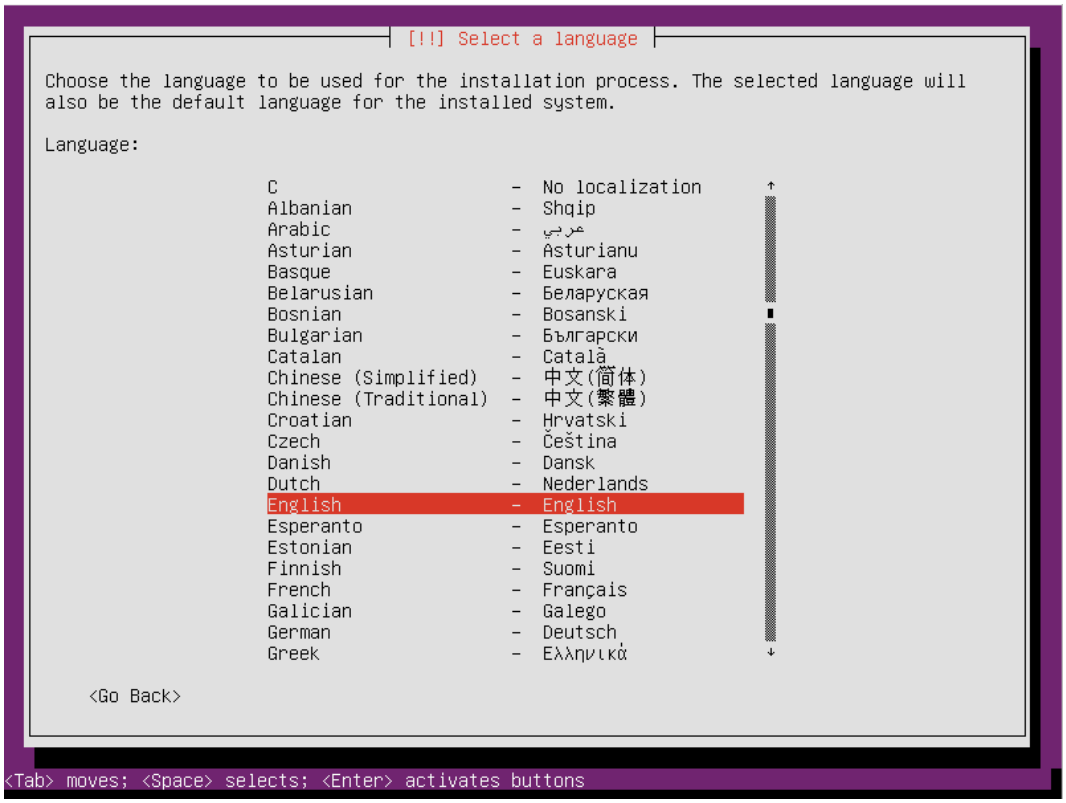
Установку ОС можно производить на жетский диск, с помощью систем виртуализации, либо путём создания загрузочного образа на съёмном носителе.



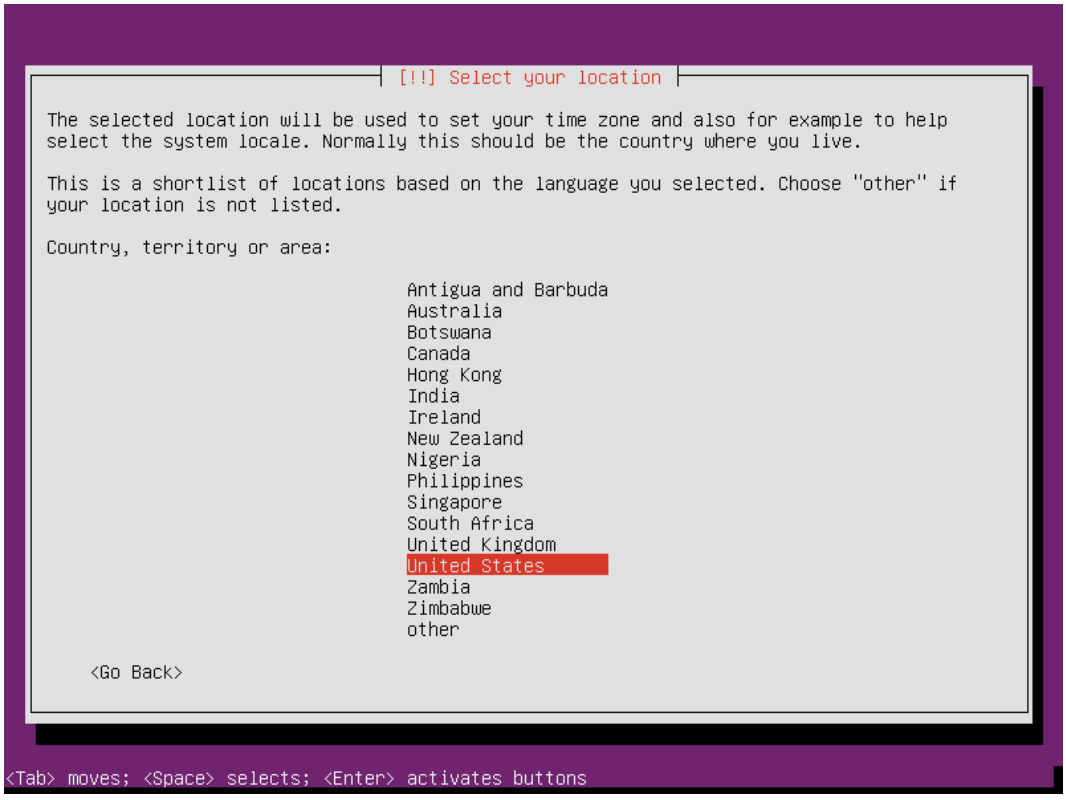
В начале установки предлагается выбрать язык. Выбираем **English**.



Запускаем установку Ubuntu Server, нажимаем **Enter**.

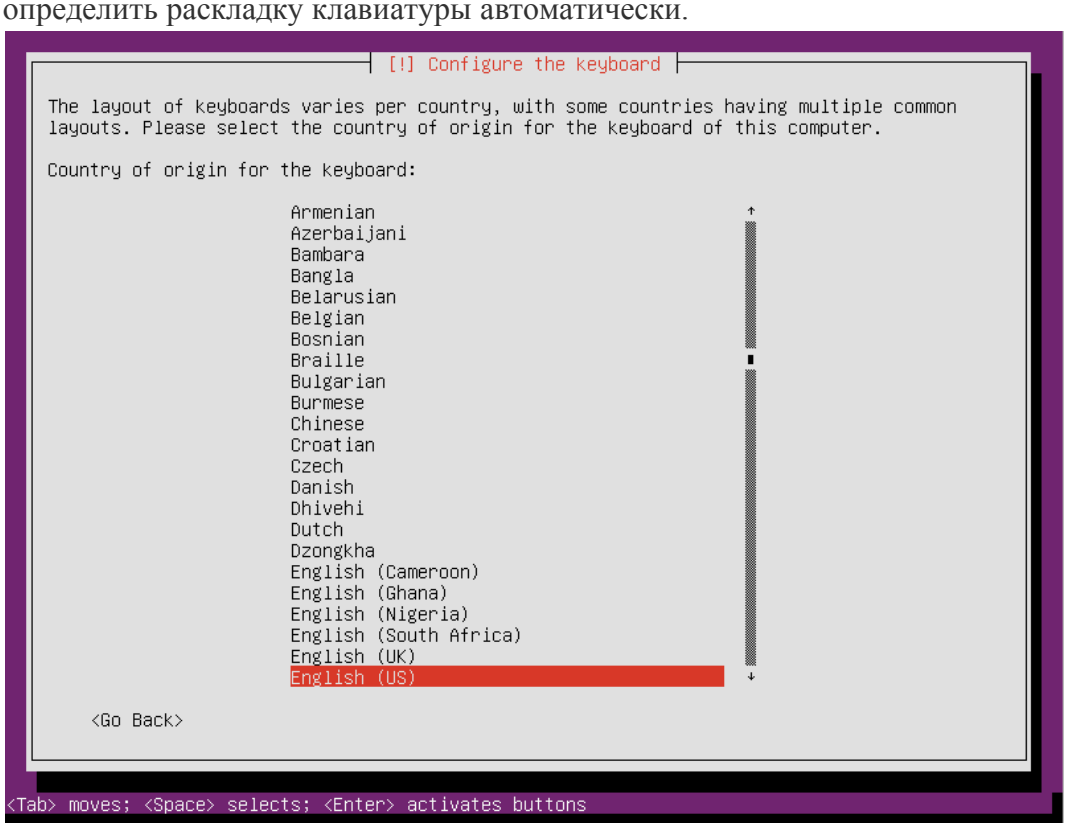


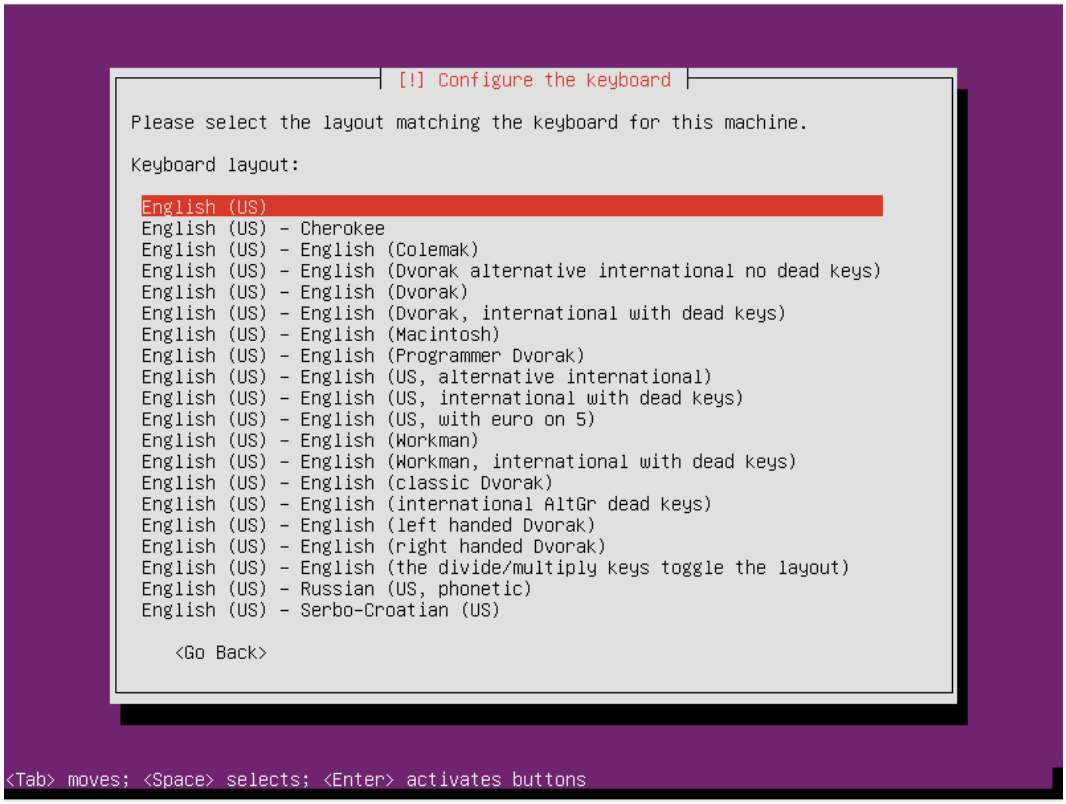
Выбираем язык установки.



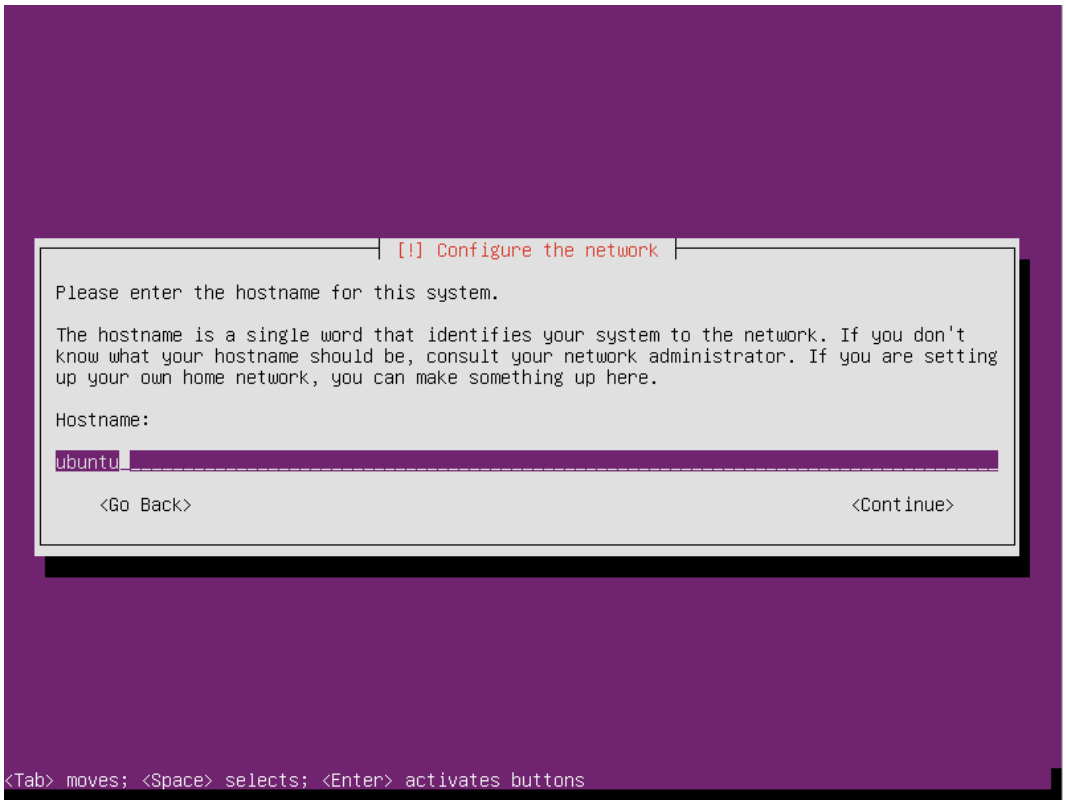
Выбираем страну.

Предлагается определить раскладку нажимая на клавиши клавиатуры. Выбираем **<No>**, чтобы определить раскладку клавиатуры автоматически.

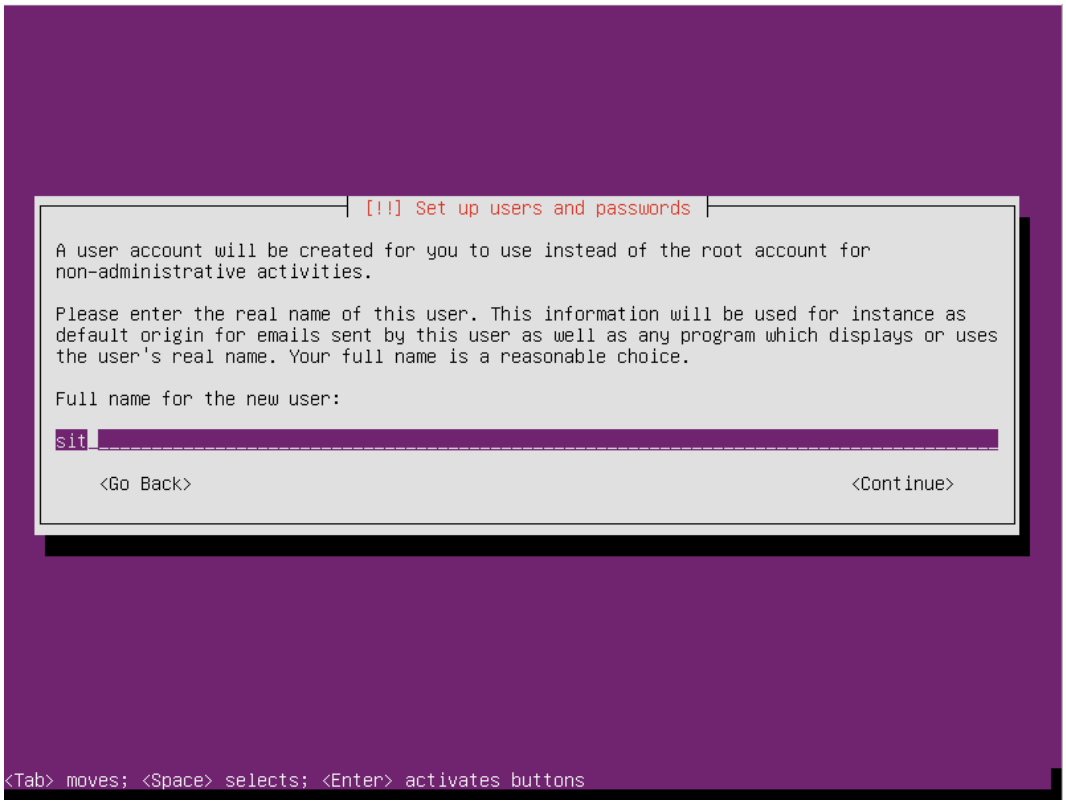




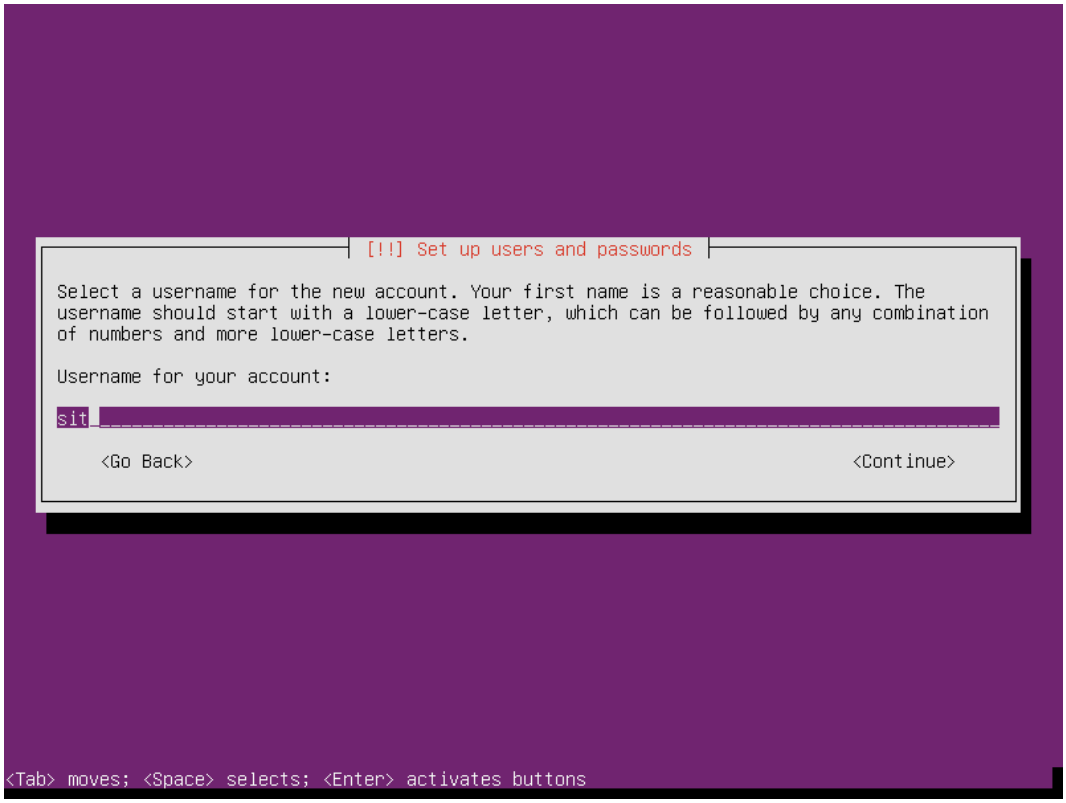
Выбираем раскладку клавиатуры (**English (US)**).



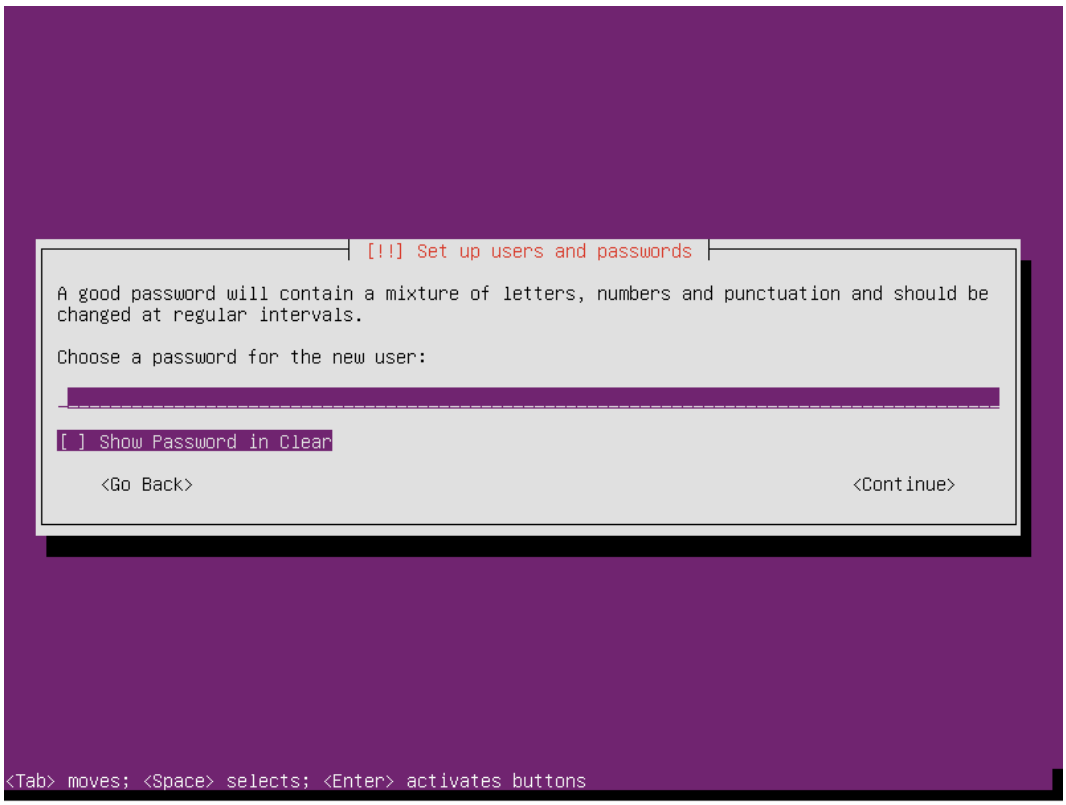
Вводим имя сервера (например **ubuntu**) и нажимаем **<Continue>**.



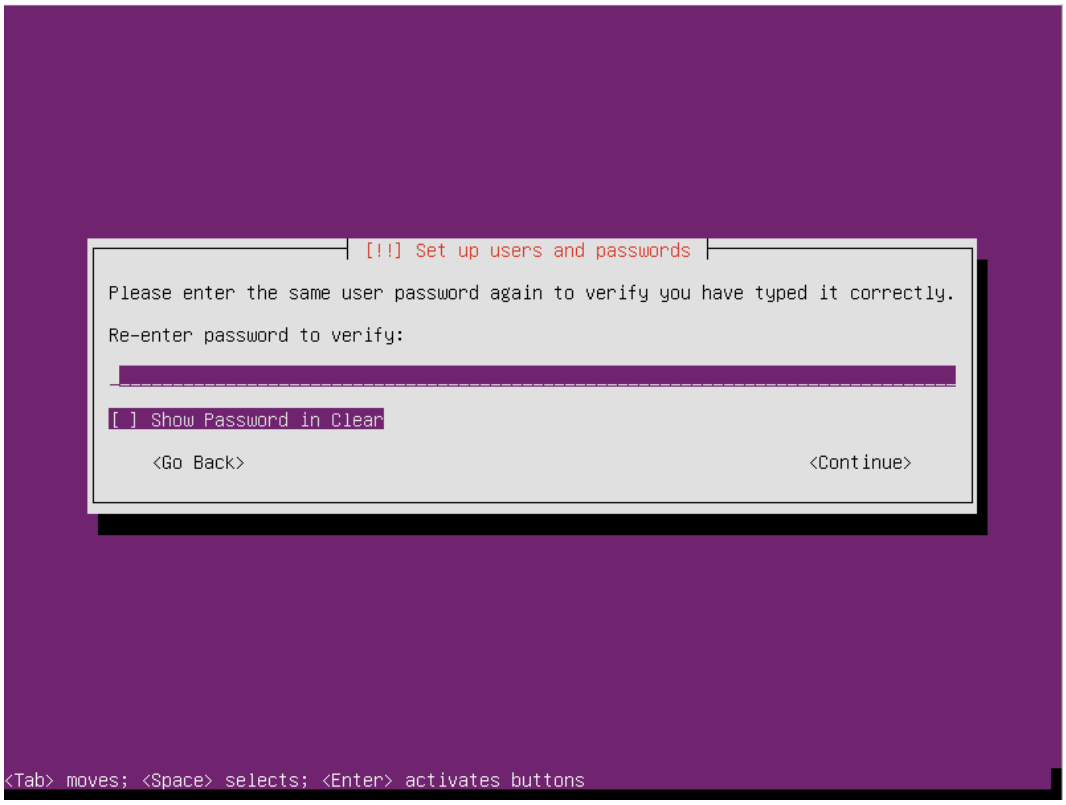
Вводим имя пользователя (например **sit**) и нажимаем **<Continue>**.



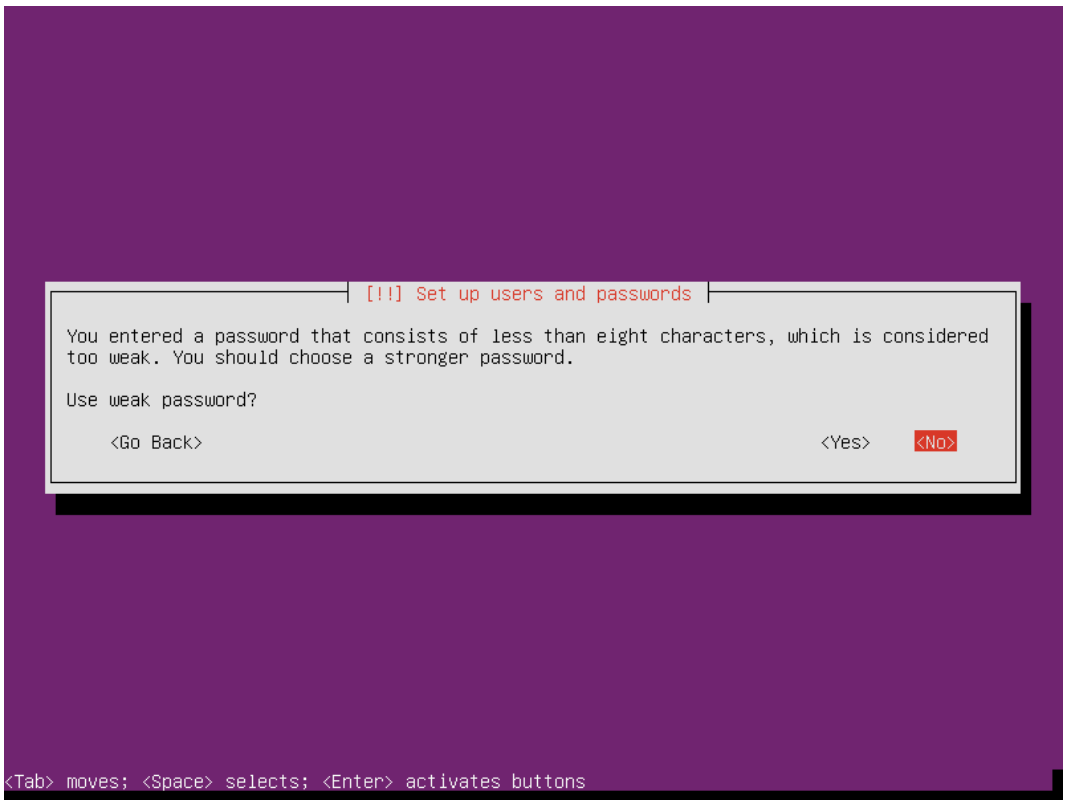
Повторяем ввод имени пользователя (например **sit**).



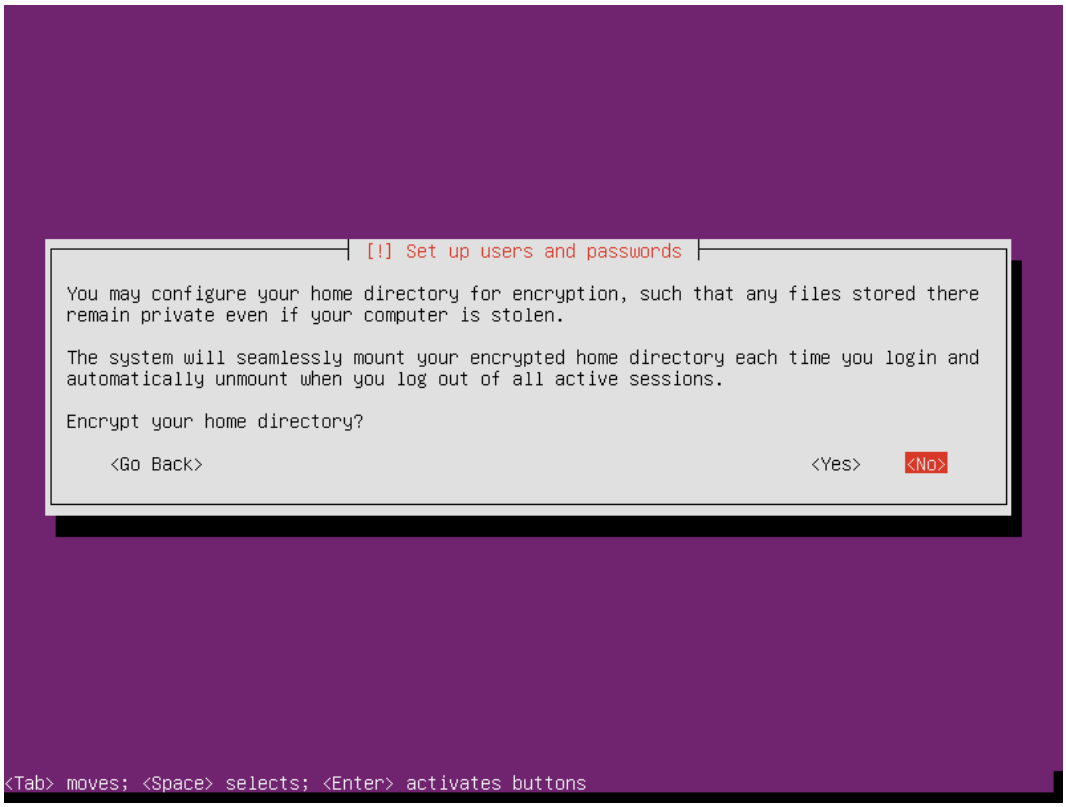
Вводим пароль (например **sit**).



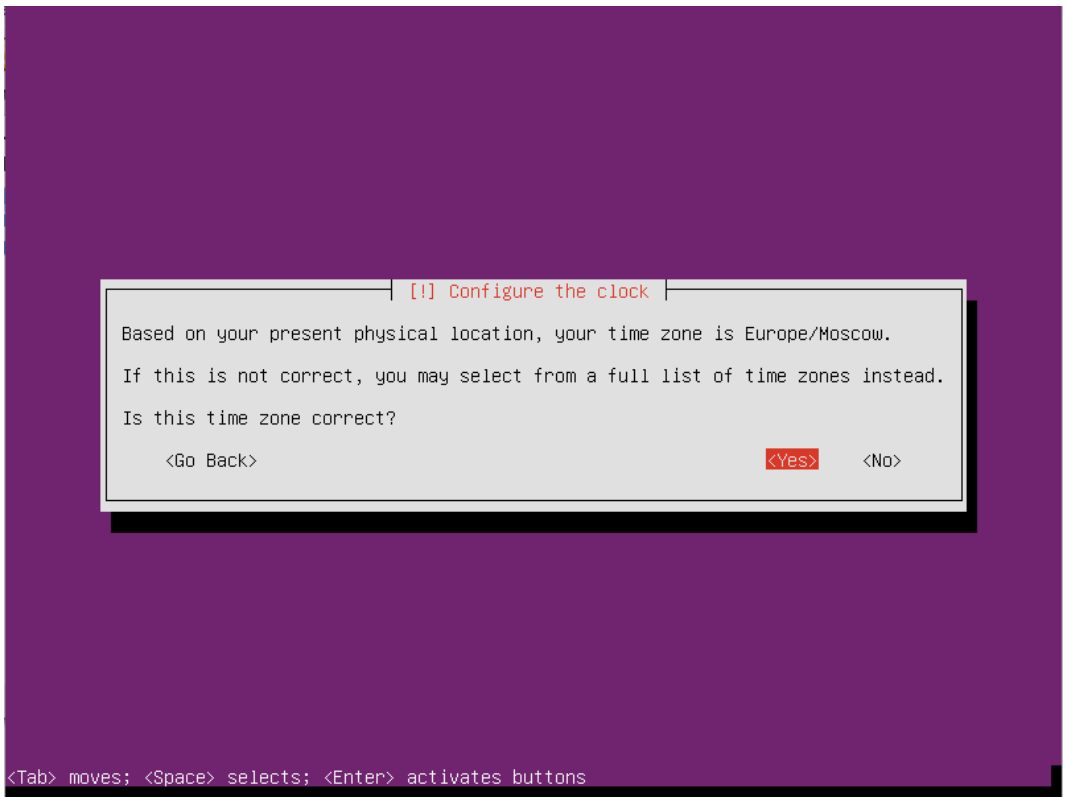
Повторно вводим пароль.



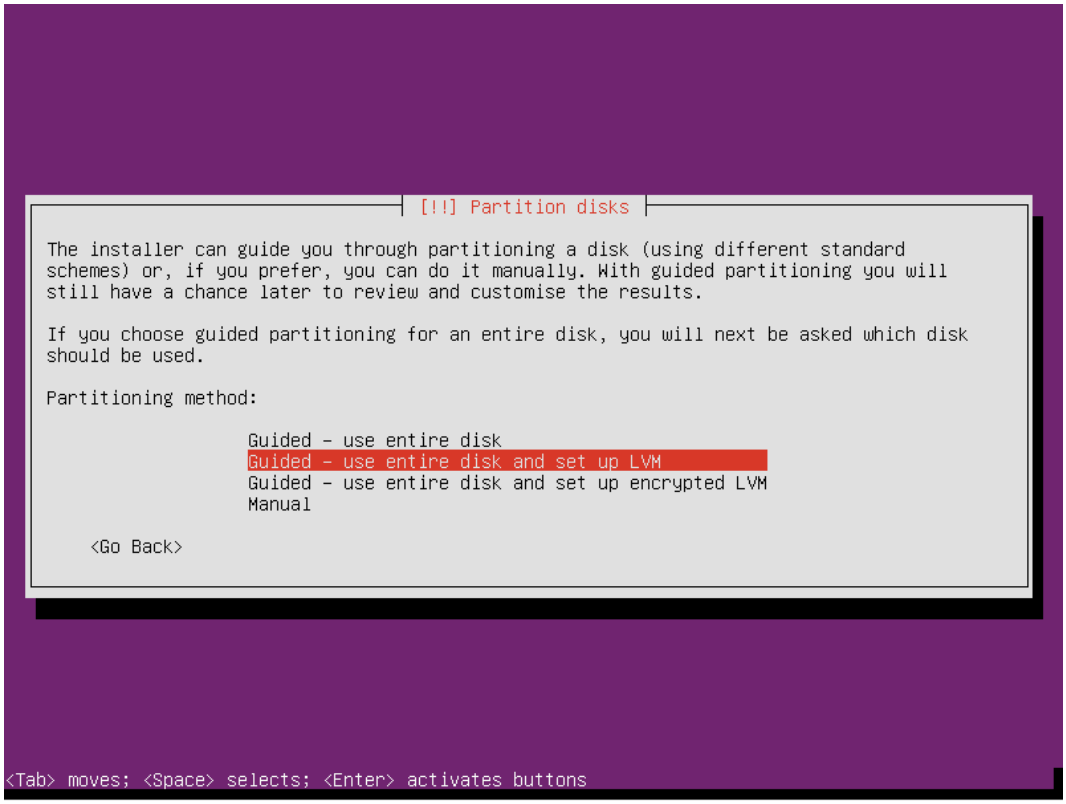
Если вы выбрали слабый пароль, система предупредит вас об этом. И спросит, точно ли вы хотите использовать слабый пароль. Чтобы продолжить с нашим паролем, выбираем **<Yes>**:



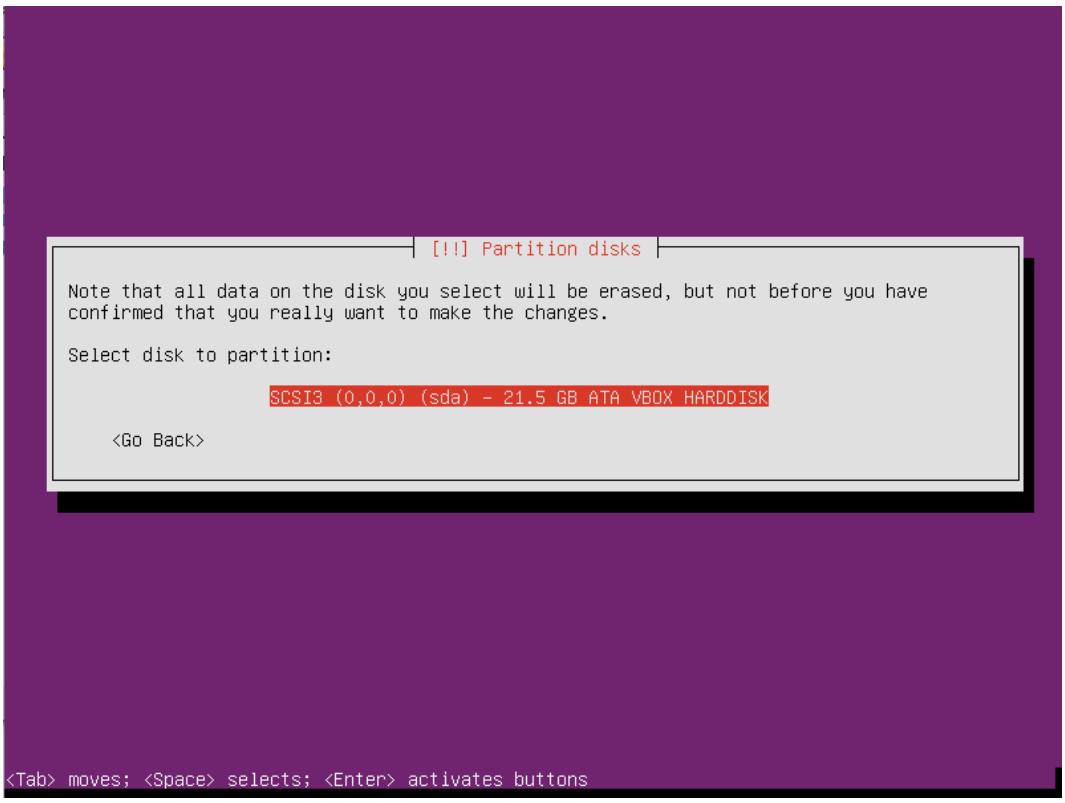
Не соглашаемся шифровать домашний каталог, иначе придется вводить пароль при загрузке сервера. Если придется перезагрузить сервер удаленно, сервер не загрузится без ввода пароля для подключения шифрованного раздела.



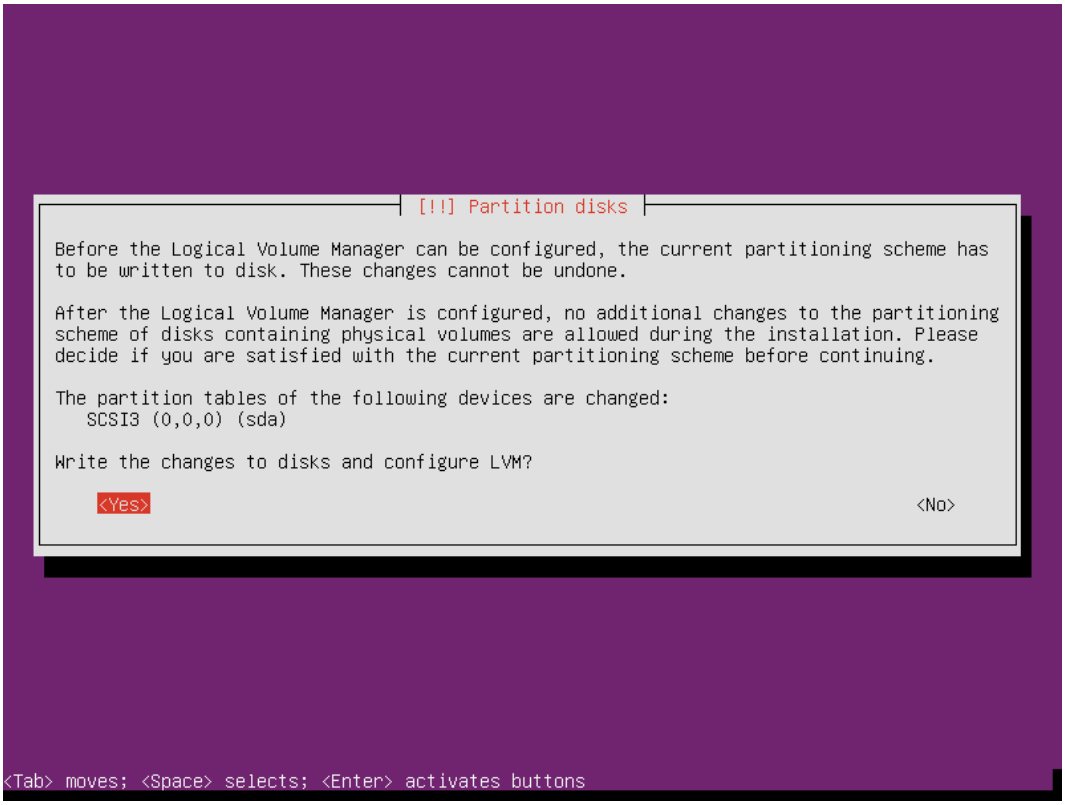
Если часовой пояс выбран корректно, нажимаем **Yes**.



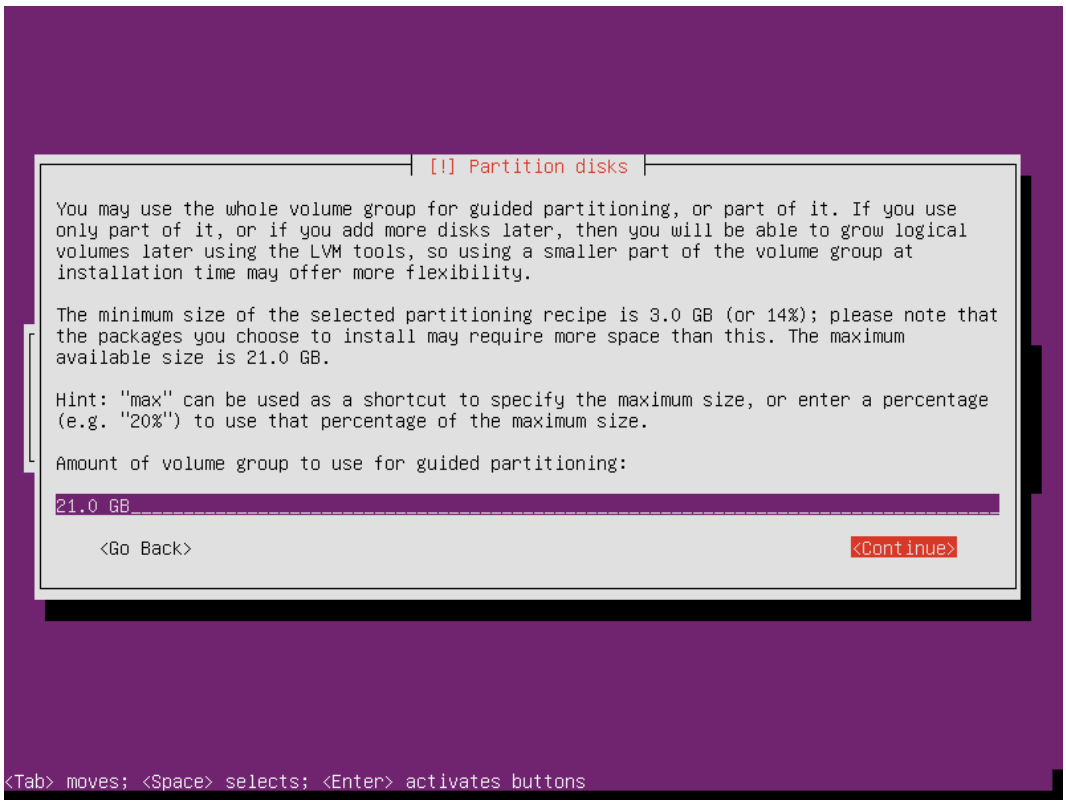
Переходим к разметке диска. Выбираем автоматическую разметку диска с настройкой диспетчера логических томов (LVM).



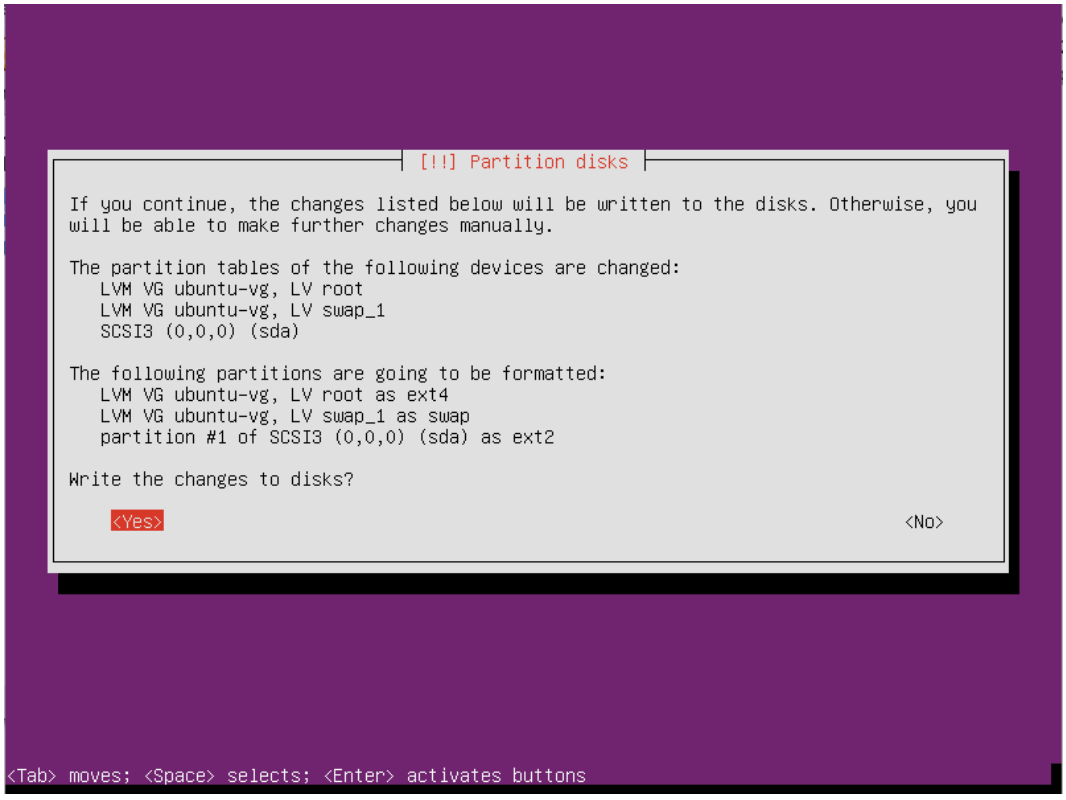
Выбираем диск, нажимаем Enter.



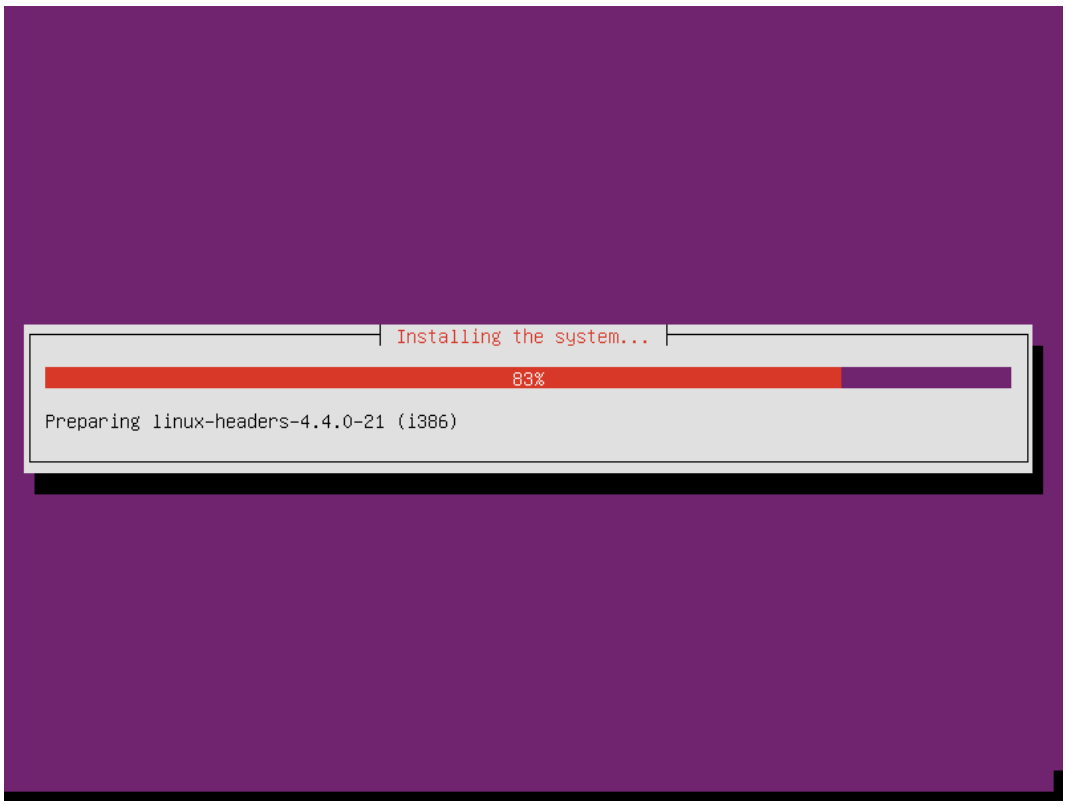
Соглашаемся записать изменения на диск.



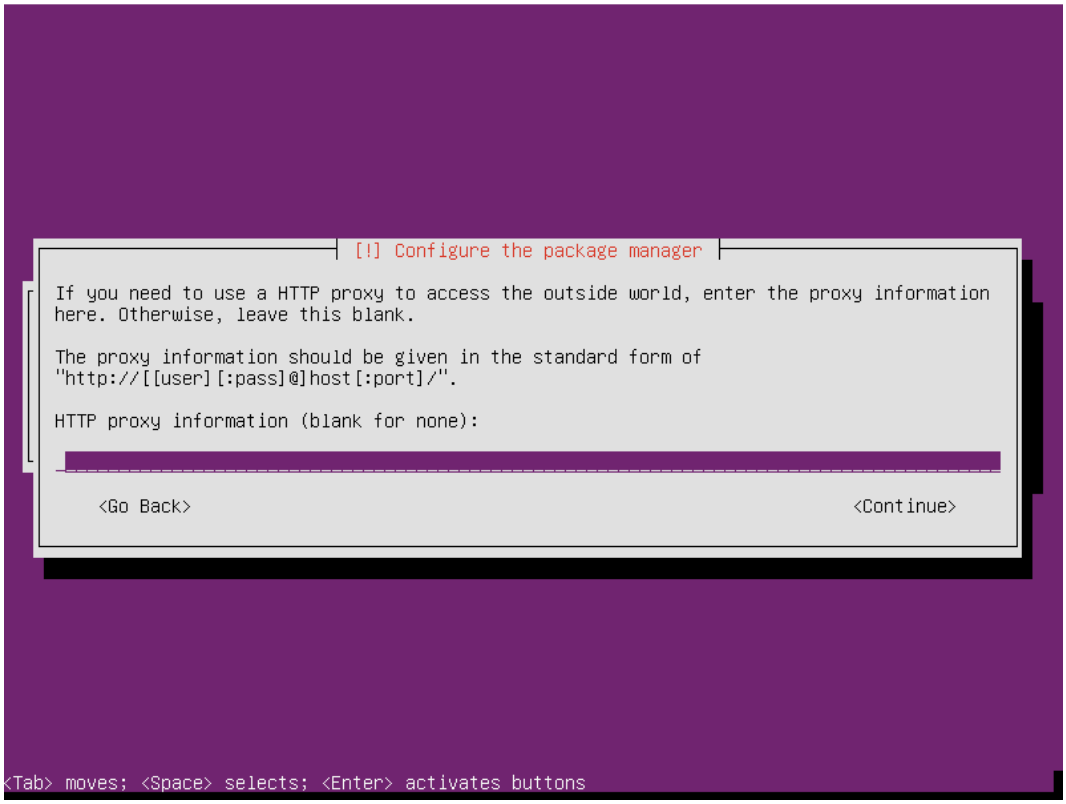
Нажимаем **<Continue>**



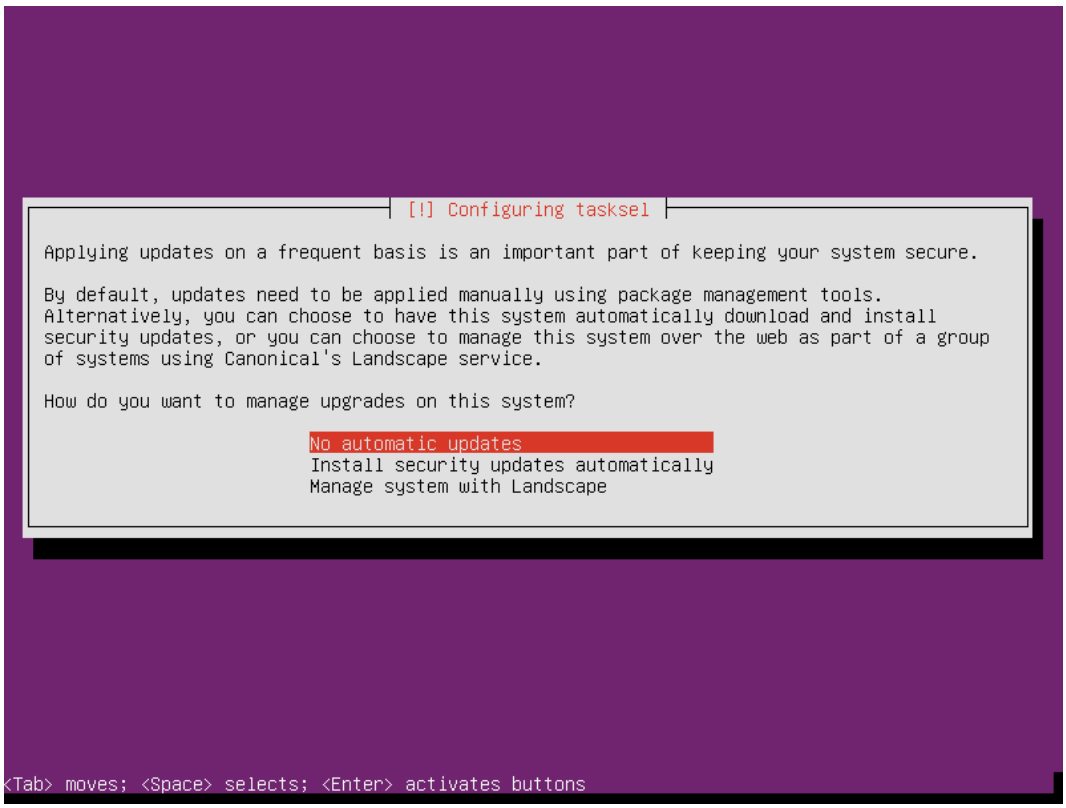
Проверяем автоматическую разметку и записываем изменения на диск нажав **<Yes>**



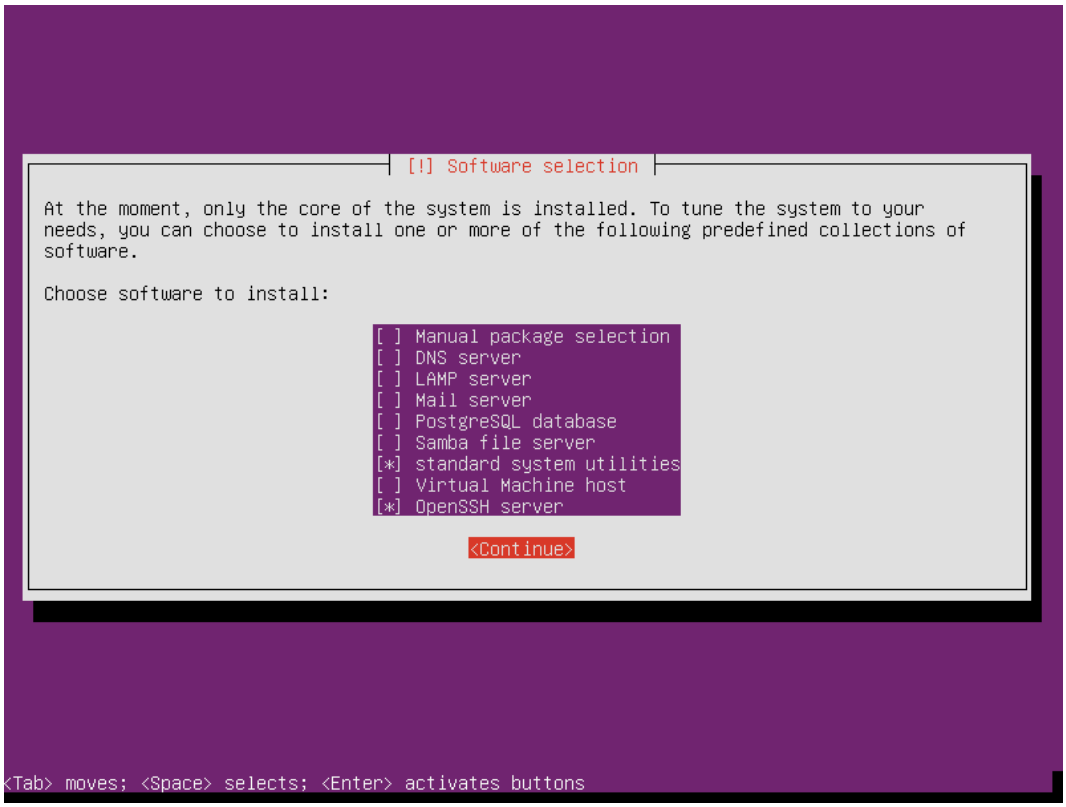
В дальнейшем будет запущен процесс установки, это может занять некоторое время.



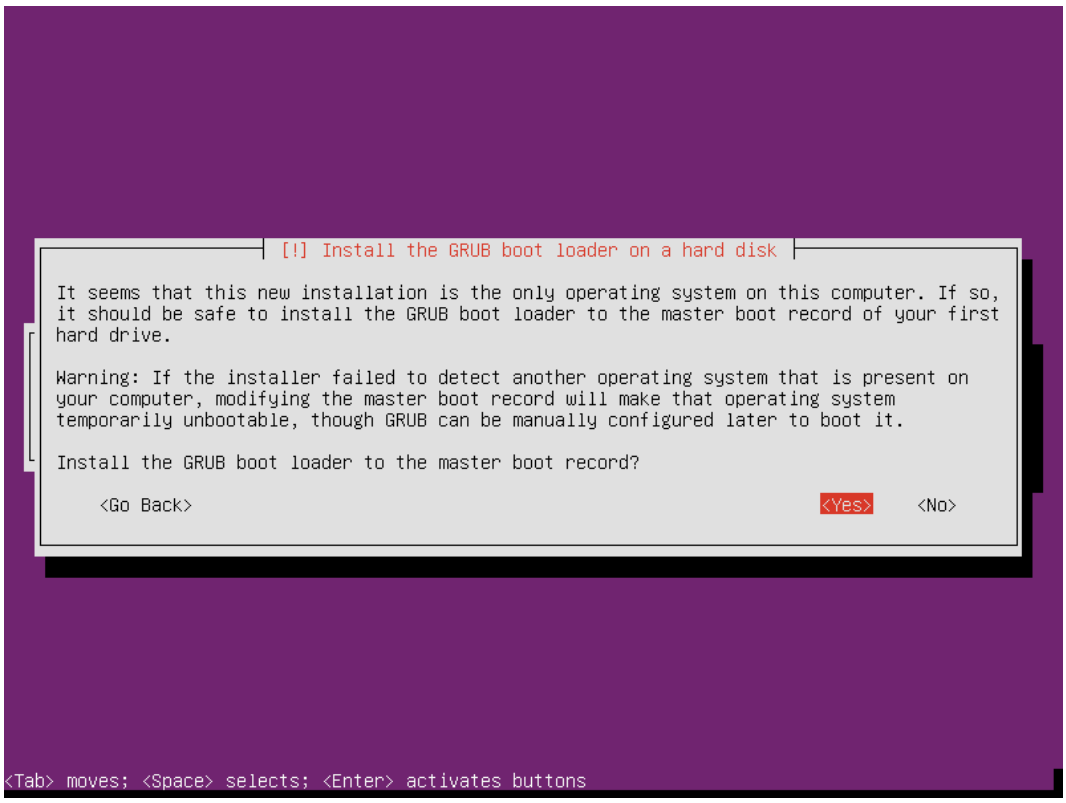
Нажимаем **Continue**, оставляя поле пустым, если для выхода в интернет вам не требуется прокси (Если требуется настройка прокси, заполните поле, согласно инструкции на экране).



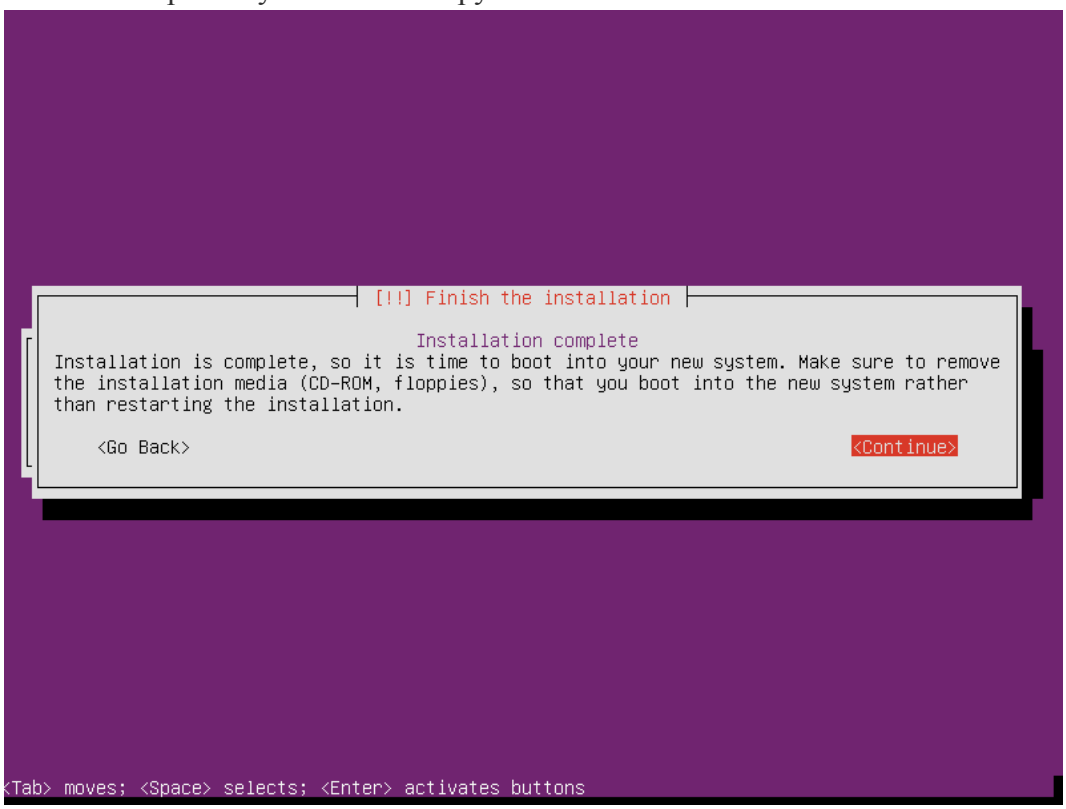
Выбираем **No automatic updates** - обновления устанавливаются вручную. Если необходимо автоматическое обновление выбирете соответствующий пункт.



В окне выбора установки програмного обеспечения выбираете нужные, исходя из ваших потребностей и нажмите **Continue**.



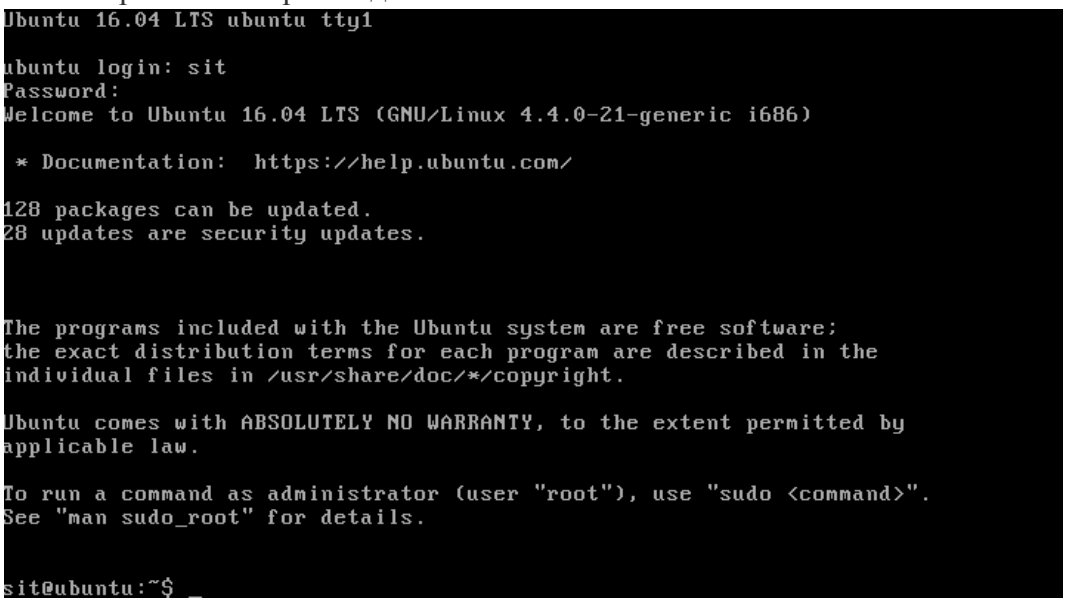
Соглашаемся с запросом установить загрузчик GRUB.



После завершения установки нажимаем **Continue**.

Произойдет перезагрузка виртуальной машины. После загрузки операционной системы, необходимо ввести логин и пароль (в нашем случае **sit** и **sit**).

Пароль не отображаеться при вводе.



Задача по установке Ubuntu 16.04 Server выполнена. Можно приступить к выполнению лабораторных работ.

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.

Все, с чем Вы встретитесь в операционной системе Linux, - это файлы. Абсолютно все! Очевидно, что текстовый документ - это файл. Изображения, аудиоданные в формате MP3 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 MacOS 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 MacOS, 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

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 с помощью команды man:
- Определите текущий каталог, в котором вы находитесь командой pwd:
- Ознакомьтесь с возможностями команды cd с помощью команды man:
- Перейдите в корневой каталог командой cd
- Ознакомьтесь с возможностями команды ls с 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 с 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 с помощью команды man:
- Создайте файл «text» в каталоге «test2» используя команду touch:
- Ознакомьтесь с возможностями команды mv с помощью команды man:
- Переименуйте файл «text» в «textSIT» используя команду mv
- Ознакомьтесь с возможностями команды cp с помощью команды man:
- Скопируйте файл «textSIT» в каталог «test2» под именем «copy.txt», используя команду cp:
- Ознакомьтесь с возможностями команды ln с 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.

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» с 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.

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).

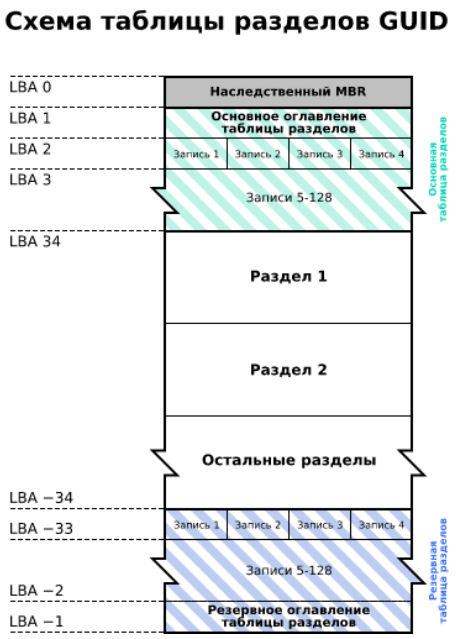
Структура GPT

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).

Структура

Название	Адрес	Описание
Наследственный MBR	LBA 0	Первые 512 байт диска отведены под "фейковый MBR". В нём из записей есть только индентификатор диска, стандартная сигнатура **0x55AA** в конце и единственный фейковый раздел типа **0xEE** (указание, что используется GPT), внутри которого находится настоящая разметка диска и все пользовательские данные. Остальное забито нулями, кода загрузчика нет. Наследственный MBR служит для предотвращения потери данных из-за программ, которые не понимают GPT.
Основная таблица разделов GPT	LBA 1	Оглавление таблицы разделов. Содержит GUID диска, адреса основной и резервной таблиц и данные о размере и количестве записей о разделах (стандартно — 128 штук). И контрольную сумму, которую проверяет EFI. "Благодаря" этой контрольной сумме **ручное редактирование разделов GPT невозможно**.
	LBA 2 ... LBA 33	Записи данных о разделах. Каждая запись занимает 128 байт, то есть в один LBA вмещается 4 записи. Первые 16 байт записи — GUID типа раздела, следующие 16 байт — его UUID, уникальный идентификатор, остальное место занимает информация о его границах и атрибутах.
Данные	LBA 34 ... LBA *	Собственно, содержимое разделов.
Резервная таблица разделов GPT	LBA -33 ... LBA -2	Полная копия описания разделов
	LBA -1	Полная копия оглавления.



Примечание

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.

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:~$

Примечание
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 с 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>
Примечание

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
Примечание
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
```

Примечание

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

Примечание

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.

Примечание

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.

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 C.
- 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, либо используйте для восстановления данные со съемного носителя, либо восстановите файлы с жесткого диска, которые были удалены при выполнении предыдущих лабораторных работ.
- 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/.
- Удалите данные с созданного каталога.
- С помощью Extundelete или Foremost восстановите данные.

Лабораторная работа №6. Шифрование данных.

Основные теоретические сведения

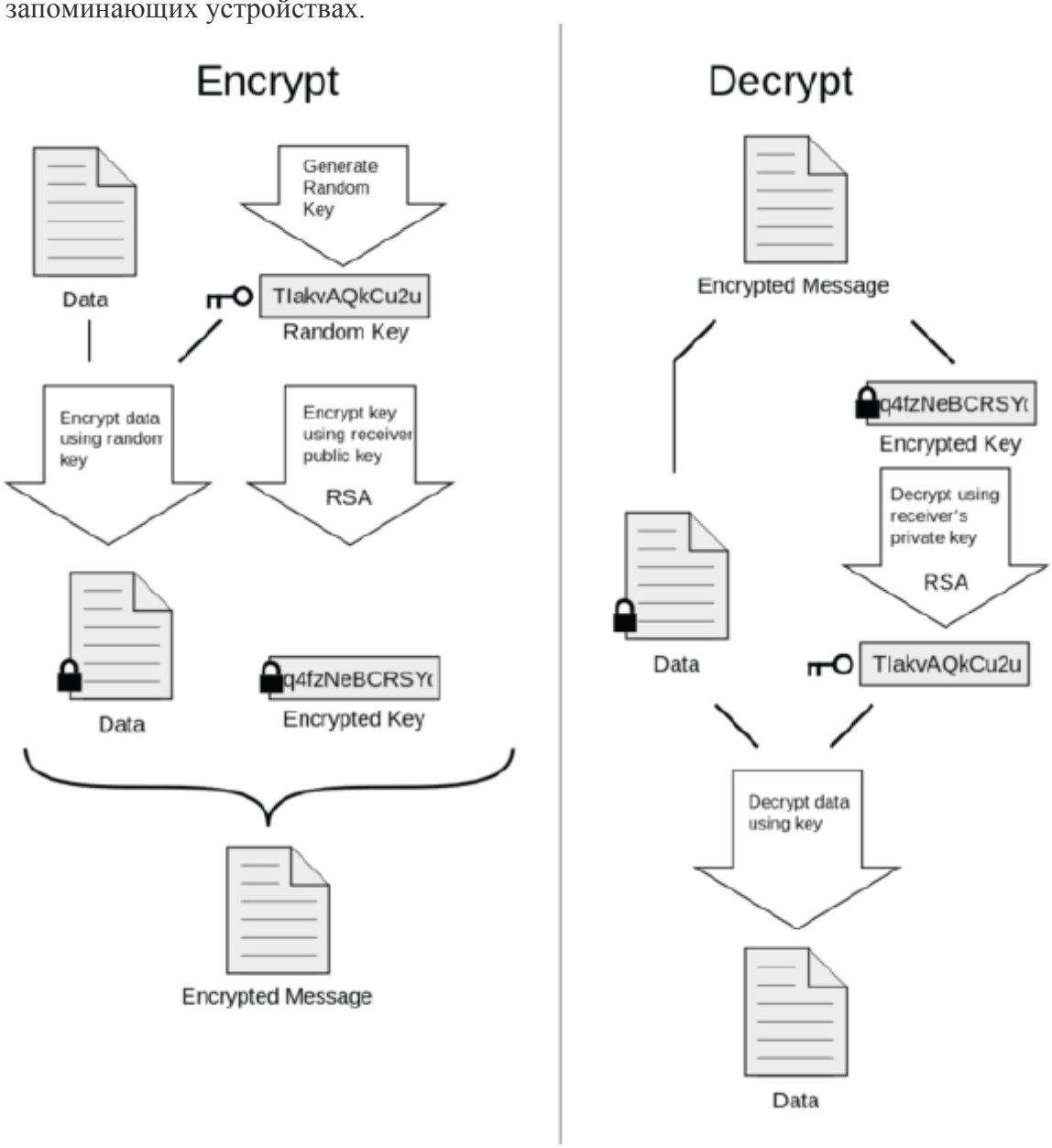
Цель: Получение теоретических и практических навыков работы с программными средствами шифрования данных.

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 7.1a. Скачать её можно здесь или здесь.
- 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-нибудь 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>

Лабораторная работа №7. 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	iptables -L

	таблица filter всех цепочек).	
–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, –	IP адрес источника пакета. Может быть определен	iptables -A INPUT -s 10.10.10.3

src)	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	
–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
- -A — 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

- Необходимо установить набор програмного обеспечения LAMP (Linux Apache MySQL PHP) с помощью следующих команд:
  sudo apt-get update
  sudo apt-get install tasksel

sudo 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>

Лабораторная работа №8. 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 октета вашего ip-адреса, например, это 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.
- Используйте различные методы сканирования nmap (используйте -sS, -sT, -sN, -sU, -sX, -sF и посмотрите, как реагирует Snort:

  nmap <IP-адрес вашего компьютера> -v –sT –p <диапазон портов> — для сканирования методом с полным циклом подключения (метод Connect);

  nmap <IP-адрес вашего компьютера > -v –sS –p <диапазон портов> — для сканирования с неполным циклом подключения (метод SYN);
  nmap <IP-адрес вашего компьютера > -v –sN –p <диапазон портов> — для сканирования при помощи TCP-пакета со сброшенными флагами (метод NULL);
  nmap <IP-адрес вашего компьютера > -v –sX –p <диапазон портов> — для сканирования при помощи TCP-пакета со всеми установленными флагами (метод XMAS);

- 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;)
- 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/

Лабораторная работа №9. 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 по ссылке: 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), 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 №7, вы уже познакомились IDS Snort. Так что, установите и настройте Snort, так же как в лабораторной работе №7. 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.