1.5 “Настройка безопасного удаленного доступа к серверу. Защита передаваемых данных”

В данной работе будет проведена комплексная конфигурация безопасности доступа к серверу, сервисов Asterisk Nginx и тд.

Защита загрузчика

В современных дистрибутивах Linux для загрузки ОС используется загрузчик GRUB2 (GRand Unified Bootloader 2 версии). По умолчанию загрузчик обычно не защищен, т.е. можно изменить параметры загрузки ядра при наличии локального доступа (или удалённой KVM консоли) к серверу/ВМ во время загрузки системы и, например, получить доступ с root правами.

Продемонстрируем способ получения такого доступа, не зная ни имен пользователей ни паролей. Запустите ВМ с CentOS, когда появится меню загрузчика нажмите e. Отобразятся параметры загрузочного устройства, загрузки и инициализации ядра и тд. Найдите строку с параметрами загрузки ядра (строка начинается с `linux16` или `linuxefi`), замените параметр `ro` на `rw` для монтирования файловой системы в режим чтения-записи, добавьте в самый конец параметров `init=/bin/sh` и нажмите Ctrl+X для загрузки системы с этими параметрами. CentOS загрузится в терминал от имени пользователя root. Теперь Вы можете совершать любые действия в системе, добавить или просмотреть пользователей в системе (`cat /etc/passwd`), сменить пароль любому пользователю (`passwd` *`username`*), внести любое изменение в работу ОС. ЕСЛИ в CentOS БЫЛ включен SELinux в активном режиме (это не наш случай, сведения приведены просто для справки), выполните команду `touch /.autorelabel` для создания в корне файловой системы пустого скрытого файла, который является маркером SELinux, что при загрузке необходимо обновить метки всех файлов (после загрузки проверьте, что этот файл удалился, иначе удалите вручную).

Для продолжения загрузки ОС введите `exec /sbin/init`, для перезагрузки `exec /sbin/reboot`

Пароль на GRUB устанавливается следующей командой:

```
grub2-setpassword
```

Теперь для доступа к параметрам загрузки потребуется авторизоваться под именем пользователя root с заданным паролем (это не системный root, просто имя совпадает).

Смена порта SSH

Использовать SSH на стандартном 22 порту, особенно при наличии доступа к серверу из сети Интернет, не рекомендуется ввиду частого проведения атак злоумышленниками по этому порту с целью получить доступ к ОС и использовать сервер в своих целях (часто незаметно для владельца). С целью усложнения проведения атак также рекомендуется запретить доступ к системе по SSH пользователю root, не использовать стандартные имена пользователей вроде admin.

Для смены порта на другой, к примеру, 30000, и установления запрета на вход пользователя root, требуется отредактировать в файле `/etc/ssh/sshd_config` следующие параметры:

```
Port 30000
PermitRootLogin no
```

Далее для успешного прохождения сетевого трафика нужно разрешить его в `firewalld`

```
sudo firewall-cmd --add-port=30000/tcp --permanent
```

Перезапустите демон sshd и перечитайте правила `firewalld` без разрыва текущего соединения

```
sudo systemctl restart sshd && sudo firewall-cmd --reload
```

Существующее соединение SSH не будет разорвано, но все новые соединения будут приниматься только по новому порту. Проверьте возможность подключения, в PuTTY в поле Port смените 22 на 30000.

Если SSH сервер успешно принимает подключения, закройте 22 порт в firewalld

```
sudo firewall-cmd --permanent --remove-service=ssh
```

Не забудьте, что теперь подключение клиента SCP также осуществляется на новый порт.

Защита сервера с помощью fail2ban

Fail2ban - утилита, отслеживающая в логах разных служб определенные события и при происшествии их определенного количества за определенное время выполняющая указанные в конфигурации действия. Используется для защиты от подбора учетных данных (атак типа bruteforce) и атак типа DoS. может отслеживать неудачные попытки подключения на указанных в конфигурации сервисах и портах и соответственно временно создавать блокирующее правило для отбрасывания трафика с IP адреса.

Установите пакет

```
sudo yum install fail2ban-firewalld
```

Базовая настройка параметров блокирования содержатся в файле `/etc/fail2ban/jail.d/00-firewalld.conf`

```
[DEFAULT]
maxretry = 4
findtime = 10m
bantime = 1m
banaction = firewallcmd-ipset[actiontype=<multiport>]
```

* где

- **maxretry —** количество событий, которые могут произойти до срабатывания триггера (наложение бана)
- **findtime —** время в секундах, в течение которого подсчитывается количество событий;
- **bantime —** время, на которое будет заблокирован IP-адрес;
- **banaction** — действия, которое будет выполняться в случае срабатывания триггера;

Для каждого сервиса обычно создается отдельная конфигурация в отдельном файле:

`/etc/fail2ban/jail.d/ssh.conf`

```
[ssh]
enabled = true
port = 30000
filter = sshd
logpath = /var/log/secure
```

/etc/fail2ban/jail.d/asterisk.conf

```
[asterisk]
enabled = true
port = 5060,5061
protocol = all
filter = asterisk
action = firewallcmd-ipset[name=asterisk, port="5060,5061",
protocol=all]
logpath = /var/log/asterisk/messages
```

/etc/fail2ban/jail.d/phpmyadmin.conf

```
[phpmyadmin]
enabled = true
filter = apache-myadmin
port = http,https
action = firewallcmd-ipset[name=phpmyadmin, port="http,https",
protocol=tcp]
logpath = /var/log/nginx/error.log

[nginx-http-auth]
enabled = true
filter = nginx-http-auth
action = firewallcmd-ipset[name=nginx, port="http,https",
protocol=tcp]
logpath = /var/log/nginx/error.log
```

Просмотр настроенных правил

```
fail2ban-client status
```

заблокированных IP

```
fail2ban-client status ssh
```

Пример Удалить из списка заблокированных клиентов для сервиса asterisk все адреса из подсети 192.168.0.0/16

```
sudo fail2ban-client set asterisk unbanip 192.168.*
```

Настройка аутентификации SSH по ключу

Доступ к системе с парольной защитой хорош ввиду простоты и отсутствия особых требований к конфигурации клиента и сервера, однако он также может представлять серьезную уязвимость в системе безопасности перед атаками злоумышленников при использовании ненадежных паролей. Всегда есть шанс, что кто-то будет использовать простые короткие пароли, которые несложно подобрать методом простого перебора при достаточной вычислительной мощности. Кроме того, пароль можно забыть, его могут подсмотреть и тд. Для решения этой проблемы используется аутентификация с использованием пары ключей шифрования. Для обеспечения защиты данных, обычно шифруют защищаемые данные открытым, расшифровывают приватным ключом, в алгоритмах цифровой подписи наоборот - подписывают данные приватным, публичным расшифровывают (проверяют подпись). Исходя из вышеизложенного, повторим, что ключи в асимметричных алгоритмах шифрования используются парами:

- открытый (публичный) ключ,
- закрытый (приватный) ключ.

Протокол SSH использует асимметричные алгоритмы шифрования, в которых, как уже было сказано, данные/соединение передачи данных шифруется одним ключом, расшифровывается другим. открытый ключ помещается на сервер SSH, а закрытый хранится у пользователя, указывается в конфигурации клиента SSH. Как только открытый ключ будет помещен в список доверенных ключей и если разрешено использование аутентификации по ключу в конфигурации SSH сервера, вы сможете подключиться по SSH к серверу, используя приватный ключ для аутентификации, без необходимости вводить пароль для входа в систему. Хотя пара ключей может быть сгенерирована без дополнительной защиты, лучшая рекомендация с точки зрения безопасности - использовать ключи с фразой-паролем.

Традиционно для SSH используется криптоалгоритм RSA, но в последнее время в OpenSSH по умолчанию используются криптоалгоритмы, основанные на вычислениях по эллиптических кривым Монтгомери - Curve25519 (обмен ключами) и Эдвардса - EdDSA (генерация ключей). Наиболее распространенный вариант реализации EdDSA, Ed25519 обеспечивает лучшую безопасность по сравнению с RSA при меньшей длине ключа, что также означает и большую производительность при генерации пар ключей. Из недостатков стоит отметить, что к сожалению, нельзя использовать открытый ключ Ed25519 для шифрования/подписи файлов, как в случае с открытым ключом RSA.

Далее Вам предлагается настроить в рамках работы аутентификацию SSH по ключу без парольной фразы. Сгенерируйте новый ключ командой (после ключа C подставьте произвольный комментарий, обычно он поясняет принадлежность ключа)

```
ssh-keygen -t ed25519 -C "pbx server"
```

не заполняйте (оставьте значения по умолчанию) запрашиваемые параметры, место сохранения и парольную фразу (passphrase) во время процесса генерации пары ключей.

Убедитесь, что оба ключа сгенерированы (есть файлы `id_ed25519` `id_ed25519.pub` в директории `~/.ssh`)

Просмотр приватного ключа

```
cat ~/.ssh/id_ed25519
```

Просмотр публичного ключа

```
cat ~/.ssh/id_ed25519.pub
```

Для разрешения использовать данную пару ключей для аутентификации необходимо создать файл `~/.ssh/authorized_keys` и скопировать в него данные публичного ключа (вывод команды `cat ~/.ssh/id_ed25519.pub`) .

Для работоспособности аутентификации SSH по ключу требуется также обеспечить корректные права доступа на директорию и файл с публичными ключами.

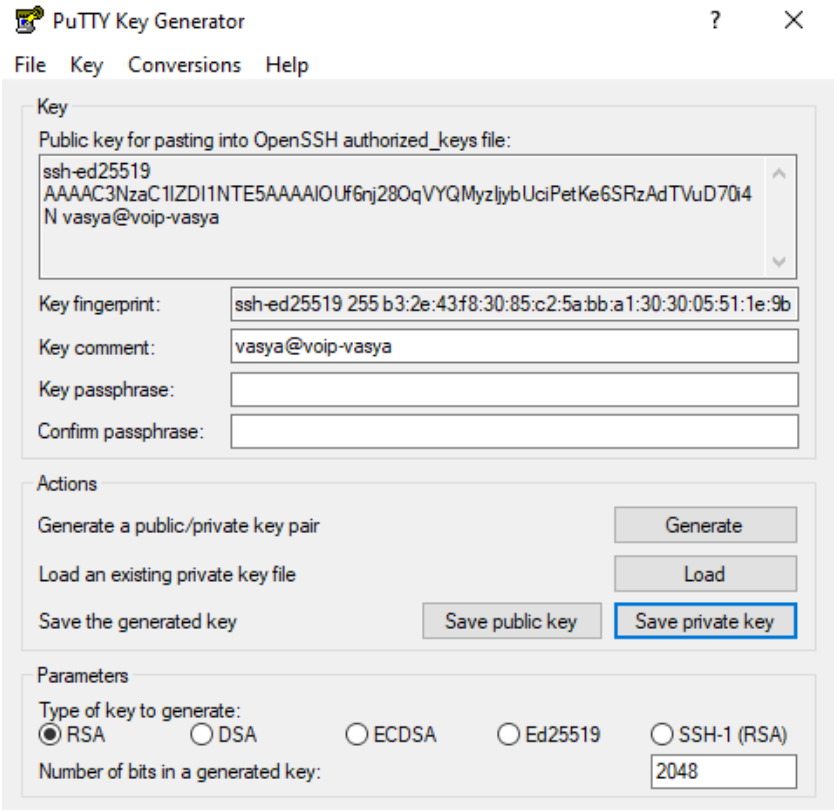
```
chmod 700 ~/.ssh
chmod 700 ~/.ssh/authorized_keys
```

Перезапустите демон OpenSSH

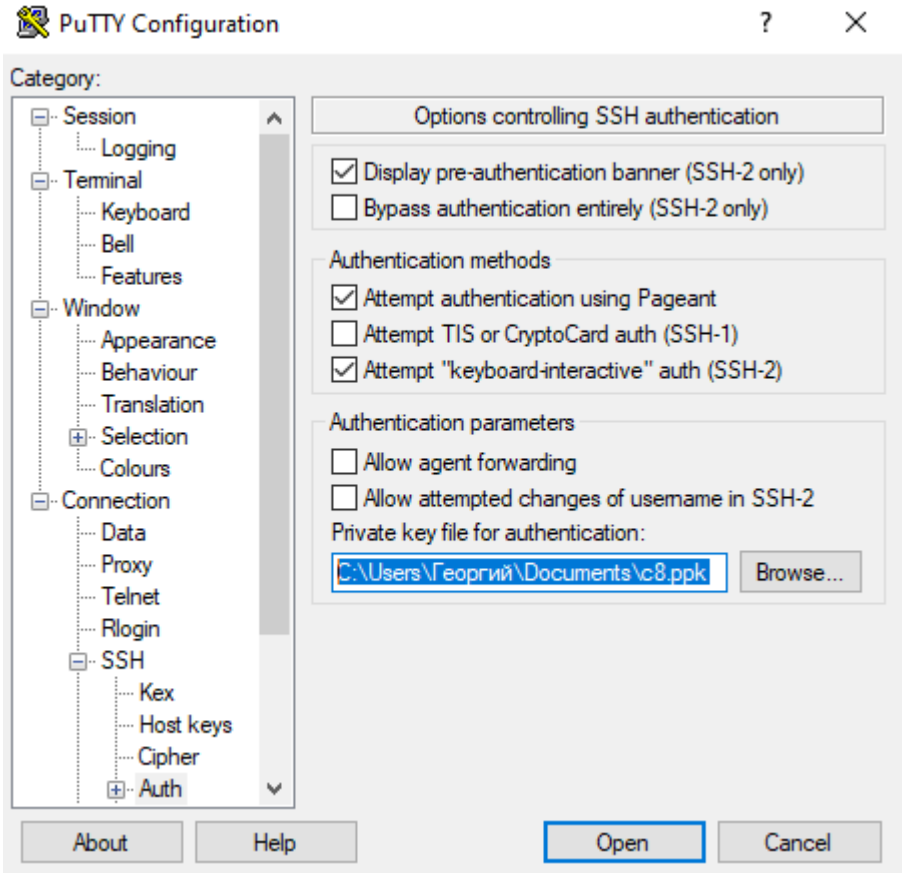
```
sudo systemctl restart sshd
```

Для следующего шага потребуется перенести файл приватного ключа на компьютер, с которого осуществляются подключения. Можно просто скопировать все содержимое из вывода файла приватного ключа в новый файл, но в рамках данной работы будет полезно показать, как можно передавать файлы по сети. Чтобы сделать это, воспользуемся протоколом SCP и непосредственно программой WinSCP в качестве клиента.

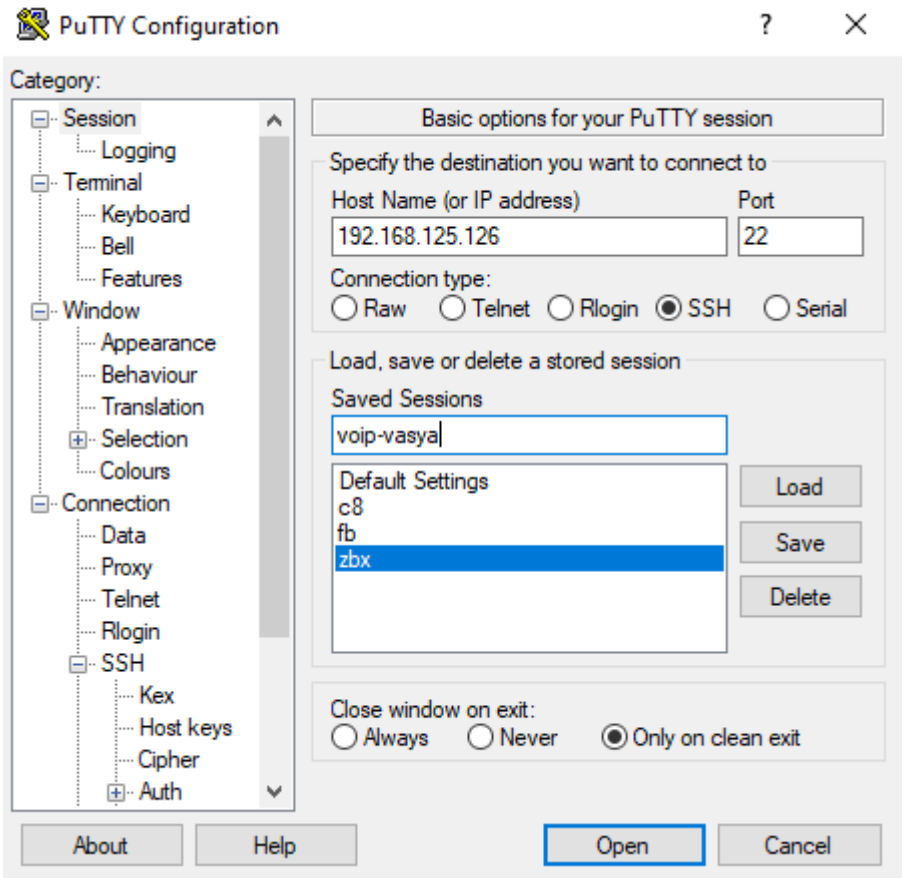
Загрузите на свой компьютер файл `id_ed25519` из `.ssh/`. Формат файлов ключей, используемый OpenSSH не подходит для PuTTY, поэтому сначала полученный приватный ключ придется сконвертировать в используемый PuTTY формат ppk. Запустите программу PuTTYgen, выберите Conversions - Import key, укажите файл с приватным ключом, нажмите Save private key, согласитесь на сохранение приватного ключа без парольной фразы, укажите размещение файла с приватным ключом в формате ppk, который будет использовать PuTTy для аутентификации.



Откройте новое окно PuTTY, слева в пункте Connections разверните подпункт SSH, в нем выберите подпункт Auth и в поле Private key for authentication укажите путь к сконвертированному файлу приватного ключа в формате ppk.



Перейдите в пункт Session, введите IP адрес CentOS, укажите протокол SSH, в поле Saved Sessions укажите имя, под которым хотите сохранить это подключение, нажмите Save.



Теперь вы можете подключаться к серверам, используя заранее настроенные параметры сессии, двойным щелчком мыши по соответствующей сохраненной сессии. Попробуйте подключиться к CentOS, если все было сделано правильно, после ввода имени пользователя появится сообщение об аутентификации с публичным ключом "комментарий вашего ключа" и вы получите доступ к системе без запроса пароля. Этот же файл закрытого ключа можно использовать и в WinSCP.