Министерство образования и науки Республики Марий Эл
Государственное бюджетное профессиональное образовательное
учреждение Республики Марий Эл
«Марийский радиомеханический техникум»

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

ПП.02.01.09.02.06-00.17

ПМ.02 Организация сетевого администрирования

Выполнил
студент группы КС-41
_____Свинцов А.А.

Руководитель практики
_____ Глозштейн А. М.

Руководитель практики
от предприятия
_____Спиридонова М.С.

Йошкар-Ола
2023

ГБПОУ Республики Марий Эл

«Марийский радиомеханический техникум»

«УТВЕРЖДАЮ»

Зам. директора по УМР

_____Бурханова И.Ю.

«__»_____20__г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

На период производственной практики с «*19*» *01 2023г. по «22» 02 2023г.*

студента группы *КС-41* специальности *09.02.06 Сетевое и системное администрирование*

по *ПМ.02 Организация сетевого администрирования*

Вопросы, подлежащие изучению:

- Ознакомительный этап
- Охрана труда и техники безопасности на рабочем месте
- Описание рабочего места
- Для чего нужны SSH туннели
- Настройка организации туннелей в Windows
- Переброс удаленного порта на локальную машину (Remote TCP forwarding)
- Реализация SSH туннелей
- Работа с программой PuTTY

Руководители практики:

от предприятия	Спиридонова Мария Сергеевна, Исполнительный директор
	(фамилия, имя, отчество, должность)
от техникума	*Глозштейн Александр Моисеевич*
	(фамилия, имя, отчество, должность)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Ситуация, когда требуется срочный доступ к домашнему компьютеру или внутренней корпоративной сети, знакома каждому. К сожалению, качество и безопасность общественных сетей часто оставляют желать лучшего. Важную информацию по таким сетям лучше не передавать. Кроме того, для организации доступа часто требуется внешнее ПО (Team Viewer и другие). Для системного администрирования существует способ использования публичных сетей и при этом создания безопасного подключения к необходимым узлам без использования того же VPN. Речь о SSH-туннелировании.

SSH (от англ. **Secure Shell** — «безопасная оболочка») — сетевой протокол, который используется для удаленного управления ОС и проксировании TCP-соединений. Выполняет шифрование всего трафика (сюда же относятся пароли и другие важные для корпоративной безопасности данные). SSH-серверы работают с большинством сетевых операционных систем, представленных на рынке.

Подключиться по этому протоколу можно практически к любому серверу. Для организации безопасного соединения можно использовать так называемые SSH-туннели. Но с точки зрения терминологии — это не те туннели, о которых обычно идет речь, когда говорят о системном администрировании. Само наименование, SSH tunnel, сформировалось среди сисадминов для простоты обозначения технологии — SSH Port Forwarding (проброса портов). В ней реализовано сразу несколько возможностей сетевого протокола SSH, а именно — передача TCP-пакетов и трансляция IP-заголовка во время передачи информации при условии существования заранее заданного правила.

1. Ознакомительный этап

1.1. Структура организации.

Структура «ЭР Телеком Холдинг» представлена на рисунке 1. Деятельность организации направлена на подключение компании а также частных клиентов к сети Интернет.

Рисунок 1 – Структура организации «ЭР Телеком Холдинг»

1.2. Охрана труда и техника безопасности на рабочем месте.

Акционерное общество «ЭР Телеком Холдинг» (далее Компания) предоставляет услуги широкополосного доступа в Интернет, телефонии, цифрового ТВ, доступа к сетям Wi Fi , VPN, LoRaWAN , видеонаблюдения и комплексных решений на базе технологий промышленного Интернета вещей (IoT), осознает характер и масштабы влияния своей деятельности, продукции и услуг, и понимает свою ответственность за обеспечение безаварийной производственной деятельности, безопасных условий труда работников и сохранность здоровья населения, проживающего в районах деятельности Компании.

Компания при осуществлении всех видов деятельности признает приоритет жизни и здоровья работников по отношению к результату производственной деятельности.

В цели Компании входит предотвращение производственных травм и ухудшения здоровья работников, обеспечение безопасных условий с учетом безопасности труда и охраны здоровья для снижения негативного влияния на нее в процессе деятельности Компании.

Высшее руководство Компании рассматривает систему управления охраной труда и промышленной безопасностью в качестве необходимого элемента управления производственными рисками, воздействующими на здоровье работников, оборудование и имущество, и берет на себя следующие обязательства:

обеспечивать безопасные рабочие условия труда и охраны здоровья для предотвращения травматизма и нанесения вреда здоровью работников Компании в процессе их трудовой деятельности:

- проходить обучение безопасным способам и приемам выполнения работ, и оказанию первой помощи пострадавшим на производстве, инструктаж и проверку знаний по охране труда, стажировку на рабочем месте;

- немедленно извещать своего непосредственного или вышестоящего руководителя о любой ситуации угрожающей жизни и здоровью людей, о каждом несчастном случае, происшедшем на производстве, или ухудшении состояния своего здоровья;

- проходить обязательные предварительные (при поступлении на работу), периодические (в течение трудовой деятельности) внеочередные (по направлению работодателя) медицинские осмотры.

Основными направлениями функционирования системы охраны труда медицинской организации, как правило, являются:

- управление охраной труда;

- обучение и профессиональная подготовка работников в области охраны труда;

- предупреждение несчастных случаев и нарушения здоровья работников;

- контроль соблюдения нормативных требований охраны труда.

1.3. Описание рабочего места.

В помещении, где находилось мое рабочее место, стояли три стола, на 2 из них сидят инженеры в том числе и главный инженер. Рядом со мной располагалось три сервера организации и оборудование для сетевого администрирования. Основные задачи, которые мне полагалось выполнять:

- обеспечение информационной безопасности
- устранение неполадок в системах или компьютерах
- установка нового аппаратного оборудования и программ
- обновление системы и компьютерных программ

- создание резервных копий данных
- периодическое обновление
- создание пользовательских учётных данных
- мелкий ремонт периферийного оборудования

1.4. Виды туннелей Windows.

Прямой SSH-туннель

Прямой ssh-туннель применяется для того, чтобы перенаправить соединение с заданным TCP-портом на локальном (клиентском) хосте на порт удаленного хоста (сервера). Всякий раз, когда устанавливается соединение с локальным портом или сокетом, соединение пересылается по защищенному каналу, и с удаленного компьютера устанавливается соединение либо с хост-портом, либо с сокетом Unix.

Обратный SSH-туннель

Обратный SSH-туннель применяется для того, чтобы на удаленном хосте (ssh-сервере) открыть сокет и перенаправлять соединения, устанавливаемые с этим сокетом, на порт локального хоста (ssh-клиента).Вся полученная при сканировании информация может быть выведена в отчёты, которые легко экспортируются в любой удобный для вас формат.

Динамический SSH-туннель

Динамические SSH-туннели создают соединение, работающее как SOCKS-прокси. На локальном хосте создается динамический сокет, который превращает ваш компьютер в прокси сервер, переадресующий весь трафик на удаленный ssh-сервер.

1.5. Secure Shell – безопасная передача данных.

Secure Shell состоит из зашифрованного туннеля, созданного через соединение протокола SSH. Пользователи могут настроить туннели SSH для передачи незашифрованного трафика по сети через зашифрованный канал. Например, компьютеры Microsoft Windows могут обмениваться файлами с использованием протокола сервера сообщений (SMB), незашифрованного протокола. Если вы хотите удаленно подключить файловую систему Microsoft Windows через Интернет, кто-то, следящий за соединением, может видеть переданные файлы. Чтобы безопасно подключить файловую систему Windows, можно установить туннель SSH, который направляет весь SMB-трафик на удаленный файловый сервер через зашифрованный канал. Несмотря на то что сам протокол SMB не содержит шифрования, зашифрованный SSH-канал, через который он перемещается, обеспечивает безопасность.

1.6. Типы переадресации портов.

Переадресация портов — это широко поддерживаемая функция, обнаруживаемая во всех основных клиентах и серверах SSH. С функцией перенаправления портов SSH можно осуществлять передачу различных типов интернет-трафика через сеть. Это используется для того, чтобы избежать сетевой слежки или с целью обхода неправильно настроенных маршрутизаторов в Интернете.

Существует три типа переадресации портов с SSH:

- локальная — соединения с SSH-клиента перенаправляются на SSH-сервер, а затем на целевой сервер.
- удаленная — соединения с SSH-сервера перенаправляются через SSH-клиент, а затем на целевой сервер.
- динамическая — соединения из различных программ пересылаются через SSH-клиент, затем через SSH-сервер и, наконец, на несколько целевых серверов.

Локальная переадресация портов является наиболее распространенным типом и, в частности, позволяет обойти брандмауэр компании, который блокирует "Википедию". Удаленная переадресация портов встречается реже. Позволяет подключиться с вашего SSH-сервера к компьютеру в интрасети вашей компании. Динамическая переадресация портов используется также нечасто. Позволяет обойти брандмауэр компании, который полностью блокирует доступ к Интернету. Требует много работы для настройки, и обычно проще использовать локальную переадресацию портов для определенных сайтов, к которым вы хотите получить доступ.

Чтобы использовать переадресацию портов, вам необходимо убедиться, что перенаправление портов включено на вашем сервере. Вы также должны указать своему клиенту номера портов источника и назначения. Если вы используете локальную или удаленную переадресацию, вам нужно сообщить клиенту конечный сервер. Если применяется динамическая переадресация портов, потребуется настроить свои программы на использование прокси-сервера SOCKS. Опять же, то, как это сделать, зависит от того, какой клиент SSH вы используете, поэтому может потребоваться более детально ознакомиться с документацией.

2. Практическая часть.

2.1. Защищенный доступ к RDP через SSH туннель

В этом режиме вы создаете на своем компьютере локальный TCP порт, подключения к которому перенаправляются через SSH туннель на указанный порт удаленного сервера. В этом примере мы создадим локальный порт 8888,

при подключении к которому выполняется перенаправление на RDP порт 3389 на удаленном компьютере. Общая схема подключения выглядит так:

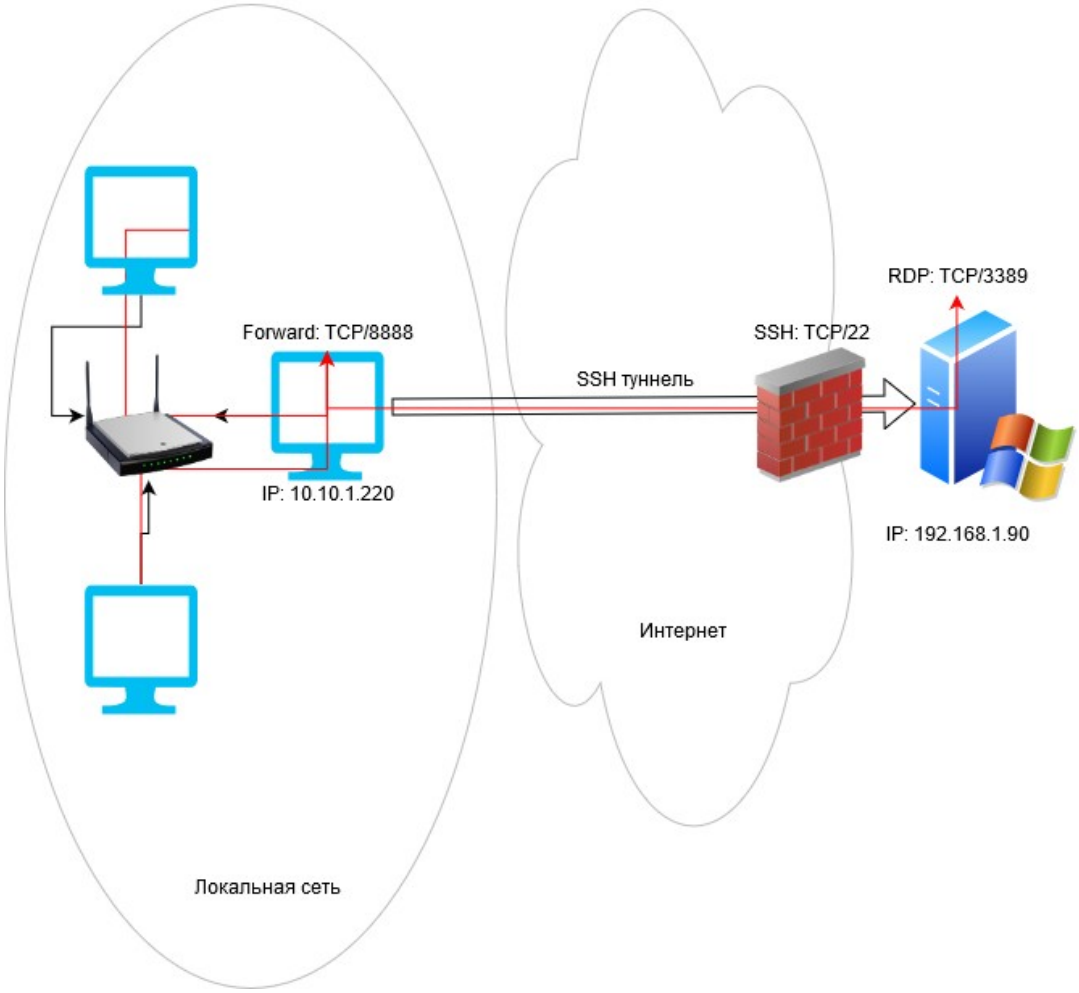

Рисунок 2 – схема подключения

Для проброса портов нам потребуется SSH клиент. Можно использовать сторонний клиент (например, Putty), но я буду использовать встроенный SSH клиент в Windows. Чтобы установить клиенте OpenSSH, выполните в консоли PowerShell команду:

Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client*'

Чтобы создать SSH туннель с удаленным компьютером 192.168.1.90, выполните команду:

ssh -L 8888:192.168.1.90:3389 root@192.168.1.90

В этом примере используется формат LOCAL_PORT:DESTINATION:DESTINATION_PORT и USER@DEST_SERVER_IP (имя пользователя и адрес удаленного SSH сервера)

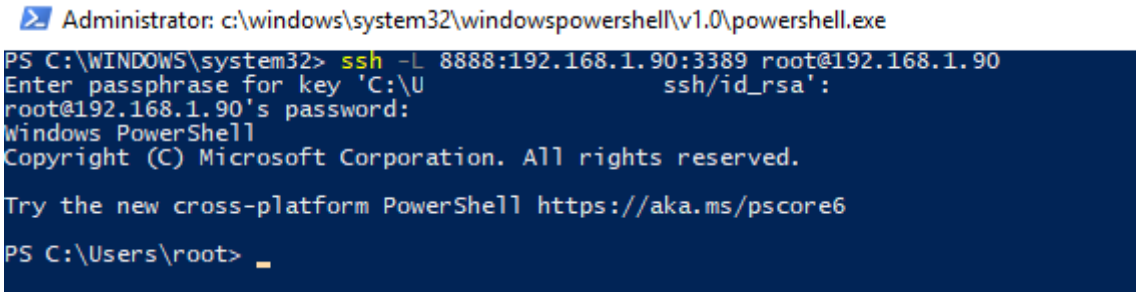

Рисунок 3 – настройка туннеля

Теперь, чтобы подключится к удаленному компьютеру по RDP через SSH туннель, вам запустить RDP-клиент (mstsc.exe) и подключиться на локальный порт 8888 своего компьютера:

127.0.0.1:8888

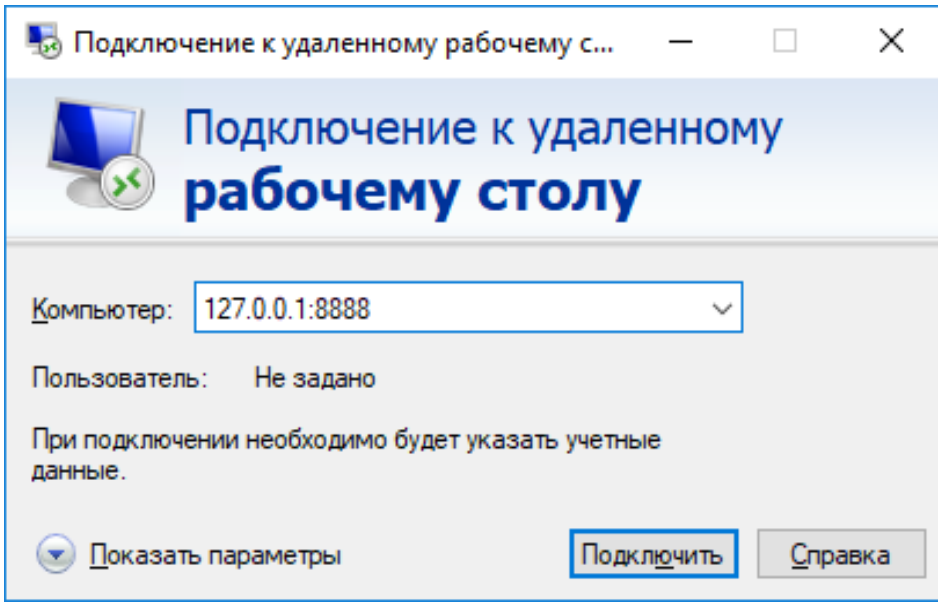

Рисунок 4 – подключение к удаленному компьютеру

Выполните аутентификацию на удаленном компьютере и можете спокойно работать в RDP-сессии. С помощью команды Get-NetTCPConnection или утилиты TCPView вы можете убедиться, что RDP

подключение установлено локально (RDP подключение инициировано запущенным локально SSH сервером):

```
Get-NetTCPConnection -State Established|where {$_.localport -eq "3389"}|fl
```

Защита RDP подключения с помощью SSH туннеля может быть хорошей альтернативой VPN для доступа к публичным Windows хостам в Интернете. В этом случае вам не нужно открывать прямой доступ к порту RDP/3389 из Интернета к хосту Windows. Достаточно открыть только порт SSH/22, что защитит вас от атак подбора пароля по RDP и эксплуатации 0-day RDP уязвимостей.

2.2. SSH туннель в Windows с помощью PuTTY

Рассмотрим, как настроить SSH туннель в Windows с помощью популярного SSH клиента Putty.

1. Запустите PuTTY и перейдите в раздел Connection -> SSH -> Tunnels;
2. В поле Source port укажите локального порта (в нашем примере это 8888);
3. Укажите IP адрес сервера SSH и порт на удаленном хосте, на который нужно выполнить переадресацию: 192.168.31.90:3389

4. Выберите Local в качестве Destination и нажмите Add;

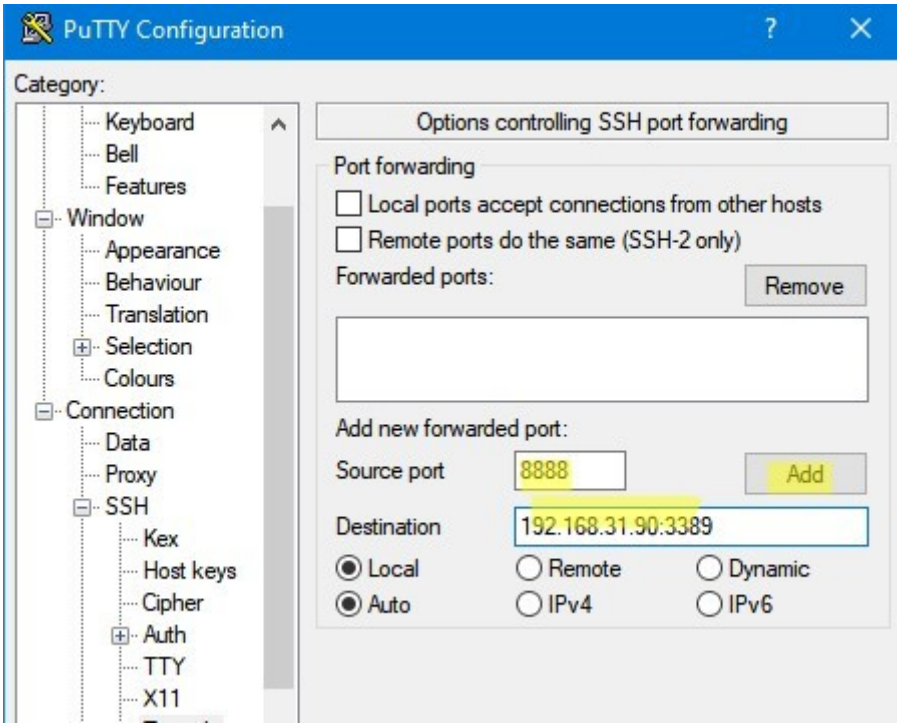

Рисунок 5 – настройка PuTTY.

5. Чтобы не открывать shell удаленного хоста при подключении через туннель, включите опцию Don’t start a shell or command at all в разделе SSH;

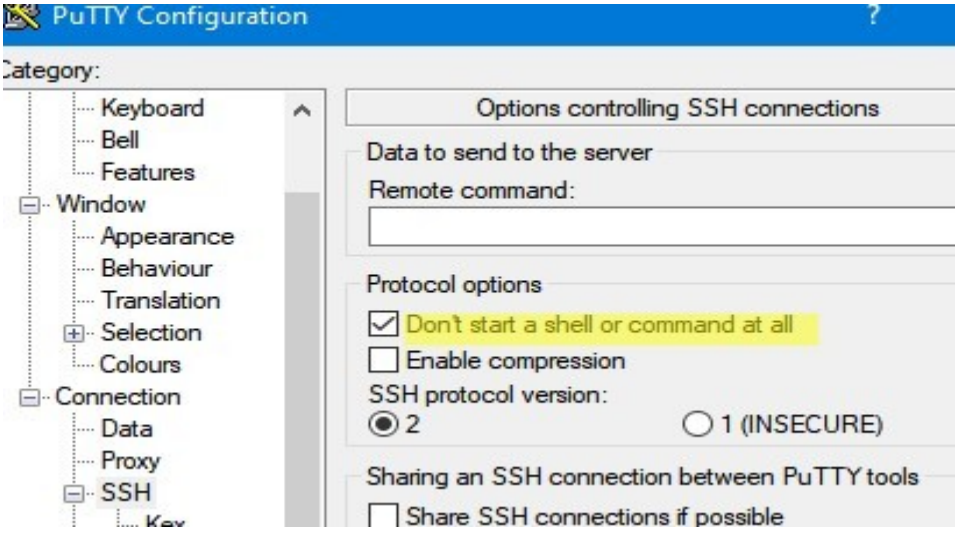

Рисунок 6 – включение опции Don’t start a shell or command at all

6. Вернитесь на вкладку Session, укажите имя или IP адрес удаленного SSH хоста и порт подключения (по умолчанию порт 22). Чтобы сохранить настройки подключения, укажите имя сессии в поле Saved Session и нажмите Save;

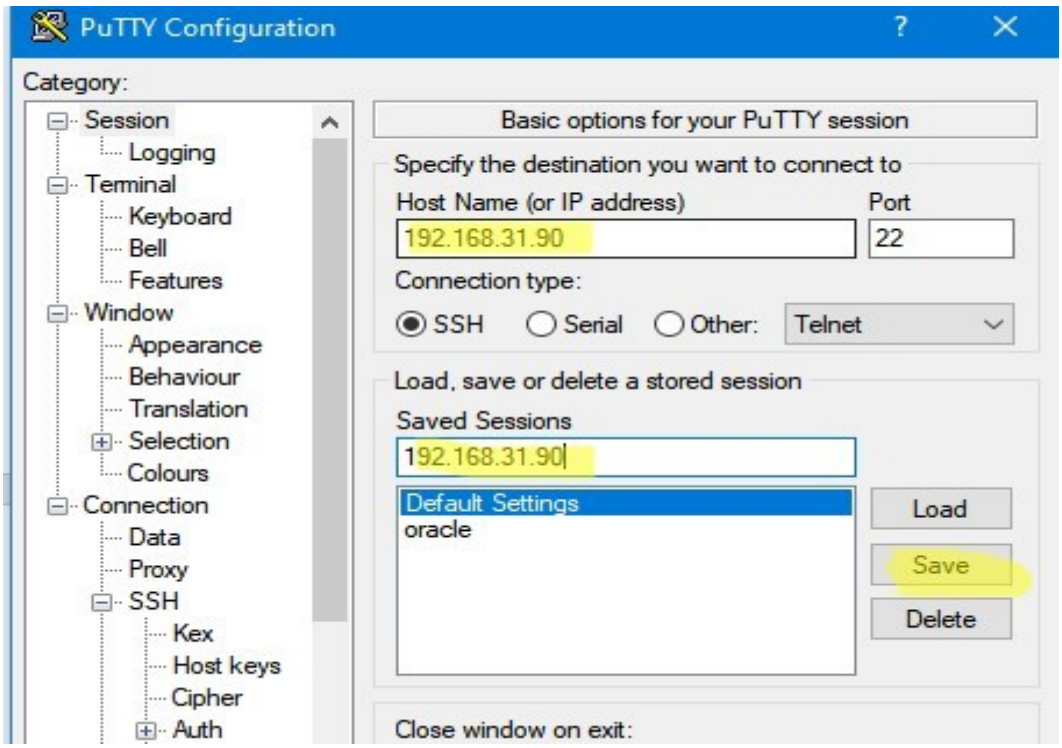

Рисунок 7 – настройка IP адреса.

7. Теперь вы можете запустить сохраненный SSH туннель в Putty прямо и панели задач Windows.

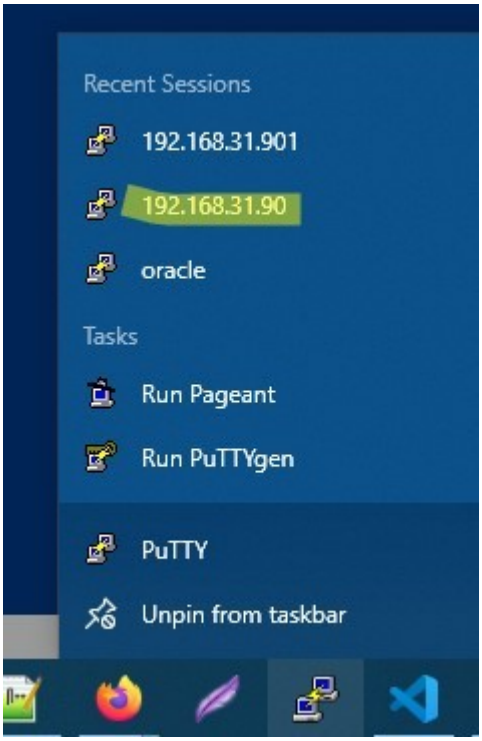

Рисунок 8 – панель задач.

2.3. Переброс удаленного порта на локальную машину.

Есть еще один вариант применения SSH туннеля – remote TCP forwarding. Через SSH туннель вы можете открыть доступ удаленному серверу к локальному порту на вашем компьютере или порту на другом компьютере в вашей локальной сети. Например, вы хотите, чтобы внешний сервер (192.168.1.90) получил доступ к вашему Интранет сайту (не опубликованному в Интернете). Для создания обратного туннеля, используйте такую команду:

ssh -R 8080:internalwebsever:80 user@192.168.1.90

Теперь, чтобы на удаленном SSH сервер получить доступ к веб серверу internalwebsever достаточно в браузере набрать адрес http://localhost:8080.

С помощью SSH туннелей вы можете строить целые цепочки для форвардинга портов. Включить или отключить SSH туннелирование можно в конфигурационном файле OpenSSH (sshd_config) с помощью директив:

AllowStreamLocalForwarding yes

AllowTcpForwarding remote

PermitTunnel no

2.4. Подключение к серверу без пароля через программу PuTTY.

Не всегда удобно каждый раз вводить пароль для входа на удаленное устройство. Комфортнее, когда вход выполняется автоматически и без лишних усилий. В PuTTY существует такой способ – использование авторизации по ключу SSH. Настроить его можно следующим образом:

1. Для выполнения данной операции нам потребуется отдельная утилита под названием «PuTTYgen». Она автоматически устанавливается вместе с PuTTY, поэтому зайдем в меню «Пуск» и запустим ее оттуда.
2. Далее перемещаемся в меню «Key» и устанавливаем значение «SSH-2 RSA key». После этого кликаем по кнопке «Generate key pair».
3. Как только ключ будет сгенерирован, сохраним его в публичном и приватном варианте.

4. Теперь перемещаемся в PuTTY и открываем раздел «Auth» — в него добавляем приватный ключ.
5. Открываем раздел «Session» и сохраняем в нем пресет для подключения, как мы делали ранее. Затем подключаемся к серверу по своему адресу.
6. Осталось подключиться к серверу и отправить на него открытый ключ. Для этого вставляем его в конец файла /root/.ssh/authorized_keys. Ключ мы можем скопировать напрямую из утилиты, в которой он был сгенерирован.

Теперь при повторном входе на сервер авторизация не потребуется. Главное – сохранить настройки сессии, чтобы не вводить ключ каждый раз.

2.5. SSH-туннели на службе системного администратора.

Любому системному администратору приходится постоянно работать удаленно, но случаются ситуации, когда нужно срочно подключиться к узлам внутренней сети, доступ к которым снаружи закрыт. Хорошо если есть доступ на другие узлы данной сети, но бывает, что доступа из глобальной сети нет вообще, в этих случаях обычно используют TeamViewer или аналогичное ПО, но если к такой сети есть возможность подключиться через SSH или установить соединение с промежуточным SSH-сервером, то можно быстро и просто организовать доступ без привлечения стороннего ПО.

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на углубленном курсе по администрированию MikroTik. Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.

Очень часто, когда речь заходит об удаленном доступе к сетям с низким уровнем безопасности, первое что приходит на ум — это VPN, однако ради административных соединений поднимать VPN не всегда целесообразно,

особенно если речь идет об аутсорсинге или «приходящем админе». Кроме того, не всегда условия связи позволяют поднять устойчивое VPN-соединение, особенно если приходится использовать мобильные сети.

При этом практически в любой сети можно найти устройство или сервер, к которому возможен доступ по SSH, либо имеется такой промежуточный сервер, например, VPS в глобальной сети. В таком случае отличным решением будут SSH-туннели, которые позволяют с легкостью организовывать безопасные каналы связи в том числе и через промежуточные узлы, что снимает проблему наличия выделенного IP-адреса.

Строго говоря, SSH-туннели не являются полноценными туннелями и это название следует рассматривать как сложившееся в профессиональной среде устойчивое наименование. Официальное название технологии — SSH Port Forwarding — это опциональная возможность протокола SSH, которая позволяет передать TCP-пакет с одной стороны SSH-соединения на другую и произвести в процессе передачи трансляцию IP-заголовка по заранее определенному правилу.

Также, в отличие от VPN-туннелей, которые позволяют передавать любой трафик в любом направлении, SSH-туннель имеет точку входа и может работать только с TCP-пакетами. По факту это больше всего похоже на проброс портов (о чем и говорит официальное название), только поверх протокола SSH.

Рассмотрим работу SSH-туннеля более подробно. В качестве примера возьмем классический случай обеспечения доступа к некоторому удаленному серверу по протоколу RDP.

Допустим, что в удаленной сети существует целевой сервер с адресом 192.168.0.105, но доступа к нему из внешней сети нет, единственное устройство к которому мы можем подключиться — это маршрутизатор с адресом 192.168.0.1 с которым мы можем установить SSH-соединение.

Остановимся на очень важном моменте: все параметры SSH-туннеля устанавливает инициатор подключения, он же SSH-клиент, вторым концом туннеля всегда является сервер, к которому мы подключаемся, он же SSH-сервер. В данном контексте следует понимать клиент и сервер сугубо как стороны соединения, например, в роли SSH-клиента может выступать VPS-сервер, а в роли SSH-сервера ноутбук админа.

Точка входа может располагаться с любой стороны соединения, там открывается TCP-сокет с указанными параметрами, который будет принимать входящие подключения. Точка выхода принимать соединения не может, а только маршрутизирует пакеты в соответствии с правилами трансляции.

Рассмотрим схему выше. Мы установили SSH-туннель с локальной машины к удаленному маршрутизатору, указав локальную точку входа 127.0.0.1:3389 и правило трансляции 192.168.0.105:3389. Еще раз обращаем ваше внимание, что правило трансляции не указывает на точку выхода, а определяет узел, которому будут посланы пакеты по выходу из туннеля. Если вы укажете недействительный адрес, либо узел не будет принимать соединения, то SSH-туннель установится, но доступа к целевому узлу не будет.

Согласно указанной точке входа служба SSH создаст локальный TCP-сокет, который будет ожидать подключений на порт 3389. Поэтому в окне RDP-подключения указываем в качестве назначения localhost или 127.0.0.1, RDP-клиент открывает динамический порт и отсылает пакет с адресом назначения 127.0.0.1:3389 и адресом источника 127.0.0.1:61256, о реальном назначении получателя пакета ему ничего не известно.

С точки входа данный пакет будет отправлен на другую сторону SSH-туннеля, а адрес назначения согласно правила трансляции будет изменен на 192.168.0.105:3389, и SSH-сервер примет дальнейшее решение согласно собственной таблицы маршрутизации, заменив адрес источника на свой собственный, в противном случае целевой сервер попытается отправить ответный пакет на локальный адрес.

Таким образом RDP-клиент работает с локальным сокетом и далее этого узла RDP-пакеты не уходят, внутри туннеля они передаются поверх протокола SSH в зашифрованном виде, а вот между SSH-сервером и RDP-сервером в сети 192.168.0.0 устанавливается обычное RDP-соединение, в открытом виде. Это следует учитывать при использовании небезопасных протоколов, твердо запомнив, что если правило трансляции указывает за пределы узла с точкой выхода, то такое соединение (между точкой выхода и узлом назначения) не защищается посредством SSH.

Разобравшись в общих чертах как работают SSH-туннели перейдем к практическим вариантам их использования, в качестве платформы мы будем рассматривать Linux-системы семейства Debian/Ubuntu, но все нижеизложенное с небольшими поправками будет справедливо для любой UNIX-подобной системы.

2.6. SSH-туннель с локальной точкой входа.

Туннели с локальной точкой входа используются для получения доступа к узлам в удаленной сети при возможности установить SSH-соединение с одним из ее узлов, что предполагает наличие у удаленной сети выделенного IP-адреса.

Мы будем рассматривать все тот-же вариант, RDP-подключение к удаленному серверу, оранжевым пунктиром на схеме обозначено безопасное SSH-соединение, синими стрелками — обычное TCP-подключение.

В самом простом варианте мы просто устанавливаем соединение с клиентского ПК к маршрутизатору в удаленной сети, указывая в правиле трансляции целевой узел:

Ключ -L указывает на то, что точка входа расположена локально, затем через двоеточие указываются адрес и порт точки входа и адрес, порт правила трансляции. Точкой выхода является узел, к которому мы подключаемся, т.е. rt.example.com

2.7. SSH-туннель с удаленной точкой входа.

Туннель с удаленной точкой входа позволяет наоборот опубликовать любую локальную службу в удаленной сети, одно из наиболее частых применений — доступ в сети без выделенного IP-адреса, однако это требует «белый» IP со стороны сети администратора.

В первом варианте удаленный сервер сам устанавливает подключение с маршрутизатором локальной сети. Это можно сделать простой командой:

Ключ -R указывает открыть точку доступа с удаленной стороны туннеля, затем указываем порт для TCP-сокета и трансляцию, так как приходящие на точку выхода пакеты следует обрабатывать локально, то также указываем локальный интерфейс.

Внимательный читатель заметит, что мы не указали ключ **-g**, да, это так. Дело в том, что для туннелей с удаленной точкой входа данный ключ неприменим и следует использовать опцию

на стороне SSH-сервера.

В целях безопасности мы рекомендуем применять данную настройку с политикой по-умолчанию DROP для цепочки INPUT, это позволит избежать случайной публикации на внешнем интерфейсе внутренних служб и ресурсов. В минимальной конфигурации следует добавить в самое начало цепочки INPUT четыре правила:

Первое из них задает запрещающую политику по умолчанию для входящих пакетов, второе разрешает входящие пакеты, инициированные самим хостом (ответы на исходящие пакеты), а третье разрешает все подключения из локальной сети. Наконец четвертое правило открывает 22 порт для входящих SSH-подключений, таким же образом можно открыть любой другой порт для внешних подключений.

Второй вариант представленный на схеме предусматривает, что SSH-туннель поднимают маршрутизаторы сетей, в этом случае команда будет выглядеть следующим образом:

Снова обратим ваше внимание, что точка входа у нас располагается на противоположной стороне туннеля, поэтому трансляция указывается для стороны инициатора туннеля, т.е. в данном случае SSH-клиент устанавливает два соединения: одно SSH с rt.example.com, а второе RDP с 192.168.0.105, тогда как при туннеле с локальной точкой входа инициатор устанавливает единственное соединение с SSH-сервером.

2.8. Организация туннеля и VPN при помощи OpenSSH.

Если вы не хотите искать легких путей, тогда можете воспользоваться стандартной программой OpenSSH для организация туннеля загрузок по сетевому протоколу SSH. Этот способ хорошо подойдет тем, у кого установлена операционная система Ubuntu или Linux. Для начала вам необходимо инсталлировать OpenSSH. Для этого в консоле введите следующую команду: sudo aptitude install openssh-server.

Дело в том, что любой туннель, обратный, прямой и даже многоканальный, можно создать при помощи стандартных возможностей OpenSSH. Но для этого нужно разбираться в возможностях этого приложения и уметь настраивать его конфигурации. Для создания туннеля, вам нужно как минимум дать добро на туннелирование внутри файла config, который определяет настройки SSH протокола. Чтобы разрешить туннелирование, введите следующую строку в файл: PermitTunnel point-to-point. После этого вам нужно будет перезагрузить программу-сервер OpenSSH. Для этого введите следующую команду: service ssh restart.

Сразу учтите, что есть одно большое но в подобной организации туннеля. И заключается оно в том, что для подключения туннеля, вы обязаны будете зайти на хост через аккаунт суперадминистратора. А как известно, это грубое нарушение правил безопасности SSH протокола. И хоть в настройках по умолчанию root-пользователь активирован, это совсем не безопасно. Если пароль будет украден, то вы лишитесь всего — сайт буквально ограбят и

выпотрошат. Потому либо делайте очень сложный шифрованный пароль, либо активируйте аутентификацию посредством публичных ключей.

После того, как вы зайдете в root-пользователя, вы сможете создать туннель посредством командной строки. Вам нужно будет прописать команду через sudo или при помощи root-a. А прописать нужно будет действие вида -w локальный_туннель:обратный_туннель. Вместо локального и обратного туннеля укажите цифры. Можно прописать два нуля — тогда будет создан туннель tun0 и для сервера, и для клиента.

Следующим шагом нужно настроить два туннеля, чтобы они могли передавать данные между собой. Вот пример настройки туннелей: для серверного туннеля — ifconfig tun0 10.0.0.1/30 pointopoint 10.0.0.2, и для клиентского — ifconfig tun0 10.0.0.2/30 pointopoint 10.0.0.1.

Но на этом еще не все. Чтобы создать автоматическую загрузку данных через туннель, его нужно указать в настройках, как шлюз по умолчанию. Но в таком случае потеряется путь к DNS и серверу. Потому текущие шлюзы нужно прописать в таблице маршрутизации через команду route add -host XX.XX.XX.XX gw ЧЧ.ЧЧ.ЧЧ.ЧЧ (XX — это IP DNS в одной строке, и IP сервера в другой; а ЧЧ — это в обоих командах IP текущего шлюза, который нужно удалить).

Теперь удаляем текущий шлюз и добавляем новый. Удаляем при помощи строки route del default. А прописываем новый при помощи аналогичной функции: route add default gw 10.0.0.1 (в вашем случае IP может быть другим, смотря что указывали при создании туннеля).

После определения таких настроек, все, что идет не по стандартным каналам, автоматически перенаправляется на защищенный сетевой протокол по туннелю. То есть таким образом был создан автоматический туннель, пропускающий через себя все данные и трафик. Но остается еще одна проблема — на сервер трафик попадает, но дальше с ним ничего не происходит. А все потом, что необходимо настроить трансляцию сетевых адресов NAT для SSH

клиента. Чтобы это реализовать нужно добавить новое правило: iptables -t nat -A POSTROUTING -s 10.0.0.2 -j MASQUERADE. Теперь осталось всего лишь включить ip-форвардинг через ядро и настроить его активацию каждый раз при запуске. После этого туннелирование можно считать успешным! Как видите, с OpenSSH все гораздо сложнее, но тем не менее, если постараться, то все реально.

2.9. Какие еще есть способы создания прямого и обратного туннеля через SSH.

Описанные методы решения данного вопроса — это не весь перечень средств, которые есть у программиста по умолчанию. Через все тот же Linux можно создать туннель несколькими способами. И сейчас мы разберем очень простой метод, как подключить компьютер к удаленному серверу, а также как настроить подключение другого компьютеру, подключенному в SSHD.

Итак, допустим вам нужно подключиться к интернет-сервису, размещенному по адресу 10.10.2.1:80. При этом хост работает через определенный домен, к примеру, site.ru. Все, что вам нужно сделать — это пробросить туннель через сервер к нужному IP-адресу. И делается это при помощи команду -f, -N и -L. Первая объясняет протоколу, что нужно будет уйти в background после активации соединения, вторая отменяет все последующие команды, третья перенаправляет соединения на определенный хост к нужному нам IP-адресу. И вот как будет звучать команда: ssh -f -N user@site.ru -L 8080:10.10.2.1:80.

Как видите, это достаточно простое решение. Какой метод выберите вы — это зависит от поставленных задач и ваших навыков!

На сервере homeserver запустите autossh со следующими аргументами с тем, чтобы создать постоянный туннель SSH, действующий в направлении сервера relayserver .

Homeserver~$ autossh -M 10900 -fN -o "PubkeyAuthentication=yes" -o "StrictHostKeyChecking=false" -o "PasswordAuthentication=no" -o

"ServerAliveInterval 60" -o "ServerAliveCountMax 3" -R 1.1.1.1:10022:localhost:22 relayserver_user@1.1.1.1

Параметр "-M 10900" указывает порт на сервере relayserver , для которого будет осуществляться мониторинг и который будет использоваться для обмена тестовыми данными при контроле сессии SSH. Этот порт не должен на сервере relayserver использоваться какой-либо другой программой.

Параметр "-fN" перенаправляется в команду ssh , что позволит туннелю SSH работать в фоновом режиме.

Параметр "-o XXXX" сообщает команде ssh следующее:

Использовать ключ аутентификации, а не парольную аутентификацию.

Автоматически принимать (неизвестные) ключи хоста SSH

Каждые 60 секунд обмениваться сообщениями keep-alive.

Отправлять до трех сообщений keep-alive без получения каких-либо ответов.

Остальные параметры обратного туннелирования SSH те же самые, что и в предыдущих примерах.

Если вы хотите, чтобы туннель SSH автоматически поднимался при загрузке системы, вы можете в /etc/rc.local добавить указанную выше команду autossh .

Заключение

В этой статье было рассказано о том, как можно использовать обратный туннель SSH для доступа к серверу Windows. Мы показали вам, как настроить туннели SSH и пересылать трафик через безопасное соединение SSH. Для простоты использования вы можете определить туннель SSH в файле конфигурации SSH или создать псевдоним Bash, который будет настраивать туннель SSH. Использование PuTTY позволяет подключаться по протоколу SSH и удаленно работать с компьютером на операционной системе Windows. Такой способ позволяет легко администрировать устройство и всегда быть в курсе возникающих проблем.

Список Использованных Источников

1. https://danplay.ru/android/podrobnyi-analiz-teorii-i-praktiki-ispolzovaniya-ssh-tunnelei.html?ysclid=lfi03b13w4941625963 [Электронный ресурс]
2. https://compsovet.com/ispol-zuyem-ssh-tunnel-windows/?ysclid=lfhzlab0v292829091 [Электронный ресурс]
3. https://fb.ru/article/356772/ssh-tunneli-nastroyka-ispolzovanie?ysclid=lfhy1rgbe3570050631 [Электронный ресурс]
4. https://routerus.com/how-to-setup-ssh-tunneling/?ysclid=lfhy17pqfu394123842 [Электронный ресурс]
5. https://selectel.ru/blog/ssh-tunnels/?ysclid=lfhwuqnpg4965768599 [Электронный ресурс]
6. https://winitpro.ru/index.php/2019/10/29/windows-ssh-tunneling/?ysclid=lfhwmisw8i936674717 [Электронный ресурс]
7. https://timeweb.com/ru/community/articles/vse-o-putty-ustanovka-nastroyka-osnovnye-komandy [Электронный ресурс]