Mason Competitive Cyber

Pivoting and Offensive Networking

Upcoming Competitions

- #cyber-fasttrack -- April 5-7
  - Lots of scholarship $$$ for this one

- #UMDCTF2021 -- April 16-18
  - Historically a good event

Club News

- Guest talk next week from Sounil Yu!

- PatriotCTF 2021 will be held on May 8, 2021!
  - Signup things coming this weekend

Agenda

What is Pivoting?

Pivoting philosophies

SSH tunnels

Proxychains

revsocks

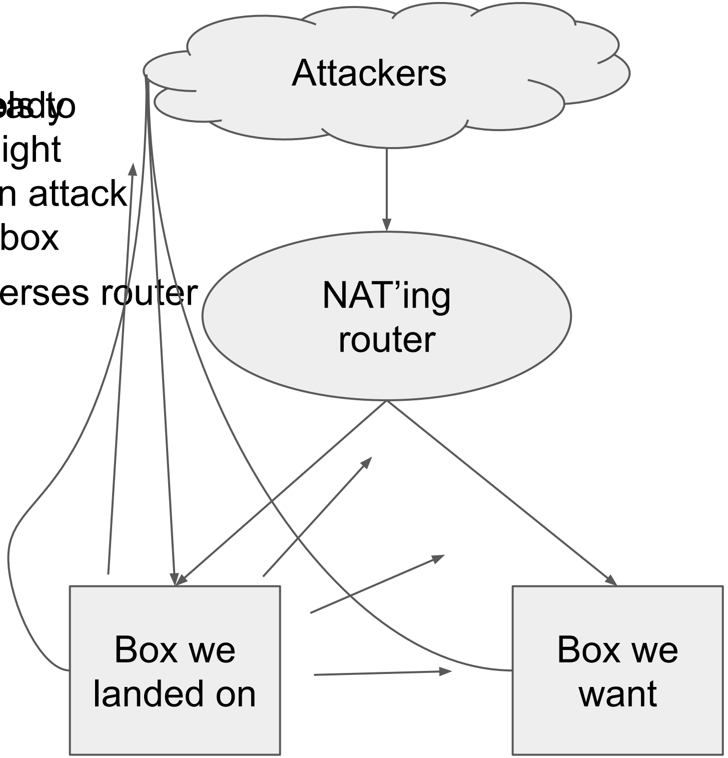
Meterpreter Pivoting

What is pivoting?

...and why should I care?

- Offensive security bread+butter
    - HR Workstation where Doris opened my Word macro ✕
    - NFS server full of credit card numbers ✓

Network POV



Option 1: Use already existing or lightweight tools to network an attack through the initial box

Option 2: Drop tools to initial box

Beacon traverses router thru NAT

Attacker takes control of initial box

Network enumeration

Option 1 looks like:

- Dropping python scripts/powershell scripts/attack environments to the compromised machine
- Creating new users
- In the most extreme case, doing something like pulling metasploit onto your newly compromised box (probably don’t do that)

Pros/Cons to Option 1

Pros:

- Less network hops between exploit and target, might be more reliable
- Generally is pretty straightforward to do

Cons:

- Spending time anchoring in on a less-lucrative system
- Have to set up an environment to run attacks out of
- Logs Logs Logs
- Lots of things potentially dropped to disk

Option 2 looks like:

- Using already present binaries to launch exploits and move laterally
- Dropping small binaries to facilitate networking from attacker into network (sshd, for instance)
- Leave little behind on the initial box

Pros/Cons to Option 2:

Pros:

- Spend less time in HR, more time in the bank vault
- Potentially less artifacts left on the system than Option 1
- Allows attacker to use their already built up attack environment

Cons:

- Can be harder to implement
- Required binaries may be missing, meaning more time troubleshooting
- More hops between attacker and target

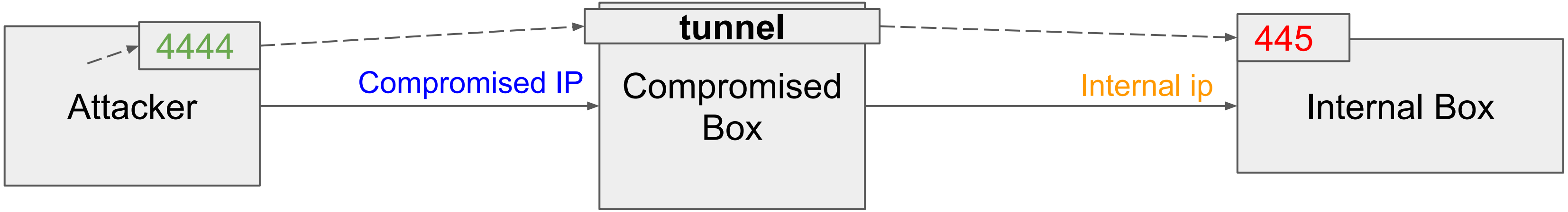
SSH Tunneling

Laying remote tunnels is the best thing ever

-L (standard) tunneling:

Command run on attacker’s box:

```
ssh user@{compromised ip} -L 4444:{internal ip}:445
```



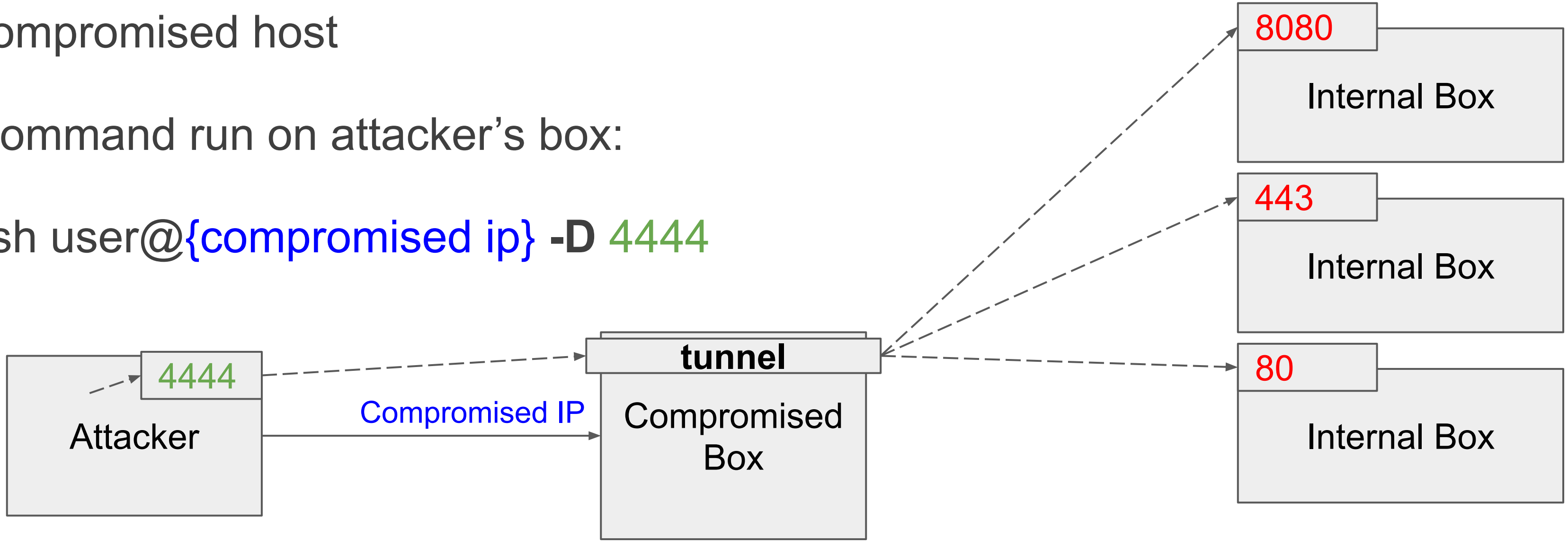
SSH Tunneling

Laying *dynamic* remote tunnels is the best thing ever

-D (Dynamic) tunneling: opens a SOCKS proxy, sending traffic through the compromised host

Command run on attacker's box:

ssh user@{compromised ip} **-D** 4444



Interacting with a -D tunnel

Directly through a web browser:

Configure your browser’s proxy or use an extension (FoxyProxy works well for FF)

Proxying command line tools:

proxychains is your best friend!

Proxychains 101

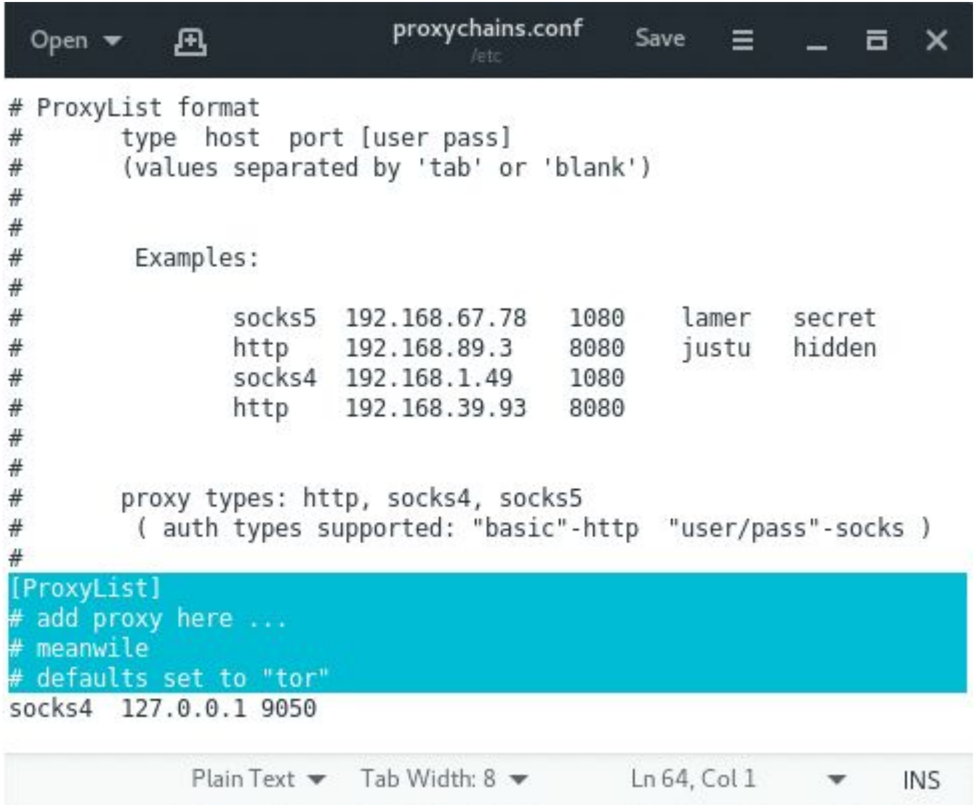
What does it do:

Basically shims a custom networking library before you invoke your program so that network traffic flows via a *chain* of *proxies* (get it?) instead of just flowing straight to the destination

Most linux distros (especially attack distros!) have it in their package managers, so something like an `apt-get install proxychains` should *probably* work

if you want to nmap through proxychains, I’ve had the most luck with proxychains3, which you might have to compile from source!

Sample Proxychains Config



```
# ProxyList format
#       type  host  port [user pass]
#       (values separated by 'tab' or 'blank')
#
#
#        Examples:
#
#               socks5  192.168.67.78   1080    lamer   secret
#               http    192.168.89.3    8080    justu   hidden
#               socks4  192.168.1.49    1080
#               http    192.168.39.93   8080
#
#
#       proxy types: http, socks4, socks5
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4  127.0.0.1 9050
```

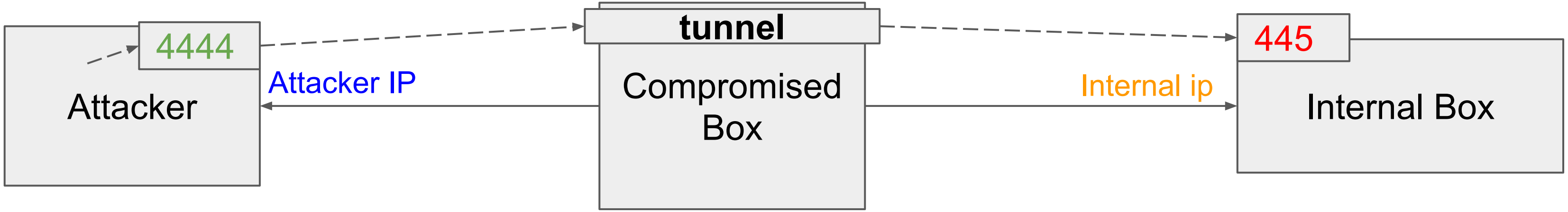
SSH Tunneling

What if you have to get through a firewall or NAT?

-R (reverse) tunneling: tells sshd to open a port on the remote system

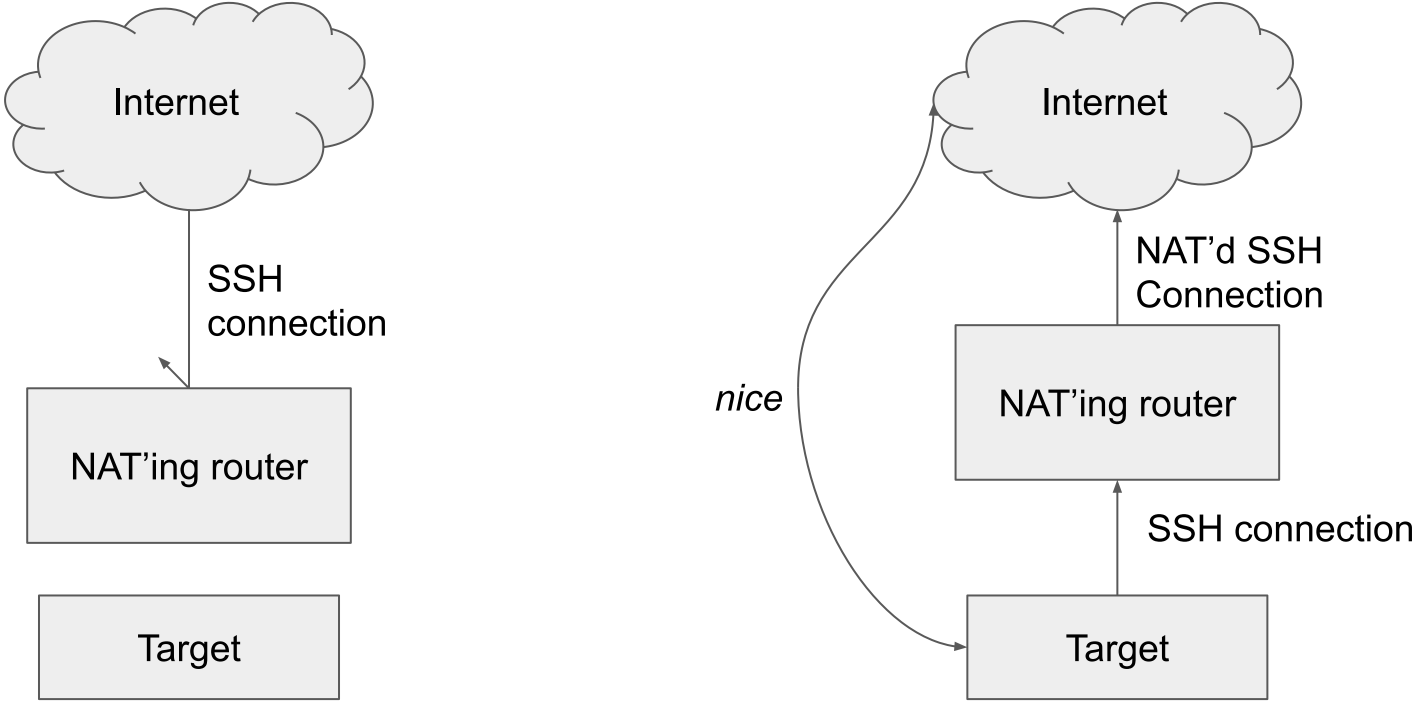
Command run on attacker’s box:

```
ssh user@{attacker ip} -R 4444:{internal ip}:445
```



SSH Tunneling

Why use reverse instead of standard tunneling?



Combining Reverse with -D

SSH doesn’t support this natively, but it’s pretty easy to build a tool (in golang, of course!) to accomplish this!

https://github.com/boba8710/revsocks

Provided for you to play with (don’t do bad things)

Compiling the binaries is an exercise left to the reader (but if you get stuck, I’d be happy to help out!)

Leveraging Meterpreter for Pivoting

Everyone’s favorite free C2, Meterpreter

This is literally just a bad summary of https://www.offensive-security.com/metasploit-unleashed/pivoting/

tl;dr is hook post/multi/manage/autoroute up to your meterpreter session

Leveraging Meterpreter for Pivoting

A live demo of meterpreter tunneling!

Proud Sponsors

CACI

EVER VIGILANT

VOLGENAU SCHOOL

MASON

ENGINEERING