Computer Network Security

COMP 178 | Spring 2025 | University of the Pacific | Jeff Shafer

Penetration Testing: Post-Exploitation

We ran an exploit and have a shell on the target host

Shell vs Terminal

- Normal “legit” remote access (SSH, Telnet) provides a **terminal**
  - Standard input / output, *plus*
  - Character sets
  - Adjustable console window size
  - Color output
  - Redrawing / clearing the screen
  - This is all handled by special control sequences
- Most exploits provide just a shell
  - Standard input / output **only**

Shell vs Terminal

- Troublesome Linux examples requiring a ***terminal***
  - `top` - shows the current CPU usage & list of processes, and keeps redrawing the screen to update
  - Text editors: `vi`, `emacs`, … and text viewers: `more`
  - `sudo` and `su` (the password prompts)
  - `ssh` and `telnet` (the password prompts)
- Additional challenge: Control characters
  - CTRL-C: Will that be passed through to the target application, or will CTRL-C cause your exploit to halt?

Shell vs Terminal

- Do I have a terminal? Or just a shell?
  - Linux: Use `tty` command (“teletype”)
    - Result “`not a tty`”? You just have a shell ☹
    - Result “`/dev/XXXX`”? You have a terminal ☺
- What do I do if I only have a shell?
  - Find substitute commands that do function in shell
  - Investigate alternate access options
    - Start an SSH server / Telnet sever / Remote Desktop server / etc with your own login?
    - Try harder to acquire credentials and log in as a normal user?

Post-Exploit Inspection

Post-Exploit Inspection

- Do the **rules of engagement** allow you to inspect the exploited system for interesting files and data?
- Related questions
  - Can you transfer files *from* the exploited system?
  - Can you transfer files *to* the exploited system? (typically additional pentesting tools)

Post-Exploit Inspection: Network

- Are there machines on the LAN that the target has recently communicated with?
  - *Check ARP cache*

- Are there other local networks that the target has access to?
  - Uncommon for clients
  - A server in a datacenter may have several NICs and access to multiple networks - *exciting discovery!*
  - *Check network configuration and routing table*

Post-Exploit Inspection: Applications

- What applications are installed on the target?
  - *Numerous "software inventory" tools exist, or you could briefly look in the usual installed locations*
- May discover additional data here on a per-application basis
  - DNS server: Zone files? *(list of other hostnames)*
  - Web server: Behind-the-scenes scripts or databases?
  - Mail server: Full list of email accounts? *(useful for social engineering)*

File Transfer

- Numerous ways to get files to or from target system
- Push files **to** target
  - FTP, SCP, NFS, SMB, Meterpreter
  - *Only if firewall permits it*
- Target pulls files **from** tester machine
  - HTTP, HTTPS
  - Practically every system will have either a GUI and/or command-line HTTP client
  - *Firewall more likely to allow web-like traffic*

Post-Exploit Inspection: Passwords

- Hashed login passwords
  - Linux: `/etc/passwd`, `/etc/shadow`
  - Windows: SAM (security account manager) database
- Crypto keys
  - SSH public (and private!) keys
  - PGP keys
- Microsoft Credential Manager logins
- Scripts and programs with hard-coded passwords
- Wireless client profiles (pre-shared keys?)

Passwords

Obtaining Passwords

- Exploits are fun & dramatic, and a key part of penetration testing
- Drawbacks
  - Not every system has a current exploit (whether a polished Metasploit module or in “raw” form on the web)
  - Exploits can crash the target (application or OS), or just be flaky and unreliable (i.e. must try 10 times to get it to work)

Goal of any pen tester is to leverage an exploit into **usernames and passwords**, which can then be used to gain access to more systems and retain access after a vulnerability is patched

Obtaining Passwords : Methods

Online Attack

- Generate password *guess* and send it to target to verify
- Pros
  - Will work if you have *no other choice*
- Cons
  - Slow (network latency + target throttling)
  - Can lock out legitimate users due to repeated failures
  - Can set off security alarms you would rather not trigger

Offline Attack

- Generate password *guess*, hash it, and compare to hashed password you previously obtained via exploit
- Pros
  - Dramatically faster!
    - No network latency
    - No target throttling
    - Parallelizable
  - No risk of account lockouts
  - Less risk of attack being detected

Obtaining Passwords : Cracking

- Brute force password cracking (either online or offline) requires **wordlist** + set of permutations on the wordlist
  - Engine just tries every possible word + permutation and checks result
- The larger the wordlist, the longer it will take to test
  - Speed also affected by available parallelism (GPUs?) and complexity of the password hashing algorithm (more on cryptography later!)
- Vary size based on specific scenario
  - Shorter wordlists for online attacks?
  - Longer wordlists for offline attacks?

Obtaining Passwords : Cracking

- Kali has a number of small and medium wordlists available
  - `/usr/share/metasploit-framework/data/wordlists/`
  - `/usr/share/wordlists/`
- Larger wordlists can be obtained online
  - https://crackstation.net/crackstation-wordlist-password-cracking-dictionary.htm (15GB uncompressed)

Obtaining Passwords : Cracking

- What about a **better** wordlist as opposed to a **larger** wordlist?
  - Suggestion: Edit your password permutations to ensure they match known corporate password policies (One uppercase, one lowercase, one symbol…)
  - Suggestion: Crawl the corporate website and make a custom dictionary of words from that website
  - Tool: CeWL (Custom WordList Generator)
    - https://github.com/digininja/CeWL/
    - Already in Kali!

Obtaining Passwords : Cracking

- The quality of Linux password hashing algorithms has improved over time
  - Implemented in `crypt()` / glibc library
- Numbers in `/etc/passwd` represent the algorithm used
  - `$1$` - MD5 (oldest & fastest to brute force)
  - `$2$` - Blowfish
  - `$5$` - SHA-256
  - `$6$` - SHA-512
  - `$y$` - Yescrypt (based on scrypt, newest & slowest to brute force)
- Example from Metasploitable2:
  `root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::`
  - Uses the MD5 hash (a very weak algorithm), but the password is highly random and thus still resistant to cracking

Obtaining Passwords : Cracking

- You’re being paid `#RealMoney` ($$) for a pentest
- Put some of it to use!
  - Build a cluster of password cracking servers with the latest GPUs
  - Rent some servers (w/GPUs) in the cloud

Passwords : Can Never Have Enough

- In an enterprise environment, a single password might be usable on myriad systems (Windows domain, single sign-on, etc…)
- Even for stand-alone systems, how many users employ the same password everywhere?
  - *Best practices are a password manager plus random password*
- Suggestion: Obtain hashes and start cracking passwords immediately even when you have full root/Administrator access to the target system and that system is otherwise uninteresting
  - You never know **where else** that password might work

Passwords : After The Test

- Document time it took to crack each password and the relatively complexity
  - 100 hosts w/4 GPUs running for 100 hours?
  - Or 1 host with no GPU guessed the password in 10 minutes?
- Provide list of all exploited accounts to client with expectation that users will be changing their passwords *immediately*
- Don’t keep copies of these cracked passwords after the end of the pentest period – considered privileged information

Meterpreter

Meterpreter

- **Met**asploit Int**erpreter**
- Powerful tool that aims to provide a consistent command-line interface to the target host
- It’s a shell, but instead of Windows CMD shell, Windows PowerShell, Linux BASH shell, it’s a *Meterpreter* shell
  - Target platforms: Windows (x86, x64), Linux (x86, x64, arm), OS X (x64), Python, PHP, Java, iOS, Android

Meterpreter: Stealth

- Resides entirely in memory
  - No files written to disk for AV to detect
- No new processes are created
  - Injected into existing running process, and can migrate to different processes on demand
- Encrypted communication
  - *Outbound* TLS connection from target host back to Metasploit (will resemble web traffic)

Meterpreter: Navigation

- Basic set of file navigation and manipulation commands
  - `cd` – Change directory (on target host)
    - `lcd` – Change directory (on local Kali host)
  - `pwd` – Print working directory
    - `lpwd` – Print directory (on local Kali host)
  - `ls` – List files
  - `cat` – Display file
  - `download` / `upload` – Transfer files between Kali and target
  - `edit` – Edit file on target (fun with vi!)

https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

Meterpreter: Search

- `search` – Look for files on target host

```
meterpreter > search -d c:\\documents\
and\ settings\\administrator\\desktop\\ -
f *.pdf
Found 2 results...
 c:\documents and
settings\administrator\desktop\operations
_plan.pdf (244066 bytes)
 c:\documents and
settings\administrator\desktop\budget.pdf
(244066 bytes)
meterpreter >
```

https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

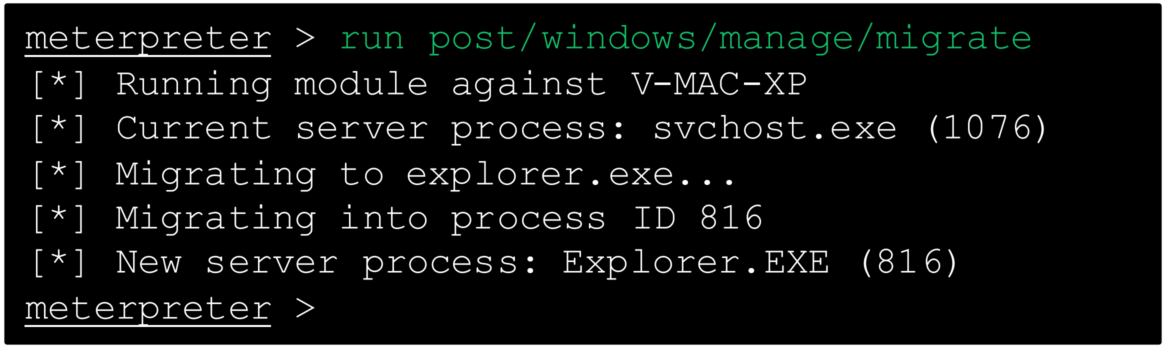
Meterpreter: Processes

- `getpid` – Get PID that Meterpreter is running inside of
  - Remember, Meterpreter is *injected* into an existing process for stealth!
- `getuid` – Get UID of user running the process that Meterpreter is inside of
- `ps` – Process list
- `kill` – Terminate a process
- `execute` – Run another process

https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

Meterpreter: Processes

- `migrate` – Move Meterpreter to another running process
  - More stable? (e.g. web server would be better than notepad.exe)
  - Hiding from scanners?
  - Looking at files owned by that process?



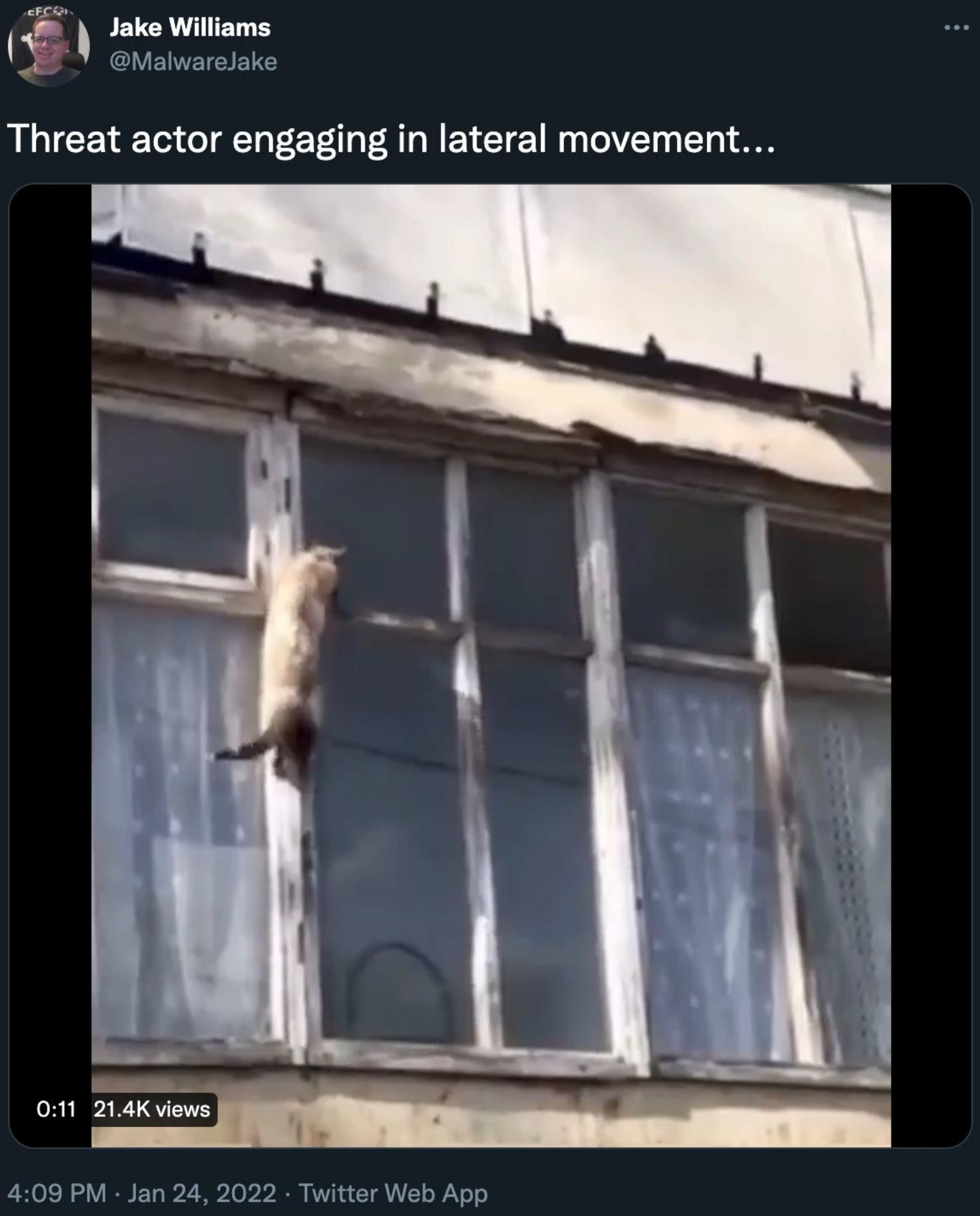
```
meterpreter > run post/windows/manage/migrate
[*] Running module against V-MAC-XP
[*] Current server process: svchost.exe (1076)
[*] Migrating to explorer.exe...
[*] Migrating into process ID 816
[*] New server process: Explorer.EXE (816)
meterpreter >
```

https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

Meterpreter: Other Functions

- Some *fun* Meterpreter functions may be **<u>legally dubious</u>** for penetration testers *(or go beyond testing scope)*
  - Screenshot of desktop?
  - Disable keyboard & mouse for in-person user?
  - Keystroke logger?
  - Webcam pic grab?
  - Microphone audio sample?

Pivoting

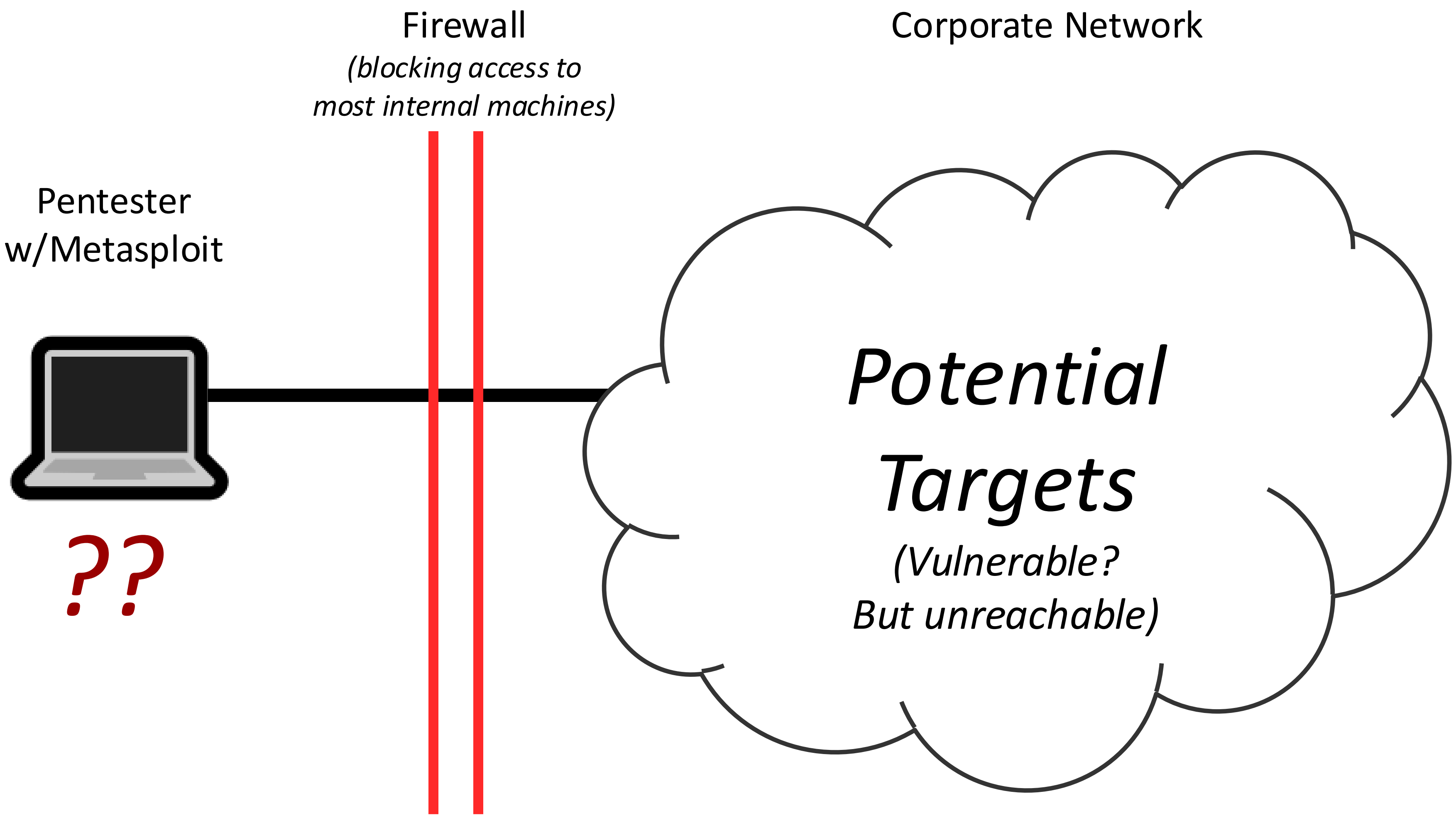


Jake Williams
@MalwareJake

Threat actor engaging in lateral movement…

4:09 PM · Jan 24, 2022 · Twitter Web App

Pivoting



Firewall

(blocking access to most internal machines)

Corporate Network

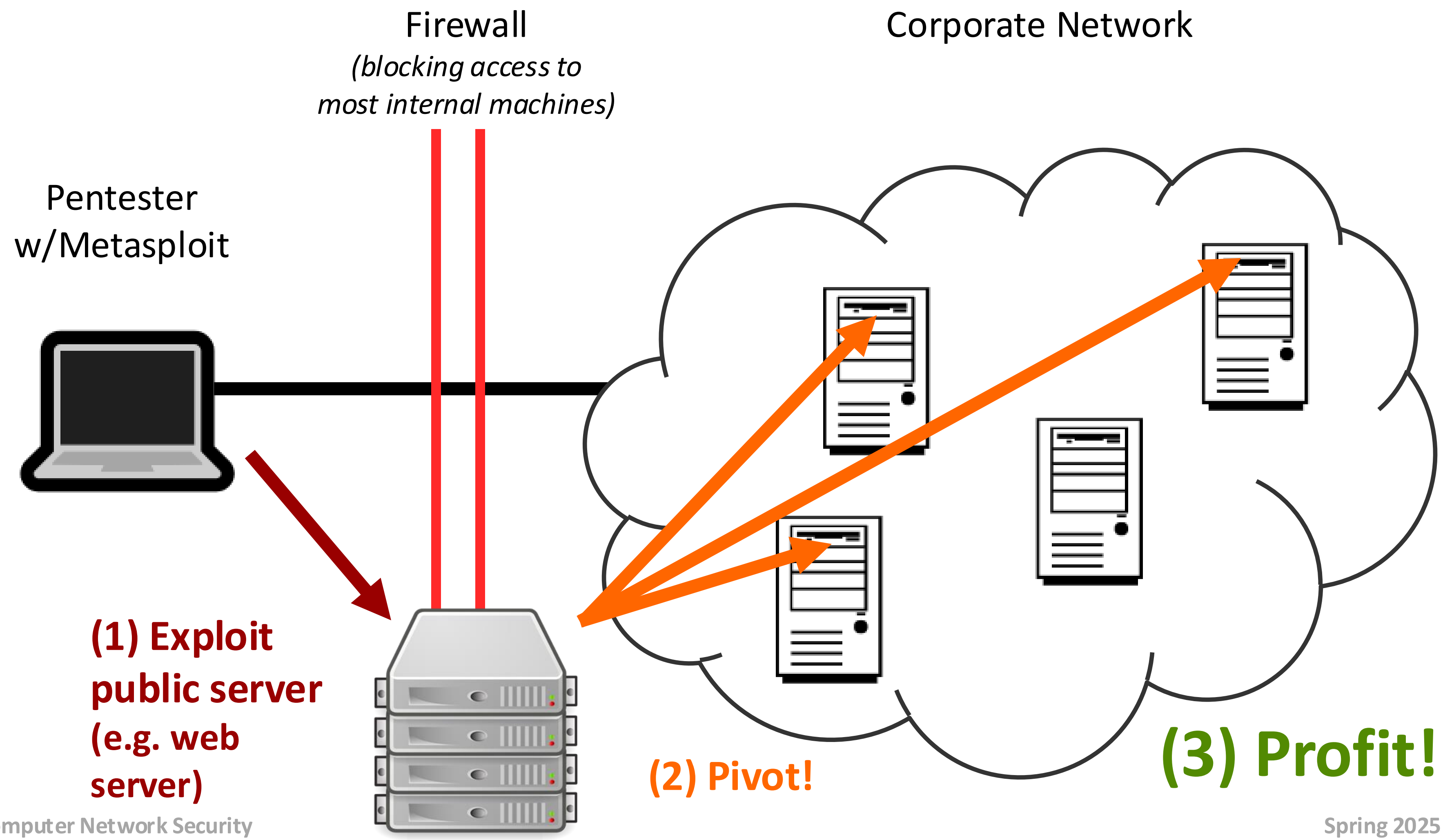
Pentester w/Metasploit

??

Potential Targets

(Vulnerable? But unreachable)

Pivoting


Firewall
(blocking access to
most internal machines)
Corporate Network
Pentester
w/Metasploit
(1) Exploit
public server
(e.g. web
server)
(2) Pivot!
(3) Profit!

Pivoting

"Nesting"

- Install all your favorite pen testing tools on the pivot system
- Use remote desktop or SSH to connect to pivot system
- Pen testing equivalent of "moving right in" (to the system)
- Is installing software on targets allowed by your rules of engagement?

"Live Off the Land"

- Install **nothing** on the pivot machine
  - No files on disk!
  - *Potentially allow some in-memory software, e.g. Meterpreter*
- Approaches
  - Run commands and use software already installed on pivot
  - Tunnel network traffic (exploits, scans) *through* the pivot
- Advantages
  - Easier to cleanup after pentest
  - Harder for adversaries to detect

Meterpreter: Port Forwarding

Remote Desktop App

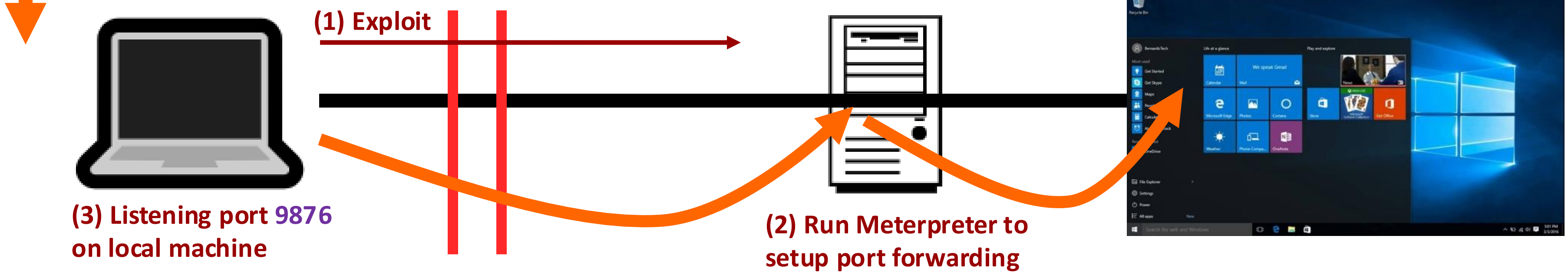
- TCP tunneling
  - Firewall bypass? For purposes of system access or even exploit targeting
  - `portfwd add -l 9876 -p 3389 -r w.x.y.z`

Pentester w/Metasploit

Firewall *(blocking access to most internal machines)*

Exploitable *internal* target not blocked by firewall

Windows 10 Target **w.x.y.z** *(with remote desktop, port 3389)*


(1) Exploit
(3) Listening port 9876 on local machine
(2) Run Meterpreter to setup port forwarding

https://www.offensive-security.com/metasploit-unleashed/portfwd/

Meterpreter: Port Forwarding

- `portfwd <command> <options>`
- Commands
  - `add` / `delete` / `list` / `flush`
- Options
  - `-l` – Listening port (on Metasploit host)
  - `-p` – Destination port (on target host)
  - `-r` – Destination IP (on target)

https://www.offensive-security.com/metasploit-unleashed/portfwd/

Metasploit : Exploit Forwarding

- Metasploit (**not Meterpreter)** has an easy way to forward exploits *through* a Meterpreter tunnel

```
msf5> use [exploit]
msf5> set RHOST [pivot]
msf5> set PAYLOAD windows/meterpreter/bind_tcp
msf5> exploit

// Meterpreter runs – do CTRL-Z to send to background
// Note Meterpreter session ID

msf5> route add [pivot subnet] [netmask] [sessionID]
msf5> use [exploit2]
msf5> set RHOST [victim2]
msf5> set PAYLOAD [payload2]
msf5> exploit
```

You can accomplish the same thing via "autoroute" command inside of Meterpreter *https://www.offensive-security.com/metasploit-unleashed/pivoting/*

Bonus Resources

Bonus Resources on Class Website

Linux Post-Exploit Cheat Sheet

Windows Post-Exploit Cheat Sheet

Tools! More Tools! Yet More Tools!
(For enumeration, privilege escalation, “living off the land”, …)