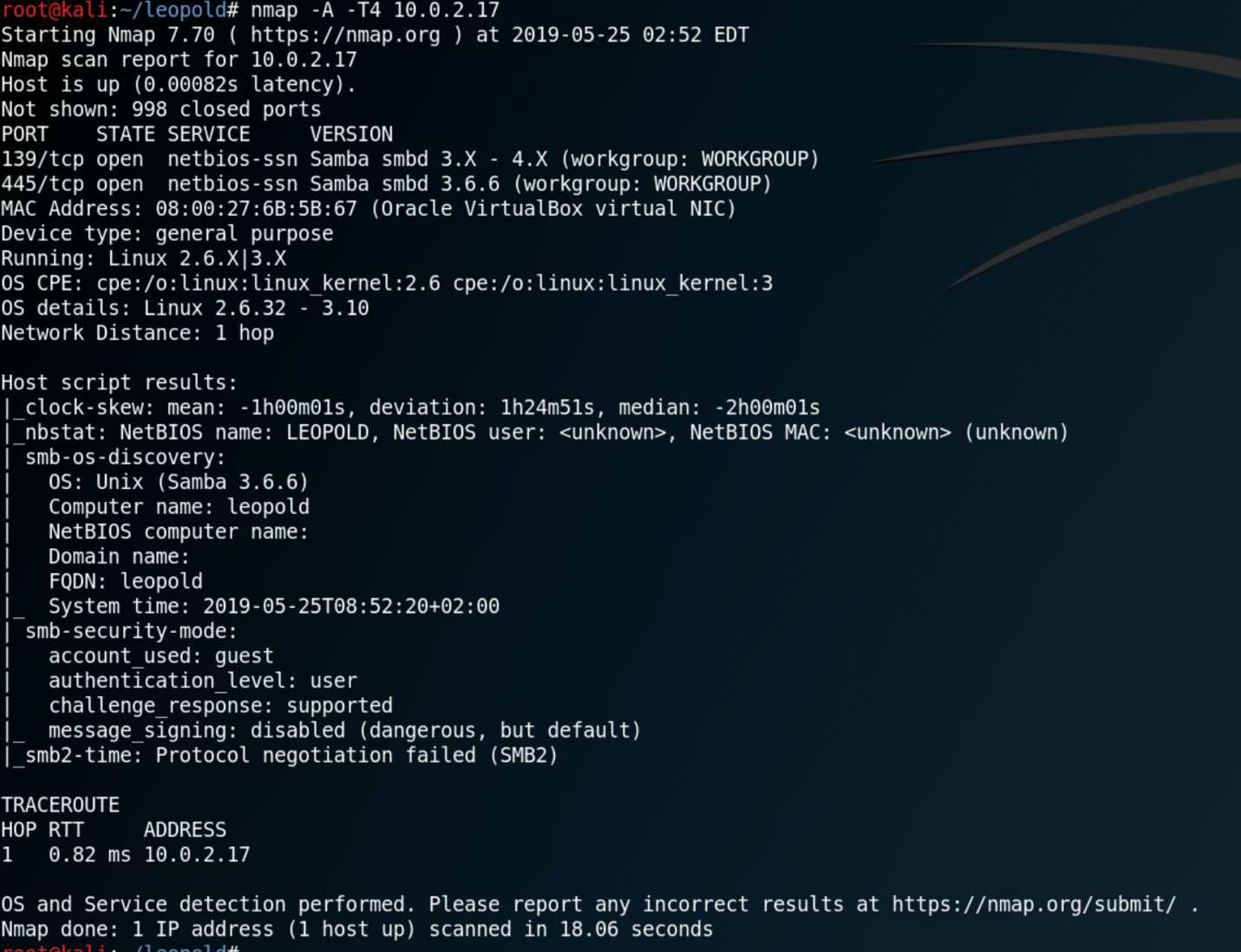
Leopold Walkthrough

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:



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):

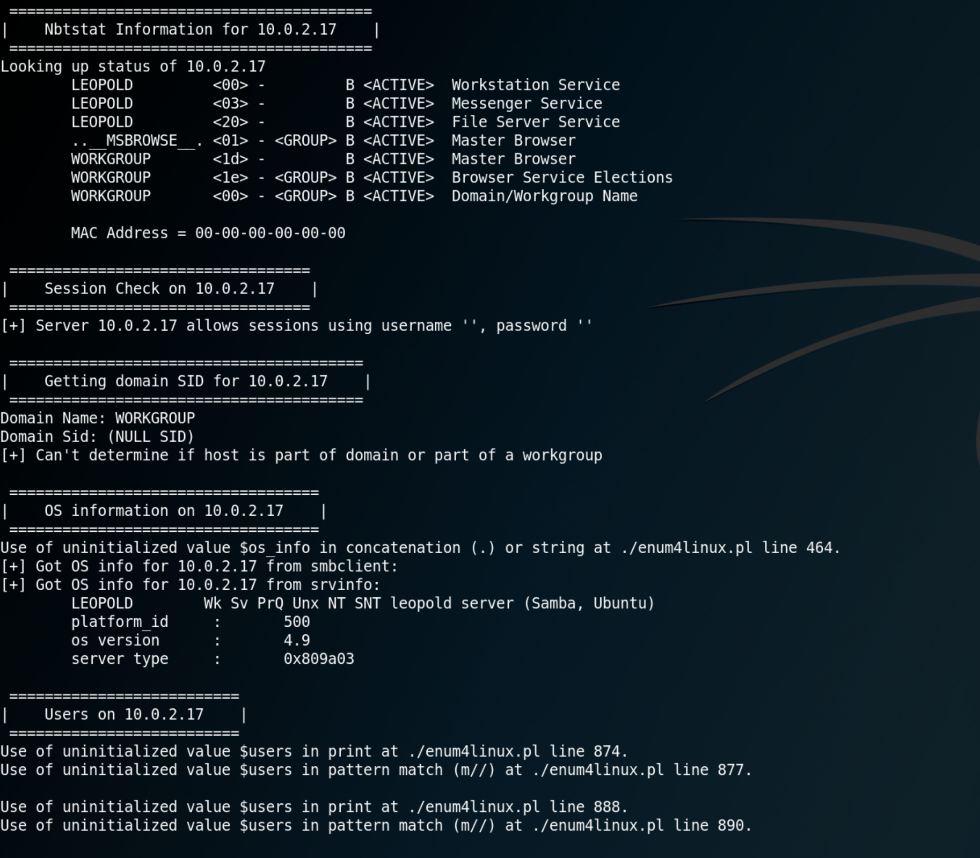
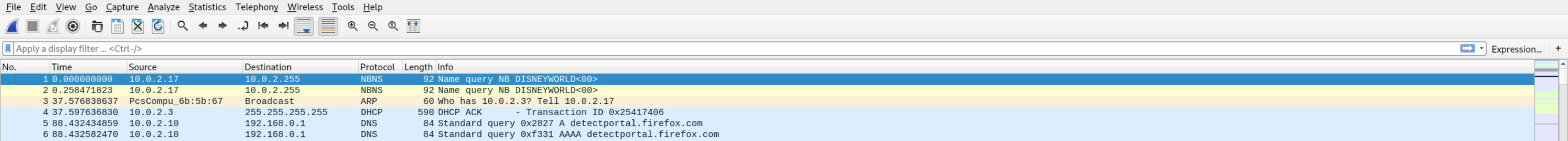
Figure 2 (on left), Enum4linux shows that null sessions are allowed. In addition, the tool yields more information pertaining to the underlying operating system.

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.



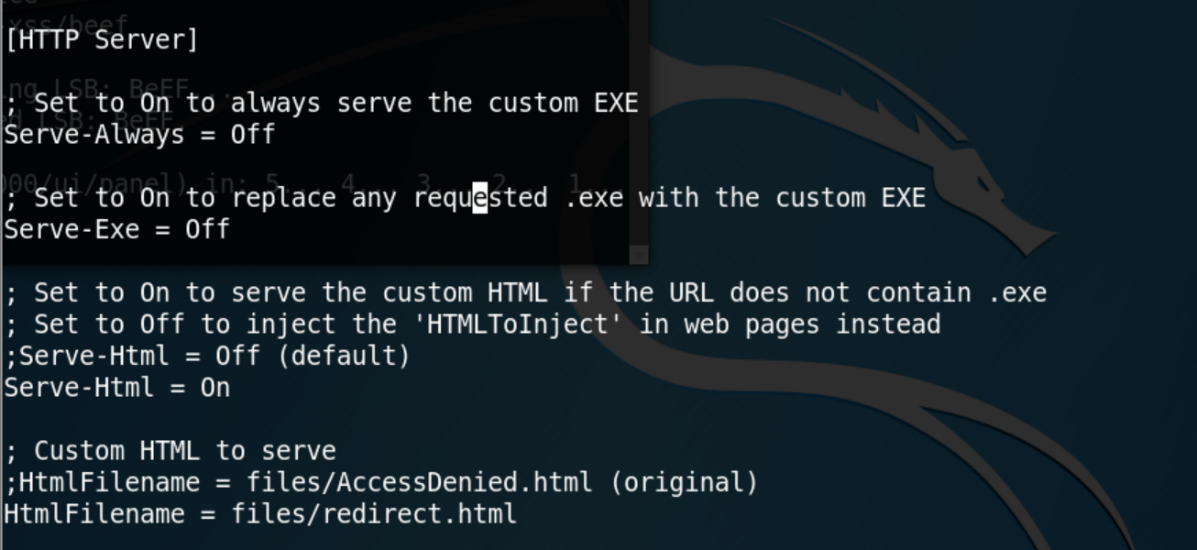
The victim machine is using netbios-ssn. Netbios-ssn allows machines on the same subnet to identify each other when DNS fails. Since the attacking machine is on the same network, Wireshark (or any other sniffer) can be used to see if any interesting information is being sent over the network. After waiting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



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**

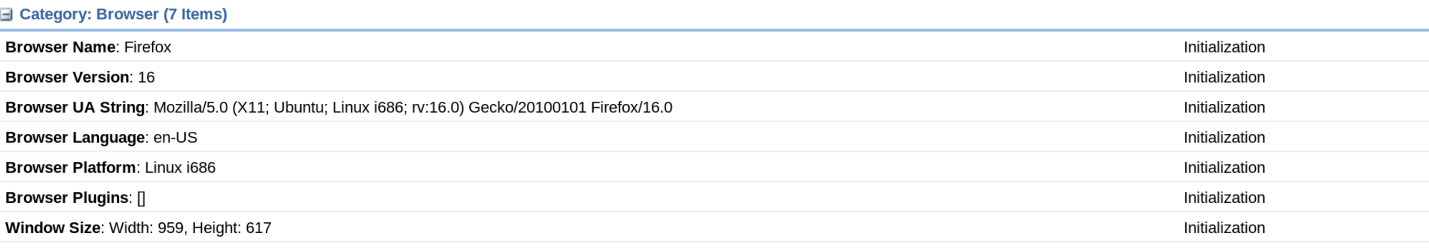
The tool Responder can be used respond to the NBNS request. Responder can be used to redirect the victim’s machine to the Beef framework, which will hook the victim’s browser (hooking the browser will give us the version of the browser which can be used to further exploit the machine).

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.

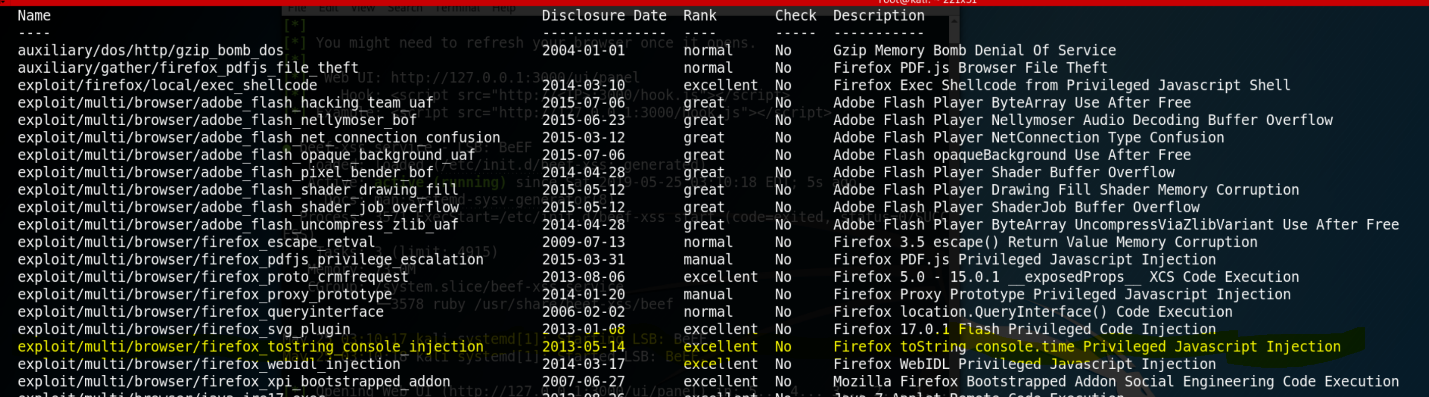
Navigate to /usr/share/responder/files, create a file called redirect.html, and add the following to the file:



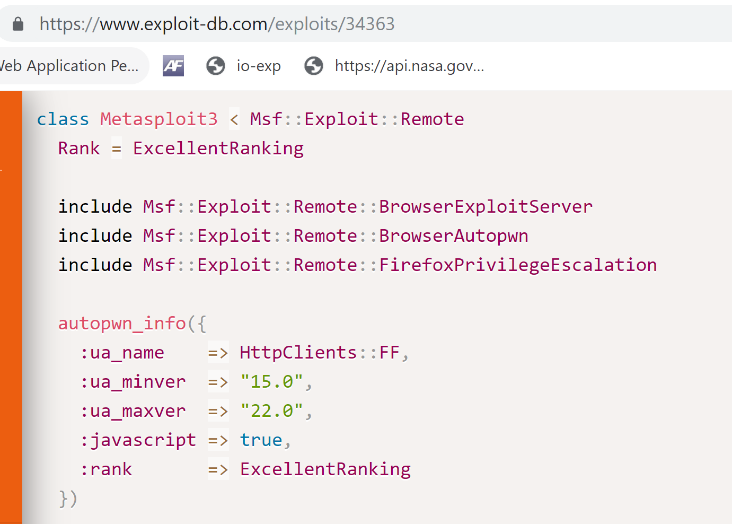
After everything is setup Responder can be run (responder -I eth1 -wrf). Beef was also started at the URL and port number shown in the above image. When the victim asks for the location of Disneyworld Responder redirects the victim’s browser to the hooked webpage (provided by Beef).



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.

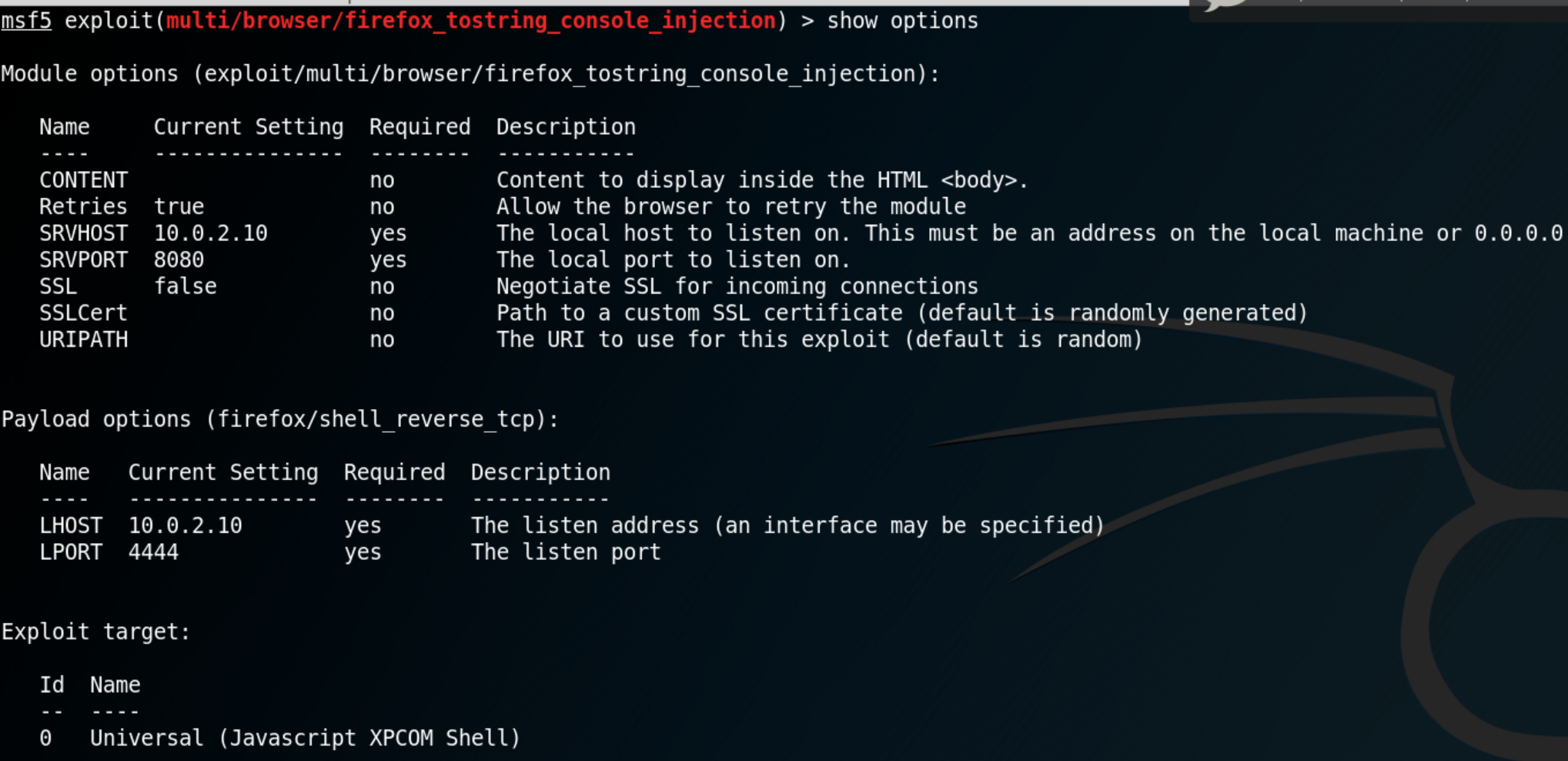


The highlighted option was ultimately chosen because the vulnerable versions matched the version of Firefox being used by Leopold.

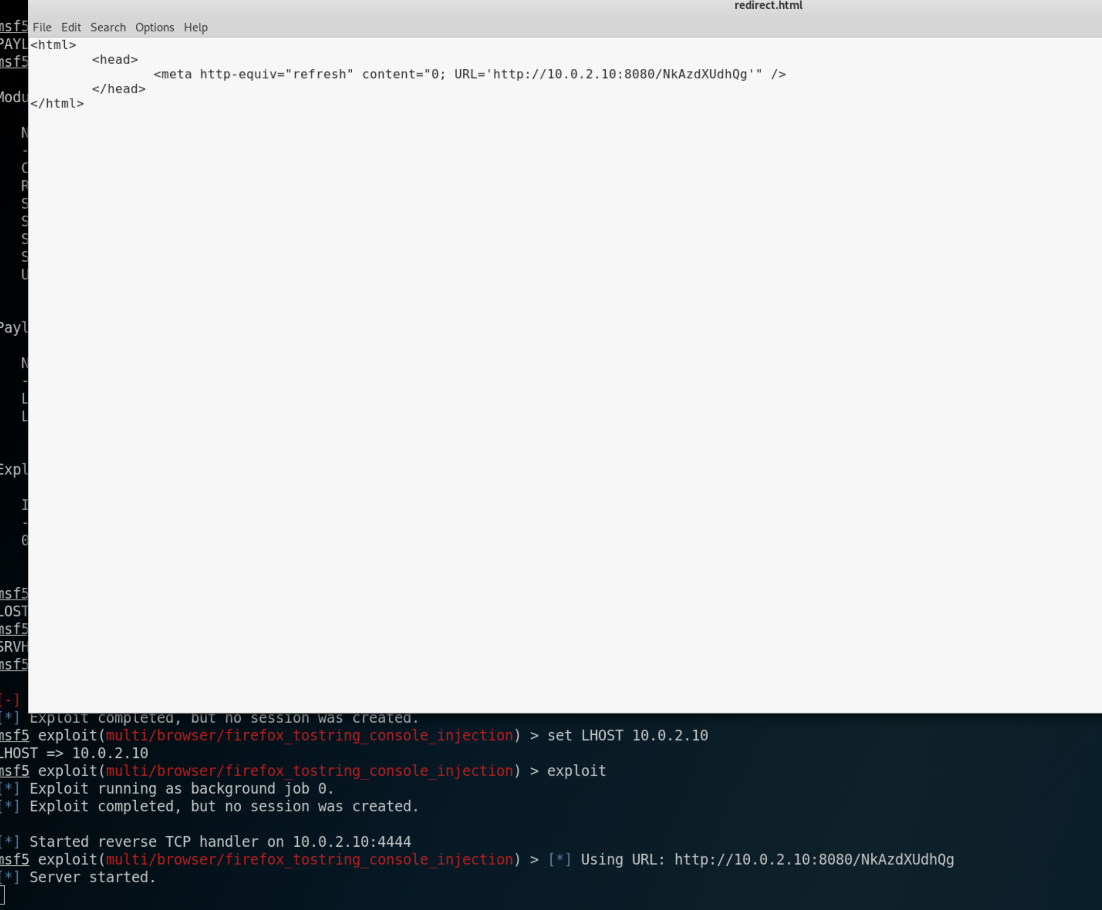
The figure on the left illustrates the minimum effected version (15.0) and the maximum effected version (22.0). Since the target is using Firefox version 16 this exploit should be effective. The exploit source code can be found by searching the exploit database for Firefox to string console injection.

**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).



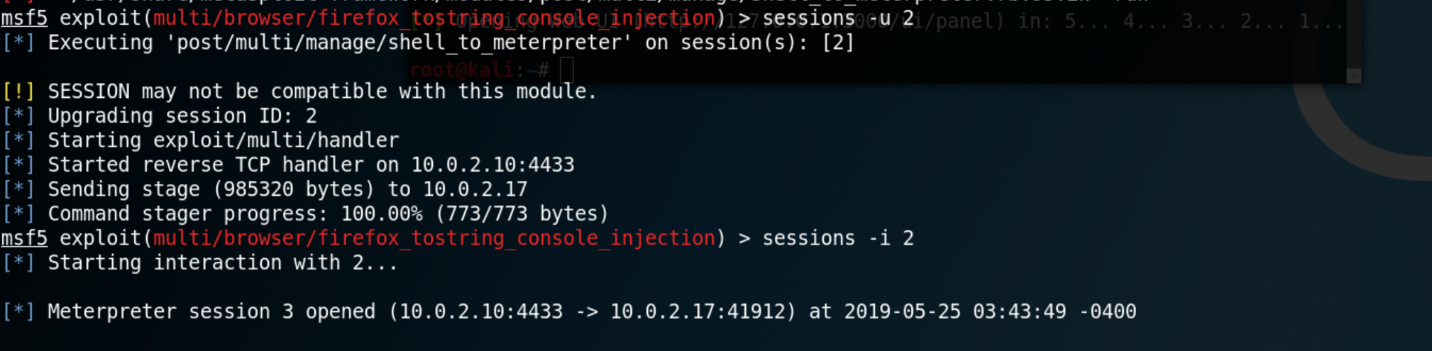
Once everything is ready to go run the exploit. However, since this exploit relies on the target connecting to the malicious payload the tool Responder must be used to redirect the victim to the payload. The set-up for this can be seen below. Please note that the top portion of the image shows the changes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



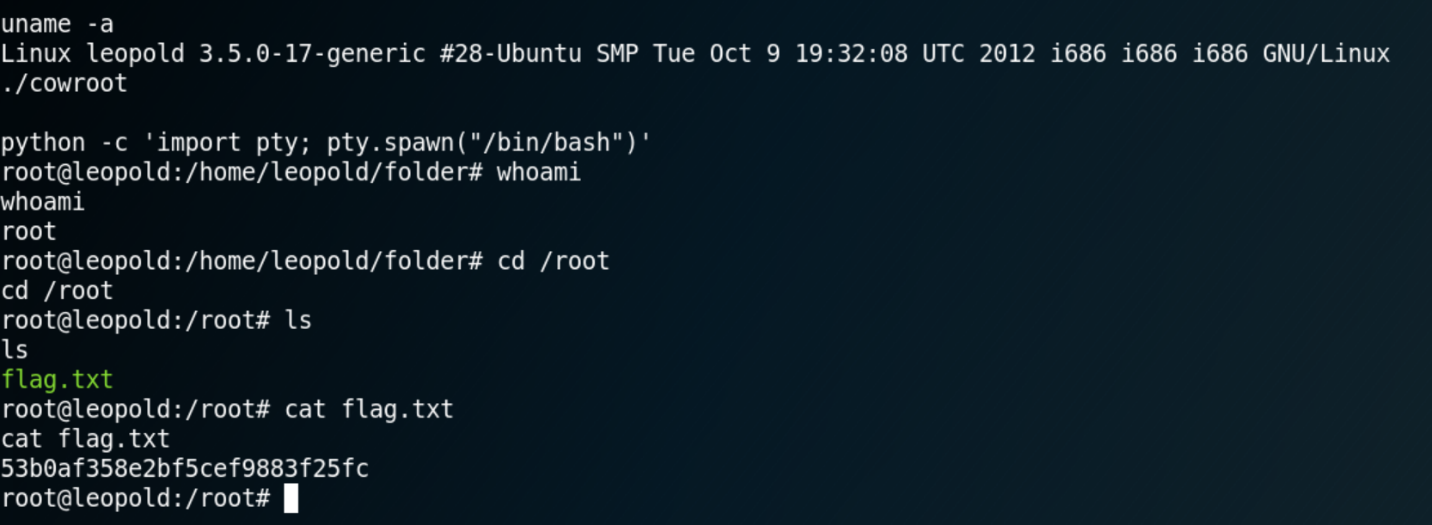
The payload works as expected and a shell is received. Unfortunately, the shell seems to die after a few commands. Repeating the process explained above gives a new shell.

**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>.



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).



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