Using Kali to gain access to Metasploitable.

Tommy Flasza – s4092115 – F4AC - Cyber Security Foundations

Part I – Checking Kalis and Metasploitable’s connection to one another

In fig 1 and 2 I have checked that I have both Kali Linux which is built from Linux and Metasploitable have been installed in Virtual box and both work

After resetting the DHCP client on the Metasploitable machine since the machine was having trouble with pinging one another I used the Linux command of “sudo dhclient” to change the Ip address to allow for it to be pinged by the main Kali machine. I then used “ping 192.168.0.169” to check and verify if there was a connection between both machines (fig 3)

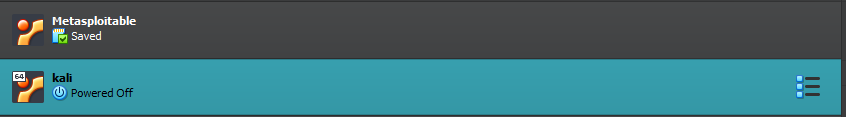


Fig 1: Kali and Metasploitable in VMware installed.

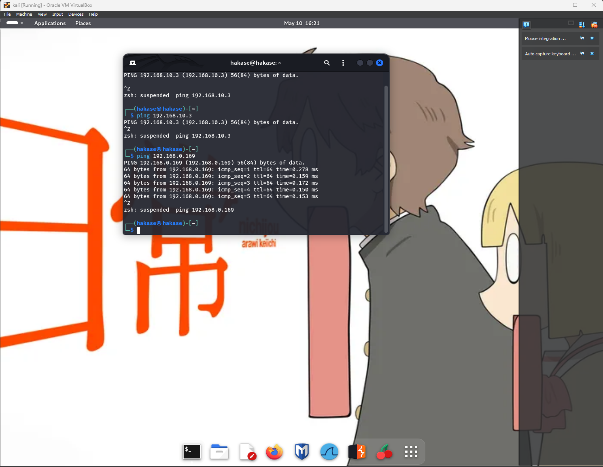
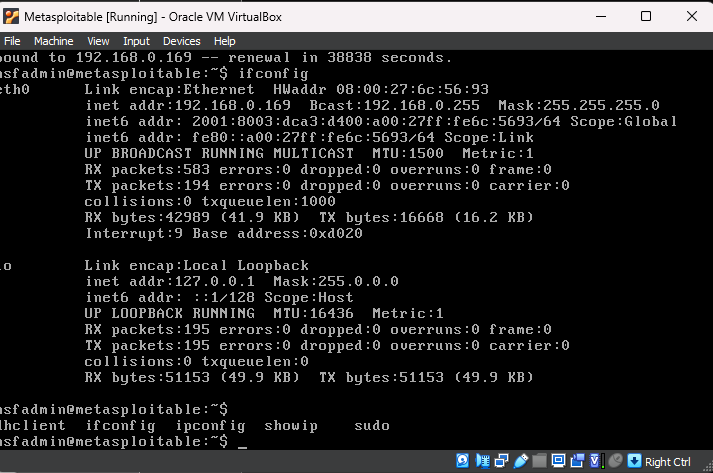


Fig 2: Kali and Metasploitable running at the same time in VMware

A screenshot of a cartoon

Description automatically generated

A cartoon of a child

Description automatically generated

Fig 3: Pinging both Machines

Part II – Generating the Target Files

I started with the text file of my name and student ID “Tommy Flasza – s4092115” (fig 4) that would be used for generation to having a ciphertext encryption. I used OpenSSL to generate a RSA key for both decryption and encryption that would protect the data in this case my name and student ID from being seen and decrypted beside from the person who I would send it to. In fig 5 we can see the terminal and the outputted encrypted file from using openssl. The command to use the rsa command to encrypt was (“openssl rsautl -encrpyt -pubin -inkey public\_key.pem -in plaintext -out info.txt”) this allows for the file to be encrypted with the key that was just made. In fig 6 I used steghide’s command (“steghide embed -ef private\_key.pem -cf secret.jpeg”) to hide the private key inside a photo of Zeus a professional league of legends player. This will allow for the person who is intended for this file to be sent the actual key to be allowed to decrypt the actual contents of the file “info.txt”. In fig 7 we can see the files being ready to be sent to the target machine. They are in “/home/hakase/” to make it easier to find and send.

A screenshot of a computer

Description automatically generated

Fig 4 – Creation of the file

A screenshot of a computer

Description automatically generated

Fig 5 – The terminal and encrypted file saved in /home/hakase

A computer code with white text

Description automatically generated

Fig 6 – Using Steghide to put the decryption Key into a photo of Zeus (Secret.jpeg).

A screenshot of a computer

Description automatically generated

Fig 7- Secret.jpeg with the payload of key and Info.txt with the encrypted credentials in /home/hakase

Part III - Establishing Remote Connectivity

In part 3 I Used the Metasploitable framework to scan the open ports on the Metasploitable machine. The command (use auxiliary/scanner/portscan/tcp) allows for all the ports to be scanned and be discovered to allow for penetration (fig 8). The main one I used here was port 22 (fig 9) which was open, and this allowed for me to use ssh\_login (“use auxiliary/scanner/ssh/shh\_login”) to attempt a login into the vulnerable machine since I knew the username and password of the machine. After inputting the command, I put in the IP of the vulnerable machine. Then the username and password of it. As I pressed run it started the brute force login and was successful (fig 10). I had to change my own SSH config to allow for the machine that needed to be penetrated to be seen as a know host after getting errors within trying to send a file. After fixing the config file and using “SSH-Keygen” that establishes a secure session both Kali and the Metasploitable machine using secure brand-new generated authentication keys. I was allowed securely into the Metasploitable machine, and I used the command of “whoami” to display the name of the machine. It outputted msfadmin (Fig 11) which was the outcome that was needed so I logged out of it and prepared to copy over the files that contained my name and student number along with the image that had the private key embedded inside of it. Since I was still connected to port 22, I used the command of scp to allow for file transfer. I sent the text file with the encrypted name and ID using the command “scp /home/hakase/info.txt [msfadmin@192.168.0.169](mailto:msfadmin@192.168.0.169):” to allow the transfer after inputting the password of the Metasploitable machine and was a success (Fig 12). I then did it again with the image with the key embedded with the command of “scp /home/hakase/secret.jpeg [msfadmin@192.168.0.169](mailto:msfadmin@192.168.0.169):”, this was also a successful transfer according the msfconsole (Fig 13). With this, all the objectives were completed successfully.

A screenshot of a computer program

Description automatically generated

Fig 8 – Using portscan to check for open ports

A screenshot of a computer screen

Description automatically generated

Fig 9 – All of the open ports with the key one of 22 being open

A screenshot of a computer

Description automatically generated

Fig 10 – successful SSH login to allow for a brute force using port 22. The username and password of msfadmin is the target machines credentials.

A screenshot of a computer screen

Description automatically generated

Fig 11 – using ssh-keygen to update the keys to allow the Kali machine to access the Metasploitable machine and use its own terminal. The command of whoami is used to verify the machine and if it was properly penetrated.

A computer screen shot of a cartoon

Description automatically generated

Fig 12 - successful file transfer from the Kali machine to the Metasploitable machine of the encrypted student name and credentials

A screenshot of a computer

Description automatically generated

Fig 13 – Successful file transfer of the Image of Zeus (secret.jpeg) carrying the private RSA key needed to decryption embedded into it.