Seth Ayers

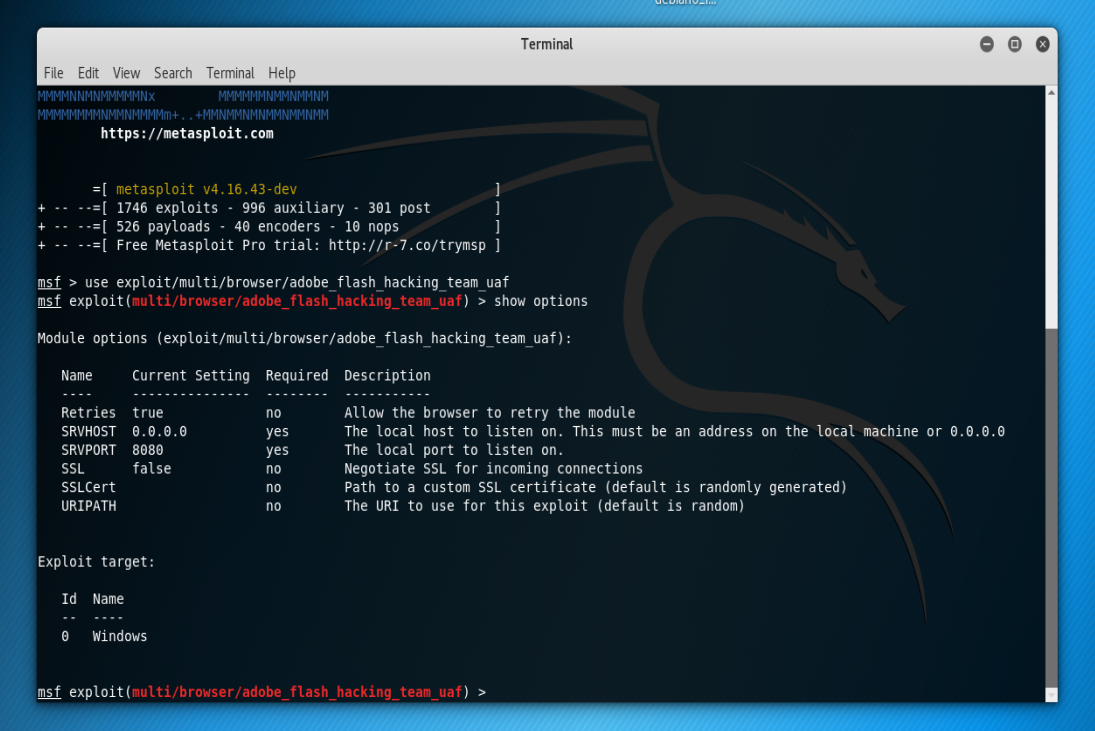
03/29/2018

IT 430 – 02

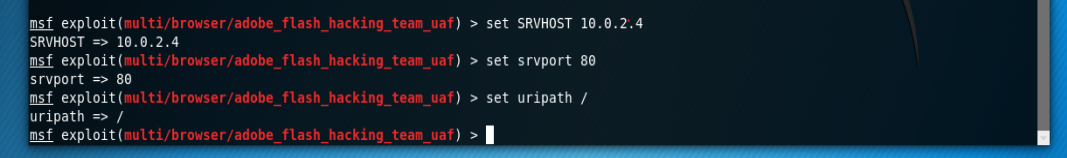
Lab 06

SAyers\_Lab06.docx

**Lab 06: Attacking a Windows Machine (Part A)**



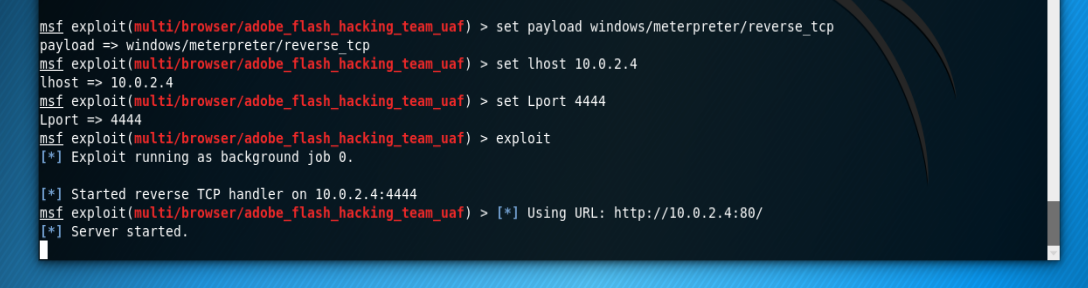
*Initialize adobe flash hack and display available options.*



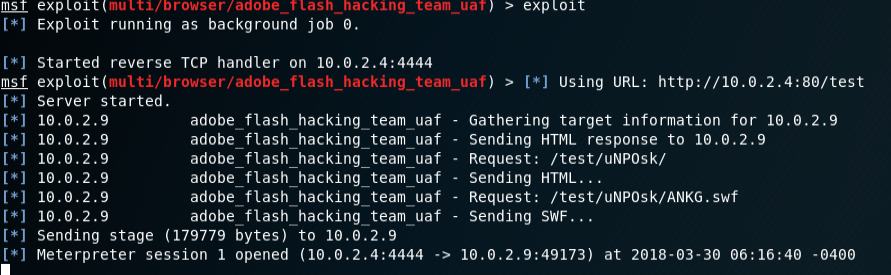
*Configuring the Host, port, and URL parameters.*



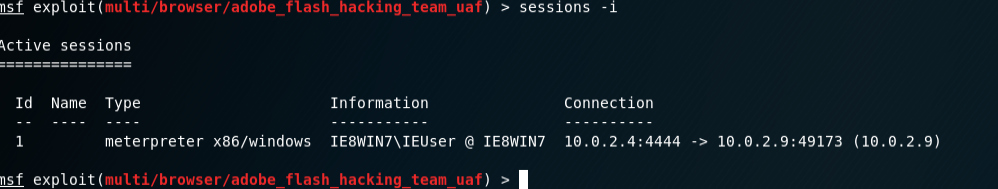
*List of available targets and selection of the windows VM machine.*



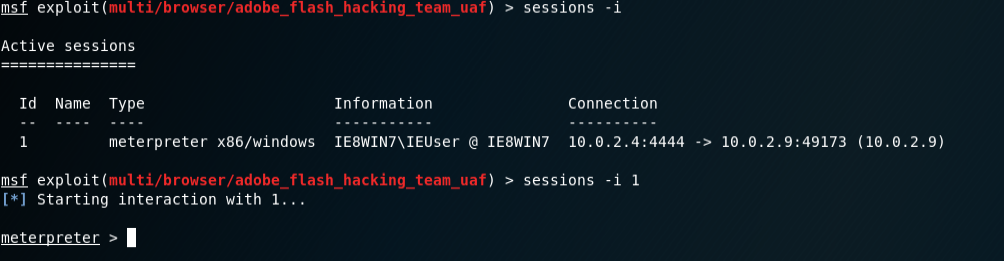
*Setting the payload with required options*



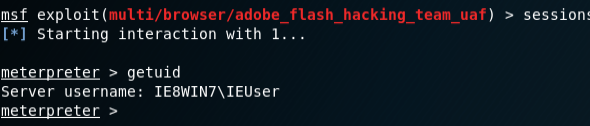
*We have access!*



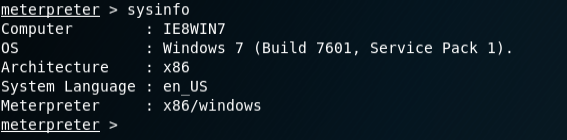
*Displaying the sessions we have available. In this case, there is only one for our connection to the WIN 7 machine.*



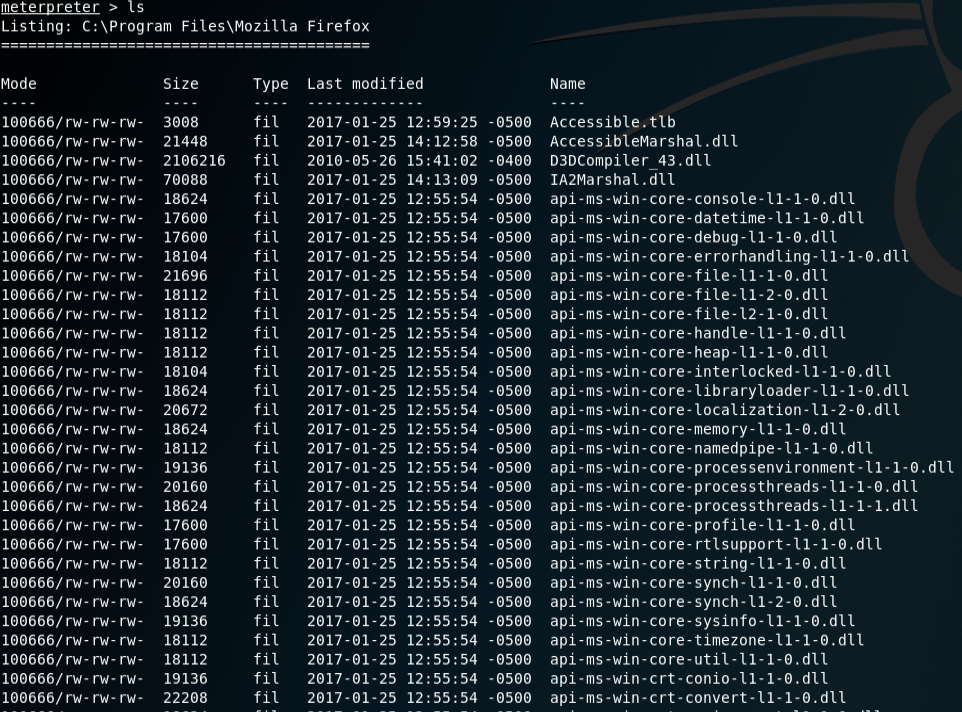
*Interaction with the WIN 7 machine.*



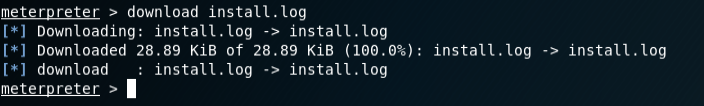
*Shows the User ID of currently logged in accounts (WIN 7 machine)*



*System info of the WIN 7 machine we’re connected to.*



*All the wonderful files we have access to.*



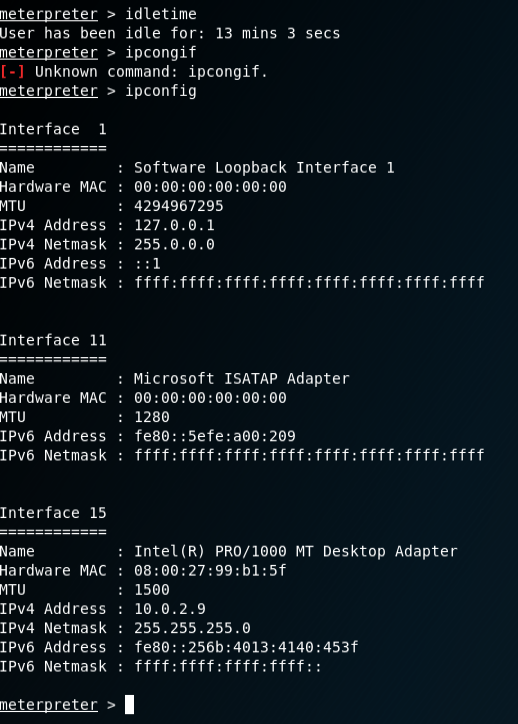
*Download of install log.*



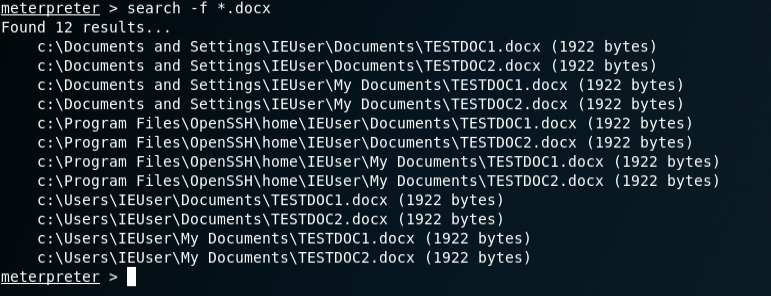
*Running processes on the WIN7 machine. This is vital to locating additional attack vectors in which a foothold can be more securely established.*



*Displays the idle time of the user on the WIN7 machine. This is important information because if the user has connected to our “website” and left the browser open, we have more time to preform a deep search of the attacked machine. However, if the user is shown as having a low idle time, we must gather all “low hanging fruit” and establish a more permanent foothold (rootkit/backdoor) as quickly as possible.*



*Displays the IP and MAC information of the WIN7 machine.*



*Searches the document that was made on the WIN7 machine using the wildcard operator. If more word documents were on the machine, the list above would be more substantial.*

**Reflections and observations for the adobe\_flash\_hacking\_team\_uaf**

**What have you learned from this lab?**

Overall, the lab provided a fantastic step by step in successfully attacking a venerable host. A few things stood out for me, most importantly, the critical part that reconnaissance plays in the implementation of an attack. Discovering and honing down reconnaissance information for an an attack vector is not only important to the success of the attack but is also critical in keeping an “invisible presence”. Meaning, if you know specific factors of the target (OS, applications, services, versions), an exploit can be used that would ensure success. Otherwise, the use of ineffective exploits can almost guarantee detection which would most certainly result in complete failure. The importance of reconnaissance is clearly shown in the lab, as the attack vector is an older version of Adobe Flash, which would haven’t been discovered unless diligent recon actions preceded the attack.

Another point of interest was the nature of the adobe\_flash\_hacking\_team\_uaf exploit. It’s not just restricted to individual targets and if another target were to connect, another session would open if the target machine had an older version of Flash. This opens quite a few possibilities in attacking multiple targets utilizing other means of exploitation. For instance, time can be sent preforming precision footprinting, seeking out targets with specific versions of Adobe Flash and obtaining email addresses of the users. Once a robust list is compiled, phishing emails can be composed sent to the targets with spoofed URL’s. These spoofed URL’s will contain the link to the created session. If a target connects, an automated script can detect the connection through a flag function (the exploit has this function in the code) and dump a backdoor/rootkit into the target system in a matter of seconds. This eliminates the possibility of disconnecting via browser closure and maintains persistence.

**What did you find challenging?**

As mentioned above, I really learned the importance of footprinting. This proved to be the biggest challenge in the lab as I spent roughly two hours troubleshooting an issue I had with the meterpreter command line. Basically, I couldn’t get the commands to process, receiving the message “Invalid command”. I knew that the attack vector of the exploit was the Adobe Flash player, but I didn’t connect the fact that my old version of Flash wasn’t as old as it should have been. I troubleshot the problem by looking up the exploit online to get some details and discovered that, although my version of Flash was old, it wasn’t old enough. I corrected the problem and proceeded with the lab.

**What are the major steps in exploiting a target machine?**

The major steps are as follows:

Footprinting/Reconnaissance: Obtaining as much information about the target system as possible (IP/MAC addressed, Operating Systems, Applications, Services running, Ports open, Network Topography, Versions)

Selecting/Building an exploit: Depending on the information gained from the footprinting/reconnaissance, a viable exploit must be selected. To take into consideration: Goal of the attacker’s motives, resources, time.

Commencing attack: Setting up parameters for the attack, and initiate. Confirm connections.

Ensure persistence: Quickly install backdoor/rootkits/malware/keyloggers deep into the system to ensure continuous access, regardless if the exploitation fails. Delete logs.

**What would have happened if the target closed the browser or shutdown the computer during the exploitation? Would you still gain access to the target after it reboots? If no, what could you do to maintain access to the target?**

In the event of a target shutdown, access would be disrupted and would not return if the user were to reboot. In order to prevent the loss of potential data of the attacker, a number of exploits can be used to backdoor the targets system. These rootkits, trojans, malware, and keyloggers can assist in maintaining an “on demand” connection to the targets machine. These can be executed by placing the active session in the background.

**What meterpreter command is used to wipe out the target logs?**

WINDOWS:

Clearev: This will wipe the Application, System, and Security log on a windows system.

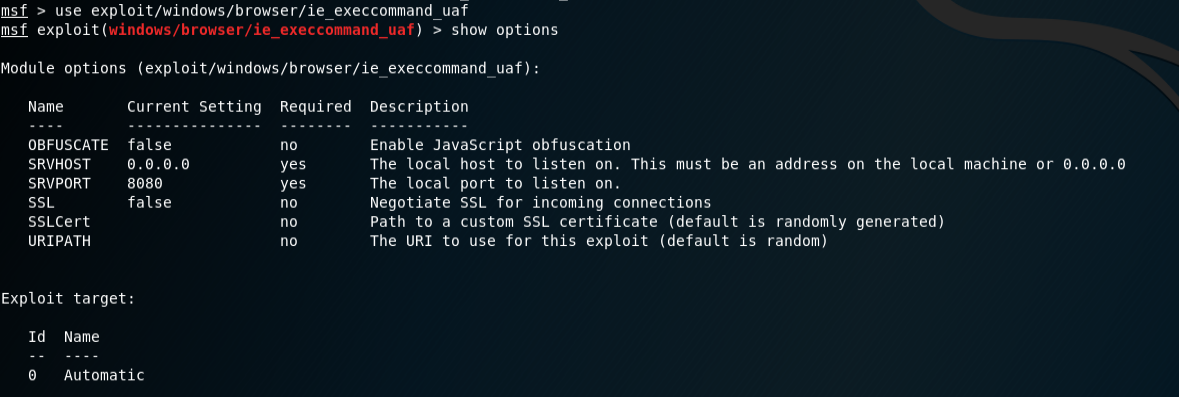
Clearlogs.exe -sec: Same effect as above, only with elevated privilege.

LINUX:

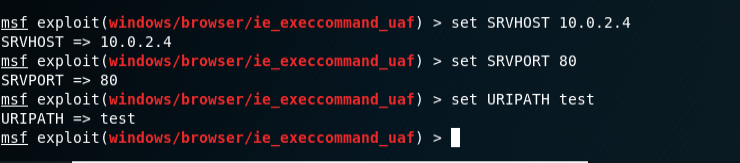
Kwrite /var/log/messages: This opens the log file. It requires a manual deletion of the log inputs.

Shred -zu root/.bash\_history: This command shreds and overwrites the history portion of Unix, which is used to document the input commands of the system.

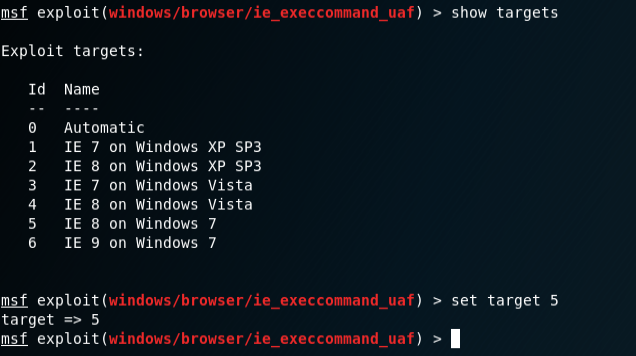
**Part B: ie\_execommand\_uaf**



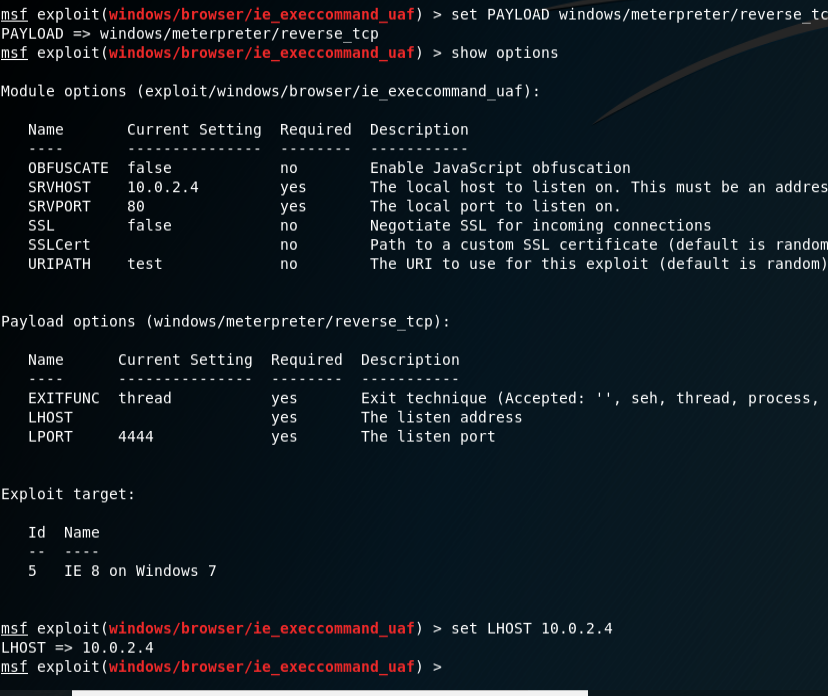
*Displaying the options of the ie\_execommand\_uaf exploit.*



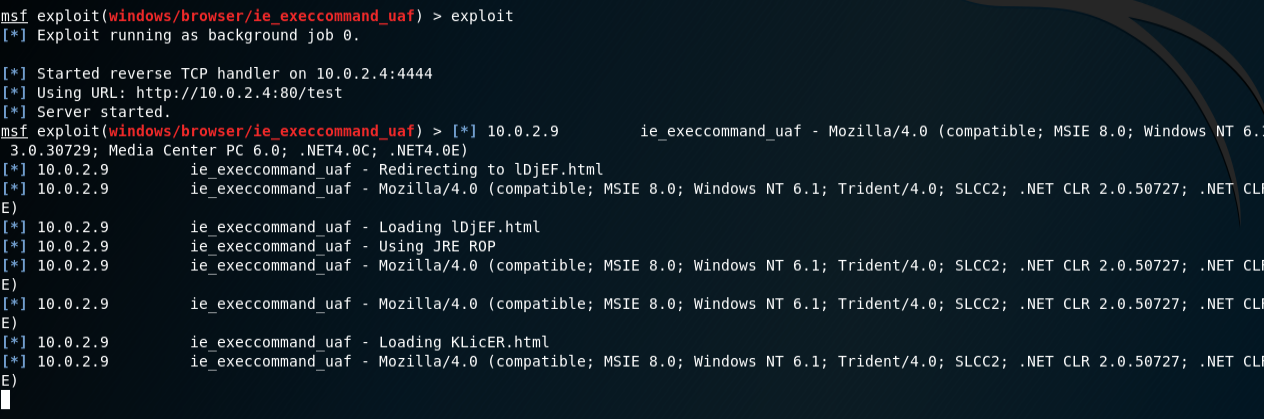
*Setting exploit options.*



*Shows the selection of targets that the exploit is valid with. I am using a WIN7 IE 8 target, so that is selected.*



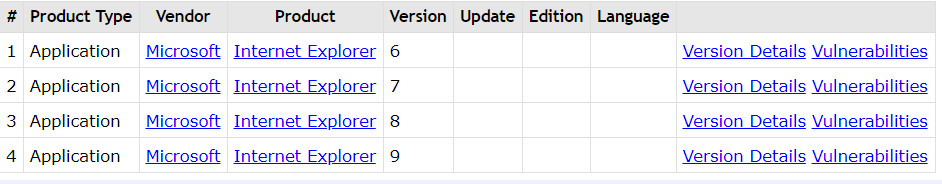
*Setting the payload options.*



*Exicuting the exploit, accessing the web browser from the target.*

**Observations for Part B:**

Unfortunately, the exploit never processed past the KLicER.html injection. I’ve reviewed the documentation for the exploit on various exploit information sites and the parameters seem to be valid:



This may require a deeper understanding of the exploit with examination of the code and interactions with other programs. I disabled the firewall via GUI and manually stopped the firewall services process, with no avail. Oh well, can’t win them all.