**Assignment:** Working with firewalls and Suricata.

**Required Resources**

* Ubuntu Apache Server created in previous assignment.
* Documentation as posted in BrightSpace Course Resources

**Professional Documentation**

All documentation must be done in a **professional style**. It must include:

* Title page
* **Updateable** Table of Contents
* Document introduction
* Section introductions or description, each section must be clearly identified
* Graphics or screenshots MUST include a title with a short description
* Any direct or copied quotes or graphics MUST be properly credited in a footnote
* ALL sources MUST be properly cited (APA format) and placed at the end of your document in a bibliography.
* **NO** embedded, zipped or compressed files. \*\* All scripts must be converted to text before including them in your documentation. \*\*
* **1 Professional Word Document ONLY.**

**Research and documentation sections** -Please complete all research and question responses in your own words. Research sections not completed in your own words may result in a mark of 0 for the section.

**NOTE:** Please do NOT copy and paste responses from internet, **even with a citation**. I expect each section or response to be in your own words. Be prepared to explain your responses and demonstrate your comprehension.

**No marks** will be given for cited or credited information included in document.

***\*\* I recommend completing any research section before completing any required task listed below as you will have a much better understanding of the material and data.***

**Evaluation:** This assignment is markedas per the attached Rubric (marks will be deducted for deviating from Requirements). \*\*You may be asked to demonstrate some of your assignment to show your comprehension of the material.

**Marking and Assignment Notes:**

* ScreenshotsMUST include user or device identifying information.
* Screenshots MUST be added to your document in the order of creation.
* Documentation must meet Professionalism requirements.
* **Automatic mark of 0 - Assignment not submitted or work not original.**

<http://www.nscc.ca/docs/about-nscc/policies-procedures/policy-studentcodeofconduct.pdf>

<https://www.nscc.ca/docs/about-nscc/policies-procedures/policy-academicintegrity.pdf>

**NOTE: This assignment may require some adaption, research and troubleshooting.**

**Task 1** – Working with Firewalls

Now we will created a dev or testing environment we will use it to practice our firewall settings and configurations.

* Run the following command:

**sudo iptables -L -v**

* Use the information and details from that command above along with the internet to answer the following questions:

1. What would the **full** command be to Replace rule number 3 from the Ouput chain?
2. What would the **full** command be to Delete a matching rule number 1 from the Input chain?
3. What would the **full** command be to Redirect a port to a different destination port in the nat table?
4. What would the **full** command be to Create a new chain called Sam1?
5. What would the **full** command be to Change the name of the Sam1 chain to Sample?
6. What would the **full** command be to Delete all the rules in the nat table?

Before we proceed we will review what rules are already applied.

* Check what rules are currently applied

**sudo iptables -L -n -v**

* Ping your Ubuntu server from your host machine to confirm your icmp packets are being accepted and returned.
* Now use your iptables to block all icmp traffic both incoming and outgoing. Note: you will need to add an argument to your protocol to identify what type of icmp traffic to include in the rule. You may use the argument “ **--icmp-type any**”.
* **Stop**. Take a screenshot of your iptables filter table rules that include your new icmp traffic rules.
* **Stop**. Take a screenshot of your host machine icmp packets no longer being accepted and returned.
* Now that we are confident that we add new rules to our iptables. Use the commands required to delete the icmp rule you created.
* **Stop**. Take a screenshot of your iptables filter table rules that shows your icmp traffic rules removed. Also, take a screenshot of your host machine able to ping your Ubuntu server.

Now that we have looked at blocking traffic based on protocol let’s look at redirecting traffic. One of the largest security issues when it comes to protocols and ports is the link to default ports. Similar to “administrator” or “root” users most protocols have default ports they use. It is always the best security process to not keep any default usernames, password, ports, etc.

Since we installed SSH which uses port 22 to pass through let’s look at modifying it to use an alternate port. In order to have our SSH request use an alternate port we will need to configure SSH to use the new port as well as create a firewall rule to close our old port and open a new port.

* First we confirm the original port 22 is listening for SSH requests by using the following command to checking our listening ports.

**ss -tulpn**

* View your listening ports
* Narrow your search by adding a grep

**ss -tulpn |grep 22**

* Now that you have confirmed you are using port 22 as a listening port we will confirm our server is responding to SSH requests.
* Confirm you can successfully SSH from host machine with your previously created user, if it is not, make any changes required to allow the ssh connection from your host machine.
* If successful log out of your SSH session.

Now that we have confirmed we can successfully SSH from our host machine to Ubuntu we will configure our Ubuntu server to use a new port for all SSH packets.

* First we will need to change to default port for SSH by modifying the following file using nano.

**sudo nano /etc/ssh/sshd\_config**

* remove the # from Port 22
* Change to Port 2222 # Modify default port “date” - “team name”
* Now restart the SSH service with the following command

**sudo systemctl restart sshd**

Now that we have set SSH to use a new port we will need to create some new firewall rules to listen and respond on the new port.

Using your iptables run the following commands:

**iptables -A INPUT -p tcp -m tcp --dport 2222 -j ACCEPT**

**iptables -A OUTPUT -p tcp --sport 2222 -m state --state ESTABLISHED -j ACCEPT**

* **Stop**. Run the command to view your listing ports and capture a screenshot to show you Ubuntu is now listening on port 2222.
* We will also test our new port by connecting from our host machine to Ubuntu, remember we will now require the port.

Now that we have created some new rules it is recommended that we create a copy of the rules for reference, records and back if required.

* Backup and export your iptables rules

**Sudo iptables-save >> IPTablesBK”YourTeamName”.txt**

Ex. …… >> IPTablesBKTeamMD.txt

* **Stop**. Add a copy of your iptables backup file to your documentation. Remember to keep it in a text format. NOTE: You may open your file and copy the text and paste it into Notepad++ on your workstation for format consistency.
* For further backup we will create a copy of all the Linux command we ran on our server. Use the following command to create a backup copy of your Linux Commands:

**History > HistoryReport”YourTeamName”.txt**

* **Stop**. Add a copy of your History backup file to your documentation. Remember to keep it in a text format. NOTE: You may open your file and copy the text and paste it into Notepad++ on your workstation.

We have spent some time looking at the Linux firewall environment, let’s look at the Windows firewall as well.

* Complete hands-On Projects 9-2 and 9-3 (pages 281-282) in your textbook with the following modifications:
* Project 9-2 (questions 7-9)
  + Question 7
    - Select Allow another app…
  + Question 8
    - Browse to %windir%/system32/mstsc.exe
  + Question 9
    - Select Network Types… answer the questions from the textbook.
* **Stop**. Capture the required screenshot to demonstrate successful complete of Project 9-2 and 9-3 of your text book and add them to your documentation. Make sure to correctly label the images to identify what part of the project your are showing.

**Task 2 – Install and configure a Suricata on your Ubuntu Server**

* Firstly, you need to add 3rd-party repo into the operation system. Just authorize as superuser and execute:

**sudo add-apt-repository ppa:oisf/suricata-stable**

Press **enter** to include entire repository.

* Now that we have added our repository lets run the command to update and upgrade
* Remember, if asked to always allow all required services to restart.
* Now we will update your packages and install our Suricata and start the suricate service

**sudo apt update && sudo apt install suricata -y**

* If asked, select Y to keep your current version.

**sudo systemctl enable suricata**

* Use the following command to ensure Suricata service s enabled and active

**sudo systemctl status suricata**

* Now that we are confident Suricata will start with our system startup we want to temporarily stop the service while we complete our configurations.

**sudo systemctl stop suricata**

* Before we configure suricata let’s look at the files in their different locations, it is important when working with linux application to understand the required files an file locations.

**Find / -type d -name suricata 2> /dev/null**

* Now let’s take a minute to review our suricata configuration file, we will use nano to make the file easier to read (colour codes). Review the type of information located in the file.

**nano /etc/suricata/suricata.yaml**

* While in your suricata.yaml file use the Where is command to find the default-path location of your rules directory.
* Suricata is a rules-based IDS, let’s look at what rules already exist in our rules directory with the following command.

**ll default-rule-path**

Now that we have looked through our rules directory and configuration file let’s make a few modifications.

* First use the ip command to identify your network, netmask and interfaces.
* Now that we have identified our network, we will start our configuration changes. Open your suricata configuration file in with elevated permissions in an editor to start making your changes.
  + Modify your HOME\_NET to match your current network.

\*Do NOT modify any of the secondary #HOME\_NET networks as they are not required.

Hint: you only need to modify the line to include your current network.

*Example “[192.168.xxx.xxx/xx]”*

* + Modify your file to identify the correct interface for both captured packets and pcap.

**af-packet:**

**– interface: “interface here”**

**pcap:**

**– interface: “interface here”**

* Save and exit your Yaml file.
* Now that we have made some change let’s update our suricata one more time to ensure we have the most up to date rules.

**sudo suricata-update**

Since we are working in a rules based environment we may want to add, remove and edit rules live with Suricata but we must setup that option first.

* We want to make sure any new rules are automatically included so we will add a requirement to auto reload to our configuration.
* Edit your suricata.yaml to add the following lines to the **bottom**.

Hint: Alt+/ will take you to the end of your document in nano.

**#Include new rules automatically – Date – Team Member Name**

**detect-engine:**

**rule-reload: true**

* First let’s update our Suricata with our additional change and then view our available rules

**sudo suricata-update**

**sudo suricata-update list-sources**

Now that we have updated and viewed available rule sets let’s select a couple of rule sets to add to your suricata but we will need to ensure the License is Open Source such as GPL or MIT.

* Add **tgreen/hunting, oisf/trafficid** plus **two** other open source rule sets.

**sudo suricata-update enable-source sourcenamehere**

* Now update suricata to ensure you have the most current version of the new rule sets.

Notice at the end of our update suricata does a small run test to check that your update was successful.

* Let’s **test** our suricata in a more verbose way by running it in test mode to ensure our new configuration is installed and configured correctly.

**sudo suricata -T -c /etc/suricata/suricata.yaml -v**

* If everything tested fine lets’ restart the suricata service

**sudo systemctl start suricata.service**

* We have now installed Suricata, configured it and added some additional rule sets. Let’s start working with our Suricata.
* Let’s have a look at the logs or files that are created to store or capture data or packets.

**ll /var/log/suricata**

* Confirm it is created the required log files we will use for this assignment.

**fast.log**

**eve.json**

* Now that we have installed, configured and added rule sets to our Suricata let’s test one of our rules to ensure our rules are working.

*Additional Learning: curl (Client URL) is a command tool used in Linux that allows data transfer over various protocols by communicating with a web server or application server using a specified URL and identifying the data that needs to be received or sent. Curl is similar to wget. Curl is not necessarily installed in all distros but may be installed if required.*

* We will run a couple of commands to Test Our IDS with a remote site.

**curl** [**http://testmynids.org/uid/index.html**](http://testmynids.org/uid/index.html)

* You should receive a response that identifies your uids as root. Now we will run an addition command to help us generate more test traffic.
* Let’s check our log to confirm we captured some data.

**sudo cat /var/log/suricata/fast.log**

Now we have have installed, configured and tested our Suricata and rules, lets get creative and create our own rule set.

* First we must stop our suricata server to make modifications.
* Use the command to confirm you have stopped the Suricata service
* Now we will create a new rule
* Use the following site to create the one custom **alert** rule that includes your name in the message.

<https://docs.suricata.io/en/latest/rule-management/adding-your-own-rules.html>

\*\* Make sure to create and save your rule file to the same location as your suricate yaml file.

* **Stop.** Create a section in your document to capture your rule and the details such as
* **Action** = alert
* **Headers =**
* **Rule Options =**
* **Rule Description =**

EXAMPLE:

**Rule:**

**alert icmp any any -> $HOME\_NET any (msg: “Ping Attempt”; sid:1; rev:1;)**

* **type** = alert
* **icmp any any** = any pings from any network or address
* **$HOME\_NET** = into our home network
* **Any** = any port
* **(…..)**  = message followed by message SID and version.
* **Rule Description** = This rule will create an alert for any incoming or outgoing ICMP requests by creating a message Ping Attempt and listing the message id and version.
* **Run the test** to confirm there is no issue with your custom rule or configuration modification. Fix any issues that arise.
* **Stop.** Capture a screenshot of the successful test run and add it to your documentation.
* If you test is success *“Configuration provided was successfully loaded. Exiting.”* then we will need to start our suricata.service
* Now you can test your new rule and confirm it was successful.
* **Stop**. Be prepared to test your custom rule and demonstrate the correct alert response was captured.

Let’s view our captured information in a different file.

* Open your captured data in your /var/log/suricata/eve.json file using the cat command and pipe it through he command to find only your alerts. Wow, that is a lot of information and it seems to change rapidly. Let’s see if there is a better way.
* To view json file with different details we must first install a json interpreter

**sudo apt install jq**

* Now that we have installed an interpreter let’s take another look at the file.
* Now run the following command to view our ping captures. \*NOTE: They are backticks not single quotes.

**sudo tail -f /var/log/suricata/eve.json | jq ‘select(.event\_type==”alert”)’**

* Notice the additional information we identified and how much easier it can be to read.
* Review and **record** the following information for the rule you created:
  + Message
  + Timestamp
  + Interface
  + Source IP
  + Destination IP
  + Protocol
  + Revision
  + Signature
  + Severity

It is important to keep an up to date record of all changes and modifications made to your server and have a reliable copy available as backup.

* Take a snapshot of your Ubuntu Server in the OFF state
* Create a “Gold” copy of your Ubuntu Server and include a screenshot of your Gold Copy properties with all required information.
* Make sure to update your change log for your web server and submitted the updated log.
* Include all requested **exports, transcripts, scripts and reports** in your documentation.
* Add all supplementary documentation, questions and screenshots to your professional document.
* **Upload your professional documentation to Brightspace.**

**Marking Rubrics**

|  |  |
| --- | --- |
| **Value** | **Task** |
|  | **Task 1-3 –** **Machine install and Firewall practice** |
| 2 | Question 1 |
| 2 | Question 2 |
| 2 | Question 3 |
| 2 | Question 4 |
| 2 | Question 5 |
| 2 | Question 6 |
| 3 | Screenshot of your iptables filter table rules that include your new icmp traffic rules. |
| 2 | Screenshot of your host machine icmp packets no longer being accepted and returned. |
| 4 | Screenshot of your iptables filter table rules that shows your icmp traffic rules removed. Also, take a screenshot of your host machine able to ping your Ubuntu server. |
| 2 | Screenshot to show you Ubuntu SSH is now listening on port 2222. |
| 4 | IPTables backup file with correct rules. |
| 4 | Linux History backup file which includes correct commands |
| 4 | Textbook Hands-On project 9-2 (completed with required modifications) |
| 2 | Textbook Hands-On project 9-3 |
| 5 | Suricate Rule creation details. |
| 4 | Screenshot of the successful test run and add it to your documentation. |
| 10 | Demonstration of rule test and rule will run and capture the required alert and data. |
| 10 | Alert Capture details from eve.json log:   * + Message   + Timestamp   + Interface   + Source IP   + Destination IP   + Protocol   + Revision   + Signature   + Severity |
| 1 | Web Server Snapshot |
| 5 | Change Log with all required content. |
| 1 | Gold copy screenshot with required details |
| 2 | Documentation professionalism |
| **75** | **Total Assignment Marks.** |