Lab 2: Signature based Detection using Suricata (Security Onion)

INFT 1202 – Dr. Sukhwant Sagar Total Marks: 75 Marks

Lab Introduction:

An **intrusion detection system** (**IDS**) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any malicious activity or violation is typically reported either to an administrator or collected centrally using a security information and event management **(SIEM)** system. A SIEM system combines outputs from multiple sources, and uses alarm filtering techniques to distinguish malicious activity from false alarms. IDS types range in scope from single computers to large networks. The most common classifications are **network intrusion detection systems** (**NIDS**) and **host-based intrusion detection systems** (**HIDS**). A system that monitors important operating system files is an example of an HIDS, while a system that analyzes incoming network traffic is an example of a NIDS.

It is also possible to classify IDS by detection approach: the most well-known variants are **Signature-based detection** (recognizing bad patterns, such as malware); and **Anomaly-based detection** (detecting deviations from a model of "good" traffic, which often relies on machine learning), another is **Reputation-based detection** (recognizing the potential threat according to the reputation scores).

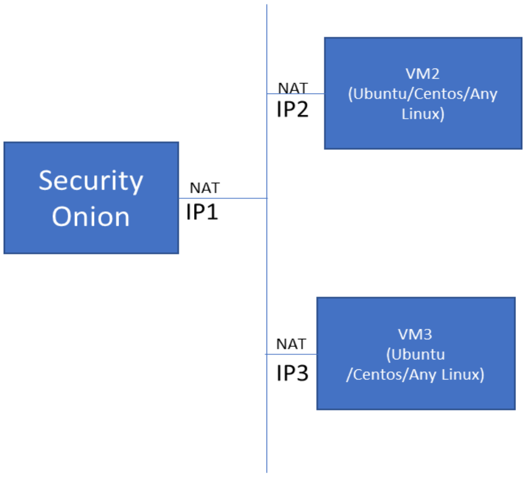
Suricata is an open-source intrusion detection and prevention system (IDPS) that can generate alerts when it detects suspicious network traffic. Suricata is a free and open source, mature, fast and robust network threat detection engine. Suricata inspects the network traffic using a powerful and extensive rules and signature language, and has powerful Lua scripting support for detection of complex threats. Suricata NIDS alerts can be found in Alerts, Dashboards, Hunt, and Kibana. Some examples of Suricata alerts include:

- A host on the network is scanning for vulnerabilities
- An attempted network intrusion
- A malicious file or payload being transmitted over the network
- A denial-of-service (DoS) attack
- Traffic from a known malicious IP address
- A suspicious protocol or application being used
- A violation of a security policy

Lab 2: Signature based Detection using Suricata (Security Onion)

INFT 1202 – Dr. Sukhwant Sagar Total Marks: 75 Marks

This lab consists of two scenarios of Signature-Based detection using Suricata in Security Onion:



Lab Requirements:

- **a)** Security Onion Virtual Machine
- **b)** Two Virtual machines having either Ubuntu/CentOS/Linux based OS
- **c)** Scapy tool for generation of traffic

Scenario 1:

1. One Virtual machine IP2 generate traffic through Scapy towards Security Onion machine IP1 with destination port 9595 and send Payload Content "Onionattack"
2. The Security Onion web interface detects the attack and show the alert "Security Onion under Attack".

Scenario 2:

1. One Virtual machine IP2 generate traffic through Scapy towards other virtual machine IP3 with source port 1236 and destination port 5002 and send Payload Content "Malicious"
2. The Security Onion web interface detects the attack and show the alert "Your Client VM under attack"

You are required to have screenshots wherever they are asked and answer questions. Please see submission requirements below in this document for the lab.

Lab 2: Signature based Detection using Suricata (Security Onion)

INFT 1202 – Dr. Sukhwant Sagar Total Marks: 75 Marks

Scenario I – Detecting an attack on Security Onion VM(port 9595) by one linux based machine [44 Marks]

1. Before configuring any rule, at Security Onion machine, (Switch to super user), go to /opt/so/rules/nids/. .
   a) What is the command to find the number of local rules? **[1 Mark]**
   b) How many rules are present in **local.rules**? **Attach the screenshot. [2 Marks]**
   c) How many rules are present in **all.rules**? **Attach the screenshot. [2 Marks]**
2. Add NIDS rule for the attack by going to Administration –> Configuration –> idstools in your Security Onion Webinterface.
   a) At the top of the page, click the Options menu and then enable the Show all configurable settings, including advanced settings. option.
   b) Then navigate to idstools –> rules –> Local Rules.
   c) Add your new rule(s) and click the checkmark to save them. **The rule is** : alert tcp Ubuntu1IP any -> SO_IP 9595 (msg: "Security Onion under Attack!"; content: "Onionattack"; flow:to_server; class_type:misc-activity; nocase; sid:your7digitid; rev:1;) **Attach the screenshot. [2 Marks]**
   d) The configuration will be applied at the next 15-minute interval or you can apply it immediately by clicking the SYNCHRONIZE GRID button under the Options menu.
   e) The next run of idstools should then merge /opt/so/rules/nids/local.rules into /opt/so/rules/nids/all.rules which is what Suricata reads from.
3. After addingAt Security Onion machine, (Switch to super user)
   a) How many rules are present now in **local.rules**? **Attach the screenshot. [2 Marks]**
   b) How many rules are present now in **all.rules**? **Attach the screenshot. [2 Marks]**
   c) Show the rule that gets added in local.rules at /opt/so/rules/nids/. . What is the command used? **Attach the screenshot. [1+2 Marks]**
   d) Update rules: sudo so-rule-update **[1 +1 + 1 + 2 Marks]**
      i. What is the location(complete path) of the loading of local file?

ii. How many total rules loaded?

iii. What is the location(complete path) where the local files are written or merged?

iv. Justify your answer with **screenshot**.

4. Perform the following steps in first Ubuntu / Centos machine: **[1 + 1 + 2 Marks]**
   - **a)** What is the version of python?
   - **b)** What is the ip address of inet Interface?
   - **c)** Install Scapy which is a Python program that enables the user to send, sniff and dissect and forge network packets by using command **pip install scapy**.
   - **d)** If you don't have pip installed in your VM, then you need to install pip by using **apt install pythonversion-pip** before step c)
   - **e)** Attach the screenshot of successful installation of scapy.

5. Packet generator/sniffer and network scanner/discovery (Python 3) Scapy is **a powerful interactive packet manipulation tool, packet generator, network scanner, network discovery, packet sniffer, etc**. It's time to craft the script for sending message to Security Onion Machine.
   - **a) From root**, go to scapy directory and type scapy and press enter. You will see prompt >>>
   - **b)** Create the script as follows(Include the comments):

     # Craft the layer 2 information.

     # The ip addresses can be random, but I would suggest sticking to RFC1918

     Ip = IP()

     ip.dst = " Security Onion Machine IP address"

     ip.src = " First Ubuntu VM IP address"

     # Craft the layer 3 information.

     # Since we specified port 9595 in our suricata rule,

     tcp = TCP()

     tcp.dport = 9595 (This has to be 9595 because you mentioned in your suricata rule)

     tcp.sport = 1234 (You can mention any source port other than assigned ports)

     # Set the playload

     payload = " Onionattack"

     # Use the / operator to compose our packet and transfer it with the send() method.

send(ip/tcp/payload)

send(ip/tcp/payload)

# you can send as many packets you want and once you are done, you can exit the Scapy tool

exit()

c) **Attach the screenshot** of your script. **[4 Marks]**

6. Go to web interface of Security onion . **Attach the screenshot** highlighting the attack. **[4 Marks]**
7. Answer the following: **[12 Marks]**
   - **a)** network.data.decoded
   - **b)** observer.name
   - **c)** rule.category
   - **d)** rule.metadata,policy
   - **e)** rule.name
   - **f)** rule.rule
   - **g)** rule.uuid
   - **h)** source.ip
   - **i)** source.port
   - **j)** destination.ip
   - **k)** destination.port
   - **l)** event.severity

Scenario II – Detecting an attack on one linux based machine from another linux based machine by Security Onion [31 Marks]

1. Configuring Local Rules on Security Onion Machine (Switch to super user)
2. Add NIDS rule for the attack by going to Administration –> Configuration –> idstools in your Security Onion Webinterface.
   - **a)** At the top of the page, click the Options menu and then enable the Show all configurable settings, including advanced settings. option.
   - **b)** Then navigate to idstools –> rules –> Local Rules.
   - **c)** Enter local rule in local.rules. The rule should include: **Attach the screenshot**. **[4 Marks]**

i. Alert with tcp protocol
ii. Source IP as your Ubuntu1/Centos1 VM(Scapy installed) and source port as 1236
iii. Destination IP as your second Ubuntu VM and destination port as 5002
iv. Msg as "Your client VM under attack!"
v. Content could be anything- eg – badpackets, malicious, badtraffic,etc.
vi. Sid number within the sid range.
vii. Construct your payload with whatever content you have chosen
viii. The payload should use nocase keyword to accept the content with no restriction on case of the payload.
ix. Class type needs to be bad-unknown
x. Revision of the alert can be anything from 1 to 10

d) The configuration will be applied at the next 15-minute interval or you can apply it immediately by clicking the SYNCHRONIZE GRID button under the Options menu.

e) The next run of `idstools` should then merge `/opt/so/rules/nids/local.rules` into `/opt/so/rules/nids/all.rules` which is what Suricata reads from.

3. At Security Onion machine, (Switch to super user)

a. How many rules are now present in **local.rules**? **Attach the screenshot**. **[2 Marks]**

b. How many rules are now present in **all.rules**? **Attach the screenshot**. **[2 Marks]**

c. Update rules: sudo so-rule-update

d. Show the rule that gets added in local.rules at `/opt/so/rules/nids/.`**.** What is the command used? **Attach the screenshot**. **[1+2 Marks]**

4. Perform the following steps in first Ubuntu / Centos machine[Machine where Scapy is already installed from Scenario I]:

a) From root, go to scapy directory and type scapy and press enter. You will see prompt >>>

c) Generate the traffic from first Ubuntu VM to second Ubuntu VM. Create the script **including comments** as per the rule made in step 1b). Attach the **screenshot of the script** after sending 3-4 packets. **[4 Marks]**

5. Go to web interface of Security onion . **Attach the screenshot** highlighting the attack. **[4 Marks]**

6. Answer the following:**[12 Marks]**
   - **a)** network.data.decoded
   - **b)** observer.name
   - **c)** rule.category
   - **d)** rule.metadata,policy
   - **e)** rule.name
   - **f)** rule.rule
   - **g)** rule.uuid
   - **h)** source.ip
   - **i)** source.port
   - **j)** destination.ip
   - **k)** destination.port
   - **l)** event.severity

Things to Explore:

You are welcome to explore beyond the mandatory requirements if you wish.

General Submission Requirements

- Include an opening comment with your full name and a short description of the lab. Please name your file as fistname_Lab2.pdf.
- The assignment asks you to provide response on the questions being asked. At times you have to also provide screenshot for the work you have done. Please make sure you use the **exact numbering scheme as used in the assignment** while reponding to questions. The assignment needs to be submitted in pdf with much clarity on the screenshot.
- If a particular point does not ask any question and rather it's a sequence step, please write N/A in the response of that question.

- Please make sure all your screenshots should have your name as name of virtual machine and time stamp of your machine. This is primarily needed to make sure that each student should perform the lab on their own.
- Academic Integrity violations would be treated severely.