

Project Report: The Network Intrusion Detection System using Suricata

6th February 2025

Domain : Cybersecurity

NAVNEET BIJALWAN

INTRODUCTION

In an increasingly digital world, network security has become paramount for organizations seeking to protect sensitive data and maintain system integrity. This project aimed to develop a robust Network Intrusion Detection System (NIDS) utilizing Suricata on Kali Linux. The system was designed to monitor network traffic, detect suspicious activities, and provide alerts to users. Furthermore, visualizations were employed to analyze and understand detected attacks, enabling better security posture and response strategies. The project successfully achieved its objectives, providing a comprehensive solution for network intrusion detection and analysis.

Project Objectives

The primary objectives of the project were as follows:

1. **Develop NIDS:** Create a system capable of monitoring network traffic and detecting intrusions effectively.
2. **Configure Rules and Alerts:** Set up rules that identify and respond to suspicious activities in real-time.
3. **Visualize Detected Attacks:** Utilize visualization tools to analyze and understand detected threats, allowing for proactive security measures.



Project Scope

The scope of this project included the following:

1. **Installation and Configuration of Suricata:** Setting up Suricata on Kali Linux to monitor network traffic.
2. **Rule Creation and Implementation:** Developing custom rules to detect specific network activities.
3. **Monitoring and Alerts Setup:** Configuring alerts for suspicious activities and monitoring network traffic.
4. **Visualization and Analysis:** Using Wireshark and Elastic Stack to visualize detected attacks and analyze data.
5. **Documentation:** Compiling a comprehensive project report detailing the entire process, including configurations, rules, challenges, and solutions.

This report outlines the project's objectives, scope, key deliverables, methodologies, challenges encountered, and future recommendations.

Key Deliverables

The following deliverables were produced during the project:

1. **Suricata Installation and Configuration:** A fully functional installation of Suricata on Kali Linux, configured to monitor the specified network interface.
2. **Custom Rules:** A set of custom rules created and implemented for detecting various types of network intrusions.
3. **Alerts and Monitoring:** A configured alert system that effectively notifies users of suspicious network activities.
4. **Visualization:** Utilization of Wireshark and Elastic Stack to create visual representations of detected attacks for further analysis.
5. **Project Report:** A detailed documentation of the entire process, including installation steps, rules created, challenges faced, and solutions implemented.

Project Phases

4.1 Research and Planning

- Conducted thorough research on Suricata and Kali Linux.
- Defined project requirements and planned the project structure.

4.2 Installation and Configuration

- **Installation:** Suricata was installed using the command:
bash
sudo apt-get install suricata
- **Configuration:** The Suricata configuration file was edited to configure the network interface for monitoring.

Installing suricata with ICMP log collection:

Step 1: Installing Suricata on Kali Linux

1.1 Update Your System-

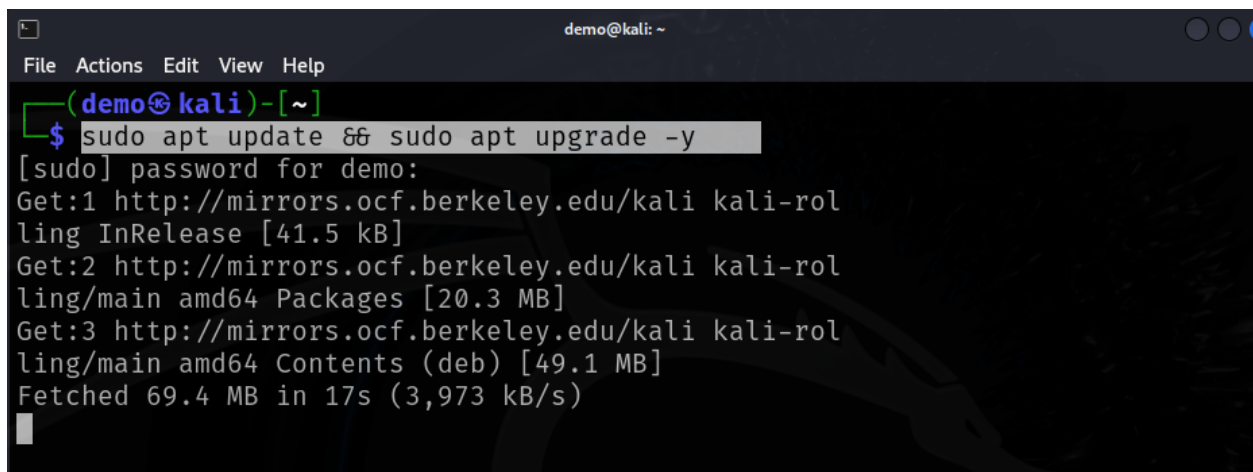
First, make sure your system is up to date.

1. Open a terminal window.
2. Update your package list:

sudo apt-get update

3. Upgrade your existing packages:

sudo apt-get upgrade



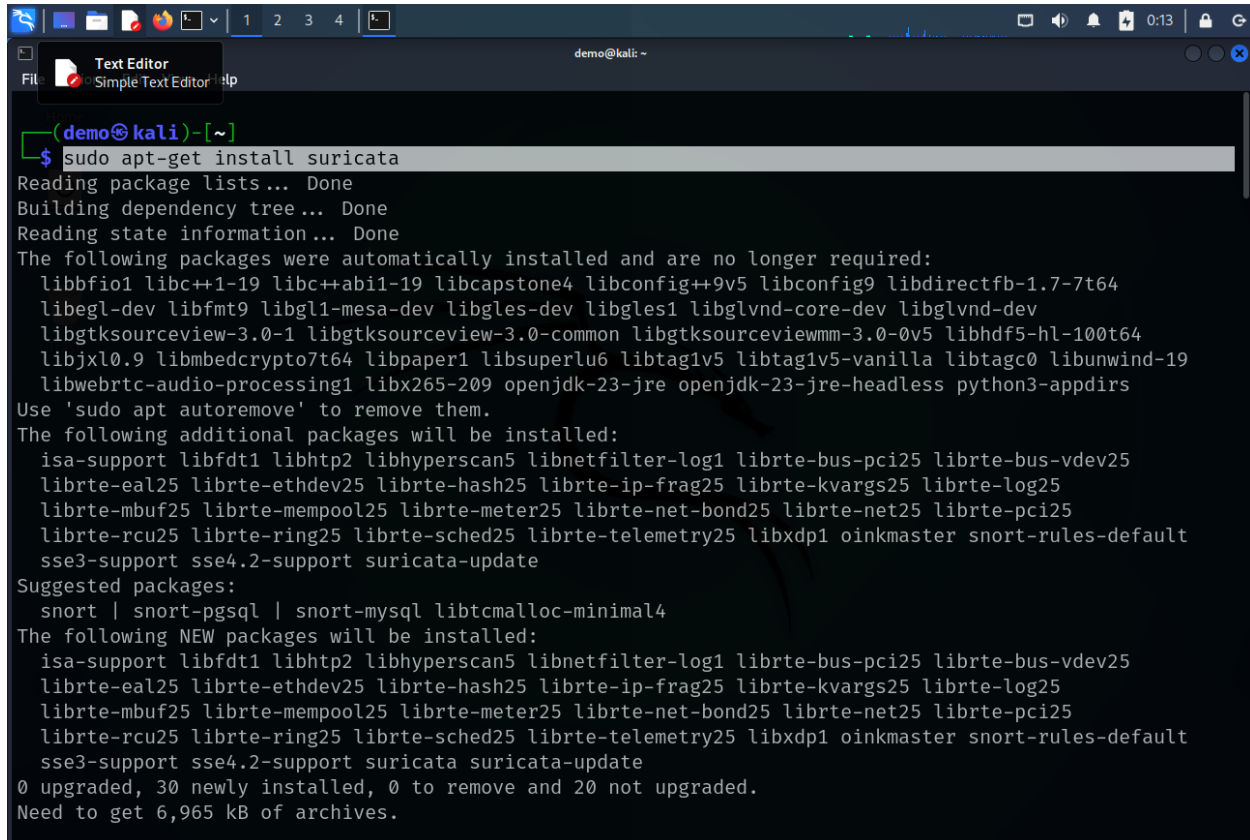
```
demo@kali: ~  
File Actions Edit View Help  
(demo@kali)-[~]  
$ sudo apt update && sudo apt upgrade -y  
[sudo] password for demo:  
Get:1 http://mirrors.ocf.berkeley.edu/kali kali-rol  
ling InRelease [41.5 kB]  
Get:2 http://mirrors.ocf.berkeley.edu/kali kali-rol  
ling/main amd64 Packages [20.3 MB]  
Get:3 http://mirrors.ocf.berkeley.edu/kali kali-rol  
ling/main amd64 Contents (deb) [49.1 MB]  
Fetched 69.4 MB in 17s (3,973 kB/s)
```

1.2 Install Suricata

Kali Linux provides Suricata in its repositories, so you can install it directly.

1. Install Suricata:

```
sudo apt-get install suricata
```



```
(demo@kali)-[~]
$ sudo apt-get install suricata
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
 libbfbio1 libc++1-19 libc++abi1-19 libcapstone4 libconfig++9v5 libconfig9 libdirectfb-1.7-7t64
 libegl-dev libfmt9 libgl1-mesa-dev libgles-dev libgles1 libglvnd-core-dev libglvnd-dev
 libgtksourceview-3.0-1 libgtksourceview-3.0-common libgtksourceviewmm-3.0-0v5 libhdf5-hl-100t64
 libjxl0.9 libmbedcrypto7t64 libpaper1 libsuperlu6 libtag1v5 libtag1v5-vanilla libtagc0 libunwind-19
 libwebrtc-audio-processing1 libx265-209 openjdk-23-jre openjdk-23-jre-headless python3-appdirs
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 isa-support libfdt1 libhttp2 libhyperscan5 libnetfilter-log1 librt-eal25 librt-ethdev25 librt-hash25 librt-ip-frag25 librt-kvargs25 librt-log25
 librt-mbuf25 librt-mempool25 librt-meter25 librt-net-bond25 librt-net25 librt-pci25
 librt-rcu25 librt-ring25 librt-sched25 librt-telemetry25 libxdp1 oinkmaster snort-rules-default
 sse3-support sse4.2-support suricata-update
Suggested packages:
 snort | snort-pgsql | snort-mysql libtcmalloc-minimal4
The following NEW packages will be installed:
 isa-support libfdt1 libhttp2 libhyperscan5 libnetfilter-log1 librt-eal25 librt-ethdev25 librt-hash25 librt-ip-frag25 librt-kvargs25 librt-log25
 librt-mbuf25 librt-mempool25 librt-meter25 librt-net-bond25 librt-net25 librt-pci25
 librt-rcu25 librt-ring25 librt-sched25 librt-telemetry25 libxdp1 oinkmaster snort-rules-default
 sse3-support sse4.2-support suricata suricata-update
0 upgraded, 30 newly installed, 0 to remove and 20 not upgraded.
Need to get 6,965 kB of archives.
```

Step 2: Configuring Suricata

2.1 Navigate to the Configuration Directory

Suricata's configuration files and rules are located in the `/etc/suricata` directory.

1. Navigate to the rule's directory:

```
cd /etc/suricata/rules
```

2.2 Create a Custom Rule File

We will create a custom rule file named `local.rules` to define our own detection rules.

1. Create and open the local.rules file:

sudo nano local.rules

2. Add the following example rules to detect ICMP (ping) and HTTP traffic:

```
Processing triggers for libc-bin (2.40-3) ...
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for kali-menu (2025.1.1) ...

(demo@kali)~$
$ cd /etc/suricata/rules

(demo@kali)~/etc/suricata/rules$
$
```

Local Rules Files

```
suricata.service is a disabled or a static unit, not starting it.
Processing triggers for libc-bin (2.40-3) ...
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for kali-menu (2025.1.1) ...

(demo@kali)~$
$ cd /etc/suricata/rules

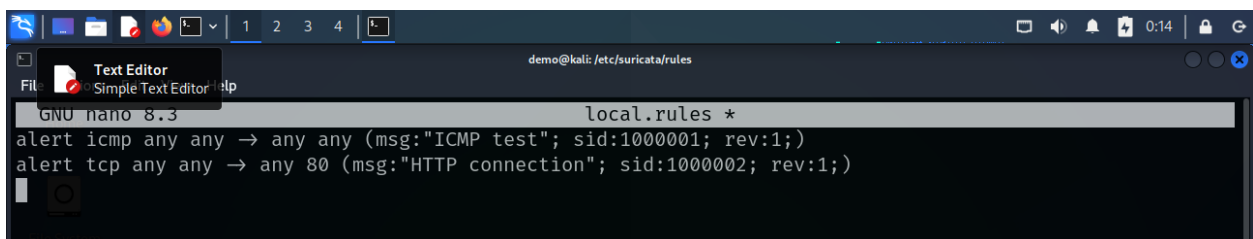
(demo@kali)~/etc/suricata/rules$
$ sudo nano local.rules

(demo@kali)~/etc/suricata/rules$
$
```

Adding rules

alert icmp any any -> any any (msg:"ICMP test"; sid:1000001; rev:1;)

alert tcp any any -> any 80 (msg:"HTTP connection"; sid:1000002; rev:1;)



```
demo@kali: /etc/suricata/rules
GNU nano 8.3 local.rules *
alert icmp any any -> any any (msg:"ICMP test"; sid:1000001; rev:1;)
alert tcp any any -> any 80 (msg:"HTTP connection"; sid:1000002; rev:1;)
```

- o alert specifies the action to take (alert in this case).
- o icmp and tcp indicate the protocols to monitor.

- any any -> any any specifies the source and destination IP addresses and ports (any means any IP or port).
- msg provides a message to display when the rule is triggered.
- sid is the unique rule identifier

2.3 Edit the Suricata Configuration File

Next, we need to configure Suricata to include our custom rule file.

1. Open the main Suricata configuration file:

```
sudo nano /etc/suricata/suricata.yaml
```

2. Add the following line under the rule-files section:

```
- /etc/suricata/rules/local.rules
```

```
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /var/lib/suricata/rules

rule-files:
- suricata.rules
- /etc/suricata/rules/local.rules

##
## Auxiliary configuration files.
##

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config
```

3. Save and close the file (Ctrl + X, then Y, then Enter).

Step 3: Running Suricata

1. Run Suricata in IDS mode:

```
sudo suricata -c /etc/suricata/suricata.yaml -i eth0
```


- `-c /etc/suricata/suricata.yaml` specifies the configuration file to use.
- `-i eth0` specifies the network interface to monitor (replace eth0 with your actual network interface).

```
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for kali-menu (2025.1.1) ...

(demo@kali)-[~]
$ cd /etc/suricata/rules

(demo@kali)-[/etc/suricata/rules]
$ sudo nano local.rules

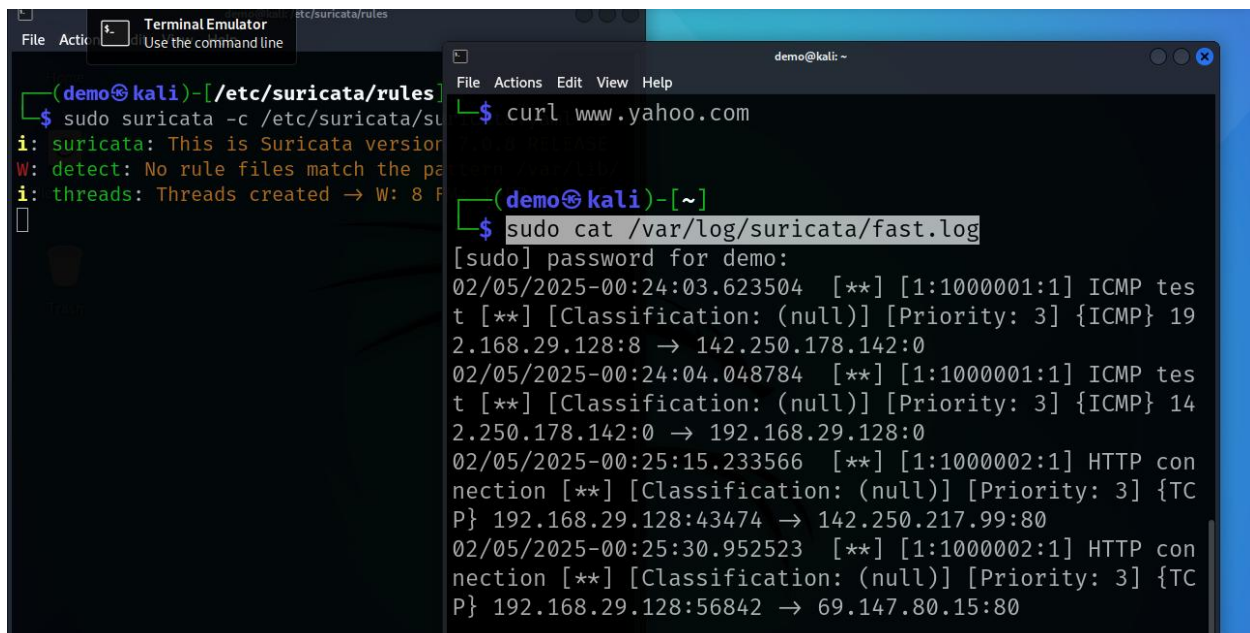
(demo@kali)-[/etc/suricata/rules]
$ sudo nano /etc/suricata/suricata.yaml

(demo@kali)-[/etc/suricata/rules]
$ sudo suricata -c /etc/suricata/suricata.yaml -i eth0
i: suricata: This is Suricata version 7.0.8 RELEASE running in SYSTEM mode
W: detect: No rule files match the pattern /var/lib/suricata/rules/suricata.rules
i: threads: Threads created → W: 8 FM: 1 FR: 1 Engine started.
```

Step 4: Analyzing Logs and Alerts

1. View Suricata alert logs:

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



```

(demo@kali)-[/etc/suricata/rules]
$ sudo suricata -c /etc/suricata/suricata.yaml -i eth0
i: suricata: This is Suricata version 7.0.8 RELEASE running in SYSTEM mode
W: detect: No rule files match the pattern /var/lib/suricata/rules/suricata.rules
i: threads: Threads created → W: 8 FM: 1 FR: 1 Engine started.

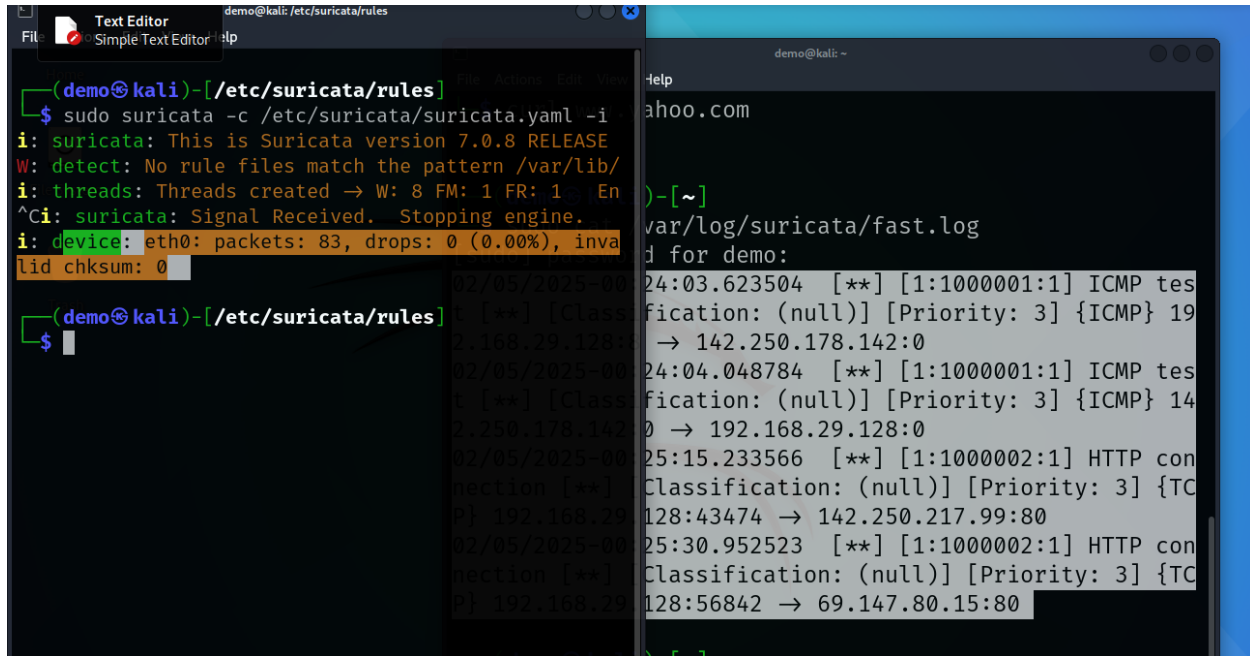
demo@kali: ~
$ curl www.yahoo.com

(demo@kali)-[~]
$ sudo cat /var/log/suricata/fast.log
[sudo] password for demo:
02/05/2025-00:24:03.623504  [**] [1:1000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.29.128:8 → 142.250.178.142:0
02/05/2025-00:24:04.048784  [**] [1:1000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 142.250.178.142:0 → 192.168.29.128:0
02/05/2025-00:25:15.233566  [**] [1:1000002:1] HTTP connection [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.29.128:43474 → 142.250.217.99:80
02/05/2025-00:25:30.952523  [**] [1:1000002:1] HTTP connection [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.29.128:56842 → 69.147.80.15:80

```

Generate Some ICMP Traffic:

ping -c 4 <target_ip>



```
(demo@kali)-[/etc/suricata/rules]
$ sudo suricata -c /etc/suricata/suricata.yaml -i eth0
i: suricata: This is Suricata version 7.0.8 RELEASE
W: detect: No rule files match the pattern /var/lib/suricata/rules
i: threads: Threads created -> W: 8 FM: 1 FR: 1 En
^Ci: suricata: Signal Received. Stopping engine.
i: device: eth0: packets: 83, drops: 0 (0.00%), invalid checksum: 0

02/05/2025-00:24:03.623504 [**] [1:1000001:1] ICMP test [**]
[Classification: (null)] [Priority: 3] {ICMP} 192.168.29.128:8 ->
142.250.178.142:0
02/05/2025-00:24:04.048784 [**] [1:1000001:1] ICMP test [**]
[Classification: (null)] [Priority: 3] {ICMP} 142.250.178.142:0 ->
192.168.29.128:0
02/05/2025-00:25:15.233566 [**] [1:1000002:1] HTTP connection [**]
[Classification: (null)] [Priority: 3] {TCP} 192.168.29.128:43474 ->
142.250.217.99:80
02/05/2025-00:25:30.952523 [**] [1:1000002:1] HTTP connection [**]
[Classification: (null)] [Priority: 3] {TCP} 192.168.29.128:56842 ->
69.147.80.15:80
```

Detailed Explanation for the First Log Entry:

02/05/2025-00:24:03.623504 [**] [1:1000001:1] ICMP test [**]
[Classification: (null)] [Priority: 3] ICMP} 192.168.29.128:8 ->
142.250.178.142:0

- **Timestamp: 02/05/2025-00:24:03.623504**

- The date and time when the event were logged (2nd February 2025, at 00:24:03.623504).

- **Alert Message:** [1:1000001:1] ICMP test

- [1:1000001:1] is the rule identifier (SID: 1000001, revision: 1) that triggered this alert.
- The ICMP test is the custom message defined in the rule to describe the alert.

- **Classification:** [Classification: (null)]

- The classification of the alert is not specified (null).

- **Priority:** [Priority: 3]

- The priority level of this alert is 3, indicating a moderate level of importance.

- **Protocol:** ICMP}

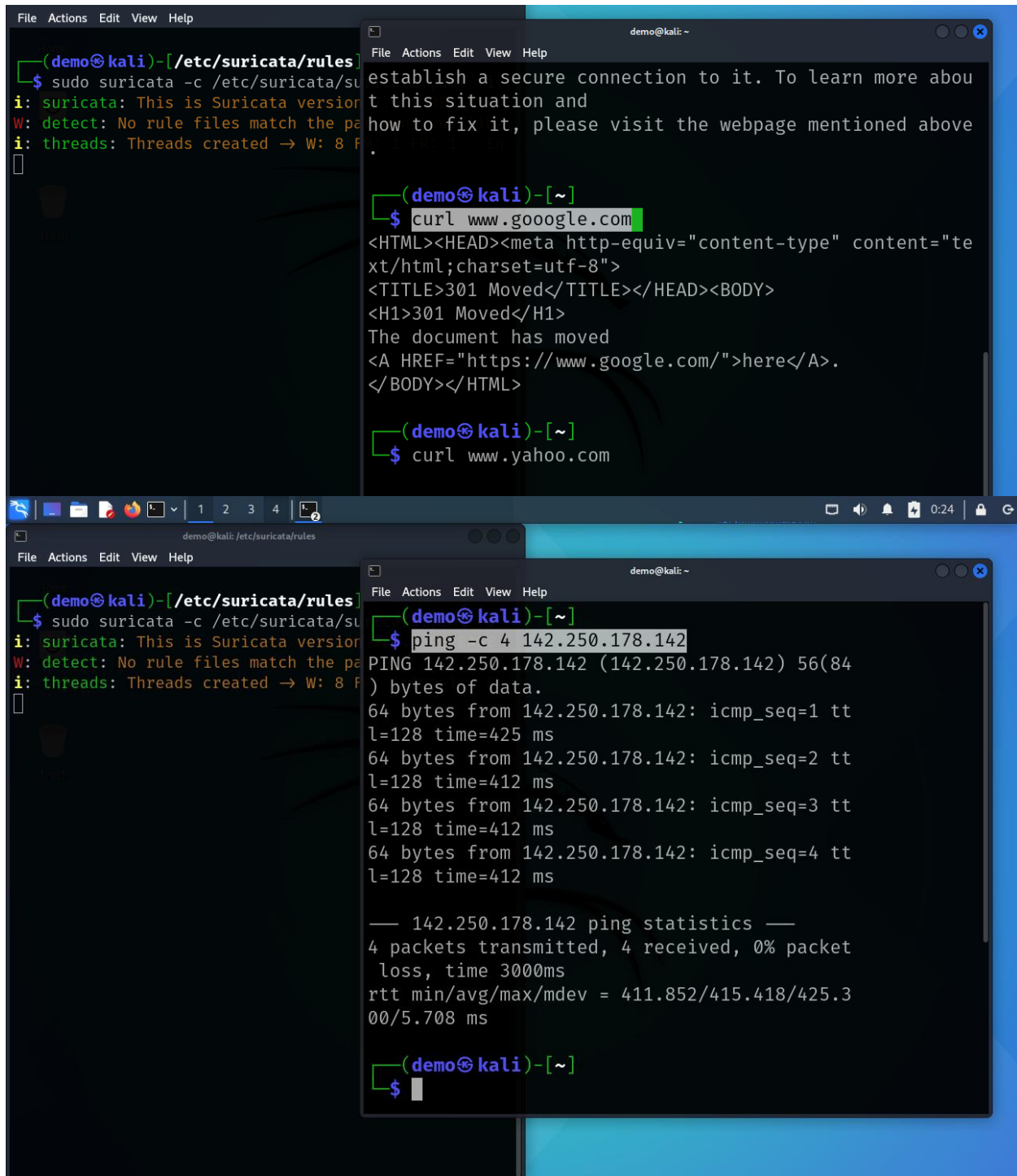
- The protocol involved in this alert is ICMP (Internet Control Message Protocol).

- **Source and Destination:** 192.168.29.128:8 -> 142.250.178.142:0

- 192.168.29.128:8 represents the source IP address (192.168.29.128) and source port (8) of the ICMP packet.
- 142.250.178.142:0 represents the destination IP address (142.250.178.142) and destination port (0) of the ICMP packet.

1. Generate Some HTTP Traffic:

`curl http://google.com <example.com>`



```
(demo@kali)-[/etc/suricata/rules]
$ sudo suricata -c /etc/suricata/suricata.yaml
i: suricata: This is Suricata version 3.0.0
W: detect: No rule files match the path /etc/suricata/rules
i: threads: Threads created -> W: 8 F

(demo@kali)-[~]
$ curl www.google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="https://www.google.com/">here</A>.
</BODY></HTML>

(demo@kali)-[~]
$ curl www.yahoo.com

(demo@kali)-[/etc/suricata/rules]
$ sudo suricata -c /etc/suricata/suricata.yaml
i: suricata: This is Suricata version 3.0.0
W: detect: No rule files match the path /etc/suricata/rules
i: threads: Threads created -> W: 8 F

(demo@kali)-[~]
$ ping -c 4 142.250.178.142
PING 142.250.178.142 (142.250.178.142) 56(84) bytes of data.
64 bytes from 142.250.178.142: icmp_seq=1 ttl=128 time=425 ms
64 bytes from 142.250.178.142: icmp_seq=2 ttl=128 time=412 ms
64 bytes from 142.250.178.142: icmp_seq=3 ttl=128 time=412 ms
64 bytes from 142.250.178.142: icmp_seq=4 ttl=128 time=412 ms

— 142.250.178.142 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 411.852/415.418/425.300/5.708 ms

(demo@kali)-[~]
$
```

Detailed Explanation for the First Log Entry:

02/05/2025-00:38:18.883975 [**] [1:1000002:1] HTTP connection [**]
[Classification: (null)] [Priority: 3] TCP} 192.168.29.128:37936 ->
92.249.39.133:80

- **Timestamp:** 02/05/2025-00:38:18.883975

- The date and time when the event was logged: February 5th, 2025, at 00:38:18.883975.

- **Alert Message:** [1:1000002:1] HTTP connection

- [1:1000002:1] is the rule identifier, where 1 is the generator ID, 1000002 is the rule ID (SID), and 1 is the revision number.
- HTTP connection is the custom message defined in the rule to describe the alert.

- **Classification:** [Classification: (null)]

- The classification of the alert is not specified (null).

- **Priority:** [Priority: 3]

- The priority level of this alert is 3, indicating a moderate level of importance.

- **Protocol:** {TCP}

- The protocol involved in this alert is TCP (Transmission Control Protocol).

- **Source and Destination:** 192.168.29.128:37936 ->
92.249.39.133:80

- 192.168.29.128:37936 represents the source IP address (192.168.29.128) and source port (37936) of the TCP packet.
- 92.249.39.133:80 represents the destination IP address (92.249.39.133) and destination port (80) of the TCP packet.

2. View Suricata Alert Logs:

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

Step 5: Visualizing Detected Attacks

5.1 Install Elasticsearch, Logstash, and Kibana (ELK Stack)

1. Install Elasticsearch:

```
sudo apt-get install elasticsearch
```

2. Install Logstash:

```
sudo apt-get install logstash
```

3. Install Kibana:

```
sudo apt-get install kibana
```

5.2 Configure Logstash to Parse Suricata Logs

1. Create a Logstash configuration file (e.g., suricata.conf):

```
sudo nano /etc/logstash/conf.d/suricata.conf
```

2. Add the right configuration to parse Suricata logs.

3. Start Logstash with the configuration file:

```
sudo service logstash start
```

4.3 Rule Creation and Implementation

- Created a custom rule file and added rules to detect suspicious activities, such as HTTP connections.
- Example rules included:

```
alert tcp any any -> 192.168.1.0/24 80 (msg:"HTTP connection detected";  
sid:1000001;)
```

4.4 Monitoring and Alerts

- Set up monitoring and alert systems to notify users of detected threats.
- Monitored logs using:

```
tail -f /var/log/suricata/fast.log
```

4.5 Visualization and Analysis

- **Nmap or Wireshark:** Captured traffic and applied filters to analyze specific traffic patterns.
- **Elastic Stack:** Configured to visualize logs and create dashboards for better insights.

Other examples for NIDS

There are several interesting and practical examples you can explore with Suricata under Network Intrusion Detection System (NIDS). Here are some ideas to get you started:

1. Detecting Port Scanning

- **Objective:** Identify and alert port scanning activities, which are often used by attackers to find open ports and vulnerabilities.
- **Rule Example:**

```
alert tcp any any -> any any (msg:"Possible Port Scan"; flags:S;  
threshold:type both, track by_src, count 20, seconds 10; sid:1000003;  
rev:1;)
```

2. Detecting SSH Brute Force Attacks

- **Objective:** Detect multiple failed SSH login attempts, which may indicate a brute force attack.
- **Rule Example:**

```
(alert tcp any any -> any 22 (msg:"Possible SSH Brute Force Attack";  
flow:to_server,established; content:"SSH"; depth:3; detection_filter:track  
by_src, count 5, seconds 60; sid:1000004; rev:1;)
```

3. Detecting Malicious DNS Requests

- **Objective:** Identify DNS queries to known malicious domains.
- **Rule Example:**

```
(Alert udp any any -> any 53 (msg:"Malicious DNS Query";  
content:"maliciousdomain.com"; sid:1000005; rev:1;)
```

4. Detecting SQL Injection Attempts

- **Objective:** Detect SQL injection attempts in web application traffic.
- **Rule Example:**

```
alert http any any -> any any (msg:"SQL Injection Attempt";  
content:"SELECT"; nocase; http_uri; sid:1000006; rev:1;)
```

5. Detecting Suspicious File Transfers

- **Objective:** Monitor and alert suspicious file transfers, such as large files or files with certain extensions.
- **Rule Example:**

```
(alert ftp any any -> any any (msg:"Suspicious File Transfer";  
content:".exe"; flow:to_server,established; sid:1000007; rev:1;)
```


6. Detecting Heartbleed Exploit Attempts

- **Objective:** Detect attempts to exploit the Heartbleed vulnerability in OpenSSL.
- **Rule Example:**

```
(Alert tls any any -> any any (msg:"Heartbleed Exploit Attempt";  
content:"|18 03 02|"; depth:3; content:"|0B 00|"; within:2; sid:1000008;  
rev:1;)
```

7. Detecting Ransomware Traffic

- **Objective:** Identify traffic patterns associated with ransomware, such as communication with known ransomware command and control (C2) servers.
- **Rule Example:**

```
(Alert http any any -> any any (msg:"Ransomware Traffic";  
content:"/api/v1/encryption"; http_uri; sid:1000009; rev:1;)
```

8. Detecting Data Exfiltration Attempts

- **Objective:** Detect attempts to exfiltrate sensitive data from your network.
- **Rule Example:**

```
(alert http any any -> any any (msg:"Data Exfiltration Attempt";  
content:"confidential"; http_client_body; sid:1000010; rev:1;)
```

9. Detecting Beaconing Activity

- **Objective:** Identify beaconing behavior, where malware communicates with its C2 server at regular intervals.
- **Rule Example:**

```
(alert http any any -> any any (msg:"Beaconing Activity";  
content:"/beacon"; http_uri; sid:1000012; rev:1;)
```

Challenges and Solutions

5.1 Challenges Faced

- **Configuration Errors:** Incorrect configuration could lead to ineffective detection.
- **Rule Complexity:** Creating complex rules posed challenges in detection accuracy.
- **Data Volume:** Handling large volumes of network traffic data for analysis.

5.2 Mitigation Strategies

- **Testing and Validation:** Conducted thorough testing to validate configurations.
- **Utilization of Resources:** Leveraged examples and resources from trusted sources for rule creation.
- **Efficient Tools:** Employed efficient data management and visualization tools to handle large datasets.

Conclusion

The project successfully developed a Network Intrusion Detection System using Suricata on Kali Linux, effectively monitoring network traffic and identifying potential threats. The system's ability to detect and alert users of suspicious activities, combined with powerful visualization tools, provides a comprehensive solution for enhancing network security. Future improvements could involve refining detection rules, integrating advanced visualization techniques, and automating responses to detected threats.

References

- **Suricata Documentation:** [Suricata Docs](#)
- **Wireshark User Guide:** [Wireshark Docs](#)
- **Elastic Stack Documentation:** [Elastic Docs](#)
- **Nmap -** <https://nmap.org/>
- **Kali Linux Documentation:** [Kali Docs](#)

THANK YOU