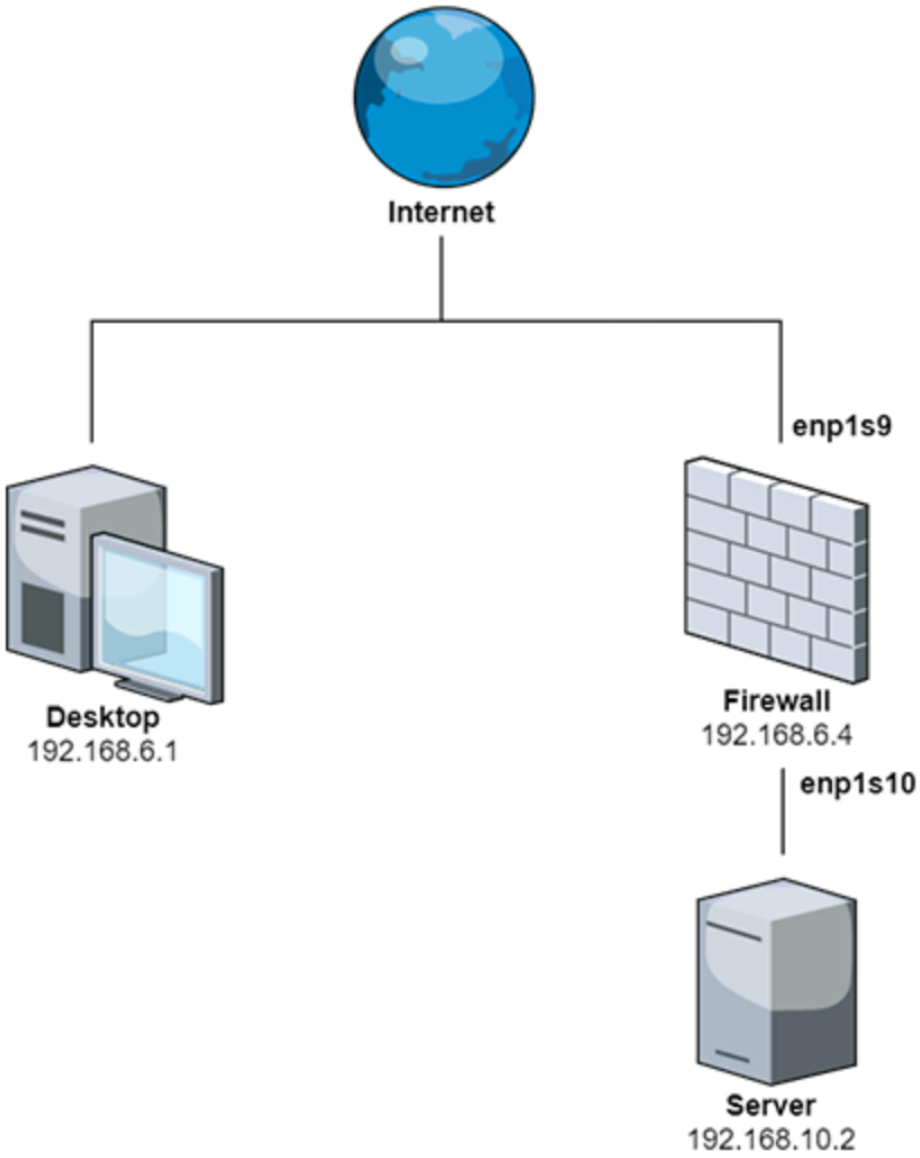
Snort IDS/IPS Configuration and Analysis

Project Overview

This report details the configuration, execution, and analysis of a Snort-based Intrusion Detection and Prevention System (IDS/IPS). The project's objectives included setting up Snort on a firewall, configuring the environment, enabling logging, and analyzing network traffic to identify malicious activities, specifically detecting a suspicious User-Agent string.

Network Topology

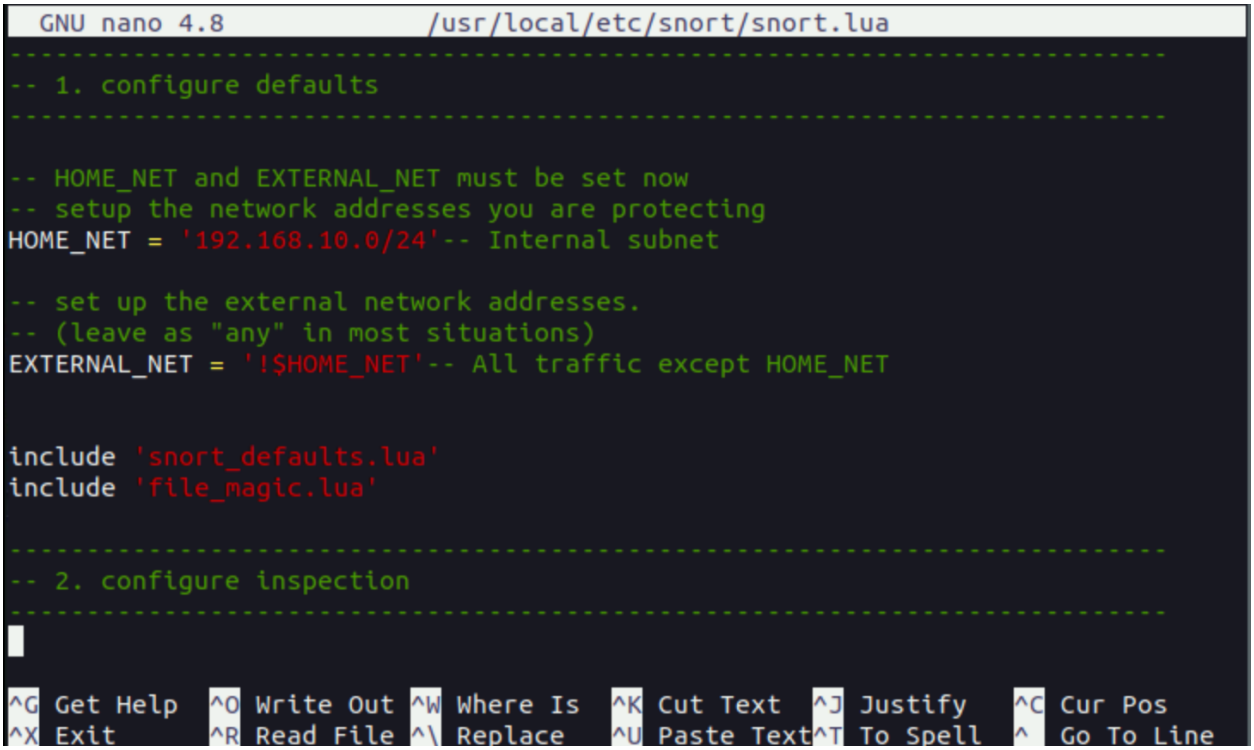


The network used for this project consisted of the following components:

- **Desktop:** 192.168.6.1 (used for management)
- **Firewall:**
    - External Interface: 192.168.6.4
    - Internal Interface: 192.168.10.254
- **Server:** 192.168.10.2 (hosted behind the firewall)

All network traffic between the desktop and server was routed through the firewall, making it an ideal location to deploy Snort IDS/IPS for monitoring.

Snort Configuration



1. Internal and External Network Variables

The following steps were taken to configure Snort's network environment:

- Edited the Snort configuration file: /usr/local/etc/snort/snort.lua
- Defined the **HOME_NET** variable to represent trusted internal networks:

HOME_NET = '192.168.10.0/24' -- Internal subnet

- Defined the **EXTERNAL_NET** variable to represent untrusted networks (all except HOME_NET):

EXTERNAL_NET = '!$HOME_NET' -- All traffic except HOME_NET

```
Finished /usr/local/etc/snort/snort.lua:
--------------------------------------------------
pcap DAQ configured to passive.

Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting
```

2. Including Rules

```
ips =
{
 include = '/usr/local/etc/rules/community.rules',
    variables = default_variables
    -- use this to enable decoder and inspector alerts
    --enable_builtin_rules = true,

    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    --include = 'snort3-community.rules',

}
```

- Configured Snort to use the community rules provided by the system administrator.
- Added the inclusion path in the ips table within the snort.lua file:
- ips = {
- include = '/usr/local/etc/rules/community.rules',
- variables = default_variables

}

3. Enabling Logging

- Configured Snort to log alerts to a file by modifying the configuration file:
- alert_fast = {
- file = true

}

- Verified that Snort logged events to /var/log/snort/alert_fast.txt.

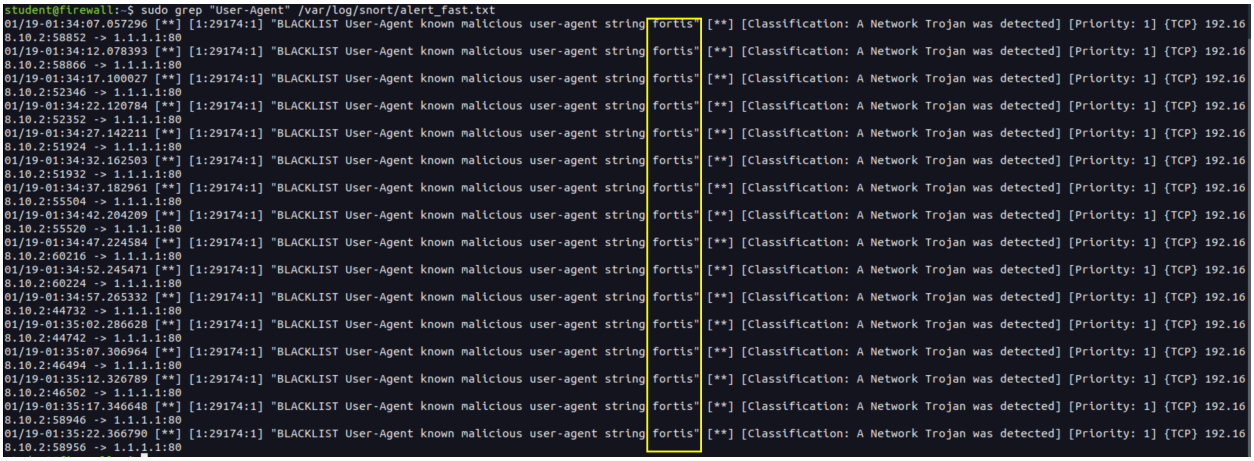
Execution and Validation

Starting Snort

Snort was started with the following command to enable daemon mode and logging:

sudo snort -c /usr/local/etc/snort/snort.lua -q -D -i enp1s10 -l /var/log/snort -k none

- **-c:** Specifies the configuration file path.
- **-D:** Runs Snort in daemon mode.
- **-q:** Suppresses startup information.
- **-i:** Defines the network interface to monitor.
- **-l:** Specifies the log directory.
- **-k:** Disables checksum verification.



Analysis and Results

Malicious Activity Detected

- The Snort alert log file (/var/log/snort/alert_fast.txt) was analyzed using:

grep "User-Agent" /var/log/snort/alert_fast.txt

- Multiple alerts were generated for a malicious **User-Agent** string:
- User-Agent: fortis
- This activity was classified as a **Network Trojan**.

Validation

Screenshots of the configuration files and logs were taken to validate the setup and results.

Conclusion

This project successfully demonstrated the setup and operation of a Snort IDS/IPS on a network. The following objectives were achieved:

1. **Configuration of Snort:** Network variables, rule inclusion, and logging were configured.
2. **Traffic Monitoring:** Snort was executed in daemon mode to monitor traffic.

3. **Detection of Malicious Activity:** The system detected a malicious User-Agent string (‘fortis’) and logged it appropriately.

The Snort IDS/IPS proves to be a valuable tool for monitoring and securing network environments against malicious activities.