**Phishing Prevention and Detection Project**

**Table of Contents**

1. **Introduction**
2. **Project Setup**
   * Hardware & Software Configuration
3. **Phishing Attack Simulation**
   * Malware Creation
   * Phishing Email Composition
   * Malware Hosting and Link Shortening
   * Listener Setup
4. **Detection and Monitoring Setup**
   * Suricata Installation and Configuration
   * Splunk Enterprise Setup
   * Splunk Alerts and Monitoring Rules
   * Splunk Universal Setup on vm
   * syslog on vm
5. **Incident Response and Findings**
6. **Challenges Faced**
7. **Conclusion and Recommendations**

# 1. Introduction

The objective of this project was to simulate a phishing attack using reverse shell malware, detect it using Suricata and Splunk, and monitor network activities for any suspicious behavior. This simulation aimed to reinforce both offensive and defensive cybersecurity practices.

The project team consisted of three members: Amina, Nicolai, and Benoit.

The timeline for the project was set to two weeks, with the final goal being to configure a phishing attack and successfully detect it using SIEM tools.

# 2. Project Setup

**Hardware & Software Configuration**

Three machines were used in the project:

* **Attacker PC (PC1)**: Used to craft the phishing attack and execute reverse shell malware.
* **Victim PC (PC2)**: Installed with Suricata, configured to detect suspicious network traffic, especially on port 1234.
* **Monitoring PC (PC3)**: Set up with Splunk Enterprise to collect data from PC2, analyze network traffic, and trigger alerts.

**PC Configurations:**

* **PC1 (Attacker)**: Used phishing tools, Netcat for listener setup, and a reverse shell script to simulate an attack.
* **PC2 (Victim)**: Installed with Ubuntu OS and Suricata for intrusion detection. It was set to detect TCP traffic on port 1234, indicating a potential reverse shell.

**Suricata rule:**

**﻿**alert tcp any any -> any 1234 (msg:"Stealthy Reverse Shell Detection"; flow:to\_server,established; content:"|0a|"; sid:1000001; classtype:trojan-activity; priority:1;)

* **PC3 (Monitoring)**: A virtual machine with Splunk Enterprise running on Kali Linux, configured to monitor Suricata logs and trigger alerts when potential attacks were detected.

# 3. Phishing Attack Simulation

**Malware Creation**

A custom Python reverse shell script was developed to establish a connection between the victim machine and the attacker's PC. This script was later converted into a standalone executable for Windows systems using PyInstaller.

**Key Points of the Script:**

* The script created a socket connection to the attacker's server using the domain fixers555.duckdns.org on port 4444.
* Once executed, the malware would send a message to the attacker's machine and provide access to the victim's system.

The script was modified into a shell script (.sh) for compatibility with macOS systems.

**Phishing Email Composition**

The phishing email was crafted in Dutch and disguised as an urgent security update from the company’s IT department. It instructed users to download a fake security patch, which was actually the reverse shell malware.

Sample Email:

Beste [Recipient Name],

In het kader van onze voortdurende inspanningen om de systemen van OPenTech te beschermen tegen mogelijke beveiligingsbedreigingen, verzoeken wij u dringend om een beveiligingspatch te installeren...

Volg de onderstaande stappen om uw apparaat te beveiligen:

[Download Link]

**Malware Hosting and Link Shortening**

Due to Gmail blocking executable files, the malware was compressed into a ZIP file and uploaded to Google Drive. A shortened Bitly link was used to obscure the true nature of the file and make it seem legitimate.

Example shortened link: [bit.ly/DownloadSecurityPatch](https://bit.ly/DownloadSecurityPatch)

**Listener Setup**

On the attacker machine, a listener was configured using Netcat to capture incoming connections from the victim machine. Commands used included:

nc -lvnp 4444

For added security, the listener was later set up with SSL encryption using ncat.

# 4. Detection and Monitoring Setup

**Suricata Installation and Configuration**

On the victim PC, Suricata was installed and configured to monitor network traffic. A specific rule was created to flag any TCP traffic on port 1234, which was identified as an indicator of reverse shell activity.

**Splunk Enterprise Setup**

Splunk Enterprise was installed on the monitoring PC (PC3) to act as a centralized SIEM solution. The steps followed were:

1. **Installation**: Downloaded and installed Splunk Enterprise:

* Create account on [www.splunk.com](http://www.splunk.com/)
* In terminal:
  + wget -O splunk-9.3.0-51ccf43db5bd-linux-2.6-amd64.deb <https://download.splunk.com/products/splunk/releases/9.3.0/linux/splunk-9.3.0-51ccf43db5bd-linux-2.6-amd64.deb>
  + sudo dpkg -i splunk-9.3.0-51ccf43db5bd-linux-2.6-amd64.deb
  + sudo /opt/splunk/bin/splunk start --accept-license
  + follow prompts on screen
    - admin: becode
    - passw: becodeisfun

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

After installation and configuration, you can access Splunk using your browser:

* http://<IP\_Splunk\_server>:8000
* Now you can log in with your credentials

Afbeelding met software, tekst, Multimediasoftware, Computerpictogram

Automatisch gegenereerde beschrijving

1. **Configuration**: Splunk was configured to collect logs from Suricata running on PC2 via the Splunk Universal Forwarder.
2. **Dashboard Setup**: A monitoring dashboard was created to visualize network traffic and detect anomalies.

**Splunk Universal Forwarder:**

On the victim PC (PC2), Splunk Universal Forwarder needs to be installed so it can ingest logs and alerts to Splunk Enterprise.

extra

**Splunk Alerts and Monitoring Rules**

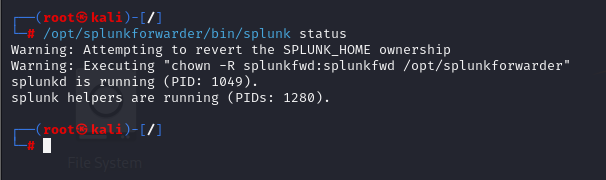
An alert was created within Splunk to monitor for Suricata logs that indicated suspicious TCP traffic on port 1234. If detected, an alert was triggered, signaling a potential reverse shell attack.

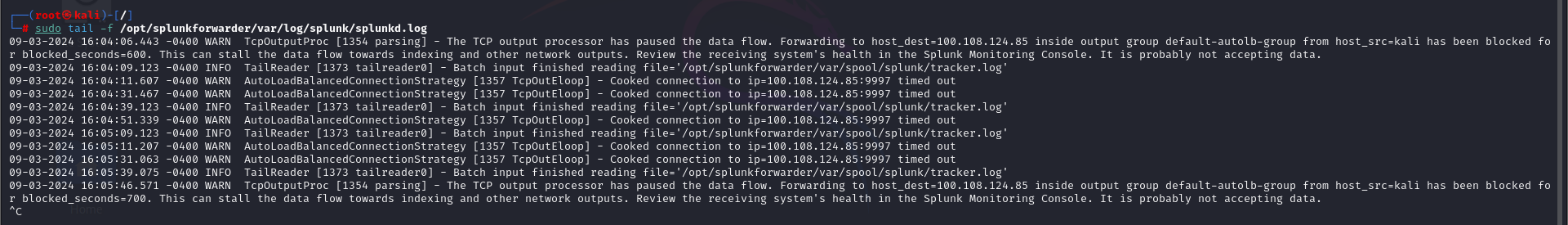
ON the virtual machine we have installed a splunk client. This is connected with the server via a VPN connection. The client sends information that we collected on the vm to the server. We have configured the client to capture log files from syslog. We have installed and configurated syslog to capture web traffic data from our thunderbird mail client which we have installed on the linux vm.

We download the universal splunk forwarder from the website or via this command “wget -O splunkforwarder-9.3.0-51ccf43db5bd-linux-2.6-amd64.deb "https://download.splunk.com/products/universalforwarder/releases/9.3.0/linux/splunkforwarder-9.3.0-51ccf43db5bd-linux-2.6-amd64.deb"”.

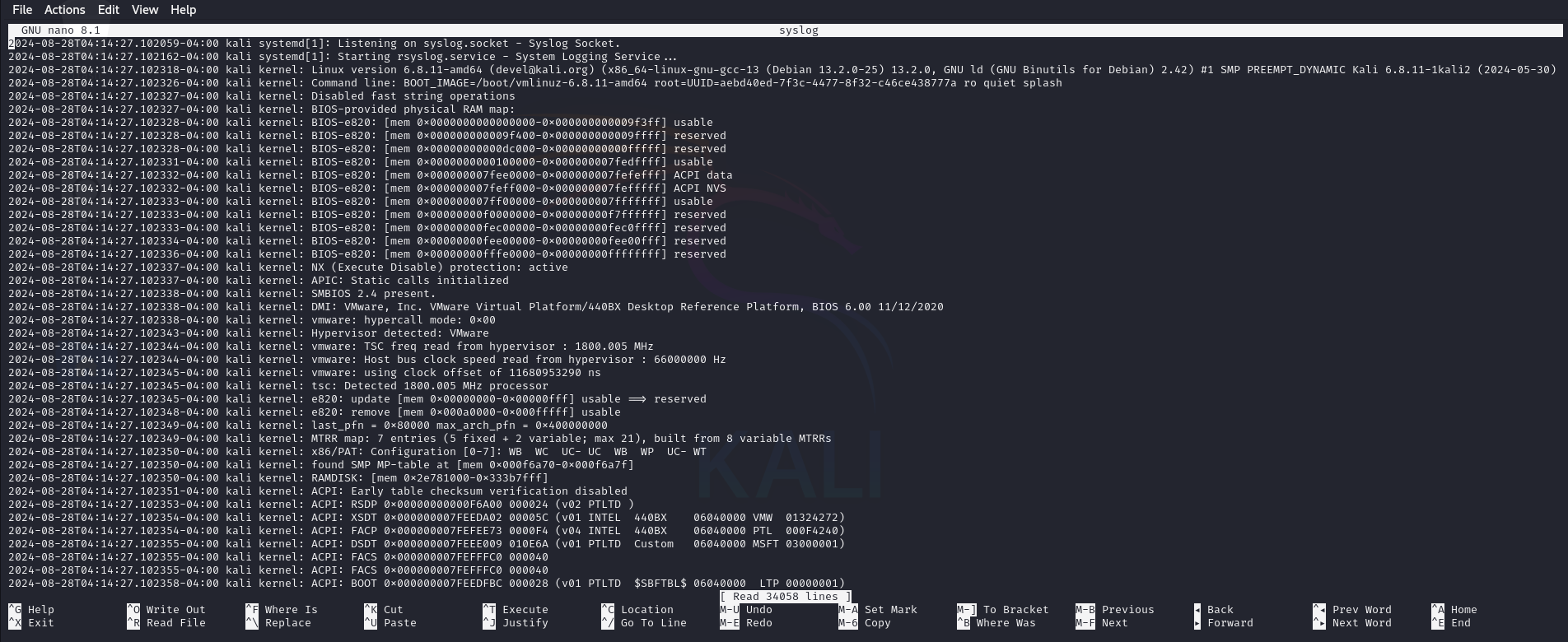
We install the software via this command “sudo dpkg -i splunkforwarder.deb”. We start and enable the program with this command “sudo /opt/splunkforwarder/bin/splunk start –accept-license sudo /opt/splunkforwarder/bin/splunk enable boot-start”.

We configure the client with these commands “sudo /opt/splunkforwarder/bin/splunk add forward-server 100.108.124.85:9997” and “sudo /opt/splunkforwarder/bin/splunk add monitor /var/log”. We forward information from the splunk client to the server. With the add monitor command we add extra files in splunk that can be monitored by syslog.

We check the status of the splunk client. We see it is installed and working. We see information like the PID.



We see in the log info, warning files. Connection is timed out since the splunk server was offline at that moment.



We see a lot of information in the syslog file which we forward to the splunk server.

# 5. Incident Response and Findings

When the phishing email was executed on the victim machine, Suricata detected the reverse shell activity, and the log data was sent to Splunk for analysis. Splunk triggered an alert based on the custom rules, allowing for a swift response to the simulated attack.

**Findings:**

* The simulated phishing attack was successfully detected.
* The alerts in Splunk proved effective in identifying potential malware activity.
* Monitoring TCP traffic for known attack patterns (e.g., reverse shell ports) can be an effective early detection method.

**Suricata alerts:**

In fast.log from suricata, the triggered alert is visible:

Afbeelding met tekst, schermopname, Lettertype, informatie

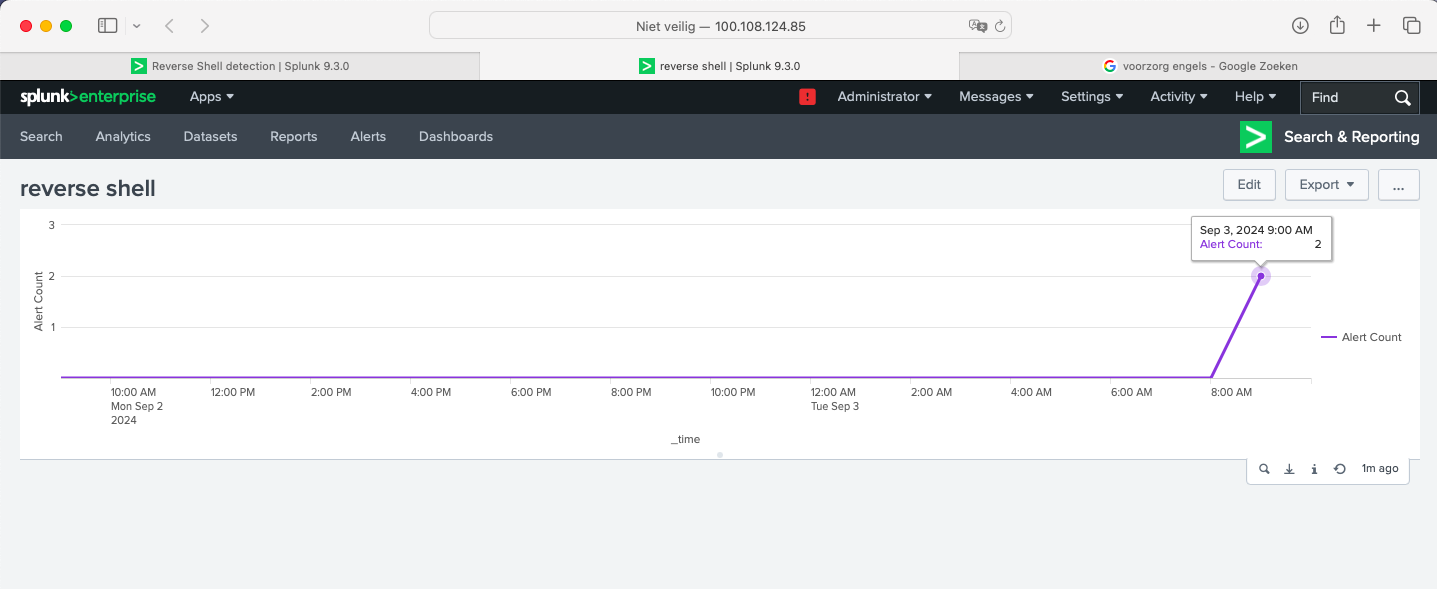
Automatisch gegenereerde beschrijving

**Splunk Enterprise alert:**

In Splunk Enterprise, we can see the same alert (both in alerts and in dashboard):

Afbeelding met schermopname, tekst, software, Webpagina

Automatisch gegenereerde beschrijving



In above screenshots, we can clearly see that the installed alert was triggered twice in the last 24 hours.

As a precaution, these alerts require further investigation to determine if these were malicious or not.

# 6. Challenges Faced

* **Executable Detection by Gmail**: Gmail blocked the initial malware executable, which was mitigated by compressing the file and hosting it on Google Drive.
* **Network Issues**: During testing, issues with the listener and IP configurations arose but were resolved by double-checking network settings and using Ngrok for tunneling.
* **CPU Usage:** Splunk enterprise requires a lot of CPU usage to run smoothly. As in the project, the Splunk Enterprise server was running in a virtual machine, with only 4GB of RAM, the system often slowed down or even crashed.

# 7. Conclusion and Recommendations

This project provided valuable insights into both phishing attack methodologies and defensive detection mechanisms using tools like Suricata and Splunk. While the simulation successfully detected and alerted the team to the reverse shell activity, our setup revealed certain limitations in fully preventing and detecting phishing attempts.

One key takeaway is that our configuration—though functional—was not the most robust for preventing phishing emails and reverse shell attacks. The absence of a dedicated mail server limited our ability to comprehensively monitor and analyze email traffic. If we had used a mail server like Microsoft Exchange, we would have gained complete access to mail logs, making it easier to trace phishing attempts and suspicious attachments. This setup would also have allowed for better integration with SIEM tools like Splunk for enhanced detection.

Additionally, the lack of a web server restricted our ability to monitor web traffic more effectively. A web server would have allowed us to capture and analyze HTTP/HTTPS requests in real-time, further improving our ability to detect phishing campaigns that rely on malicious links or websites.

Challenges like Gmail’s detection of malware and the resource intensity of Splunk in a virtualized environment highlighted areas for potential improvement in scaling up monitoring infrastructure and avoiding network bottlenecks. The combination of these factors suggests that a more comprehensive setup—integrating mail and web servers—would yield a stronger defense against phishing and reverse shell attacks.

#### Recommendations:

* **Deploy a Mail Server**: Implementing a dedicated mail server, such as Microsoft Exchange, would allow for deeper monitoring and more efficient detection of phishing emails. Having complete access to mail logs would facilitate better correlation with SIEM tools, enabling faster identification of threats.
* **Incorporate a Web Server**: Introducing a web server into the environment would enhance our ability to monitor web traffic. This would allow for more comprehensive detection of malicious links and suspicious browsing activity, critical in many phishing scenarios.
* **Broaden Detection Rules**: Expand the Suricata rule set to monitor for additional attack vectors, such as web-based exploits or other network-based threats beyond reverse shell activity. This would help catch a wider range of potential intrusions.
* **Enhance Incident Response**: Automating parts of the incident response process, such as isolating compromised machines or alerting users when a threat is detected, would improve the speed and efficiency of response actions.
* **Resource Allocation**: Consider dedicating more resources to the SIEM infrastructure, such as increasing the CPU and memory allocations for Splunk, to ensure smoother operation and faster data processing in production environments.
* **User Training**: Continue to emphasize phishing awareness training for users. A well-informed user base is often the first line of defense against phishing attempts, and regular security training can help minimize the likelihood of sucessful attacks.