Pi-Based Cybersecurity Monitoring System Using Suricata and Zeek

By Grey Brewer

Introduction

My journey into cybersecurity and networking has been driven by a hands-on approach to learning. As I progressed through my studies in Linux, systems administration, and cybersecurity, I developed a deeper appreciation for the interplay between hardware and software. My previous work in hardware hacking, particularly my research into embedded systems and IoT security, laid the foundation for this project: building a Raspberry Pi-based cybersecurity monitoring system. This initiative serves as both a practical application of networking concepts and a structured way to reinforce the knowledge required for the **CompTIA Network+ and Security+** certifications.

By integrating a TP-Link managed switch, Power over Ethernet (PoE) functionality, and Docker-based deployments of Suricata and Zeek, this system enables real-time network traffic analysis. The following write-up provides a comprehensive walkthrough of the setup process, detailing the hardware and software configurations required to build a fully operational home lab for network security monitoring.

1. Hardware Configuration

1.1 Deploying the TP-Link SG108PE Managed PoE Switch

This project began with the selection and deployment of a TP-Link SG108PE managed PoE switch. Understanding network infrastructure, including switch configurations and VLAN segmentation, is one of the many learning objectives of the **CompTIA Network+** certification.

- Unboxed and verified the TP-Link SG108PE along with all required cables.
- Connected the switch to the primary router (NetGear Orbi System) using a LAN port and plugged it into Port 8 on the switch.
- Configured VLAN settings on both the Orbi and the TP-Link switch to isolate traffic.
- Enabled PoE functionality on specific ports to supply power to the Raspberry Pi.

1.2 Setting Up the Raspberry Pi 4B with a PoE Hat

The Raspberry Pi serves as the core computing unit of this monitoring system, offering a cost-effective platform for learning networking and security fundamentals

- Installed the Waveshare PoE Hat (C) on the Raspberry Pi.
- Ensured proper alignment of the PoE hat and the heat sink.
- Connected the Raspberry Pi to a PoE-enabled port on the switch.
- Confirmed successful power delivery and system boot via PoE.

2. Software Installation and Configuration

2.1 Installing Ubuntu Server 24.04 LTS on Raspberry Pi

To establish a functional server environment, we installed Ubuntu Server 24.04 LTS and configured SSH for remote access.

- Flashed Ubuntu Server 24.04 LTS (64-bit) onto a microSD card using Raspberry Pi Imager.

Configured headless SSH access and remotely connected to the Raspberry Pi:

```
ssh username@<Your-Pi-IP>
```

Updated system packages and installed essential dependencies:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y wget curl git nano
```

2.2 Installing Docker and Docker-Compose

Docker provides an efficient way to deploy and manage security tools in a modular environment, reinforcing containerization concepts relevant to modern IT infrastructure.

Installed Docker:

```
sudo apt install -y docker.io
sudo systemctl enable --now docker
```

Added the current user to the Docker group:

```
sudo usermod -aG docker $USER
```

Installed Docker-Compose:

```
sudo apt install -y docker-compose
```

3. Deploying Suricata and Zeek for Network Traffic Monitoring

3.1 Installing Suricata (Intrusion Detection & Prevention System)

Suricata is a powerful intrusion detection and prevention system (IDS/IPS), offering deep packet inspection capabilities that align with both exams' objectives related to network security and threat monitoring.

Created a persistent directory for Suricata logs:

```
mkdir -p ~/docker/suricata/logs
```

Deployed Suricata in a Docker container:

```
docker run -d --name suricata \
     --net=host \
     --privileged \
      -v ~/docker/suricata/logs:/var/log/suricata \
     jasonish/suricata -i eth0
```

Verified Suricata logs were being generated:

```
ls ~/docker/suricata/logs/
```

3.2 Installing Zeek (Network Security Monitor)

Zeek provides network security monitoring and traffic analysis, complementing Suricata's capabilities

Created a persistent directory for Zeek logs:

```
mkdir -p ~/docker/zeek/logs
```

Deployed Zeek in a Docker container:

```
 docker run -d --name zeek \
  --net=host \
  --cap-add=NET_ADMIN --cap-add=NET_RAW \
  --privileged \
```

```
  -v ~/docker/zeek/logs:/opt/zeek/logs \
  blacktop/zeek -i eth0
```

Verified Zeek logs were being generated:

```
ls ~/docker/zeek/logs/current/
```

4. Testing and Validation

4.1 Capturing Network Traffic

Ran a packet capture test using tcpdump:

```
sudo tcpdump -i eth0 -c 10
```

Simulated network activity from a Kali Linux machine:

```
nmap -sV <Raspberry Pi IP>
```

Verified Suricata alerts:

```
tail -f ~/docker/suricata/logs/fast.log
```

Checked Zeek connection logs:

```
tail -f ~/docker/zeek/logs/current/conn.log
```

4.2 Running Zeek on a Sample PCAP File

Downloaded the sample Heartbleed attack PCAP file:

```
wget
https://github.com/blacktop/docker-zeek/raw/master/pcap/heartbleed.pca
p
```

Executed Zeek on the PCAP file:

```
docker run --rm \
  -v $(pwd):/pcap \
  blacktop/zeek -r heartbleed.pcap
```

Reviewed Zeek's processed logs:

```
cat conn.log | less
```

4. Troubleshooting and Lessons Learned

This project was far from a smooth, linear process. Each step introduced its own set of challenges that forced me to rethink my approach, troubleshoot, and adapt my understanding of network security and system administration.

4.1 Network Connectivity Issues

The first major issue arose with the TP-Link switch. Despite connecting everything correctly, the switch wasn't appearing in the network. At first, I assumed it was a cabling issue, so I swapped cables and even tested with a separate switch. No luck. Diving deeper, I discovered that the Orbi router was still handling DHCP, which was interfering with the switch's role in the network. After modifying the DHCP settings, I was finally able to get the switch recognized and properly configured.

4.2 Zeek Not Capturing Traffic

Once Zeek was deployed, I ran a quick check to verify that logs were being written, only to find… nothing. I tried restarting the container, recreating it, and even verifying that `eth0` was up. Running `tcpdump -i eth0` finally revealed the issue—Zeek was not seeing any network traffic because Suricata had exclusive control over the interface. The solution was adjusting Suricata's configuration to allow traffic to be shared, after which Zeek immediately began logging packets as expected.

4.3 Docker Volume Mounting Issues

Another unexpected roadblock came when I realized that logs were not persisting across reboots. At first, I assumed it was a permissions issue, but after confirming proper ownership and testing with other containers, I narrowed it down to an issue with Docker's volume bindings. Reconfiguring the volume mounts and restarting the containers resolved the issue, ensuring logs remained intact even after reboots.

4.4 Lessons Learned

These troubleshooting experiences reinforced several crucial lessons:

- The **importance of network topology planning**—misconfigurations can cascade into major issues.
- How IDS/IPS solutions interact with network interfaces—especially when dealing with **packet mirroring**.
- The role of system logging and troubleshooting—**logs are often the key to solving technical mysteries**.

6. Using This Setup to Learn Networking and Cybersecurity Concepts

This Raspberry Pi-based monitoring system has given me an engaging learning experience while preparing for the CompTIA Network+ certification. Some key concepts reinforced through this setup include:

- **Network Infrastructure:** Configuring a managed switch, VLAN segmentation, and PoE functionality.
- **Packet Capture and Analysis:** Using Suricata and Zeek to inspect network traffic and detect anomalies.
- **Security Best Practices:** Implementing IDS/IPS solutions and monitoring live network environments.
- **“Basic” Linux System Administration:** Deploying and managing servers, working with Docker, and automating network monitoring.

By deploying, troubleshooting, and managing this system, other learners can hopefully gain valuable hands-on experience applicable to both Network+ certification topics and real-world cybersecurity roles.

7. Conclusion

This hands-on project transformed a Raspberry Pi into a fully functional **network security monitoring system**, reinforcing key networking and cybersecurity concepts. The system facilitates real-time intrusion detection, packet analysis, and forensic investigations through Suricata and Zeek. By configuring a PoE-enabled Raspberry Pi, integrating network monitoring tools, and analyzing traffic patterns, I gained practical experience applicable to real-world IT and security roles.

This experience has deepened my understanding of network security and intrusion detection. The next phase will involve integrating visualization tools like Grafana to better interpret logs and enhance the system’s usability for threat monitoring. As with my previous hardware hacking work, this project is another stepping stone in my learning journey, bridging theoretical knowledge with practical application. It serves as both an educational exercise and a foundation for more advanced network / security research. Yet again, I reach the unsettling paradox of learning how much there is yet to be learned in this industry. The journey continues.