



Incident Report Analysis for NixGuard

Prepared by:

Name: Laurentius Valdi Adela

Position: Cybersecurity Analyst

Executive Summary

This report details the simulated detection and analysis of suspicious network activities using **NixGuard**, a GUI-based security tool, along with supporting cybersecurity utilities. The primary objectives of this exercise were:

1. Demonstrate **cybersecurity analysis skills** using NixGuard and other security tools.
2. Analyze **network traffic**, identify potential threats, and **detect anomalies**.
3. Develop a **structured incident report** based on the findings.
4. Provide **constructive feedback** on the experience of using NixGuard.

The virtual environment consisted of the following machines:

- **NixGuard Server**: Windows 10 OS (acting as the monitoring system).
- **Employee Workstations**:
 - **Windows 10 PC**
 - **Ubuntu 24.04.1 PC**
- **Attacker System**: Kali Linux (VirtualBox-amd67).

The supporting security tools used in this analysis included:

- **NixGuard**
- **Wireshark** (for network traffic monitoring)
- **Nmap** (for network scanning)
- **Metasploit** (for penetration testing)
- **Windows PowerShell & Command Prompt**
- **Kali Linux Terminal Emulator**
- **Firefox Browser**

Initial Alerts and Detection

On **November 26, 2024, at 10:20 PM**, NixGuard began generating multiple alerts regarding **system error events**. These events persisted, with key timestamps recorded at:



- **10:20 PM** – Initial system error events detected.

```
38) timestamp: November 26, 2024 at 10:20:14 PM PST
rule:
  description: Multiple System error events
  firedTimes: 1
  mail: false
  level: 10
```

- **10:35 PM** – Repeated alerts indicating **SessionEnv was unavailable** to handle notification events.

```
33) timestamp: November 26, 2024 at 10:35:35 PM PST
    rule:
      description: SessionEnv was unavailable to handle a notification event.
      firedTimes: 3
      mail: false
      level: 5
```

- **11:05 PM & 11:10 PM** – Additional alerts for **SessionEnv** and **WSearch** failures.
- **11:07 PM** – A **significant spike** in log activity was observed:
 - The system's **normal log activity** averaged around **272 entries**, but suddenly

surged to **1,523 entries** within a short timeframe.

- The description for these logs was **undefined**, which raised suspicion.

```
23)
timestamp: November 27, 2024 at 11:07:42 PM PST
log: The average number of logs between 7:00 and 8:00 is 272. We reached 1523.
rule:
  description: undefined
  firedTimes: undefined
  mail: false
  level: 4

24)
timestamp: November 27, 2024 at 11:07:09 PM PST
rule:
  description: CIS Microsoft Windows 10 Enterprise Benchmark v1.12.0: Ensure 'Rese
changed from passed to 'not applicable'
  firedTimes: 1
  mail: false
  level: 5
```

Preliminary Hypothesis

Initially, the **SessionEnv notifications** appeared to be regular system maintenance or health-check alerts. However, the **persistence of these logs** combined with **anomalous log volume surges** suggested potential **suspicious activity within the network**.

Threat Identification & Analysis

Step 1: Reviewing NixGuard Logs

A new log branch titled "**Vulnerabilities**" was identified. These logs repeatedly highlighted two key security concerns:

1. **Windows OS** vulnerabilities present on the NixGuard server.
2. **Python package** vulnerabilities within the program files.

```

Vulnerability
logs:
  )
timestamp: 2024-05-29T02:15:39Z
rule:
  firedTimes: 775
  mail: false
  level: 3
manager:
  name: nix-guard-674185331c4974d96b078470
vulnerability:
  reference: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-29335
  severity: High
  cve: CVE-2023-29335
  packages:
    name: Microsoft Windows 10 Enterprise Evaluation
    version: 10.0.19045.5131
    architecture: x86_64
  published: 2023-05-09T18:15:13Z
  enumeration: CVE
  title: CVE-2023-29335 affecting Microsoft Windows 10 Enterprise Evaluation was solved
  type: Packages
  cve:
    cve3:
      base_score: 7.500000
  updated: 2024-05-29T02:15:39Z
  status: Solved

```

```
timestamp November 26, 2024 at 06:36:16 PM PST
role
fredTomas: 773
mail: false
level: 3
manager
name: ms-guard-6741865331-6974d985d76470
version: 0.0.0
references:
https://github.com/python/python-openssl/commit/125286,
https://github.com/python/python-openssl/commit/4ee48891c1258b3c7bdc6d95076ac83920a906,
https://github.com/python/python-openssl/commit/745c5e027248b3c40496408c108819700643,
https://github.com/python/python-openssl/commit/044349c49779c306542d89c3bd6dd4170c3c8f8d,
https://github.com/python/python-openssl/commit/3d3a4c3078a4423999164e71d2532373,
https://mail.python.org/archives/list/security-announce@python.org/thread/PT7H93ZDTLWQZ81CZD3W4H4ACTV/,
https://github.com/python/python-openssl/commit/741906a830747813c35d0733e884384364e,
https://github.com/python/python-openssl/commit/64225ca91547a6d73c3ea391614f8a2b5d3c877
summary: High
cve:CVE-2024-6232
package:
name: Python 3.12.4 (64-bit)
version: 3.12.4.0.0
architecture
published: 2024-09-03T13:15:05Z
cveid: CVE-2024-6232 affecting Python 3.12.4 (64-bit) was solved
type: Packages
cve:
cveid:
score: 7.500000
updated: 2024-09-04T12:15:42
status: Solved
```

Given that vulnerability scans often precede an attack, this raised concerns about reconnaissance activity by an unauthorized actor.

Step 2: Network Traffic Analysis (Wireshark)

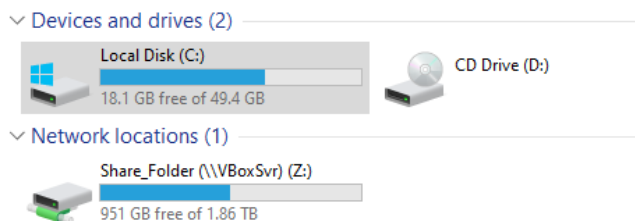
To confirm possible reconnaissance, Wireshark was used to analyze network traffic. The following anomalies were discovered:

- Multiple **ARP requests** were observed, with a **"Who has"** request being sent sequentially across the subnet.
 - The **source IP** of these ARP requests: **192.168.10.8**.
 - The ARP protocol is commonly used for **mapping IP addresses to MAC addresses**, but sequential scanning suggests host enumeration.
 - This indicates that an **internal actor** was actively scanning for live hosts.

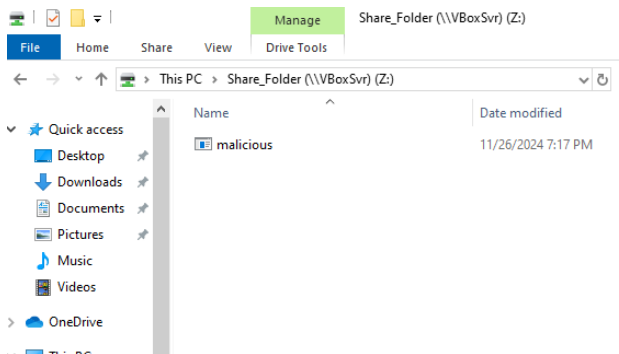
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.51?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.52?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.53?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.54?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.55?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.56?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.57?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.4?	Tell	192.168.10.8
PCSSystemtec_7b:5d:...	PCSSystemtec_ad:25:...	ARP	42	192.168.10.4	is	at	08:00:27:7b:5d:27	
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.60?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.61?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.62?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.63?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.64?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.67?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.68?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.69?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.70?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.71?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.72?	Tell	192.168.10.8
PCSSystemtec_ad:25:...	Broadcast	ARP	60	who	has	192.168.10.73?	Tell	192.168.10.8

Step 3: File Inspection

Given the possibility of post-reconnaissance activity, a **file integrity check** was conducted in the **Share Folder**.



- A suspicious file named **"malicious"** was found inside the folder.



- NixGuard did not flag it as a malicious file, prompting further investigation.
- A scan using **VirusTotal** confirmed that the file was empty (suggesting it may have been a placeholder or an attempt to bypass security measures).

Step 4: Attribution

Using **Wireshark's packet capture** analysis, the MAC address of the scanning entity was identified as:

08:00:27:ad:25:87 (corresponding to 192.168.10.8).

```
> Frame 304: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{805;  
▼ Ethernet II, Src: PCSSystemtec_ad:25:87 (08:00:27:ad:25:87), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)  
  > Source: PCSSystemtec_ad:25:87 (08:00:27:ad:25:87)  
    Type: ARP (0x0806)  
    [Stream index: 0]  
    Padding: 0000000000000000000000000000000000000000000000000000000000000000  
  > Address Resolution Protocol (request)
```

Given this information, it was concluded that:

- The malicious activity originated from within the **internal network** (an insider threat scenario).
- The actor used **ARP scanning to map** live hosts.
- They likely used **vulnerability assessment techniques** to identify system weaknesses.
- They attempted to introduce a file into the system, but it was either unsuccessful or a test probe.

Incident Response & Mitigation

1. Protection Measures

To **prevent similar incidents** in the future, the following security measures are recommended:

1. Cybersecurity Awareness & Training

- Employees should be educated about **common attack techniques** (phishing, reconnaissance, privilege escalation).
- Regular **security workshops** should be conducted.

2. Access Control & Least Privilege

- Ensure **strict access control policies** are implemented.
- Follow the **Principle of Least Privilege (PoLP)** to restrict unnecessary system access.

3. Insider Threat Detection Tools

- Deploy **behavioral monitoring solutions** to detect **anomalous internal**

activity.

- Implement **enhanced logging and user access auditing.**
-

2. Detection Enhancements

To **improve threat detection capabilities**, the following modifications should be made to NixGuard's configuration:

1. Enhanced Alerting System

- Configure NixGuard to **immediately alert on critical events** instead of aggregating logs without priority filtering.

2. Log Filtering & Categorization

- Implement **custom log filtering** to separate **system health events** from **security incidents**.
- Prioritize **security-related logs** to improve **incident detection time**.

3. File-Based Threat Detection

- Improve NixGuard's **file analysis module**.
 - Conduct **regular penetration tests** to ensure security tools function properly.
-

3. Incident Containment & Response

To **limit potential damage**, the following actions were taken:

1. Immediate Network Disconnection

- The **compromised PC was disconnected** from the network to **prevent further reconnaissance**.

2. File Removal & Firewall Hardening

- The **suspicious file was deleted**.
- **Firewall policies were updated** to restrict traffic:
 - **Only allow traffic from the default gateway.**
 - **Block the attacker's MAC address** to prevent **further exploitation attempts**.

4. Recovery & Post-Incident Monitoring

To ensure the **integrity of the system and prevent future attacks**, the following steps were taken:

1. Full System Scan & Integrity Check

- Conducted a **thorough antivirus scan** to ensure no other malicious files were present.
- Verified **Share Folder integrity**.

2. Reconnection & Continued Monitoring

- The system was **gradually reintroduced** to the network.
- **Additional monitoring** was conducted using NixGuard to **watch for any follow-up attacks**.

3. Incident Documentation & Reporting

- Findings were **reported to the supervisor**.
- A **post-mortem analysis** was conducted to identify **areas of improvement** in security posture.

Final Thoughts & Key Takeaways

This incident highlights the **importance of proactive monitoring and rapid incident response**. The key lessons learned include:

- **NixGuard is useful for log-based detection**, but **requires additional configuration for optimal security monitoring**.
- **Wireshark proved invaluable** in detecting **internal network reconnaissance**.
- **ARP scanning remains a common tactic** for attackers mapping networks—proper detection and response mechanisms must be in place.
- **Regular vulnerability assessments and firewall restrictions** can prevent similar incidents in the future

