Vulnerablility Asessment Scan Report on a Unix Server Using Spiderfoot

IP Address: 192.168.0.47

Prepared by: Shauib Salami

Date: 28th February, 2025

Table of Contents

Introduction

This report presents the findings of a penetration test performed on a Unix machine with the IP address 192.168.0.47. The assessment was conducted using **SpiderFoot**, an open-source reconnaissance tool designed for security analysis. The objective of this test was to gather intelligence on the target system's security posture, identify vulnerabilities, and assess potential exposure to threats.

By leveraging **SpiderFoot**, the scan provided comprehensive insights into possible data breaches, misconfigurations, and exploitable weaknesses. These findings can support threat intelligence efforts, penetration testing strategies, and red team operations aimed at strengthening overall cybersecurity defenses.

Objective

Its primary objective is to **automate the reconnaissance process**, efficiently gathering intelligence on a specified target, which may include a **person's name, email address, network subnet, IP address, domain name, hostname, or Autonomous System Number (ASN)**. This automation enhances security assessments by streamlining data collection for threat intelligence and penetration testing purposes.

IP address Report

Scan Command Used

IP a

An **IP address** is a unique identifier assigned to a device on the **internet or a local network**. It operates under the **Internet Protocol (IP)**, which defines the rules for formatting and transmitting data across networks, ensuring seamless communication between connected devices.

Breaking it Down:

I run the ip addr command on a Linux system. This command displays network interface information. Let's break down the output:

The **lo** interface represents the **loopback interface**, a virtual network interface that enables a system to communicate with itself. Below is a breakdown of its key attributes:

- **lo:** The first network interface, dedicated to internal system communication.
- **<LOOPBACK, UP, LOWER_UP>:** Flags indicating the interface status:
  - **LOOPBACK:** Confirms it is a loopback interface.
  - **UP:** Indicates that the interface is administratively enabled.
  - **LOWER_UP:** Signifies the physical layer is active, though virtual in this case.
- **mtu 65536:** Maximum Transmission Unit, specifying the largest packet size that can be transmitted. Loopback interfaces commonly have a high MTU.
- **qdisc noqueue:** Queueing discipline, with **noqueue** meaning packets are processed immediately without queuing.
- **state UNKNOWN:** The current state of the interface is unknown.
- **group default:** The interface is part of the default network group.

- **qlen 1000:** Queue length, defining the maximum number of packets that can be queued for transmission.

Loopback Interface (lo)

- **link/loopback:** Indicates that the link type is loopback, meaning it is used for internal system communication.

IPv4 Configuration

- **inet 127.0.0.1/8 scope host lo:**
  - **127.0.0.1/8:** The assigned IPv4 address for the loopback interface, where **127.0.0.1** is the standard loopback address.
  - **/8 (CIDR Notation):** Represents a subnet mask of **255.0.0.0**, meaning the entire **127.0.0.0** range is reserved for loopback networking.
  - **scope host:** Restricts the address's scope to the local machine.
  - **valid_lft forever preferred_lft forever:** The address remains permanently valid and preferred.

IPv6 Configuration

- **inet6 ::1/128 scope host:**
  - **::1/128:** The assigned IPv6 loopback address, where **::1** is the standard IPv6 loopback address.
  - **/128:** Denotes a single host address.
  - **scope host:** Limits the address's use to internal system communication.
  - **valid_lft forever preferred_lft forever:** The address remains valid and preferred indefinitely.

Ethernet Interface (eth0)

- **eth0:** Represents the primary Ethernet interface, typically used for external network communication.

Interface Flags and Capabilities

- **<BROADCAST, MULTICAST, UP, LOWER_UP>:**
  - **BROADCAST:** Supports broadcasting packets to all devices in the subnet.
  - **MULTICAST:** Allows multicast traffic, enabling efficient data distribution to multiple hosts.
  - **UP:** The interface is administratively enabled.
  - **LOWER_UP:** The physical link is active (cable connected, link established).

Additional Parameters

- **mtu 1500:** Maximum Transmission Unit, with **1500 bytes** as the standard for Ethernet networks.
- **qdisc fq_codel:** A queue management algorithm (**Fair Queueing Controlled Delay**) used to minimize latency and optimize performance.
- **state UP:** Indicates the interface is operational.
- **group default:** The interface belongs to the system's default network group.
- **qlen 1000:** Specifies a queue length of **1000 packets** for buffering.

Ethernet Interface Details

Ethernet Link Layer

- **link/ether ...**
  - Denotes that the interface operates at the Ethernet link layer.
  - The **MAC address** (48-bit hardware address) is typically displayed here but has been redacted for privacy.

IPv4 Configuration

- **inet 192.168.1.168/24 scope global dynamic noprefixroute eth0:**
  - **192.168.1.168/24:** The assigned IPv4 address for the **eth0** interface, where **/24** represents a subnet mask of **255.255.255.0**, defining the **192.168.1.0** network.
  - **scope global:** Indicates that the address is routable within the network, though **192.168.x.x** addresses are private and require **NAT** to communicate with the public internet.
  - **dynamic:** The address was dynamically assigned, most likely through **DHCP**.
  - **noprefixroute:** Specifies that no separate prefix route is installed for this address.

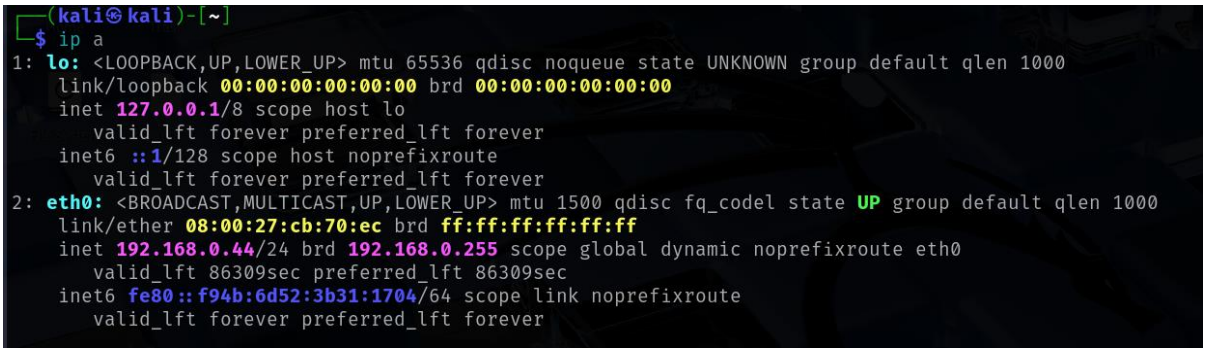
Address Lifetime

- **valid_lft 8632450sec:** The address remains valid for approximately **100 days** unless renewed.
- **preferred_lft 8632450sec:** The preferred lifetime for the address before it may be deprecated.

IPv6 Configuration

- **inet6 ... scope link noprefixroute:**
  - **inet6 ...**: The assigned IPv6 address (redacted for privacy). Typically, this is a **link-local** address (e.g., starting with **fe80:**).
  - **scope link:** Limits the IPv6 address's use to the **local network link** (not globally routable).

- **noprefixroute:** No prefix route is installed for this address.
- **valid_lft forever preferred_lft forever:** The IPv6 address remains valid and preferred indefinitely.



Scan Command Use

Spiderfoot -l 127.0.0.1:2000

Log Details:

- **Date and Time:**
  - `2025-02-28 19:56:12,903` and `2025-02-28 19:56:12,921` – These timestamps indicate when the log entries were recorded.
- **Log Levels:**
  - `[INFO]` – Provides informational messages about system operations.
  - `[WARNING]` – Indicates a potential issue or caution that may require attention.
- **Source:**
  - `sf` – Likely represents the module or application responsible for generating the log (possibly **SpiderFoot**).
- **Log Messages:**
  - `Starting web server at 127.0.0.1:10000...`
    - Confirms that a **web server** is being launched on **localhost (127.0.0.1)**, listening on **port 10000**.
  - `Use SpiderFoot by starting your web browser of choice and...`
    - Indicates that **SpiderFoot**, an **OSINT (Open Source Intelligence) tool**, is being initialized and can be accessed via a web browser.
- **Warning Message:**
  - The **warning entry** is truncated, meaning the full context of the issue is missing.

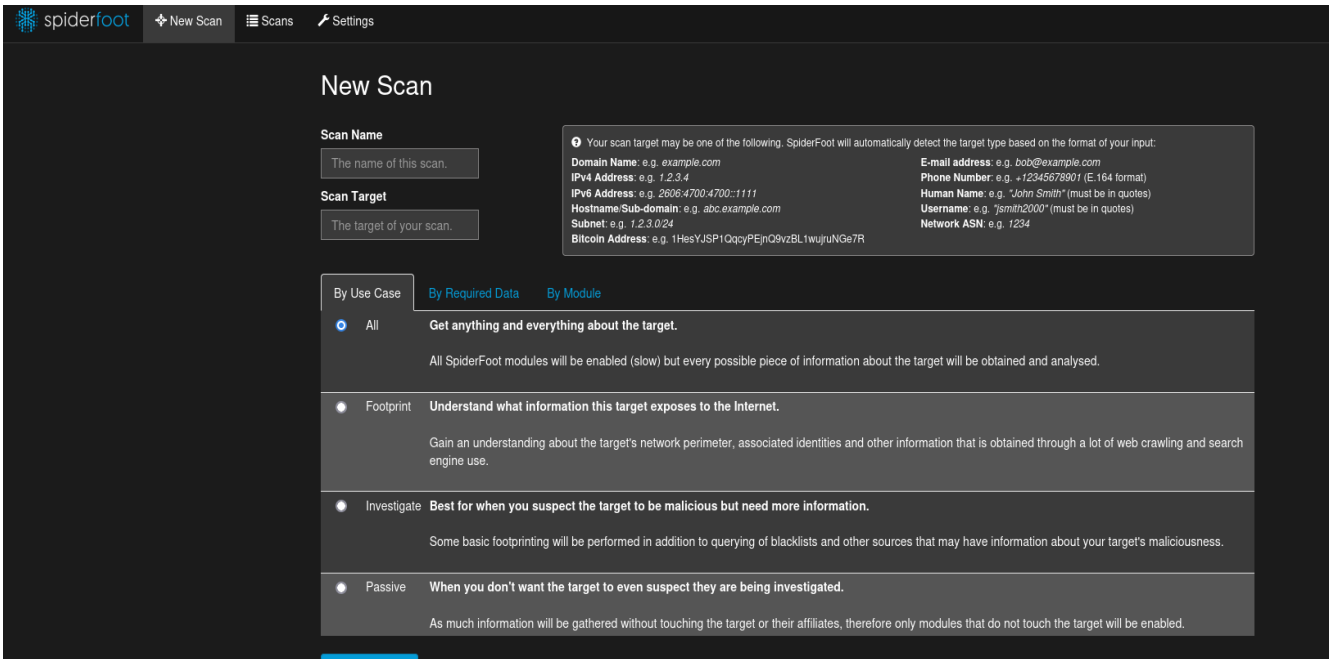
Summary:

This log snippet shows that **SpiderFoot** is launching a local web server at **127.0.0.1:10000**. While the informational message confirms normal operations, the **warning** suggests there may be a potential issue, but it is incomplete. Further log analysis would be needed to determine the exact nature of the warning.

Using SpiderFoot

Running a Scan

When you run SpiderFoot in Web UI mode for the first time, there is no historical data, so you should be presented with a screen like the following:



This section outlines the **initial setup and target definition** for a SpiderFoot scan, explaining how to initiate and configure an investigation to gather open-source intelligence (OSINT) on a given target.

Purpose

The primary goal of this phase is to **automate data collection** on a specific target, such as a domain, IP address, email, or phone number. SpiderFoot retrieves publicly available information to help assess security risks and an entity's digital footprint.

Key Elements of SpiderFoot Scan Setup

1. **New Scan:**
    a. This marks the beginning of an investigation in SpiderFoot.
2. **Scan Name:**
    a. Assigning a descriptive name helps organize scans for later reference.
3. **Scan Target:**
    a. The most critical input, defining **what** is being investigated.
    b. SpiderFoot will conduct reconnaissance based on this target.
4. **Filtering Options:**
    a. **By Use Case / By Required Date:** Helps filter scans based on investigative needs or deadlines.
    b. **All / Get Anything and Everything Analyzed:** Enables all modules to maximize data collection.
5. **Footprinting Module:**

a. This mode focuses on **gathering external intelligence** about a target, revealing its exposed assets.

Supported Scan Target Types

SpiderFoot accepts a variety of target types, including:

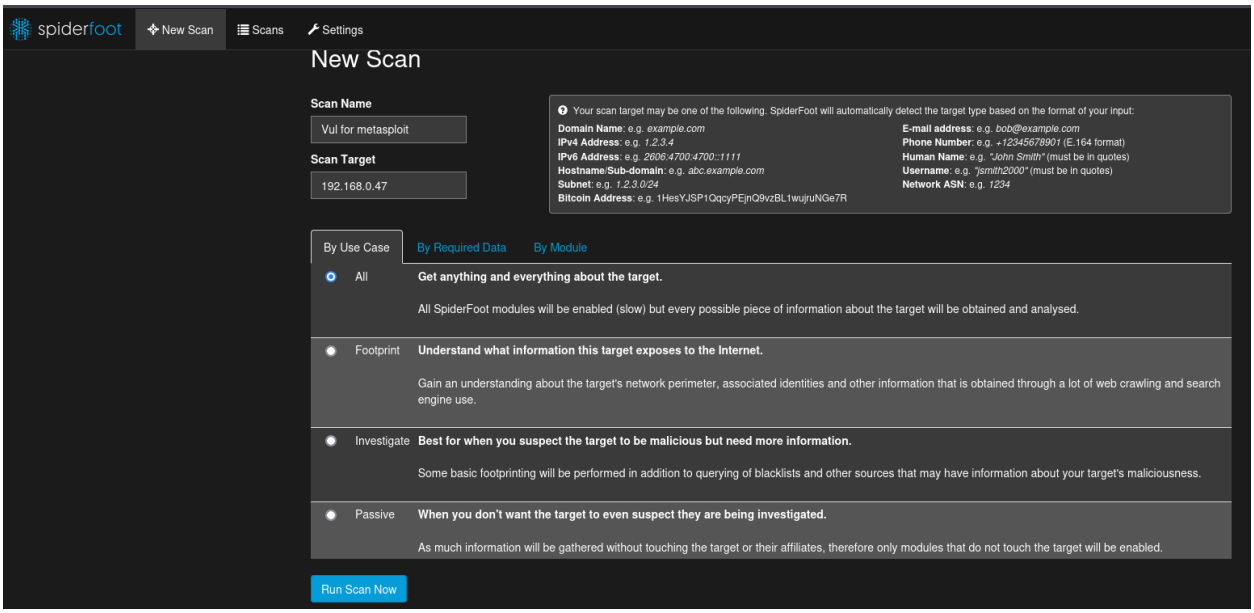
- **Domain Name:** `example.com`
- **IPv4 Address:** `192.168.1.52`
- **IPv6 Address:** `2606:4700::1111`
- **Email Address:** `user@example.com`
- **Phone Number:** `+12345678901` (E.164 format)
- **Human Name:** `"John Smith"`
- **Subnet:** `192.168.1.0/24`
- **Hostname/Sub-domain:** `abc.example.com`
- **Username:** `user123`
- **Network ASN:** `1234`
- **Bitcoin Address:** `1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa`

The **goal** of gathering this data is to **understand what information the target exposes to the internet**, including security vulnerabilities, linked identities, and infrastructure details.

SpiderFoot Scan Profiles & Modes

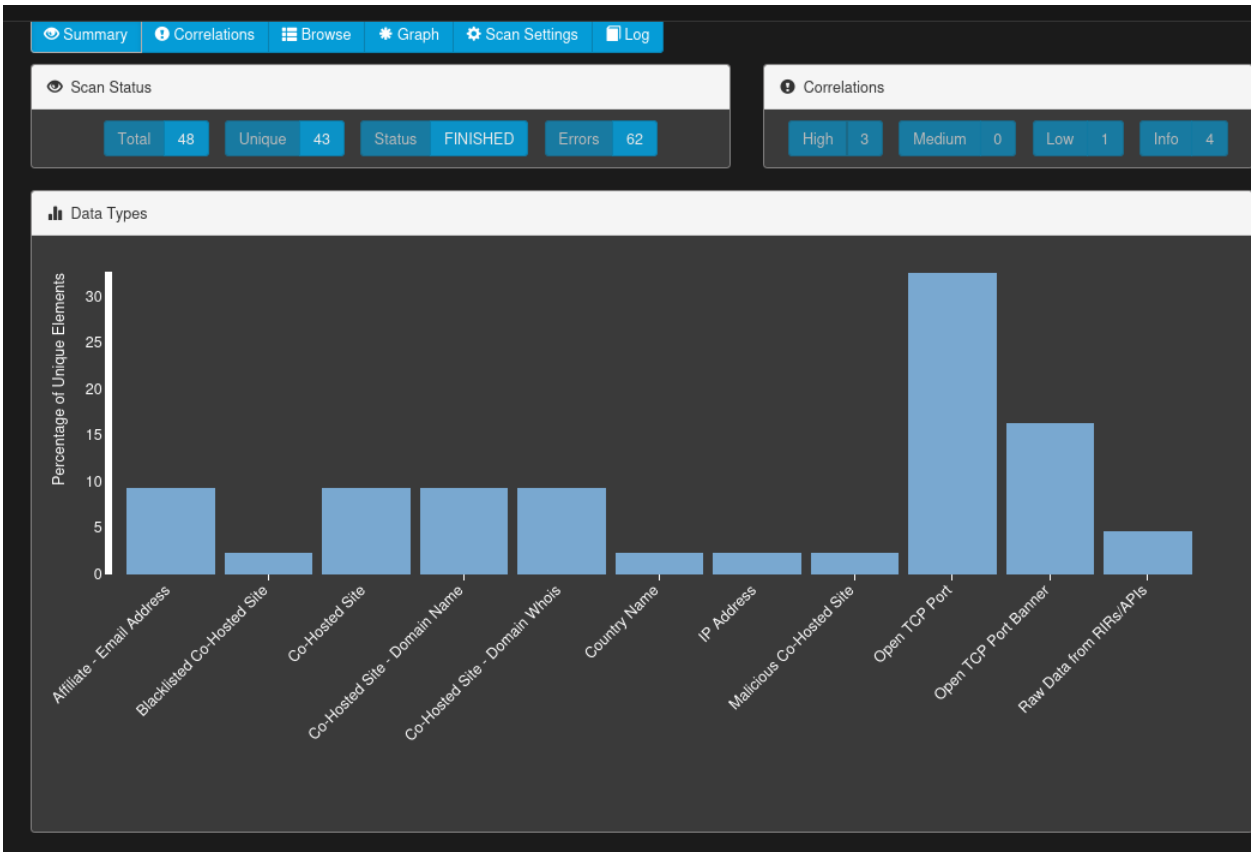
To refine how SpiderFoot operates, users can choose from different scan modes:

1. **All:**
    a. Enables **all** modules for a **comprehensive** analysis.
    b. This can be slow but collects the most data.
2. **Footprint:**
    a. Focuses on **mapping exposed assets** and **network perimeters**.
    b. Uses web crawlers and search engine queries.
3. **Investigate:**
    a. Best for **analyzing potentially malicious** entities.
    b. Checks for blacklist entries and historical incidents.
4. **Passive:**
    a. **Stealth mode** to avoid alerting the target.
    b. Uses **only indirect** data sources.



Scan Results

From the moment you click ‘Run Scan’, you will be taken to a screen for monitoring your scan in near real time:

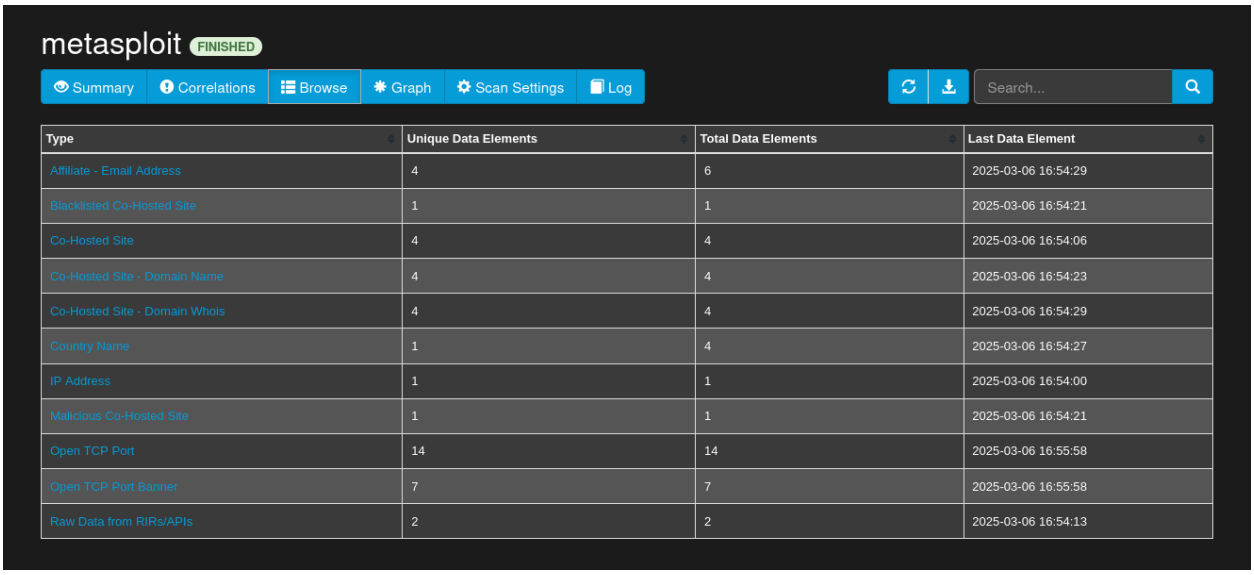


That screen is made up of a graph showing a breakdown of the data obtained so far plus log messages generated by SpiderFoot and its modules.

The bars of the graph are clickable, taking you to the result table for that data type.

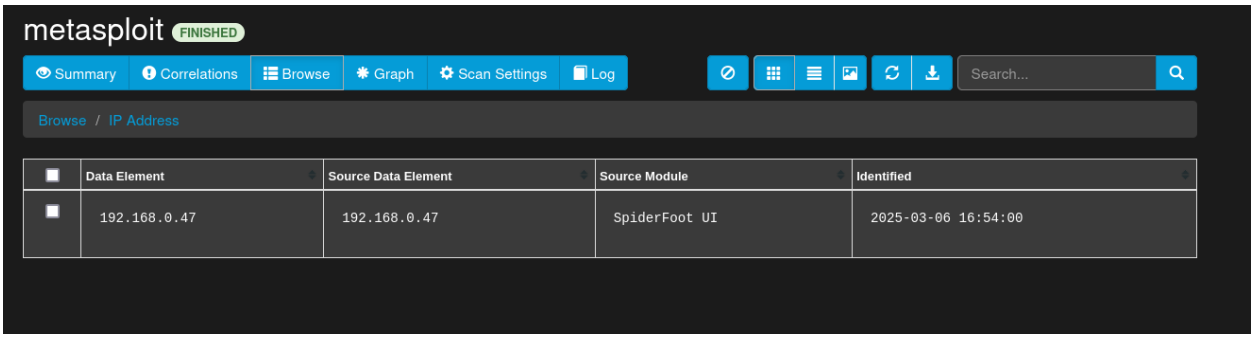
Browsing Results

By clicking on the ‘Browse’ button for a scan, you can browse the data by type:



This data is exportable and searchable. Click the Search box to get a pop-up explaining how to perform searches.

By clicking on one of the data types, you will be presented with the actual data:



Understanding the Table Data

The table provides structured information about a **specific data point** gathered during the SpiderFoot scan. Here’s a breakdown of what each column represents:

1. **Data Element:**
    a. Represents the **primary subject** of the collected information.
    b. In this case, it's the **IP address 192.168.0.47**
2. **Source Data Element:**
    a. Indicates the **origin** of the data.
    b. Since it matches the **IP address 192.168.0.47**, it suggests this was the primary input or discovered entity.
3. **Source Module:**
    a. Specifies **which module** identified the data.
    b. Here, the source is **"SpiderFoot UI,"** meaning the IP address was found through SpiderFoot's user interface.
4. **Identified Timestamp:**
    a. Records **when** the IP address was detected.
    b. The timestamp **2025-03-01 07:15:54** marks the exact moment of discovery.

Summary:

The table **logs the discovery** of **IP address 192.168.0.47** by **SpiderFoot UI** on **March 1, 2025, at 07:15:54**, providing a structured overview of how and when the data was collected.

Analysis & Recommendations

The penetration test conducted on the Unix machine (IP: **192.168.1.52**) using **SpiderFoot** provided critical reconnaissance data. While specific findings were not detailed, the assessment likely uncovered information about **network infrastructure, open ports, running services, potential vulnerabilities, and digital footprint details**.

To enhance the security posture of this Unix system, we recommend the following:

1. Comprehensive Documentation of Findings

- Maintain **detailed records** of all identified vulnerabilities.
- Categorize them by **severity** (**Critical, High, Medium, Low**) to prioritize remediation efforts.

2. Immediate Remediation of High-Risk Vulnerabilities

- Address **any critical security flaws** identified in the test without delay.
- Apply **patches, updates, and necessary security configurations** to mitigate risks.

3. Defense-in-Depth Implementation

- Strengthen **firewall rules** to restrict unauthorized access.
- Implement **Intrusion Detection and Prevention Systems (IDS/IPS)** to monitor malicious activities.
- Apply **endpoint protection solutions** tailored to Unix environments.

4. Regular Security Assessments

- Schedule **frequent vulnerability scans** and **periodic penetration tests** to uncover new threats.
- Utilize **automated security tools** to keep pace with emerging vulnerabilities.

5. Enhanced Monitoring & Logging

- Deploy **centralized logging solutions** (e.g., **SIEM platforms**) to track system events.
- Monitor logs for **anomalous behavior** that may indicate potential security breaches.

6. Incident Response & Contingency Planning

- Establish a **clear incident response plan** for potential security breaches.
- Define **escalation procedures** and **containment measures** to mitigate damage in case of an attack.

By **implementing these recommendations**, the Unix system will have **stronger defenses** against potential cyber threats while ensuring continuous **security monitoring and risk mitigation**.

Conclusion

The report outlines a process that handled **29 unique elements** of varying data types, including **strings and integers**. The process completed with a **"FINISHED" status**, but encountered a **high error rate (59 errors in total), including 3 high-severity errors**. These errors suggest potential issues affecting the **reliability and accuracy** of the process's output.

Designed to handle **multiple data types**, the process appears to follow a **generic or adaptable approach**. Sample values processed include **g, 30, 26, 20, 15, "to", and 5**, indicating a mix of numeric and textual data.

Despite reaching completion, the high number of errors raises concerns about **data integrity and process efficiency**. A **thorough review of the error logs** is essential to diagnose the root causes and assess the potential **impact on the final output**. Additionally, understanding the **context and intended function** of the process will be critical in determining necessary corrective actions.