IoT Penetration Test Report

Gagandeep Sur

gks007@shsu.edu

Contents

[1.Vulnerability Detail and Mitigation: 2](#_Toc101459305)

[2. Executive Summary: 4](#_Toc101459306)

[3. Methodology: 5](#_Toc101459307)

[4. Attack Narrative: 5](#_Toc101459308)

[5. Conclusion: 9](#_Toc101459309)

[6. Recommendations: 9](#_Toc101459310)

1.Vulnerability Detail and Mitigation:

**Vulnerabilities scanner tool Nessus was used to scan vulnerabilities on target 192.168.1.124. There were some critical and high vulnerabilities as shown in snapshots**

Graphical user interface, text, application

Description automatically generatedA screenshot of a computer

Description automatically generated

**Showing Critical, high, and medium vulnerabilities**

**CVE-2021-40438**: **Apache<2.4.49 Multiple Vulnerabilities**

**Rating**: Critical

**Description**: A crafted request Uri-path can cause mod proxy to forward the request to an origin server chosen by the remote user.

**Impact:** This issue affects Apache HTTP Server 2.4.48 and earlier.

**Remediation**: Upgrade to Apache version 2.4.49 or later

**Further Reading:** <https://nvd.nist.gov/vuln/detail/CVE-2021-40438>, <https://httpd.apache.org/security/vulnerabilities_24.html>

**CVE-2021-44224: Apache<2.4.52 Multiple Vulnerabilities**

**Rating**: Critical

**Description** A crafted URI sent to httpd configured as a forward proxy (Proxy Requests on) can cause a crash (NULL pointer dereference) or, for configurations mixing forward and reverse proxy declarations, can allow for requests to be directed to a declared Unix Domain Socket endpoint (Server-Side Request Forgery). T

**Impact:** This issue affects Apache HTTP Server 2.4.7 up to 2.4.51 (included).

**Remediation**: Upgrade to Apache version 2.4.52 or later

**Further Reading:** [https://nvd.nist.gov/vuln/detail/CVE-2021-44224](https://nvd.nist.gov/vuln/detail/CVE-2021-40438) <https://httpd.apache.org/security/vulnerabilities_24.html>

**CVE-2019-0217: Apache2.4<2.4.39 Multiple Vulnerabilities**

**Rating**: High

**Description** A crafted URI sent to httpd configured as a forward proxy (ProxyRequests on) can cause a crash (NULL pointer dereference) or, for configurations mixing forward and reverse proxy declarations, can allow for requests to be directed to a declared Unix Domain Socket endpoint (Server-Side Request Forgery).

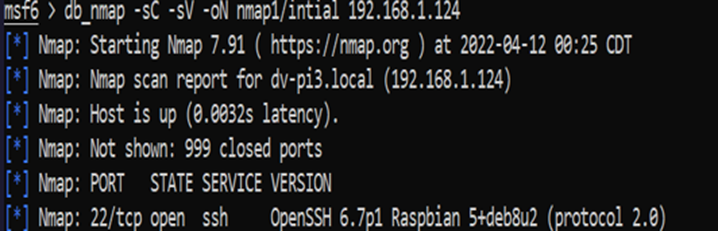
**Impact**: There can be reduced performance or interruptions in resource availability.

**Remediation**: Upgrade to Apache version 2.4.39 or later

**Further Reading:** https://www.cvedetails.com/cve/CVE-2019-0217/ <https://httpd.apache.org/security/vulnerabilities_24.html>

From the Nessus scan, I couldn’t find any critical vulnerability that I can use to exploit the remote machine. All the above critical vulnerabilities were related to Apache server that can be resolved with an upgrade to higher version of the server.

Then I looked it into NMAP scan results, Where I found SSH port 22 was open. I used this vulnerability to exploit the remote system and get it into the target system with SSH brute force attack as shown in screenshot below:



2. Executive Summary:

**Penetration** tests are carried out by simulating real-life attack scenarios without damaging the systems. Some critical vulnerabilities have been discovered during the penetration test. This report provides IoT penetration testing services and security research. This report is the result of security research conducted on our own customized IOT Device with camera, with the goals of

• Identifying if a remote attacker could gain control of the device to execute commands

• Identifying if a remote attacker could view the video feed

If attackers use these vulnerabilities, they can carry out a range of malicious acts. According to the worst-case scenario, your systems can be seized by malicious people. During the security tests, four critical, two high, three medium, and two low-level findings were detected. These findings were due to weak password usage, missing patches, misconfiguration, and improper input validation. Key recommendations are listed below:

● Setting a strong password policy: Weak passwords should not be allowed. Passwords should be changed periodically. Two-factor authentication should be used if possible.

● Making regular updates: It should regularly update the related systems.

● Input validation: All data received from the user must be properly filtered and used.

This research has been performed as part of an overall effort to improve the security of IoT devices. With massive IoT botnet attacks, hacks, and information disclosures making headlines on a regular basis.

**Summary of results**: Initial reconnaissance of IOT device network resulted in the discovery of open ports. The results provided us with a listing of specific ports to target for this assessment. After creating a custom wordlist using terms identified as password list. We were able to gain access to this interface by uncovering the password via brute-force. I was able to get access to the target system by using open port SSH 22. It did allow me to freely investigate the targets’ operating system, programs, databases, and configurations files which helped in our discovery of other vulnerabilities. I was able to get remote access view of the **Pi camera** streaming. It is unlikely for this port to be opened on the internet; therefore, it would most likely only be exploitable by someone on the same network as the camera. I was also able to get root privilege access of target system using **sudo -S su** command. It was very easy to get root privilege of the target system, which is another one of the vulnerabilities. Also, able to maintain access in target system using post module **SSHKEY\_PERSISTENCE.**

In addition, I found **/etc/shadow** file with password hashes, which I later cracked these password hashes with **John The Ripper tool**. I found passwords:

3. Methodology:

The test methodology is a 5-step process that starts with determining the scope of the test and finishes with preparing a report for the customer. The steps in this methodology can be used to understand the penetration testing process.

3.1. **Information Gathering:** This step is the most important step of the penetration test. Information about the target systems is collected. The information gathering phase is both an automated and a manual process.

3.2. **Vulnerability Assessment** Target systems are scanned by vulnerability scanning tools, and comparison is made with up-to-date vulnerability databases. Also, the information obtained in step 1 is researched on the internet for known vulnerabilities. Application or system reviews are performed. It is checked whether there are any logical or coding errors. I Vulnerability checks (automatic and manual) are performed for applications. This step includes but is not limited to:

● Network vulnerability scanning tools (NESSUS)

**3.3 Gained Access:** At this stage, a real attack will be performed based on the viable attack paths and planning in the analysis stage. For exploiting the vulnerabilities, we are employing the METASPLOIT framework. Pen testers can gain remote access to a system by exploiting it. The purpose of a pen tester is to delve as far as possible into the infrastructure to uncover valuable targets while avoiding detection. During penetration testing, the DDOS attack is prohibited to ensure the availability of the target.

**3.4. Exploitation:** This step is the second most crucial step after gathering information. Automatic tools outputs are evaluated at this step. Attacks are made like a real hacker, which means vulnerabilities are exploited without damaging the systems. This step includes but is not limited to:

● Exploitation of vulnerabilities

● Password cracking

**3.5. Reporting:** All findings are collected and analyzed. Evidence of vulnerabilities, data obtained from screenshots are matched with findings. A detailed report is created by adding solution suggestions and external sources. Risk levels of vulnerabilities were calculated according to the Common Vulnerability Scoring System (CVSS).

4. Attack Narrative:

Metasploit frame was used for this pen testing. We must start PostgreSQL database using command

**sudo systemctl start postgresql**

We can check database status using command

**sudo msfdb status**

It will show us that database is running and active

We can start msfconsole on linux from it

**Attacker Machine**: Kali Linux(started with IP address 198.168.1.178 but keeps on changing)

**Victim machine**: dv-pi 192.168.1.124 installed on Raspberry pi 3b+

**Remote access using open port SSH 22**

Using NMAP tool I found that SSH port number 22 is open, which can be exploited to get into the system.

To prepare a targeted brute-force attempt against this system, I compiled a dictionary file based on the content of Metasploit framework directory. The dictionary consisted of 1000 custom words. This dictionary file was used along with the username “pi”.

I used auxiliary(scanner/ssh/shh\_login) module that used Brute Force technique to crack the password of the ‘pi’ user to get into the system as shown in screen shots:

Text

Description automatically generated

Text

Description automatically generated



Password Brute force attack to get Success entry into the system

This brute-force attack uncovered a password of “raspberry” for the admin user. I was able to leverage these credentials to successfully gain unauthorized access to the target system

**Privilege Escalation**: Privilege escalation is a malicious attempt to abuse an app or OS bug or error of configuration at gaining unauthorized access to sensitive information by taking over a user’s account that has the necessary privileges to view or commit modifications to the said information and that wouldn’t be normally accessible to the current user.

With interactive access to the underlying operating system of the administrative obtained, I continued with the examination of the system searching for ways to escalate privileges to the administrative level.

**Sodo su** command takes me to the root user. With root user I was able to download sensitive information from remote machine(dv-pi) to local machine (Kali Linux) like password file and hashed password file (shadow file).

Text

Description automatically generated with medium confidence

By obtaining these types of rights, a malicious actor can perform a series of actions to the operating system. I was able to get root level privileges.

**/etc/shadow and /etc/password file disclosure:**

As I written above, I was able to access /etc/shadow file with root privileges. I also downloaded this file to my local machine

Text

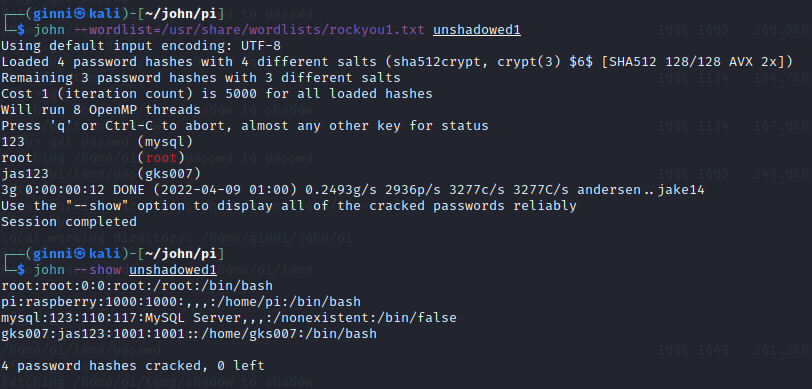
Description automatically generated

/etc/shadow file access from root user

**Cracked Password with JOHN THE RIPPER:**

Used JTR tool to crack password from shadow file. Here we can see, it cracked 4 hashes, that are highlighted below in snapshot.







**Maintained Access using SSHKey\_Persistance module:**

Maintaining Access” is a phase of the penetration testing cycle which has a very concrete purpose – to allow the pen tester to linger in the targeted systems until he acquires what information he considers to be valuable and then manages to extract it successfully from the system.

I used the **post/linux/manage/sshkey\_persistence** Metasploit module. This module will add an SSH key to a specified user (or all), to allow remote login via SSH at any time from local machine (kali linux) as shown in snapshot :

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence



remote login via SSH at any time from local machine (kali linux)

**Remotely Camera feed:**

Made a payload naming ‘streamingserver.py’ and upload into the target system directory and run in the browser using http://192.168.1.124:1981 , I was able to view live streaming of Target camera. I took a snapshot of this as below:

A picture containing text, floor, indoor

Description automatically generated

live streaming of Target’s Camera

5. Conclusion:



The specific goals of the penetration test were stated as:

• Identifying if a remote attacker could gain control of the device to execute commands

• Identifying if a remote attacker could view the video feed, In pursuit of those objectives, I was able to remotely gain control of the target system and execute arbitrary commands. I was also able to get live feed of remote camera too.

6. RECOMMENDATIONS:

Recommendations Due to the impact to the overall organization as uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner. While a comprehensive list of items that should be implemented is beyond the scope of this engagement, some high-level items are important to mention. Offensive Security recommends the following:

1. **Ensure that strong credentials are used.** The compromise of this IOT device as drastically impacted by the use of weak passwords as well as the reuse of passwords across systems of differing security levels. NIST SP 800-119 is recommended for guidelines on operating an enterprise password policy.

**2. Implement a patch management program**: Operating a consistent patch management program per the guidelines outlined in NIST SP 800-4010 is an important component in maintaining good security posture. This will help to limit the attack surface that results from running unpatched internal services.

**3. Conduct regular vulnerability assessments**. As part of an effective organizational risk management strategy, vulnerability assessments should be conducted on a regular basis. Doing so will allow the organization to determine if the installed security controls are properly installed, operating as intended, and producing the desired outcome.

**4. Timely Upgradation of system:** Most of the vulnerabilities in this device are due to the pending upgradation of server. There should be timely upgradation of system, server to avoid existing vulnerabilities.