**Penetration**

**Test Report for Victim 3**

**-Sairam Bokka**

**VX05251**

**Sbokka1@umbc.edu**

**Contents**

1. **Executive Summary-------------------------------------------------------------------------------- 3**
2. **Host Discovery -------------------------------------------------------------------------------------- 4**
3. **Information Gathering ---------------------------------------------------------------------------- 5**
4. **Exploitation ------------------------------------------------------------------------------------------ 6**
5. **Privilege Escalation -------------------------------------------------------------------------------- 8**
6. **Recommendations --------------------------------------------------------------------------------10**

**Executive Summary:**

The victim PC runs an older version of Ubuntu OS. We discover the victim PC using a simple nmap scan over the local network and discover the vulnerable machine. We do a thorough scan of the victim PC using nmap and discover all the open ports of the victim PC along with version information of all the applications running on different ports of the machine. After doing a basic network scan using nessus, we discover multiple vulnerabilities on the victim machine. We start by exploiting the ProFTPD service running on port 21 as FTP service. After running a module from metasploitable, we get a remote shell running on our machine. We also try to exploit the application running on 8181, which has an encrypted cookie. After decrypting it from URL decoding and base64 decoding, we get the cookie value. We use this value to exploit the application using metasploitable, which gives use a remote shell with root privileges.

**Host Discovery**:

A screenshot of a computer

Description automatically generated

By doing a simple nmap scan over my network, we find that we have three open devices. One is my windows machine, one is kali machine and one is unknown ubuntu machine which is Victim3.

The IP address of Victim3 was found to be 192.168.17.134.

**Information Gathering**:

Now we do a thorough scan of victim 3 using the following command: nmap -sV -p- 1-65535 192.168.17.134

-sV will help us with service version detection and -p- is used to specify the ports we are scanning.

As we can see in the screenshot, we find multiple open ports listed.

A screenshot of a computer

Description automatically generated

After doing a basic network scan using nessus, we find out a lot of high-critical vulnerabilities in the machine. We will check every port and service for vulnerabilities and try to exploit them.

**Exploitation:**

1. ProFTPD 1.3.5

We open metasploitable and search for any available exploits for ProFTPD, because according to nessus, it is a critical vulnerability that could allow for remote code execution.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

We use exploit -> unix/ftp/proftpd\_modcopy\_exec and set payload -> cmd/unix/reverse\_perl

A screenshot of a computer

Description automatically generated

After setting the RHOSTS and LHOST, we run the payload, which creates a remote shell to the victim machine.

1. Apache HTTP server:

According to CVE-2014-6271, this version of Apache http server 2.4.7 is vulnerable to remote code execution. Once we dig a little deeper, we find out that it can be exploited using multi/http/apache\_mod\_cgi\_bash\_env\_exec

We set the RHOSTS and TARGETURI to the cgi-bin and run the exploit.

A screenshot of a computer

Description automatically generated

We open a meterpreter session on the victim machine.

**Privilege Escalation**:

1. Ruby on rails application exploit:

A screenshot of a computer

Description automatically generated

We see that there is an Webrick http application running on ruby on rails on port 8181. This is interesting since there seems to be higher reward for exploiting this app.

A screenshot of a computer

Description automatically generated

We have a /flag path, which leads to something to do with cookies. We can see one \_metasploitable cookie whose value is encoded in some way.

We decode it in cyberchef using URL decode and base64 decode and we find the secret value as shown below

A screenshot of a computer

Description automatically generated

We use this value to break into the application using one of the exploits in metasploitable called rails secret deserialization which can use this cookie value to get a remote session.

A computer screen shot of a computer

Description automatically generated

Now we get shell access as root.

**Recommendations**:

1. Update and patch:

* Ensure that the Ubuntu operating system is updated with the latest security patches and updates.
* Keep the web application framework and all its dependencies up to date with the latest stable versions.

1. Secure configuration:

* Review and harden the web server configuration (e.g., Apache or Nginx) to follow security best practices.
* Disable unnecessary modules, services, and features that are not required for the application's functionality.
* Configure proper file and directory permissions to prevent unauthorized access.

1. Input validation and sanitization:

* Implement strict input validation and sanitization mechanisms to prevent common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and command injection.
* Validate and sanitize user inputs on both the client-side and server-side.
* Use parameterized queries or prepared statements when interacting with databases to mitigate SQL injection risks.

1. Authentication and authorization:

* Implement strong authentication mechanisms, such as multi-factor authentication (MFA), to protect user accounts.
* Enforce secure password policies, including minimum length, complexity, and regular password updates.
* Properly implement user roles and access controls to ensure that users can only access the resources and functionalities they are authorized for.

1. Secure communication:

* Enable HTTPS/SSL to encrypt all sensitive data transmitted between the client and the server.
* Use strong encryption algorithms and properly configure SSL/TLS certificates.
* Implement HTTP Strict Transport Security (HSTS) to enforce secure connections.

1. Error handling and logging:

* Implement proper error handling mechanisms to prevent sensitive information from being disclosed in error messages.
* Log security-related events, such as authentication attempts, access violations, and system errors, for monitoring and incident response.

1. Security headers: Implement security headers such as X-XSS-Protection, X-Frame-Options, X-Content-Type-Options, and Content-Security-Policy to protect against various web vulnerabilities.
2. Least privilege principle: Ensure that the web application runs with the least privileges necessary to perform its functions. Avoid running the application with root or administrative privileges.
3. Regular security testing: Conduct regular penetration testing and vulnerability assessments to identify and address any new security issues. Perform code reviews and security audits to identify and fix vulnerabilities in the application's codebase.
4. Security awareness and training: Provide security awareness training to developers, administrators, and users to educate them about secure coding practices, common vulnerabilities, and best practices for maintaining a secure environment.