Penetration

Test Report for

Victim 2

-Sairam Bokka

VX05251

CYBR 644

**Contents**

1. **Executive Summary------------------------------------------------------------------------------- 3**
2. **Host Discovery ------------------------------------------------------------------------------------- 4**
3. **Information Gathering --------------------------------------------------------------------------- 5**
4. **Exploitation ----------------------------------------------------------------------------------------- 7**
5. **Recommendations -------------------------------------------------------------------------------- 9**

**Executive Summary:**

We discover the victim machine running over the local network using a simple nmap scan. Once discovered, we enumerate the victim machine using nmap and discover open ports along with version detection that gives us a lot of information. We can simply login on port 512, 513, 514 using rsh client and get root access without passwords. We also have a lot of vulnerable web applications running on the machine which can give us remote code execution capabilities.

**Host Discovery**:

A screenshot of a computer

Description automatically generated

We discovered the host using a simple nmap scan over my network. After removing the IP address of Windows machine and kali linux, we can confirm that the IP address of the victim machine is 192.168.17.133

**Information Gathering:**

A screenshot of a computer

Description automatically generated

We do a thorough scan using nmap enumerating the ports and versions using the following command:

Nmap -sV -p- 1-65535 192.168.17.133

We can see a bunch of open ports with their versions displayed. This is a web application hosted on port 80.

A screenshot of a computer

Description automatically generated

As we can see, there are multiple vulnerable web applications hosted on this machine. Some of them are multidae, DVWA, and WebDAV. We can exploit almost all these applications and gain remote access to the victim machine.

**Exploitation:**

1. Gaining Access: the username and password to access the machine are displayed on the home page of the webapp. They are msfadmin:msfadmin. We can use these credentials to login into the victim machine.
2. Unreal IRCD backdoor: On port 6667, Victim machine runs the UnreaIRCD IRC daemon. This version contains a backdoor that went unnoticed for months - triggered by sending the letters "AB" following by a system command to the server on any listening port. Metasploit has a module to exploit this to gain an interactive shell, as shown below.

A screenshot of a computer program

Description automatically generated

We got access to the shell as root user.

1. rlogin vulnerability: TCP ports 512, 513, and 514 are known as "r" services, and have been misconfigured to allow remote access from any host. We can just run rlogin as root user from any linux machine to gain access to the victim machine as root user.

A screenshot of a computer

Description automatically generated

Once we gain access as root, we can simply gain access to etc/passwd and etc/shadow files to get all the users and passwords from the victim machine.

**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.