

# Penetration test report and response

*"This document outlines a report on a penetration test simulation conducted by a cybersecurity student in a virtual environment."*

## INDICE

### 1. Introduction

#### 1.1 Context and Background Information

### 2. Methodology

#### 2.1 Description of the Methodologies Used

#### 2.2 Tools and Techniques Used

### 3. Results

#### 3.1 Severity Level Associated with Each Vulnerability

#### 3.2 Detailed List of Identified Vulnerabilities

#### 3.3 Description of Potential Consequences if Exploited

### 4 Conclusion

# 1. Introduction

*This report documents the results of a penetration test conducted using Nessus Essentials on a virtual system with the IP address 192.168.0.101. The purpose of this test was to assess the security of a simulated environment in order to identify potential vulnerabilities and security risks.*

During the test, various types of vulnerabilities were detected, classified according to their severity. The detected vulnerability categories include CRITICAL, HIGH, MEDIUM, LOW, and INFO. These details allow us to assess the severity of potential threats to the system.

## 1.1 Context and Background Information:

The target system has been identified as "METASPLOITABLE," with MAC address 08:00:27:B1:7C:1D and operating system Linux Kernel 2.6 on Ubuntu 8.04 (hardy). These details provide context and basic information for understanding the test results.

The report will provide a detailed overview of the identified vulnerabilities, assessing their potential impact on the system and providing recommendations for mitigating the identified security risks.

# 1. Methodology

During the penetration test, various methodologies were employed to conduct a comprehensive and thorough analysis of the security of the target system. The main methodologies and techniques used were selected to identify a wide range of potential vulnerabilities and to assess the system's resilience to attacks.

## 2.1 Description of the Methodologies Used:

1. **Information Gathering Approach (Reconnaissance):** This phase involved gathering data and information about the target system, including network infrastructure information, running services, software versions, and other relevant information to identify potential weak points.
2. **Vulnerability Analysis:** Once basic information was gathered, a thorough analysis of vulnerabilities was conducted using tools like Nessus Essentials. This included scanning the system to identify and catalog existing vulnerabilities, including software and operating system vulnerabilities.

3. **Attack Phase:** Based on the vulnerabilities identified during the analysis phase, targeted penetration tests were developed and executed to determine if vulnerabilities could be exploited to gain unauthorized access to the system or compromise its integrity.
4. **Post-Attack and Results Analysis:** After completing the attack tests, the results were analyzed to assess the effectiveness of existing security countermeasures and to identify any necessary corrective actions. This phase also includes detailed documentation of all findings, including the exact steps followed during the attack and proof of concept (PoC) for identified vulnerabilities.

## 2.2 Tools and Techniques Used:

Several tools and techniques were employed during the test, including:

- Nessus Essentials for vulnerability scanning and identification.
- Metasploit Framework for developing and executing penetration tests.
- Nmap for port scanning and detecting running services.
- Wireshark for network traffic analysis.
- DNS enumeration for gathering information about the target system.

These tools and techniques enabled a comprehensive analysis of the security of the target system, identifying and assessing existing vulnerabilities, and providing a critical assessment of its resistance to attacks.

## 2. Results:

The scanning process revealed a comprehensive breakdown of vulnerabilities within the system. Specifically, our analysis unveiled a concerning total of 13 critical vulnerabilities, signaling immediate and high-priority security risks. Additionally, we identified 10 high-risk vulnerabilities that demand prompt attention to mitigate potential exploitation. Furthermore, the scan detected 35 medium-level vulnerabilities, highlighting areas of moderate concern that still pose significant security risks if left unaddressed. Lastly, we found 13 low-risk vulnerabilities, which, although less urgent, should not be overlooked as they could serve as potential entry points for attackers seeking to infiltrate the system. Overall, this breakdown underscores the diverse range of vulnerabilities present within the system and emphasizes the importance of implementing comprehensive security measures to safeguard against potential threats.



#### Scan Information




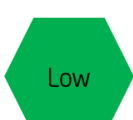
Start time: Wed May 8 12:27:32 2024  
End time: Wed May 8 12:56:53 2024

#### Host Information

Netbios Name: METASPLOITABLE  
IP: 192.168.0.101  
MAC Address: 08:00:27:B1:7C:1D  
OS: Linux Kernel 2.6 on Ubuntu 8.04 (hardy)

### 3.1 Severity Level Associated with Each Vulnerability:

#### Severity Level Description

Severity	CVSSv2 Score	Explanation
	9.0-10.0	Critical risk vulnerabilities will have a crippling effect on this service. Vulnerabilities of this level usually result in complete compromise of the affected host along with the possible network it resides on. In most instances, the exploit requires little to no knowledge and can be easily implemented.
	7.0-8.9	High risk vulnerabilities will be able to access potential sensitive information and cause denial of service (DOS) conditions. The severity is reduced as the issue is more difficult to exploit than that of a critical risk issue.
	4.0-6.9	Medium risk vulnerabilities will most often require further determination and technical ability to create a noticeable affect to an organisations business. In some cases, these issues require a high level of resourcing which can only be available by the likes of a funded project.
	0.1-3.9	Low risk vulnerabilities have very little impact on an organisation's business. Exploitation of such vulnerabilities would either require local privileged access or to be used in combination to other findings.



## Vulnerabilities

Total: 149

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	8.9	70728	Apache PHP-CGI Remote Code Execution
CRITICAL	9.8	-	51988	Bind Shell Backdoor Detection
CRITICAL	9.8	-	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	9.8	-	125855	phpMyAdmin prior to 4.8.6 SQLi vulnerability (PMASA-2019-3)
CRITICAL	10.0	-	171340	Apache Tomcat SEoL (<= 5.5.x)
CRITICAL	10.0	-	33850	Unix Operating System Unsupported Version Detection
CRITICAL	10.0*	5.1	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0*	5.1	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
CRITICAL	10.0*	5.9	11356	NFS Exported Share Information Disclosure
CRITICAL	10.0*	7.4	46882	UnrealIRCd Backdoor Detection
CRITICAL	10.0*	-	61708	VNC Server 'password' Password
HIGH	8.8	7.4	19704	TWiki 'rev' Parameter Arbitrary Command Execution
HIGH	8.6	-	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	-	42256	NFS Shares World Readable
HIGH	7.5	6.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5	-	90509	Samba Badlock Vulnerability
HIGH	7.5*	8.9	59088	PHP PHP-CGI Query String Parameter Injection Arbitrary Code Execution
HIGH	7.5*	6.7	36171	phpMyAdmin Setup Script Configuration Parameters Arbitrary PHP Code Injection (PMASA-2009-4)
HIGH	7.5*	5.9	10205	rlogin Service Detection
HIGH	7.5*	5.9	10245	rsh Service Detection

## 3.2 Detailed List of Identified Vulnerabilities:

### Vulnerabilities:

**70728 – Apache PHP-CGI Remote Code Execution**

**RISK FACTOR - CRITICAL**

#### Synopsis

The remote web server contains a version of PHP that allows arbitrary code execution.

### *Description*

The PHP installation on the remote web server contains a flaw that could allow a remote attacker to pass command-line arguments as part of a query string to the PHP-CGI program. This could be abused to execute arbitrary code, reveal PHP source code, cause a system crash, etc.

### *Solution*

Upgrade to PHP 5.3.13 / 5.4.3 or later

171340 - Apache Tomcat SEoL (<= 5.5.x)

RISK FACTOR - CRITICAL

### *Synopsis*

An unsupported version of Apache Tomcat is installed on the remote host.

### *Description*

According to its version, Apache Tomcat is less than or equal to 5.5.x. It is, therefore, no longer maintained by its vendor or provider. Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it may contain security vulnerabilities.

### *Solution*

Upgrade to a version of Apache Tomcat that is currently supported.

51988 - Bind Shell Backdoor Detection

RISK FACTOR - CRITICAL

### *Synopsis*

The remote host may have been compromised.

### *Description*

A shell is listening on the remote port without any authentication being required. An attacker may use it by

connecting to the remote port and sending commands directly.

### *Solution*

Verify if the remote host has been compromised, and reinstall the system if necessary.

```
Nessus was able to execute the command "id" using the
following request :

This produced the following truncated output (limited to 10 lines) :
----- snip -----
root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#
----- snip -----
```

## 32314 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

### RISK FACTOR - CRITICAL

#### *Synopsis*

The remote SSH host keys are weak.

#### *Description*

The remote SSH host key has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to set up decipher the remote session or set up a man in the middle attack.

#### *Solution*

Consider all cryptographic material generated on the remote host to be guessable. In particular, all SSH, SSL and OpenVPN key material should be re-generated.

## 32321 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)

### RISK FACTOR - CRITICAL

#### *Synopsis*

The remote SSL certificate uses a weak key.

### *Description*

The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to decipher the remote session or set up a man in the middle attack.

### *Solution*

Consider all cryptographic material generated on the remote host to be guessable. In particular, all SSH, SSL and OpenVPN key material should be re-generated.

**32321 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)**

**FACTOR RISK - CRITICAL**

### *Synopsis*

The remote SSL certificate uses a weak key.

### *Description*

The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to decipher the remote session or set up a man in the middle attack.

### *Solution*

Consider all cryptographic material generated on the remote host to be guessable. In particular, all SSH, SSL and OpenVPN key material should be re-generated.

**11356 - NFS Exported Share Information Disclosure**

**FACTOR RISK - CRITICAL**

### *Synopsis*

It is possible to access NFS shares on the remote host.



### *Description*

At least one of the NFS shares exported by the remote server could be mounted by the scanning host. An attacker may be able to leverage this to read (and possibly write) files on remote host.

### *Solution*

Configure NFS on the remote host so that only authorized hosts can mount its remote shares.

```
The following NFS shares could be mounted :
```

```
+ /
```

```
+ Contents of / :
```

```
- .  
- ..  
- bin  
- boot  
- cdrom  
- dev  
- etc  
- home  
- initrd  
- initrd.img  
- lib  
- lost+found  
- media  
- mnt  
- nohup.out  
- opt  
- proc  
- root  
- sbin  
- srv  
- sys  
- tmp  
- usr  
- var  
- vmlinuz
```

## 20007 - SSL Version 2 and 3 Protocol Detection

### RISK FACTOR - CRITICAL

### *Synopsis*

The remote service encrypts traffic using a protocol with known weaknesses.

### *Description*

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.
- Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients. Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely. NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

### Solution

Consult the application's documentation to disable SSL 2.0 and 3.0.

Use TLS 1.2 (with approved cipher suites) or higher instead.

```
- SSLv3 is enabled and the server supports at least one cipher.
Explanation: TLS 1.0 and SSL 3.0 cipher suites may be used with SSLv3

Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

Name           Code           KEX           Auth           Encryption           MAC
-----
EDH-RSA-DES-CBC3-SHA
SHA1
DES-CBC3-SHA
SHA1
RSA             RSA             3DES-CBC(168)

High Strength Ciphers (>= 112-bit key)

Name           Code           KEX           Auth           Encryption           MAC
-----
DHE-RSA-AES128-SHA
SHA1
DHE-RSA-AES256-SHA
SHA1
AES128-SHA
SHA1
AES256-SHA
SHA1
RC4-SHA
SHA1
RSA             RSA             AES-CBC(128)
RSA             RSA             AES-CBC(256)
RSA             RSA             AES-CBC(128)
RSA             RSA             AES-CBC(256)
RSA             RSA             RC4(128)

The fields above are :

{Tenable ciphername}
{Cipher ID code}
Kex={key exchange}
Auth={authentication}
Encrypt={symmetric encryption method}
MAC={message authentication code}
{export flag}
```

## 33850 - Unix Operating System Unsupported Version Detection

### RISK FACTOR - CRITICAL

#### Synopsis

The operating system running on the remote host is no longer supported.

### *Description*

According to its self-reported version number, the Unix operating system running on the remote host is no longer supported. Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.

### *Solution*

Upgrade to a version of the Unix operating system that is currently supported.

```
Ubuntu 8.04 support ended on 2011-05-12 (Desktop) / 2013-05-09 (Server).  
Upgrade to Ubuntu 23.04 / LTS 22.04 / LTS 20.04 .  
  
For more information, see : https://wiki.ubuntu.com/Releases
```

## 46882 - UnrealIRCd Backdoor Detection

### FACTOR RISK - CRITICAL

### *Synopsis*

The remote IRC server contains a backdoor.

### *Description*

The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.

### *Solution*

Re-download the software, verify it using the published MD5 / SHA1 checksums, and re-install it.

```
The remote IRC server is running as :  
uid=0(root) gid=0(root)
```

## 61708 - VNC Server 'password' Password

### RISK FACTOR - CRITICAL

### *Synopsis*

A VNC server running on the remote host is secured with a weak password.

### *Description*

The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

### *Solution*

Secure the VNC service with a strong password.

## 3.3 Description of Potential Consequences if Exploited:

Here's a description of the potential consequences of the vulnerabilities identified in the Nessus Essentials report:

### **1. Apache PHP-CGI Remote Code Execution (CVE-2012-1823):**

This critical vulnerability allows a remote attacker to execute arbitrary code on the Apache web server by exploiting a vulnerability in the PHP-CGI module. If successfully exploited, an attacker could gain full control of the web server and access sensitive information or compromise the system.

### **2. Bind Shell Backdoor Detection:**

This critical vulnerability indicates the presence of a "Bind Shell" type backdoor in the system. A backdoor allows an attacker to gain unauthorized access to the system, bypassing normal authentication procedures. This could allow attackers to perform malicious actions on the system, such as installing malware or compromising data.

### **3. SSL Version 2 and 3 Protocol Detection (CVE-2014-3566, CVE-2015-3197):**

These critical vulnerabilities detect the use of SSL Version 2 and 3 protocols, known to have serious security flaws. The use of these protocols exposes the system to risks of attacks such as POODLE and other downgrade-based attacks. Attackers could intercept and compromise secure communications between the server and clients, gaining unauthorized access to sensitive information.

### **4. phpMyAdmin prior to 4.8.6 SQLi vulnerability (PMASA-2019-3):**

This critical vulnerability allows an attacker to execute SQL injection (SQLi) attacks through phpMyAdmin, a popular MySQL database management tool. By exploiting this vulnerability, an attacker could manipulate database queries to access, modify, or delete sensitive data in the database, compromising data integrity and confidentiality.

## **5. Apache Tomcat SEoL (CVE-2017-12617):**

This critical vulnerability allows an attacker to execute arbitrary code on the Apache Tomcat server by exploiting a vulnerability in the default configuration. An attacker could exploit this vulnerability to gain unauthorized access to the server and compromise data hosted on the system.

These are just some of the potential consequences that could result from exploiting the vulnerabilities identified in the system. It's important to promptly and properly address these vulnerabilities to protect the system and company data from harmful attacks and financial losses.

## **4 Conclusion:**

The conducted penetration test has unveiled numerous critical vulnerabilities within the system, posing significant cybersecurity risks to the company. These vulnerabilities range from possibilities of remote code execution to exposure of backdoors and weaknesses in encryption protocols. If exploited by malicious actors, these vulnerabilities could lead to severe disruptions to business operations, compromise data confidentiality, and result in financial losses.

It is imperative that immediate measures are taken to address and remediate these vulnerabilities. This may include applying security patches, updating systems, and implementing countermeasures to mitigate identified risks. Additionally, it is essential to adopt robust security policies and procedures and provide staff training to ensure ongoing and effective protection against cyber threats.

Recognizing the gravity of the identified vulnerabilities, it is strongly recommended to act promptly and engage stakeholders to ensure that necessary corrective actions are taken expeditiously. Only through a collective commitment to cybersecurity can we effectively safeguard the company from security risks and preserve the trust of customers and stakeholders.