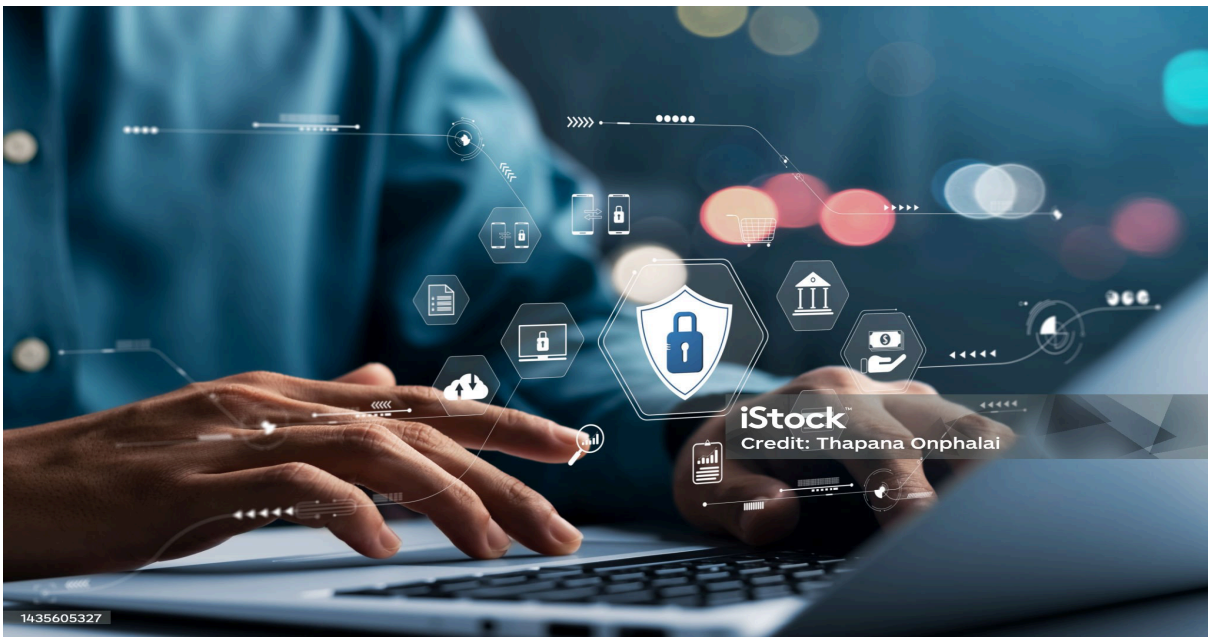


Comprehensive Analysis and Remediation of a Compromised Debian Server - Phase 2

Guillermo J. Costa H.
December, 9, 2024
4Geeks Academy

Introduction

In a digital environment where cybersecurity threats evolve constantly, detecting and addressing vulnerabilities are essential steps to protect critical systems. In this second phase, an in-depth analysis of services and configurations on our previously compromised machine was conducted. Using advanced tools and security methodologies, additional risks were identified and mitigated with secure configurations and established procedures. This document outlines the findings, corrective actions implemented, and results obtained, consolidating a more robust and secure operational environment.

1. **Service and Port Scanning**:
   - A scan was performed using `nmap` to identify vulnerable services and their versions.
   - Command used: `nmap -sV --script=vuln 192.168.56.115`.

```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.56.115
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://192.168.56.115:80/manual
|     Form id: wp-block-search__input-2
|     Form action: http://localhost/
|
|     Path: http://192.168.56.115:80/apache2;repeatmerged=0
|     Form id: wp-block-search__input-2
|_    Form action: http://localhost/
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-server-header: Apache/2.4.62 (Debian)
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
| http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
|   /robots.txt: Robots file
|   /readme.html: Wordpress version: 2
|   /wp-includes/images/rss.png: Wordpress version 2.2 found.
|   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|   /wp-login.php: Wordpress login page.
|   /wp-admin/upgrade.php: Wordpress login page.
|   /readme.html: Interesting, a readme.
|_  /0/: Potentially interesting folder
MAC Address: 08:00:27:75:E9:4E (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
debian@debian:/var$ sudo nmap -sS -p- localhost
Starting Nmap 7.93 ( https://nmap.org ) at 2024-11-20 20:16 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000080s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 65529 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
631/tcp  open  ipp
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 1.32 seconds
```

2. The scan performed with `nmap` identified active services on the machine and their respective versions. Key findings included services such as SSH on port 22, SMTP on port 25, HTTP on port 80, and MySQL on port 3306. Each of these services

presented potential vulnerabilities, which were addressed in subsequent mitigation steps.

3. **Vulnerability Analysis**:
    - A review of detected services and their configurations was conducted.
    - Associated risks and possible attack vectors were evaluated.
4. **Mitigation**:
    - Secure configurations were applied, and detected vulnerabilities were mitigated.
5. **Verification**:
    - Tests were repeated to confirm that the risks had been mitigated.

Scan Results

Detected Services

- **SSH (Port 22)**:
    - **Risks**: Brute force attacks and vulnerable default configurations.
    - **Action Taken**: The listening port was changed to 2222, and `iptables` rules were implemented to restrict access.

```
Include /etc/ssh/sshd_config.d/*.conf

Port 2222
#AddressFamily any
```

- Changing the SSH port from 22 to 2222 enhances security by reducing the likelihood of automated attacks, such as brute force attempts, which typically target default ports. Additionally, implementing `iptables` rules to restrict access only to trusted IPs prevents unauthorized connections and protects against malicious network scans. Together, these measures significantly increase the difficulty for attackers to access the system.

- **SMTP (Port 25)**:
  - **Risks**: Misuse for spam and potential exposure of information.
  - **Action Taken**: The service was disabled as it was not in use. Rules were configured for ports 25, 465, and 587, and regular audits were implemented for future activations.

```
debian@debian:~$ sudo nmap -p 25 localhost
Starting Nmap 7.93 ( https://nmap.org ) at 2024-12-10 23:17 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000062s latency).
Other addresses for localhost (not scanned): ::1

PORT   STATE  SERVICE
25/tcp closed smtp

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

- Disabling the SMTP service is relevant because, when improperly configured or unused, it can become a significant attack vector. By keeping port 25 open without protection, attackers could use it to send spam or even launch phishing attacks via the server. By disabling SMTP and adjusting `iptables` rules for ports 25, 465, and 587, this entry point for potential attackers is eliminated. This action also reduces the system's attack surface, enhancing overall security while allowing for secure reactivation in the future with mandatory authentication and other robust measures.

- **HTTP (Port 80)**:
  - **Risks**: Vulnerabilities in outdated versions of Apache and lack of SSL.
  - **Action Taken**: Apache was updated to the latest version, and SSL implementation for HTTPS was confirmed.
- To confirm SSL implementation, tests were conducted by accessing the site via HTTPS and verifying the digital certificate using tools like `openssl` and web browsers. The results showed a valid and active certificate, ensuring secure data transmission between client and server. This eliminates risks associated with unencrypted connections, such as data interception or man-in-the-middle (MITM) attacks.

- **MySQL (Port 3306)**:
  - **Risks**: Database exposure and weak passwords.
  - **Action Taken**: Excessive privileges were revoked, and secure passwords were configured for the users `root`, `user`, and `wordpressuser`.

```
MariaDB [(none)]> SELECT user, host FROM mysql.user;
+---------------+-----------+
| User          | Host      |
+---------------+-----------+
| mariadb.sys   | localhost |
| mysql         | localhost |
| root          | localhost |
| user          | localhost |
| wordpressuser | localhost |
+---------------+-----------+
5 rows in set (0.180 sec)
```

```
MariaDB [(none)]> SHOW GRANTS FOR 'root'@'localhost';
+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
----------------+
| Grants for root@localhost

                |
+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
----------------+
| GRANT ALL PRIVILEGES ON *.* TO `root`@`localhost` IDENTIFIED VIA mysql_nativ
e_password USING '*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9' OR unix_socket WI
TH GRANT OPTION |
| GRANT PROXY ON ''@'%' TO 'root'@'localhost' WITH GRANT OPTION

                |
```

```
MariaDB [(none)]> SHOW GRANTS FOR 'user'@'localhost';
+-----------------------------------------------------------------------------
-------------------------------------------------------------+
| Grants for user@localhost
                                                             |
+-----------------------------------------------------------------------------
-------------------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO `user`@`localhost` IDENTIFIED BY PASSWORD '*2
470C0C06DEE42FD1618BB99005ADCA2EC9D1E19' WITH GRANT OPTION |
+-----------------------------------------------------------------------------
-------------------------------------------------------------+
1 row in set (0.004 sec)
```

```
MariaDB [(none)]> SHOW GRANTS FOR 'wordpressuser'@'localhost';
+-----------------------------------------------------------------------------
-----------------------------------------+
| Grants for wordpressuser@localhost
                                         |
+-----------------------------------------------------------------------------
-----------------------------------------+
| GRANT USAGE ON *.* TO `wordpressuser`@`localhost` IDENTIFIED BY PASSWORD '*6
BB4837EB74329105EE4568DDA7DC67ED2CA2AD9' |
| GRANT ALL PRIVILEGES ON `wordpress`.* TO `wordpressuser`@`localhost`
                                         |
```

```
File  Edit  View  Search  Terminal  Help
debian@debian:~/john/run$ ./john --format=mysql-sha1 --wordlist=/usr/share/wor
dlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (mysql-sha1, MySQL 4.1+ [SHA1
 128/128 SSE4.1 4x])
Cracked 1 password hash (is in ./john.pot), use "--show"
Remaining 1 password hash
Warning: no OpenMP support for this hash type, consider --fork=2
Note: Passwords longer than 10 [worst case UTF-8] to 32 [ASCII] rejected
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
password         (?)
1g 0:00:00:00 DONE (2024-11-25 21:04) 6.250g/s 25.00p/s 25.00c/s 25.00C/s 1234
56..password
Warning: passwords printed above might not be all those cracked
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
debian@debian:~/john/run$ ./john --show hash.txt
?:123456
?:password
?:123456

3 password hashes cracked, 0 left
debian@debian:~/john/run$
```

- Changes in passwords and privileges significantly enhance the system's overall security by addressing several key risks:

1. **Weak Passwords**: Initial passwords were vulnerable to brute force and dictionary attacks, allowing potential attackers to gain access. Replacing them with strong, complex passwords significantly reduced the likelihood of exploitation.
2. **Excessive Privileges**: Some users, such as `wordpressuser` and `user`, had unnecessary permissions that could have been used to escalate privileges or compromise other areas of the system. By revoking these permissions and restricting them only to necessary databases, the scope of possible attacks was minimized.
3. **Improved Auditing**: The changes provide a solid foundation for implementing regular security audits, ensuring configurations remain aligned with best practices.

Together, these modifications increase the system's resilience and significantly limit attack opportunities.

- **IPP (Port 631)**:
  - **Risks**: Exposure of the printing service.
  - **Action Taken**: Access was restricted to local IPs, and HTTPS usage was enforced.

Permission Reviews

1. **Critical Directories and Files**:

```
File  Edit  View  Search  Terminal  Help
-rw-r--r--  1 www-data www-data 19915 Dec 31  2023 license.txt
-rw-r--r--  1 www-data www-data  7409 Jun 18 07:59 readme.html
-rw-r--r--  1 www-data www-data  7387 Feb 13  2024 wp-activate.php
drwxr-xr-x  9 www-data www-data  4096 Sep 10 11:23 wp-admin
-rw-r--r--  1 www-data www-data   351 Feb  6  2020 wp-blog-header.php
-rw-r--r--  1 www-data www-data  2323 Jun 14  2023 wp-comments-post.php
-rw-------  1 www-data www-data  3017 Sep 30 12:02 wp-config.php
drwxr-xr-x  5 www-data www-data  4096 Oct  8 16:49 wp-content
-rw-r--r--  1 www-data www-data  5638 May 30  2023 wp-cron.php
drwxr-xr-x 30 www-data www-data 12288 Sep 10 11:23 wp-includes
-rw-r--r--  1 www-data www-data  2502 Nov 26  2022 wp-links-opml.php
-rw-r--r--  1 www-data www-data  3937 Mar 11  2024 wp-load.php
-rw-r--r--  1 www-data www-data 51238 May 28  2024 wp-login.php
-rw-r--r--  1 www-data www-data  8525 Sep 16  2023 wp-mail.php
-rw-r--r--  1 www-data www-data 28774 Jul  9 11:43 wp-settings.php
-rw-r--r--  1 www-data www-data 34385 Jun 19  2023 wp-signup.php
-rw-r--r--  1 www-data www-data  4885 Jun 22  2023 wp-trackback.php
-rw-r--r--  1 www-data www-data  3246 Mar  2  2024 xmlrpc.php
```

```
debian@debian:/var/log$ ls -l
total 1620
-rw-r--r--  1 root              root              8621 Nov 18 20:44 alternativ
es.log
-rw-r--r--  1 root              root             48068 Sep 30 12:14 alternativ
es.log.1
drwxr-x---  2 root              adm               4096 Nov 30 17:02 apache2
drwxr-xr-x  2 root              root              4096 Nov 25 20:56 apt
-rw-------  1 root              root                 0 Nov 20 19:22 boot.log
-rw-------  1 root              root              9794 Nov 20 19:22 boot.log.1
-rw-------  1 root              root              8902 Nov 18 17:59 boot.log.2
-rw-------  1 root              root             56408 Nov 15 18:14 boot.log.3
-rw-------  1 root              root             79043 Nov 13 18:09 boot.log.4
-rw-rw----  1 root              utmp                 0 Nov 13 18:09 btmp
-rw-rw----  1 root              utmp              2688 Oct  8 16:43 btmp.1
drwxr-xr-x  2 root              root              4096 Nov 13 20:38 chkrootkit
drwxr-xr-x  2 clamav            clamav            4096 Nov 30 17:02 clamav
drwxr-xr-x  2 root              root              4096 Nov 30 17:02 cups
-rw-r--r--  1 root              root            203577 Nov 25 20:56 dpkg.log
```

```
debian@debian:/var/www/html$ ls -l
total 244
-rwxrwxrwx  1 www-data www-data 10701 Sep 30 10:44 index.html
-rwxrwxrwx  1 www-data www-data   405 Feb  6  2020 index.php
-rwxrwxrwx  1 www-data www-data 19915 Dec 31  2023 license.txt
-rwxrwxrwx  1 www-data www-data  7409 Jun 18 07:59 readme.html
-rwxrwxrwx  1 www-data www-data  7387 Feb 13  2024 wp-activate.php
drwxrwxrwx  9 www-data www-data  4096 Sep 10 11:23 wp-admin
-rwxrwxrwx  1 www-data www-data   351 Feb  6  2020 wp-blog-header.php
-rwxrwxrwx  1 www-data www-data  2323 Jun 14  2023 wp-comments-post.php
-rwxrwxrwx  1 www-data www-data  3017 Sep 30 12:02 wp-config.php
drwxrwxrwx  5 www-data www-data  4096 Oct  8 16:49 wp-content
-rwxrwxrwx  1 www-data www-data  5638 May 30  2023 wp-cron.php
drwxrwxrwx 30 www-data www-data 12288 Sep 10 11:23 wp-includes
-rwxrwxrwx  1 www-data www-data  2502 Nov 26  2022 wp-links-opml.php
-rwxrwxrwx  1 www-data www-data  3937 Mar 11  2024 wp-load.php
-rwxrwxrwx  1 www-data www-data 51238 May 28  2024 wp-login.php
-rwxrwxrwx  1 www-data www-data  8525 Sep 16  2023 wp-mail.php
-rwxrwxrwx  1 www-data www-data 28774 Jul  9 11:43 wp-settings.php
```

- Permissions for critical directories like `/var`, `/etc`, and sensitive files like `wp-config.php` were reviewed.
- Changes made:
  - Files: `sudo find . -type f -exec chmod 644 {} \;`
  - Directories: `sudo find . -type d -exec chmod 755 {} \;`

2. Before these changes, permissions on several directories and files were too permissive, potentially allowing unauthorized access by malicious users. For example, some sensitive files, such as configurations and user data, had permissions enabling read, write, and even execute access by any system user.
   After the changes:
   - Sensitive files like configurations were set to 600, allowing only the owner to access them.
   - General files now have permissions set to 644, limiting write access to the owner while maintaining read access for authorized users.
   - Directories are configured with 755 to prevent unauthorized modifications.
3. These adjustments strengthen security by protecting critical files from unauthorized access and reducing the exposure of sensitive information.

4. **WordPress**:
    - Permissions for insecure files and directories were corrected.
        - `sudo chmod 600 wp-config.php`
        - Files: 644
        - Directories: 755

```
File  Edit  View  Search  Terminal  Help
-rw-r--r--  1 www-data www-data 19915 Dec 31  2023 license.txt
-rw-r--r--  1 www-data www-data  7409 Jun 18 07:59 readme.html
-rw-r--r--  1 www-data www-data  7387 Feb 13  2024 wp-activate.php
drwxr-xr-x  9 www-data www-data  4096 Sep 10 11:23 wp-admin
-rw-r--r--  1 www-data www-data   351 Feb  6  2020 wp-blog-header.php
-rw-r--r--  1 www-data www-data  2323 Jun 14  2023 wp-comments-post.php
-rw-------  1 www-data www-data  3017 Sep 30 12:02 wp-config.php
drwxr-xr-x  5 www-data www-data  4096 Oct  8 16:49 wp-content
-rw-r--r--  1 www-data www-data  5638 May 30  2023 wp-cron.php
drwxr-xr-x 30 www-data www-data 12288 Sep 10 11:23 wp-includes
-rw-r--r--  1 www-data www-data  2502 Nov 26  2022 wp-links-opml.php
-rw-r--r--  1 www-data www-data  3937 Mar 11  2024 wp-load.php
-rw-r--r--  1 www-data www-data 51238 May 28  2024 wp-login.php
-rw-r--r--  1 www-data www-data  8525 Sep 16  2023 wp-mail.php
-rw-r--r--  1 www-data www-data 28774 Jul  9 11:43 wp-settings.php
-rw-r--r--  1 www-data www-data 34385 Jun 19  2023 wp-signup.php
-rw-r--r--  1 www-data www-data  4885 Jun 22  2023 wp-trackback.php
-rw-r--r--  1 www-data www-data  3246 Mar  2  2024 xmlrpc.php
```

5. Insecure permissions in WordPress, such as 777 on sensitive files like `wp-config.php`, allowed any system user to read, write, or even execute these files. This created a significant risk of attackers accessing critical information, such as database credentials.
   To mitigate these risks, stricter permissions were applied:
    - `wp-config.php`: Permissions adjusted to 600, ensuring only the owner can read and write the file.
    - General files: Permissions set to 644, allowing read access for others but limiting write access to the owner.
    - Directories: Configured to 755 to restrict modification capabilities to unauthorized users.
6. These modifications enhance security by limiting unauthorized access opportunities to essential WordPress files.

Directory Listing Prevention

- **Detected Issue**: Files and directories were listable in Apache.
- **Solution Applied**: The Apache configuration file was edited to disable directory listing.
    - Modified `Options Indexes FollowSymlinks` to `Options FollowSymlinks`.

apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

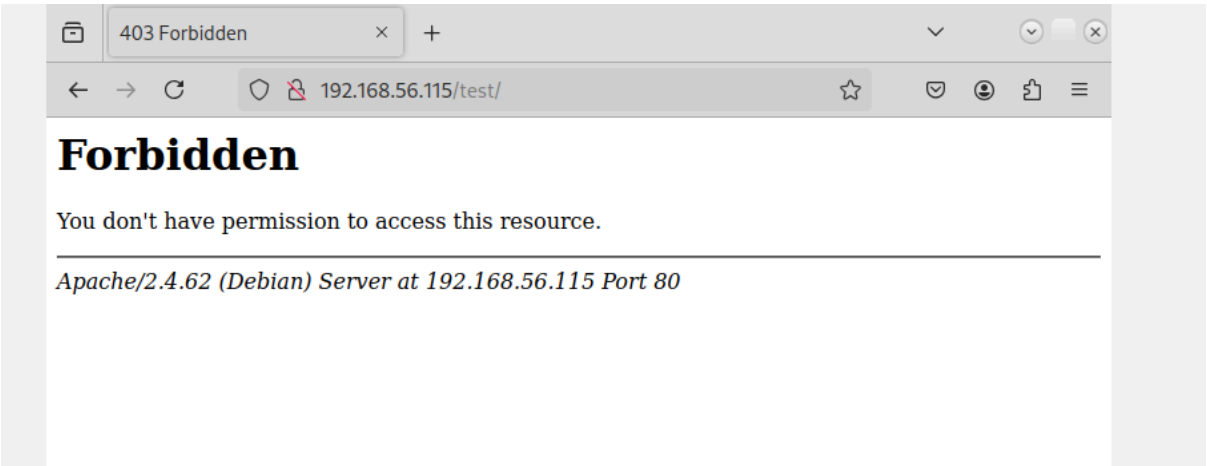
```
debian@debian:~$ curl -I http://192.168.56.115/
HTTP/1.1 200 OK
Date: Sun, 01 Dec 2024 00:31:13 GMT
Server: Apache/2.4.62 (Debian)
Last-Modified: Mon, 30 Sep 2024 14:44:22 GMT
ETag: "29cd-623573d915b52"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Content-Type: text/html

debian@debian:~$
```

```
  GNU nano 7.2                /etc/apache2/apache2.conf *
        Options FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>

<Directory /usr/share>
        AllowOverride None
        Options FollowSymLinks
        Require all granted
</Directory>

<Directory /var/www/>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
```



- To verify that directory listing was no longer available, tests were conducted by accessing various URLs on the web server that previously displayed listable content. These tests were performed using a web browser and tools like `curl`. The results confirmed that, after the changes, the server returns a 403 Forbidden error instead of displaying directory content. This validates that the current Apache configuration protects against accidental exposure of files and directories.

Conclusion

The corrective measures detailed in this phase have significantly strengthened the security of our system. From reconfiguring essential services like SSH and MySQL to correcting permissions and mitigating critical vulnerabilities in Apache and WordPress, concrete steps have been taken to reduce exploitation risks. Furthermore, focusing on password hardening and directory listing prevention ensures that detected breaches are no longer exploitable. This effort not only guarantees the current security of the system but also lays the groundwork for more proactive and resilient management against future threats. Regular audits and continuous monitoring will remain key components for maintaining a secure environment.