

PUCPR - Pontifical Catholic University of Paraná

## **Blue Team**

Implementation of Network Security Infrastructure

Bernardo Walker Leichtweis  
Cassio Viecei Filho  
Enzzo Machado Silvino

Curitiba - PR

May 2025

# Contents

<b>Executive Summary</b> . . . . .	<b>3</b>
Firewall Layer . . . . .	3
DMZ - Demilitarized Zone . . . . .	3
Internal Network . . . . .	3
Conclusion . . . . .	3
<b>Network Topology</b> . . . . .	<b>4</b>
Machine Descriptions . . . . .	4
<b>Apache Web Server</b> . . . . .	<b>5</b>
Security Directives . . . . .	5
Directory Configurations . . . . .	5
DoS Mitigation . . . . .	6
Security Configuration . . . . .	6
HTTP Virtual Host . . . . .	6
HTTPS Virtual Host . . . . .	7
Permissions . . . . .	7
<b>Web Application</b> . . . . .	<b>8</b>
General Description . . . . .	8
Command Execution Security . . . . .	10
Secure Password Storage . . . . .	10
Secure SQL with Prepared Statements . . . . .	10
Input Sanitization . . . . .	11
Favorites Functionality . . . . .	11
<b>Mailcow Email Service (Postfix + Dovecot)</b> . . . . .	<b>12</b>
Installation and Configuration . . . . .	12
Security and Accounts . . . . .	12
Email Service Operation . . . . .	12
<b>MySQL Server Database</b> . . . . .	<b>14</b>
Initial Configuration . . . . .	14
Database and Table Creation . . . . .	14
User with Restricted Permissions . . . . .	15
<b>SAMBA (SMB) File Sharing</b> . . . . .	<b>16</b>
Sharing Configuration . . . . .	16
Auditing and Logging . . . . .	16
Space Control with Quotas . . . . .	16
Sharing and Access Tests . . . . .	17
<b>Backup with Restic</b> . . . . .	<b>18</b>
Restic Advantages . . . . .	18
User and Permissions . . . . .	18
Backup Script . . . . .	18
Systemd Integration . . . . .	19

Important Paths . . . . .	19
Test User (Snapshot Reading) . . . . .	20
<b>Secure Remote Access via SSH . . . . .</b>	<b>21</b>
SSH Server Configuration . . . . .	21
Public Key Installation . . . . .	21
Bypass for Test User . . . . .	21
Connectivity Tests . . . . .	21
<b>Snort (IDS/IPS) . . . . .</b>	<b>24</b>
Rule Sources . . . . .	24
Manual Rule Update . . . . .	24
Pass Lists (Exclusion List) . . . . .	24
Monitored Interfaces . . . . .	24
Rule Policy . . . . .	25
Preprocessors . . . . .	25
Detection Tests . . . . .	25
<b>Light LDAP (LLDAP) . . . . .</b>	<b>27</b>
Configuration via Docker Compose . . . . .	27
Web Administrative Interface . . . . .	27
LDAPSEARCH Query . . . . .	28
<b>Wazuh (XDR + SIEM) . . . . .</b>	<b>30</b>
Wazuh Manager Installation . . . . .	30
Wazuh Agent Installation . . . . .	30
Wazuh Agent Configuration . . . . .	30
Audit Rules for Privileged Commands . . . . .	30
<b>YARA Integration with Wazuh (DLP) . . . . .</b>	<b>33</b>
YARA Rules . . . . .	33
Scanning Script and Scheduling . . . . .	33
Integration with Wazuh Agent . . . . .	34
<b>HashiCorp Vault (PAM) . . . . .</b>	<b>35</b>
Workflow . . . . .	35
Vault Configuration . . . . .	35
Vault Agent Configuration . . . . .	36
Apache/PHP Integration . . . . .	37
<b>Firewall Rules . . . . .</b>	<b>40</b>
Blocking Internal Network to External Machine . . . . .	40
DMZ to Internal Network Access Rules . . . . .	40
External Machine Restrictions . . . . .	40

## Executive Summary

This project proposes the implementation of a network security infrastructure structured in three main layers: **firewall**, **DMZ (demilitarized zone)**, and **internal network**. The goal is to ensure the protection of the organization's information based on the pillars of information security - *confidentiality, integrity, availability, and non-repudiation*.

### Firewall Layer

The **firewall**, based on *pfSense*, will be responsible for filtering and controlling all incoming and outgoing network traffic. For secure remote connections, *WireGuard*, a modern VPN, will be used. Real-time traffic monitoring will be performed with *Snort*, acting as an IDS/IPS to detect and mitigate intrusion attempts and denial-of-service attacks (*DoS/DDoS*).

### DMZ - Demilitarized Zone

The **DMZ** will host services aimed at the external public, logically isolated from the internal network to contain potential attack vectors. A web application and a complete email server with *MailCow* will be deployed, integrating SMTP, IMAP, antispam, and antivirus functions, providing secure and auditable communication. This separation ensures that, even in the event of a compromised exposed service, internal assets remain protected.

### Internal Network

The **internal network** will house the most critical services, such as:

- **MySQL**: Database storing critical web application information, with restricted access and encryption at rest for data protection.
- **SAMBA**: Network file-sharing system with user-based permission control and access auditing to prevent leaks.
- **Wazuh**: Centralized security monitoring platform, acting as SIEM, XDR, and DLP, using YARA rules for advanced threat detection.
- **Light LDAP and HashiCorp Vault**: Identity and access management. LDAP authenticates users, while Vault stores and automatically rotates passwords.
- **Restic**: Automated and encrypted backup tool with periodic scheduling, ensuring information availability and integrity in case of failures or incidents.

### Conclusion

This architecture ensures a secure, organized, and resilient foundation for operations, reducing risks and ensuring service continuity in compliance with information security best practices.

## Network Topology

The network is structured with the firewall as the central point, connecting the internal network, the DMZ, and external interfaces. The topology is illustrated below.

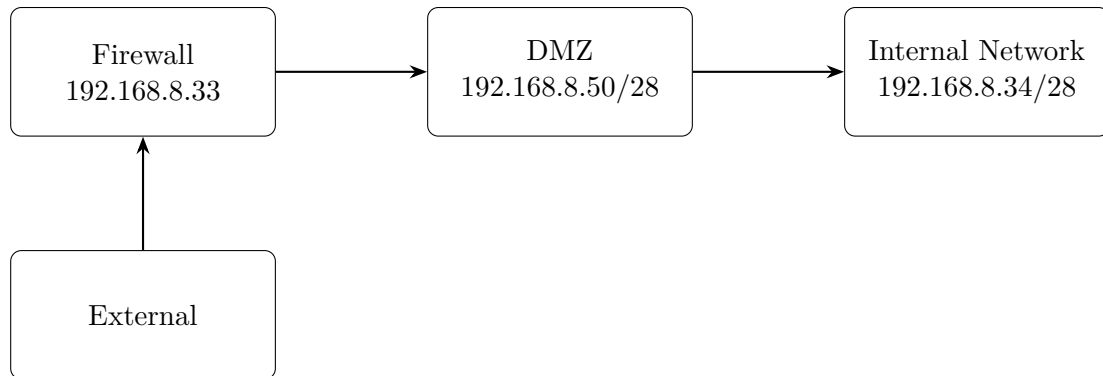


Figure 1: Network Topology Diagram

## Machine Descriptions

- **Firewall (pfSense):**
  - Address: 192.168.8.33
  - Hostname: ENE48-EP-PUCPR.home.arpa
  - Operating System: FreeBSD 14.0-CURRENT
  - pfSense Version: 2.7.2-RELEASE
- **DMZ:**
  - Address: 192.168.8.50/28
  - Hostname: ene50-ep-pucpr
  - Default Gateway: 192.168.8.49 via eth0
  - Distribution: Ubuntu 24.04.2 LTS (Noble)
  - Kernel: Linux 6.8.0-57-generic
- **Internal Network:**
  - Address: 192.168.8.34/28
  - Hostname: ene49-ep-pucpr
  - Default Gateway: 192.168.8.33 via eth0
  - Distribution: Ubuntu 24.04.2 LTS (Noble)
  - Kernel: Linux 6.8.0-57-generic

## Apache Web Server

The **Apache** web server is deployed in the DMZ to host the *sitefoda.com* website. This configuration ensures secure access via HTTP and HTTPS, automatically redirecting HTTP traffic to HTTPS. Security headers and directory restrictions have been implemented to protect against common web vulnerabilities. A self-signed SSL certificate and specific permissions are used to enhance security.

### Security Directives

The following directives are added to `apache2.conf` to minimize information disclosure and disable unnecessary features:

```
1 ServerTokens Prod
2 ServerSignature off
3 FileETag None
4 TraceEnable off
```

Listing 1: Security Directives in `apache2.conf`

- `ServerTokens Prod`: Limits server information in HTTP headers to “Apache”.
- `ServerSignature off`: Disables server signature on error pages.
- `FileETag None`: Removes ETag headers to prevent inode-based attacks.
- `TraceEnable off`: Disables the TRACE method to prevent cross-site tracing.

### Directory Configurations

Directory configurations restrict access and disable unnecessary features:

```
1 <Directory />
2 Options FollowSymLinks
3 AllowOverride None
4 Require all denied
5 </Directory>
6
7 <Directory /usr/share>
8 AllowOverride None
9 Require all granted
10 </Directory>
11
12 <Directory /var/www/>
13 Options -Indexes
14 AllowOverride None
15 Require all granted
16 <LimitExcept GET POST HEAD>
17 Require all denied
18 </LimitExcept>
19 </Directory>
```

Listing 2: Directory Configurations in `apache2.conf`

- Root directory (/): Denies all access except for symbolic links.
- /usr/share: Grants access to shared resources.
- /var/www/: Allows GET, POST, and HEAD methods, disables directory listing, and denies other HTTP methods.

## DoS Mitigation

To reduce the risk of denial-of-service (DoS) attacks, the timeout is reduced:

```
1 Timeout 60
```

Listing 3: Timeout Configuration

This sets the server timeout to 60 seconds, reducing resource exhaustion risks.

## Security Configuration

The file /etc/apache2/conf-enabled/security.conf contains additional security headers and rules:

```
1 RedirectMatch 404 /.git
2 Header set X-Content-Type-Options: "nosniff"
3 Header set X-XSS-Protection "1; mode=block"
4 Header always set Referrer-Policy "strict-origin"
5 Header always append X-Frame-Options SAMEORIGIN
```

Listing 4: Security Configuration in security.conf

- RedirectMatch 404 /.git: Prevents access to .git directories.
- X-Content-Type-Options: "nosniff": Prevents MIME type sniffing.
- X-XSS-Protection: Enables XSS filtering in browsers.
- Referrer-Policy: "strict-origin": Limits referrer information to the origin.
- X-Frame-Options SAMEORIGIN: Prevents clickjacking by restricting framing.

## HTTP Virtual Host

The file /etc/apache2/sites-available/000-default.conf redirects all HTTP traffic to HTTPS:

```
1 <VirtualHost *:80>
2 ServerName www.sitefoda.com
3 ServerAlias sitefoda.com
4 Redirect permanent / https://sitefoda.com
5 ServerAdmin webmaster@localhost
6 DocumentRoot /var/www/html
7 ErrorLog ${APACHE_LOG_DIR}/error.log
8 CustomLog ${APACHE_LOG_DIR}/access.log combined
9 </VirtualHost>
```

Listing 5: HTTP Virtual Host Configuration

This ensures all traffic is encrypted, redirecting to <https://sitefoda.com>.

## HTTPS Virtual Host

The file `/etc/apache2/sites-available/ssl.conf` configures the HTTPS virtual host with SSL:

```
1 <VirtualHost *:443>
2   ServerName www.sitefoda.com
3   ServerAlias sitefoda.com
4   DocumentRoot /var/www/html
5   ErrorLog ${APACHE_LOG_DIR}/error.log
6   CustomLog ${APACHE_LOG_DIR}/access.log combined
7   ServerAdmin webmaster@localhost
8
9   <FilesMatch "^(?:cgi|shtml|phtml|php)$">
10    SSLOptions +StdEnvVars
11  </FilesMatch>
12  <Directory /usr/lib/cgi-bin>
13    SSLOptions +StdEnvVars
14  </Directory>
15
16  SSLEngine on
17  SSLCertificateFile /etc/ssl/certs/apache.crt
18  SSLCertificateKeyFile /etc/ssl/private/apache.key
19 </VirtualHost>
```

Listing 6: HTTPS Virtual Host Configuration

- `SSLEngine on`: Enables SSL/TLS.
- `SSLCertificateFile` and `SSLCertificateKeyFile`: Specify the self-signed certificate and key.

## Permissions

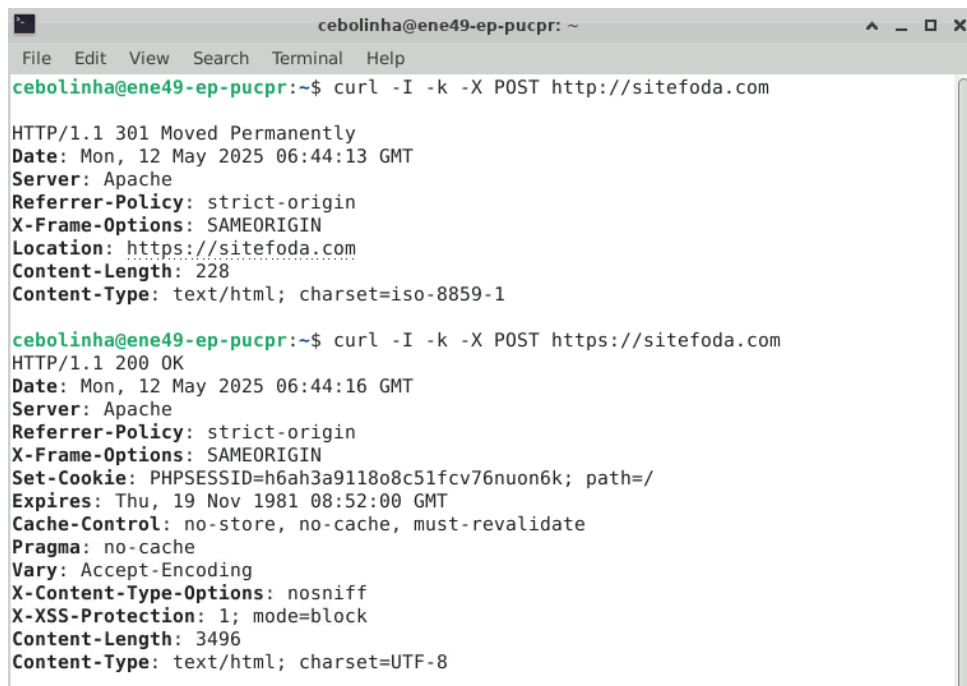
File and directory permissions are set to enhance security:

```
1 sudo chmod 750 /etc/apache2/conf*
2 sudo chown -R www-data:www-data /var/www
```

Listing 7: Permission Commands

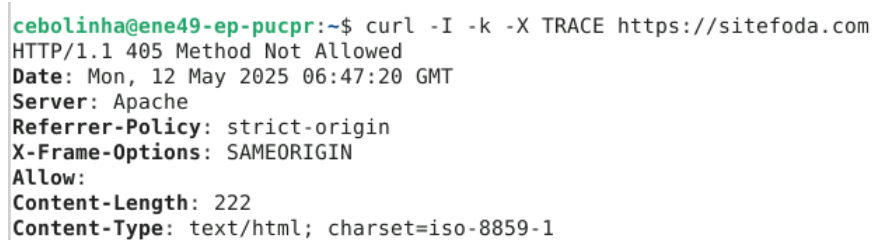
- `chmod 750 /etc/apache2/conf*`: Restricts configuration files to the owner and group.
- `chown -R www-data:www-data /var/www`: Sets the web server user as the owner of the web root.





```
cebolinha@ene49-ep-pucpr: ~  
File Edit View Search Terminal Help  
cebolinha@ene49-ep-pucpr:~$ curl -I -k -X POST http://sitefoda.com  
HTTP/1.1 301 Moved Permanently  
Date: Mon, 12 May 2025 06:44:13 GMT  
Server: Apache  
Referrer-Policy: strict-origin  
X-Frame-Options: SAMEORIGIN  
Location: https://sitefoda.com  
Content-Length: 228  
Content-Type: text/html; charset=iso-8859-1  
  
cebolinha@ene49-ep-pucpr:~$ curl -I -k -X POST https://sitefoda.com  
HTTP/1.1 200 OK  
Date: Mon, 12 May 2025 06:44:16 GMT  
Server: Apache  
Referrer-Policy: strict-origin  
X-Frame-Options: SAMEORIGIN  
Set-Cookie: PHPSESSID=h6ah3a9118o8c51fcv76nuon6k; path=/  
Expires: Thu, 19 Nov 1981 08:52:00 GMT  
Cache-Control: no-store, no-cache, must-revalidate  
Pragma: no-cache  
Vary: Accept-Encoding  
X-Content-Type-Options: nosniff  
X-XSS-Protection: 1; mode=block  
Content-Length: 3496  
Content-Type: text/html; charset=UTF-8
```

Figure 2: HTTP to HTTPS Redirect



```
cebolinha@ene49-ep-pucpr:~$ curl -I -k -X TRACE https://sitefoda.com  
HTTP/1.1 405 Method Not Allowed  
Date: Mon, 12 May 2025 06:47:20 GMT  
Server: Apache  
Referrer-Policy: strict-origin  
X-Frame-Options: SAMEORIGIN  
Allow:  
Content-Length: 222  
Content-Type: text/html; charset=iso-8859-1
```

Figure 3: TRACE Method Not Allowed

## Web Application

The web application was developed with a focus on simplicity, security, and user experience. We used **HTML**, **Tailwind CSS**, **JavaScript**, and **PHP** for its construction. The application is hosted in the DMZ, adhering to the principle of layer separation and protecting sensitive data in the internal network.

### General Description

The homepage displays *quotes* (motivational or inspirational phrases) for users. The quotes are dynamically generated using the system command **fortune**, executed securely on the server.

The system consists of four main pages:

- **Home:** Displays a randomly generated quote.
- **Register:** New user registration page.
- **Login:** Authentication page.

- **Profile:** User profile page, where users can save their favorite quotes.

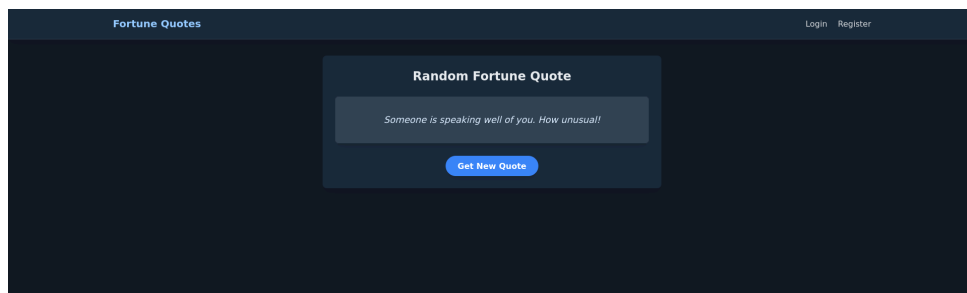


Figure 4: Home Page

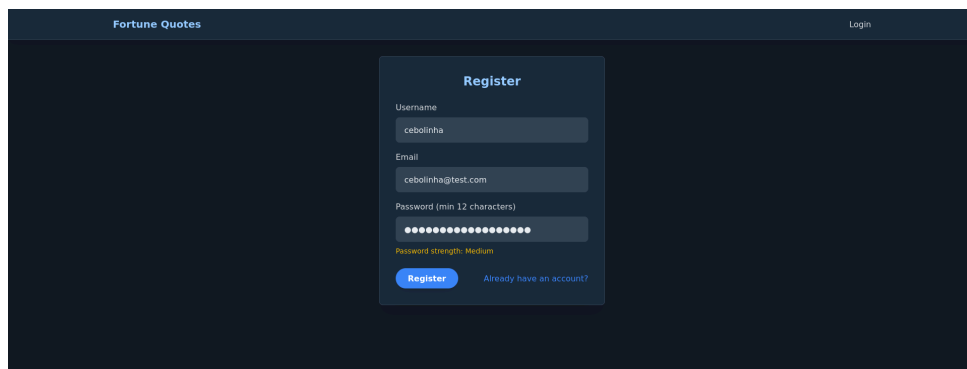


Figure 5: Register Page

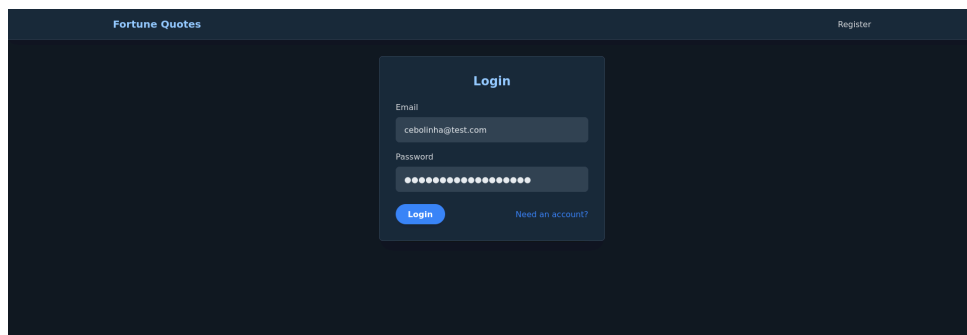


Figure 6: Login Page

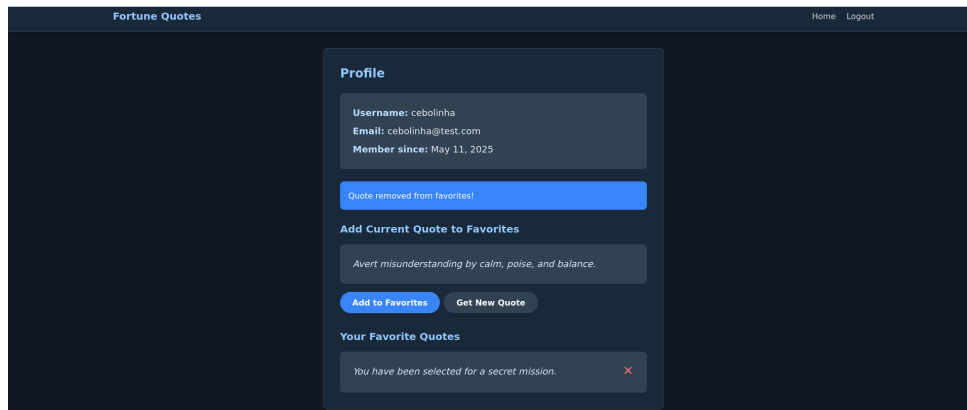


Figure 7: Profile Page

## Command Execution Security

Quotes are generated with `shell_exec` and the `fortune` command, but with rigorous sanitization to prevent arbitrary execution:

```
1 while(empty($fortune) || strlen($fortune) >= 700) {
2     if (is_executable($fortune_path)) {
3         $fortune = shell_exec(escapeshellcmd($fortune_path));
4     }
5
6     if(empty($fortune) || strlen($fortune) >= 700) {
7         $fortune = '';
8     }
9 }
10
11 if (empty($fortune)) {
12     $default_quotes = [...];
13     $fortune = $default_quotes[array_rand($default_quotes)];
14 }
15
16 return htmlspecialchars(trim($fortune), ENT_QUOTES |
    ENT_SUBSTITUTE, 'utf-8');
```

Listing 8: Sanitization of shell\_exec Command

## Secure Password Storage

During user registration, passwords are encrypted using `bcrypt` via PHP's native function:

```
1 $hashed_password = password_hash($password, PASSWORD_DEFAULT);
```

Listing 9: Password Hashing with bcrypt

## Secure SQL with Prepared Statements

All database interactions use `prepared statements` to prevent SQL injection:

```
1 $stmt = $db->prepare("INSERT INTO users (username, email,  
    password) VALUES (?, ?, ?)");  
2 $stmt->bind_param("sss", $username, $email, $hashed_password);
```

Listing 10: Insertion with Prepared Statement

## Input Sanitization

Data received via forms is sanitized using appropriate filters:

```
1 $username = filter_input(INPUT_POST, 'username',  
    FILTER_SANITIZE_SPECIAL_CHARS);  
2 $email = filter_input(INPUT_POST, 'email', FILTER_SANITIZE_EMAIL)  
    ;
```

Listing 11: Input Sanitization

## Favorites Functionality

On the profile page, users can save their favorite quotes. These actions are also protected by prepared statements and session validations.

## Mailcow Email Service (Postfix + Dovecot)

To provide a complete, secure, and auditable email service, we opted for the deployment of **Mailcow**, which integrates *Postfix* as the MTA (Mail Transfer Agent) and *Dovecot* as the MDA (Mail Delivery Agent), along with antispam, antivirus, webmail, and administrative interface modules.

### Installation and Configuration

Mailcow installation is performed via **Docker** using the automated `generate_config.sh` script provided by the official repository. This script generates the necessary configuration files based on the domain and administrator preferences.

By default, Mailcow uses the ports:

- 80 (HTTP)
- 443 (HTTPS)

To avoid conflicts with the Apache web server already present in the DMZ, these ports were changed in the `mailcow.conf` file:

```
1 HTTP_PORT=8080
2 HTTPS_PORT=8443
```

Listing 12: Port Changes in mailcow.conf

### Security and Accounts

After installation, the default administrator account password was changed to a secure password, and a new test user account was created.

We adopted a strong password policy with the following criteria:

- Minimum of 12 characters
- Must contain letters and numbers

This policy reduces the risk of unauthorized access, even if credentials are exposed.

### Email Service Operation

Mailcow offers a complete and responsive administrative interface, facilitating the management of domains, mailboxes, aliases, distribution lists, and logs. Communication between servers and clients is protected by SSL/TLS, ensuring the confidentiality and integrity of transmitted data.

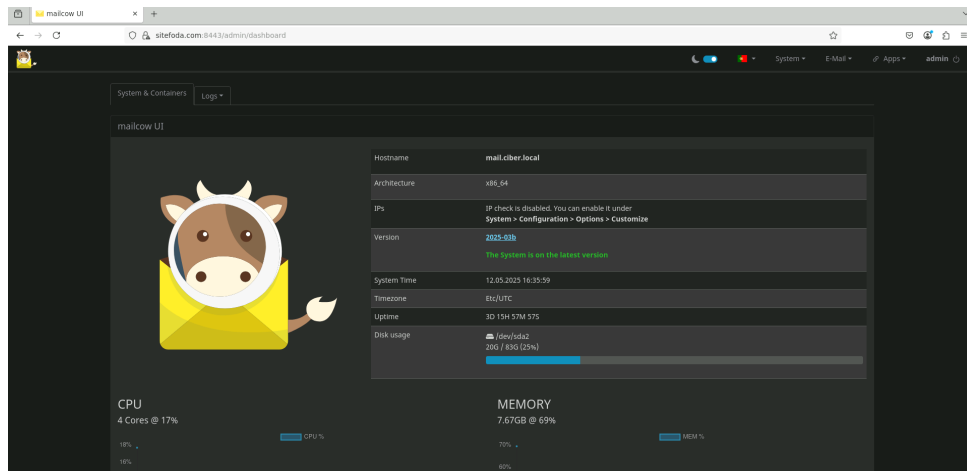


Figure 8: Mailcow Administrative Interface Dashboard

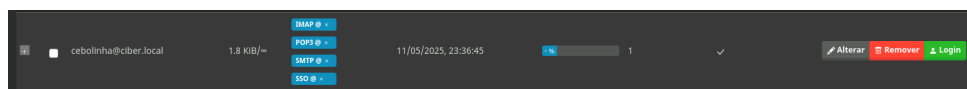


Figure 9: Test User Account Created Successfully

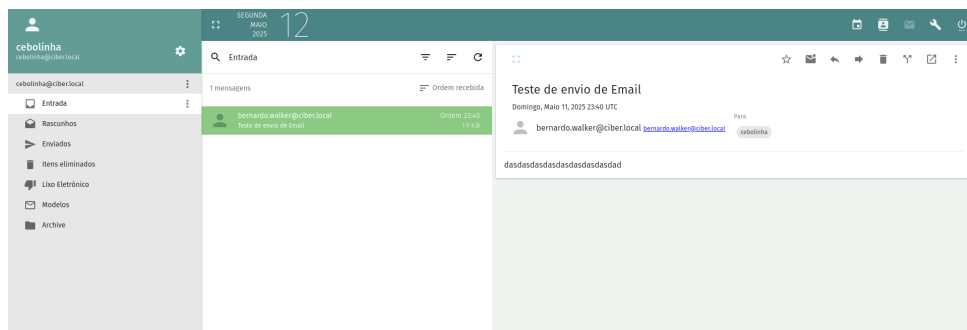


Figure 10: Email Sending and Receiving Tested Successfully

## MySQL Server Database

The **MySQL Server** database is responsible for storing sensitive web application information, including user credentials and favorite quotes. It is located in the internal network for enhanced security, accessible remotely only by the application in the DMZ.

### Initial Configuration

The first step was to change the administrator (**root**) password and create a new user specifically for the application to access the database remotely:

```
1 CREATE USER 'remote_user'@'192.168.8.50' IDENTIFIED BY '
   securePassword123';
2 GRANT ALL PRIVILEGES ON *.* TO 'remote_user'@'192.168.8.50';
3 FLUSH PRIVILEGES;
```

Listing 13: Creation of remote user with privileges

This user is permitted access only from the DMZ machine's IP (192.168.8.50), ensuring logical segmentation between network zones.

### Database and Table Creation

The **fortune\_quotes** database was created, containing two tables: one for storing user data and another for recording favorite quotes.

```
1 CREATE DATABASE IF NOT EXISTS fortune_quotes;
2 USE fortune_quotes;
3
4 -- Users table
5 CREATE TABLE IF NOT EXISTS users (
6     id INT AUTO_INCREMENT PRIMARY KEY,
7     username VARCHAR(50) NOT NULL,
8     email VARCHAR(100) NOT NULL UNIQUE,
9     password VARCHAR(255) NOT NULL,
10    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
11    INDEX (email)
12 );
13
14 -- Favorite quotes table
15 CREATE TABLE IF NOT EXISTS favorite_quotes (
16     id INT AUTO_INCREMENT PRIMARY KEY,
17     user_id INT NOT NULL,
18     quote TEXT NOT NULL,
19     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
20     FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
21     INDEX (user_id)
22 );
```

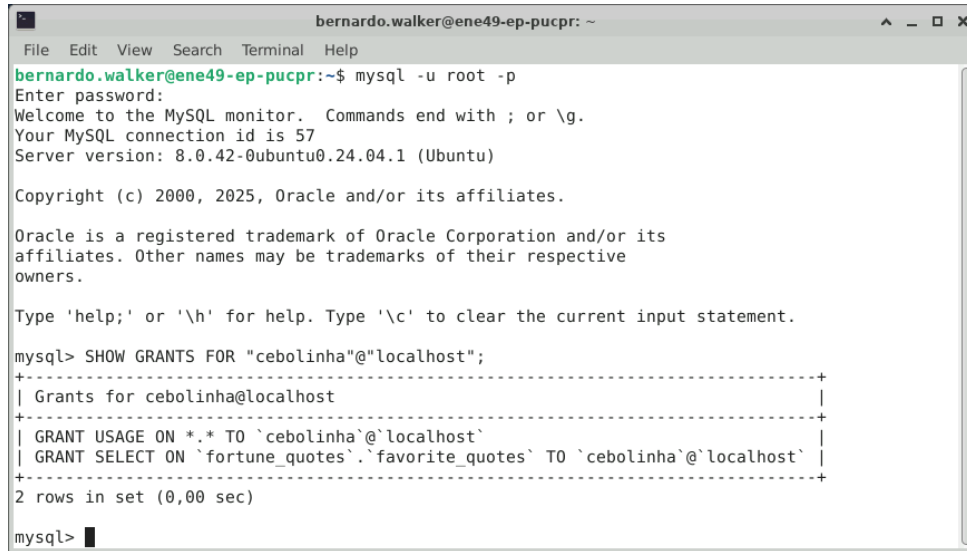
Listing 14: Database and Table Creation

The **favorite\_quotes** table has a foreign key linked to the **users** table, ensuring referential integrity and allowing automatic deletion of quotes if a user is removed.

## User with Restricted Permissions

For testing and simulations, an additional user with minimal permissions was created. This user can only perform read operations (SELECT) on the favorite quotes table.

This practice follows the principle of least privilege, essential for database security.



```

bernardo.walker@ene49-ep-pucpr: ~
File Edit View Search Terminal Help
bernardo.walker@ene49-ep-pucpr:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 57
Server version: 8.0.42-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

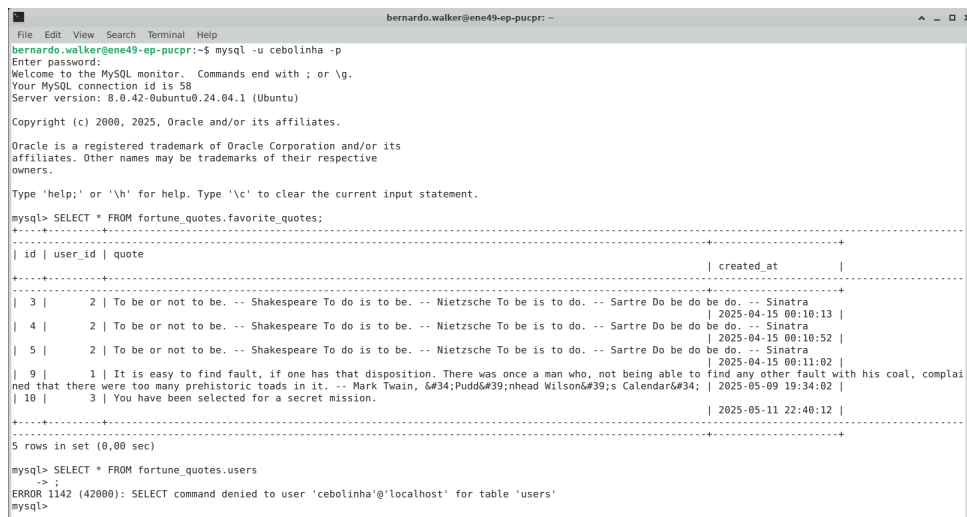
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW GRANTS FOR "cebolinha"@"localhost";
+-----+
| Grants for cebolinha@localhost                                     |
+-----+
| GRANT USAGE ON *.* TO `cebolinha`@`localhost`                    |
| GRANT SELECT ON `fortune_quotes`.`favorite_quotes` TO `cebolinha`@`localhost` |
+-----+
2 rows in set (0,00 sec)

mysql>

```

Figure 11: Test User Permissions with SHOW GRANTS



```

bernardo.walker@ene49-ep-pucpr: ~
File Edit View Search Terminal Help
bernardo.walker@ene49-ep-pucpr:~$ mysql -u cebolinha -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 58
Server version: 8.0.42-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT * FROM fortune_quotes.favorite_quotes;
+-----+
| id | user_id | quote                                                                                               | created_at |
+-----+
| 3 | 2 | To be or not to be. -- Shakespeare To do is to be. -- Nietzsche To be is to do. -- Sartre Do be do be do. -- Sinatra | 2025-04-15 00:10:13 |
| 4 | 2 | To be or not to be. -- Shakespeare To do is to be. -- Nietzsche To be is to do. -- Sartre Do be do be do. -- Sinatra | 2025-04-15 00:10:52 |
| 5 | 2 | To be or not to be. -- Shakespeare To do is to be. -- Nietzsche To be is to do. -- Sartre Do be do be do. -- Sinatra | 2025-04-15 00:11:02 |
| 9 | 1 | It is easy to find fault, if one has that disposition. There was once a man who, not being able to find any other fault with his coal, complained that there were too many prehistoric toads in it. -- Mark Twain, 6#34;Pudd#39;nhead Wilson#39;s Calendar#34; | 2025-05-09 19:34:02 |
| 10 | 3 | You have been selected for a secret mission. | 2025-05-11 22:40:12 |
+-----+
5 rows in set (0,00 sec)

mysql> SELECT * FROM fortune_quotes.users
->
ERROR 1142 (42000): SELECT command denied to user 'cebolinha'@'localhost' for table 'users'
mysql>

```

Figure 12: Query to favorite\_quotes Table with Test User



## SAMBA (SMB) File Sharing

The file-sharing service was implemented using **Samba** with the **SMB3** protocol, which offers native support for encryption of all connections. The following directives were configured in the `smb.conf` file to enhance security:

```
1 smb min protocol = 3
2 smb encrypt = required
```

Listing 15: SMB Protocol Security Configuration

These directives ensure that only SMB3 connections are accepted and that encryption is mandatory for all accesses to the Samba server.

## Sharing Configuration

The sharing was defined as follows:

```
1 [InternalFiles]
2 path = /srv/internal_files
3 browseable = yes
4 writable = yes
5 valid users = cebolinha bernardo.walker enzzo.silvino filho.
   cassio
6 create mask = 0660
7 directory mask = 0770
```

Listing 16: Sharing Block in smb.conf

The `/srv/internal_files` directory was created with restrictive permissions:

- **Permissions:** `drwxrwx---` (770)
- **Purpose:** Ensure that only authenticated users have full access.

## Auditing and Logging

The log level was adjusted to capture useful access information:

```
1 log level = 2
```

Listing 17: Log Configuration for Auditing

This allows logging of events such as connections, errors, and file accesses, aiding future audits.

## Space Control with Quotas

To prevent excessive disk usage by users, quotas were implemented per group in the filesystem. A group called `smbquota` was created, and authorized users were added:

```
1 sudo usermod -aG smbquota filho.cassio
2 sudo usermod -aG smbquota bernardo.walker
3 sudo usermod -aG smbquota enzzo.silvino
```

```
4 sudo usermod -aG smbquota cebolinha
```

Listing 18: Adding users to smbquota group

The quota was applied as follows:

```
1 sudo setquota -g smbquota 921600 1048576 0 0 /
```

Listing 19: Applying quota to smbquota group

### Defined Limits:

Limit Type	Value	Meaning
Soft Limit	900 MiB	Warning after reaching (7 days tolerance)
Hard Limit	1 GiB	Prevents writing above the limit
File Limit	Unlimited	No restriction on file count

Table 1: Quota Policy for the `smbquota` Group

## Sharing and Access Tests

Practical tests of access and file movement were conducted to ensure the correct functioning of the sharing.

```
bernardo.walker@ene49-ep-pucpr: ~
File Edit View Search Terminal Help
bernardo.walker@ene49-ep-pucpr:~$ touch arquivo_teste.txt
bernardo.walker@ene49-ep-pucpr:~$ mv arquivo_teste.txt /srv/arquivos_internos/piupiu/
bernardo.walker@ene49-ep-pucpr:~$
```

Figure 13: Creation and Movement of File to Shared Folder

```
bernardo.walker@ene49-ep-pucpr: ~
File Edit View Search Terminal Help
bernardo.walker@ene49-ep-pucpr:~$ smbclient //192.168.8.34/ArquivosInternos -m SMB3 -U cebolinha
Password for [WORKGROUP\cebolinha]:
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0 Mon Apr 14 17:46:16 2025
..               D          0 Mon Apr 14 17:46:16 2025
piupiu           D          0 Mon May 12 17:21:52 2025
80901592 blocks of size 1024. 45220212 blocks available
smb: \> cd piupiu\
smb: \piupiu\> ls
.                D          0 Mon May 12 17:21:52 2025
..               D          0 Mon Apr 14 17:46:16 2025
arquivo_teste.txt N          0 Mon May 12 17:21:45 2025
80901592 blocks of size 1024. 45220036 blocks available
smb: \piupiu\>
```

Figure 14: Access to Sharing via smbclient with Test User

## Backup with Restic

The backup system was implemented using **Restic**, a modern tool that ensures security, performance, and simplicity. This mechanism performs incremental backups with native encryption and easy integration with **systemd**.

### Restic Advantages

- **Security:** All data is encrypted with **AES-256** and authenticated with **HMAC**, ensuring protection against reading and tampering.
- **Performance:** Incremental backups avoid redundancy, saving time and space.
- **Simplicity and Portability:** Works without a daemon, ideal for scheduling via **systemd** or custom scripts.

### User and Permissions

We created the **backupuser** user, exclusively for running the backup process, with secure characteristics:

- System user type (UID below 1000)
- No login shell (**/usr/sbin/nologin**)
- No home directory
- No access password
- Group: **nogroup**

Permissions for the repository and sensitive files:

```
1 sudo chown -R backupuser:nogroup /backups/restic_repo
2 sudo chown backupuser:nogroup /etc/restic/pass
3 sudo chmod 600 /etc/restic/pass
```

Listing 20: Repository and Password Permissions

### Backup Script

The main script was moved to a secure location:

```
1 sudo mkdir -p /opt/restic
2 sudo mv /usr/local/sbin/restic_backup.sh /opt/restic/
   restic_backup.sh
3 sudo chown backupuser:nogroup /opt/restic/restic_backup.sh
4 sudo chmod 750 /opt/restic/restic_backup.sh
```

Listing 21: Backup Script Organization

### Script Functions:

- Checks if repository disk usage exceeds 50%. If so, the backup is aborted.

- Performs incremental backup of the /opt folder.
- Executes intelligent retention: only the last two snapshots are kept with `restic forget --keep-last 2 --prune`.
- All execution is logged in `/var/log/restic/backup.log`.

## Systemd Integration

**Service:** `/etc/systemd/system/restic-backup.service`

```
1 [Unit]
2 Description=Daily Restic Backup (Incremental, retention 2)
3
4 [Service]
5 Type=oneshot
6 User=backupuser
7 Group=nogroup
8 ExecStart=/opt/restic/restic_backup.sh
9 Nice=10
10 IOSchedulingClass=best-effort
11 IOSchedulingPriority=7
```

Listing 22: Restic systemd Service

**Timer:** `/etc/systemd/system/restic-backup.timer`

```
1 [Unit]
2 Description=Daily Restic Backup Scheduler
3
4 [Timer]
5 OnCalendar=*-*-* 04:30
6 Persistent=true
7
8 [Install]
9 WantedBy=timers.target
```

Listing 23: Restic systemd Timer

## Important Paths

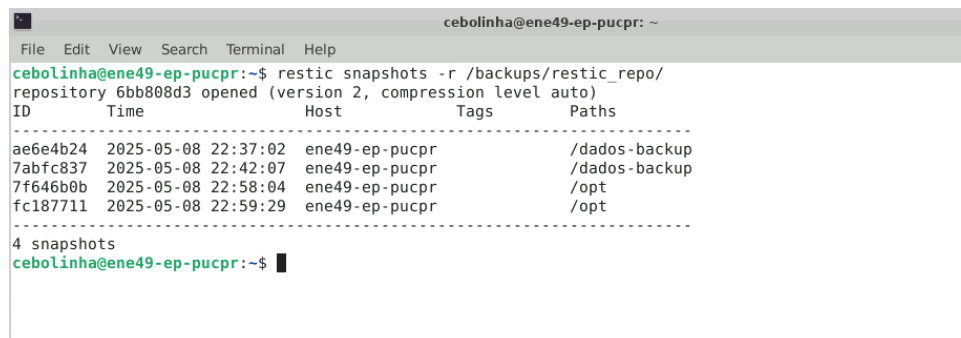
- `/backups/resticrepo` – Backup repository
- `/etc/restic/pass` – Repository password
- `/opt/restic/resticbackup.sh` – Execution script
- `/etc/systemd/system/restic-backup.service` – Systemd service
- `/etc/systemd/system/restic-backup.timer` – Systemd timer
- `/var/log/restic/backup.log` – Detailed execution log

## Test User (Snapshot Reading)

A configuration was created so that the test user can only view snapshots, without write permissions:

```
1 sudo chmod -R 550 /backups/restic_repo/  
2 sudo chown -R root:cebolinha /backups/restic_repo/
```

Listing 24: Read Permission for Test User



A terminal window titled 'cebolinha@ene49-ep-pucpr: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The user 'cebolinha@ene49-ep-pucpr' runs the command 'restic snapshots -r /backups/restic\_repo/'. The output shows the repository '6bb808d3 opened (version 2, compression level auto)' and a table of snapshots. The table has columns: ID, Time, Host, Tags, and Paths. It lists 4 snapshots. The user then presses Enter, and the prompt returns.

```
cebolinha@ene49-ep-pucpr:~$ restic snapshots -r /backups/restic_repo/  
repository 6bb808d3 opened (version 2, compression level auto)  
ID          Time                Host                Tags                Paths  
-----  
ae6e4b24    2025-05-08 22:37:02    ene49-ep-pucpr      /dados-backup  
7abfc837    2025-05-08 22:42:07    ene49-ep-pucpr      /dados-backup  
7f646b0b    2025-05-08 22:58:04    ene49-ep-pucpr      /opt  
fc187711    2025-05-08 22:59:29    ene49-ep-pucpr      /opt  
-----  
4 snapshots  
cebolinha@ene49-ep-pucpr:~$
```

Figure 15: Test User Viewing Snapshots in Restic Repository

## Secure Remote Access via SSH

The **SSH** (Secure Shell) protocol was configured with a focus on security and access control, using public key authentication and integration with multifactor authentication.

### SSH Server Configuration

The main configuration file used was `/etc/ssh/sshd_config`. The main security directives applied include:

- `PubkeyAuthentication yes` – Enables public key authentication.
- `AuthorizedKeysFile .ssh/authorized_keys` – Standard path for authorized keys.
- `PasswordAuthentication no` – Disables password authentication.
- `PermitRootLogin no` – Blocks direct login as `root`.
- `UsePAM yes` – Kept enabled for Google Authenticator integration.
- `PermitEmptyPasswords no` – Prevents authentication of users with empty passwords.

### Public Key Installation

Authorized users' public keys were added to their respective `~/.ssh` directories:

```
1 mkdir -p ~/.ssh
2 chmod 700 ~/.ssh
3 echo "XXCHAVEXX" > ~/.ssh/authorized_keys
4 chmod 600 ~/.ssh/authorized_keys
```

Listing 25: Directory Creation and Key Installation

### Bypass for Test User

For the `cebolinha` user, a Google Authenticator bypass was configured, requiring only the public key:

```
1 Match User cebolinha
2     AuthenticationMethods publickey
```

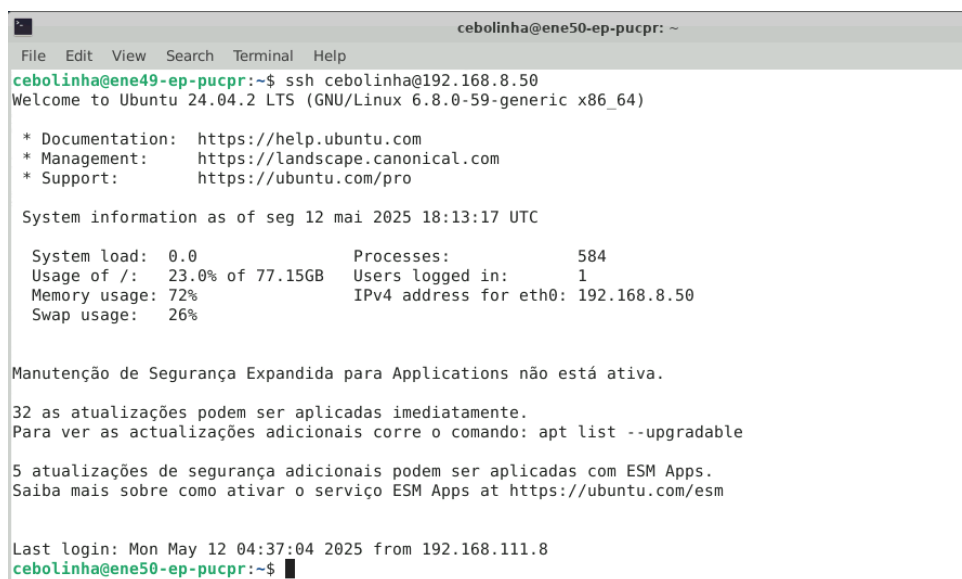
Listing 26: Match rule for test user

### Connectivity Tests

SSH connectivity tests were conducted between different network segments to verify firewall and routing policies:

- **Internal Network** → **DMZ** on port 22: **Allowed**.
- **DMZ** → **Internal Network** on port 2222: **Allowed**.
- **External Machine** → **DMZ** on port 22: **Allowed**.

- **External Machine → Internal Network** on port 2222: **Blocked** (as expected).



```
cebolinha@ene50-ep-pucpr: ~
File Edit View Search Terminal Help
cebolinha@ene49-ep-pucpr:~$ ssh cebolinha@192.168.8.50
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of seg 12 mai 2025 18:13:17 UTC

System load:  0.0                Processes:    584
Usage of /:   23.0% of 77.15GB   Users logged in: 1
Memory usage: 72%              IPv4 address for eth0: 192.168.8.50
Swap usage:  26%

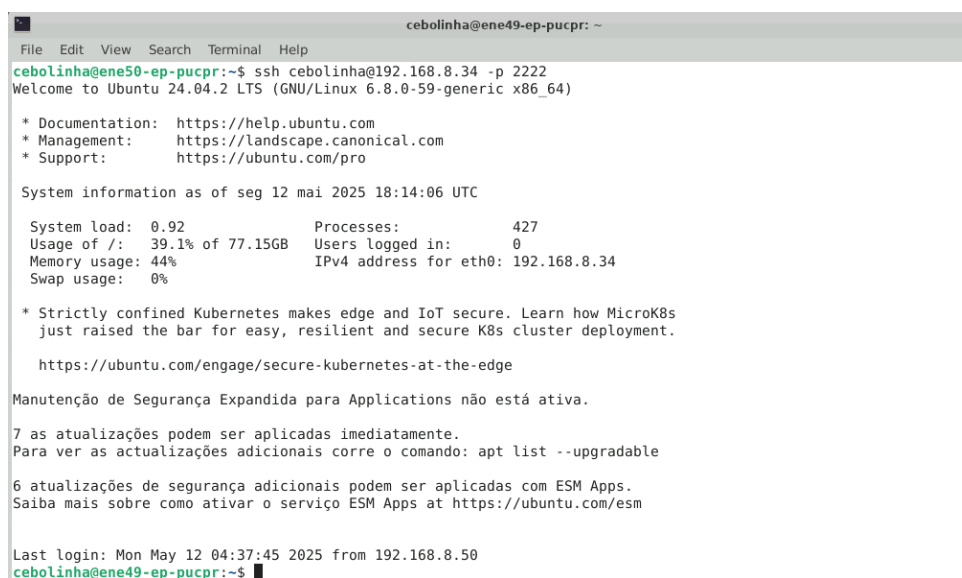
Manutenção de Segurança Expandida para Applications não está ativa.

32 as atualizações podem ser aplicadas imediatamente.
Para ver as actualizações adicionais corre o comando: apt list --upgradable

5 atualizações de segurança adicionais podem ser aplicadas com ESM Apps.
Saiba mais sobre como ativar o serviço ESM Apps at https://ubuntu.com/esm

Last login: Mon May 12 04:37:04 2025 from 192.168.111.8
cebolinha@ene50-ep-pucpr:~$
```

Figure 16: Connection Established from Internal Network to DMZ on Port 22



```
cebolinha@ene49-ep-pucpr: ~
File Edit View Search Terminal Help
cebolinha@ene50-ep-pucpr:~$ ssh cebolinha@192.168.8.34 -p 2222
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of seg 12 mai 2025 18:14:06 UTC

System load:  0.92                Processes:    427
Usage of /:   39.1% of 77.15GB   Users logged in: 0
Memory usage: 44%              IPv4 address for eth0: 192.168.8.34
Swap usage:  0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

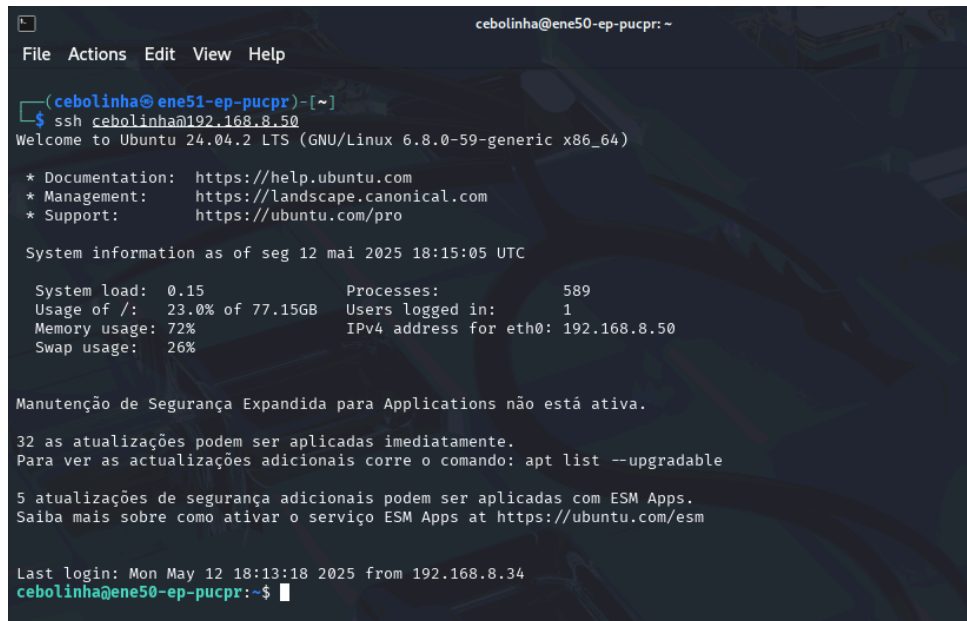
Manutenção de Segurança Expandida para Applications não está ativa.

7 as atualizações podem ser aplicadas imediatamente.
Para ver as actualizações adicionais corre o comando: apt list --upgradable

6 atualizações de segurança adicionais podem ser aplicadas com ESM Apps.
Saiba mais sobre como ativar o serviço ESM Apps at https://ubuntu.com/esm

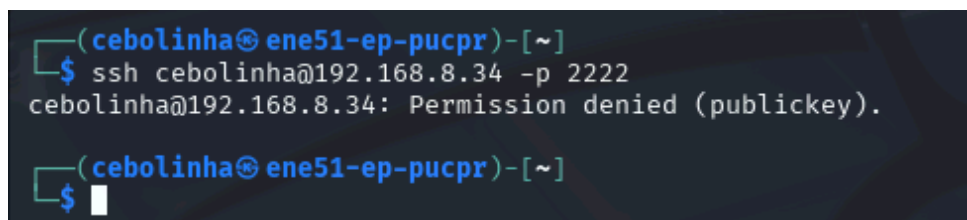
Last login: Mon May 12 04:37:45 2025 from 192.168.8.50
cebolinha@ene49-ep-pucpr:~$
```

Figure 17: Connection Established from DMZ to Internal Network on Port 2222



```
cebolinha@ene50-ep-pucpr: ~  
File Actions Edit View Help  
(cebolinha@ene51-ep-pucpr)-[~]  
$ ssh cebolinha@192.168.8.50  
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-59-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
System information as of seg 12 mai 2025 18:15:05 UTC  
  
System load:  0.15          Processes:           589  
Usage of /:   23.0% of 77.15GB Users logged in:     1  
Memory usage: 72%          IPv4 address for eth0: 192.168.8.50  
Swap usage:   26%  
  
Manutenção de Segurança Expandida para Applications não está ativa.  
  
32 as atualizações podem ser aplicadas imediatamente.  
Para ver as actualizações adicionais corre o comando: apt list --upgradable  
  
5 atualizações de segurança adicionais podem ser aplicadas com ESM Apps.  
Saiba mais sobre como ativar o serviço ESM Apps at https://ubuntu.com/esm  
  
Last login: Mon May 12 18:13:18 2025 from 192.168.8.34  
cebolinha@ene50-ep-pucpr:~$
```

Figure 18: Connection Established from External Machine to DMZ on Port 22



```
(cebolinha@ene51-ep-pucpr)-[~]  
$ ssh cebolinha@192.168.8.34 -p 2222  
cebolinha@192.168.8.34: Permission denied (publickey).  
  
(cebolinha@ene51-ep-pucpr)-[~]  
$
```

Figure 19: Connection Attempt from External Machine to Internal Network on Port 2222  
- Access Denied



## Snort (IDS/IPS)

**Snort** was configured as an Intrusion Detection and Prevention System (IDS/IPS), operating in real-time to protect the Internal Network and DMZ against scans, unauthorized access, and suspicious connections.

### Rule Sources

Several rule sources were enabled to maximize threat detection coverage:

- **Snort VRT Rules:** Requires registration on Snort.org and provision of the *Oinkmaster Code*.
- **Snort GPLv2 Community Rules:** Community-maintained rules.
- **Emerging Threats Open Rules (ET Open):** Free set maintained by Proofpoint.
- **OpenAppID Detectors:** Application detection by signature.
- **Feodo Tracker Botnet C2 IP Rules:** Blocks connections to command and control (C2) IPs.

**Rule Update Interval:** Daily at 04:00.

**Automatic IP Unblocking:** After 1 hour.

### Manual Rule Update

An initial manual update was performed in the Updates tab to ensure all sources were synchronized:

- Snort Subscriber Ruleset
- Snort GPLv2 Community Rules
- Emerging Threats Open Rules
- Snort OpenAppID Detectors
- Snort AppID Open Text Rules
- Feodo Tracker Botnet C2 IP Rules

### Pass Lists (Exclusion List)

To avoid blocking essential internal IPs, an exclusion list was created with the following IPs:

- 192.168.8.50 (Internal Network Server)
- 192.168.8.34 (DMZ Server)

### Monitored Interfaces

The configured interfaces for monitoring are:

- **LAN33 (Internal Network)**

- **LAN49 (DMZ)**

For both:

- Snort enabled on the interface
- *Block Offenders* enabled (legacy mode)
- *Kill States* enabled
- Detection mode: **AC-BNFA**, with optimized search
- Passlist applied

## Rule Policy

The **Balanced Policy** was chosen, providing a balance between security and connectivity. The **Connectivity** policy, though less restrictive, allowed unwanted scans and tests.

## Preprocessors

- **SSH Detection:**
  - LAN33: port 2222
  - LAN49: port 22
- **Portscan Detection:**
  - LAN33: **High** sensitivity
  - LAN49: **Medium** sensitivity
- **ARP Spoofing Detection:** Enabled on both interfaces

## Detection Tests

To validate Snort's functionality, the following tests were conducted:

- **Nmap scan from external machine to DMZ:**

```
1 nmap -sV -O 192.168.8.50
```

Listing 27: Nmap Scan

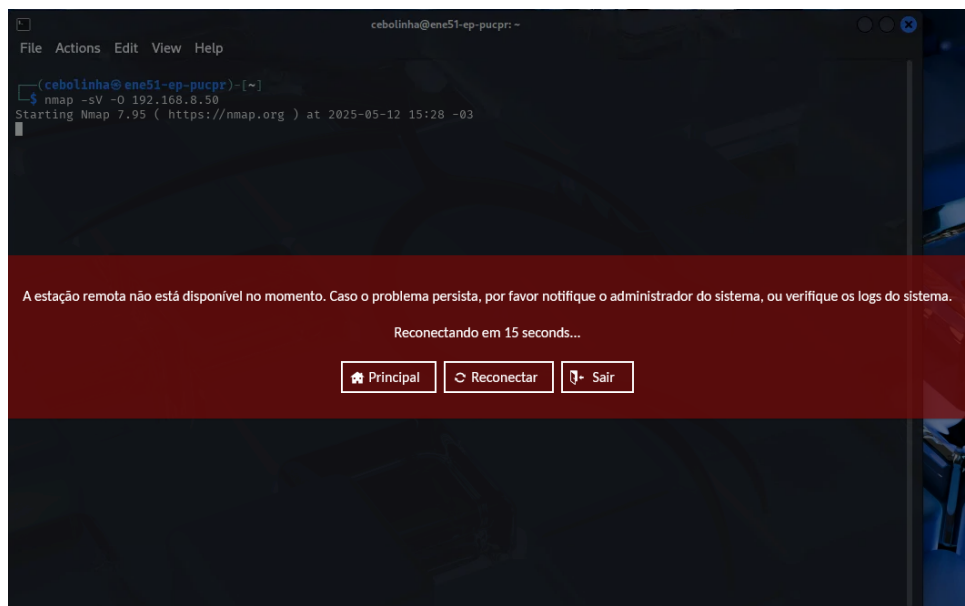


Figure 20: Nmap Scan Blocked and Stopped

Alert Log View Settings

Interface to Inspect

LAN49 (hn2)

Choose interface...

☒ Auto-refresh view

250

Alert lines to display.

Salvar

Alert Log Actions

Download




Limpar

Alert Log View Filter

8 Entries in Active Log

Data	Ação	Pri	Proto	Class	IP de Origem	SPort	IP de Destino	DPort	GID:SID	Descrição
2025-05-12 14:28:55	⚠	1	TCP	Web Application Attack	192.168.111.8 🔍📄🔴	45654	192.168.8.50 🔍📄	8080	1:2024364 📄🔴	ET SCAN Possible Nmap User-Agent Observed
2025-05-12 14:28:55	⚠	1	TCP	Web Application Attack	192.168.111.8 🔍📄🔴	34660	192.168.8.50 🔍📄	80	1:2024364 📄🔴	ET SCAN Possible Nmap User-Agent Observed
2025-05-12 14:28:55	⚠	1	TCP	Web Application Attack	192.168.111.8 🔍📄🔴	34658	192.168.8.50 🔍📄	80	1:2024364 📄🔴	ET SCAN Possible Nmap User-Agent Observed
2025-05-12 14:28:55	⚠	1	TCP	Web Application Attack	192.168.111.8 🔍📄🔴	45640	192.168.8.50 🔍📄	8080	1:2024364 📄🔴	ET SCAN Possible Nmap User-Agent Observed

Figure 21: Alert Generated by Snort During Intrusion Attempt

Blocked Hosts and Log View Settings				
Blocked Hosts		<div> Download</div>	<div> Limpar</div>	
		All blocked hosts will be saved		All blocked hosts will be removed
Refresh and Log View		<div> Salvar</div>	<div><input checked="" type="checkbox"/> Atualizar</div>	<div><input type="text" value="500"/></div>
		Save auto-refresh and view settings	Default is ON	Number of blocked entries to view. Default is 500

Last 500 Hosts Blocked by Snort (only applicable to Legacy Blocking Mode interfaces)

#	IP	Alert Descriptions and Event Times	Remover
1	192.168.111.8	ET SCAN Possible Nmap User-Agent Observed – 2025-05-12 14:28:55	✕

1 host IP address is currently being blocked Snort on Legacy Blocking Mode interfaces.

Figure 22: External IP Automatically Blocked by Snort

## Light LDAP (LLDAP)

The **LLDAP (Light LDAP)** service was used as a lightweight and modern directory solution, supporting LDAPS (LDAP over TLS) and a web administrative interface. It was implemented via **Docker**, ensuring portability and simplicity in management.

### Configuration via Docker Compose

The service was configured with the following `docker-compose.yml` file, exposing port 6360 for LDAPS connections and 17170 for the web interface:

```
1 version: "3"
2
3 services:
4   ldap:
5     image: ldap/ldap:stable
6     ports:
7       - "6360:6360"    # LDAPS
8       - "17170:17170"  # Web interface
9     volumes:
10      - ldap_data:/data
11      - ./certs/ldap.crt:/app/ldap.crt
12      - ./certs/ldap.key:/app/ldap.key
13     environment:
14       - LDAP_LDAPS_OPTIONS__ENABLED=true
15       - LDAP_LDAPS_OPTIONS__CERT_FILE=/app/ldap.crt
16       - LDAP_LDAPS_OPTIONS__KEY_FILE=/app/ldap.key
17       - LDAP_JWT_SECRET=XXXXXX
18       - LDAP_KEY_SEED=XXXXXX
19       - LDAP_LDAP_BASE_DN=dc=example,dc=com
20       - LDAP_LDAP_USER_PASS=XXXXX
21     restart: unless-stopped
22
23 volumes:
24   ldap_data:
25     driver: local
```

Listing 28: LDAP docker-compose.yml File

### Web Administrative Interface

After deployment, the administrative interface can be accessed via browser at:

`http://localhost:17170`

Through this interface, the following actions were performed:

- Created a new user with administrative privileges.
- Created the `cebolinha` user and added it to the `LDAP-STRICT-READONLY` group, ensuring restricted read-only access.

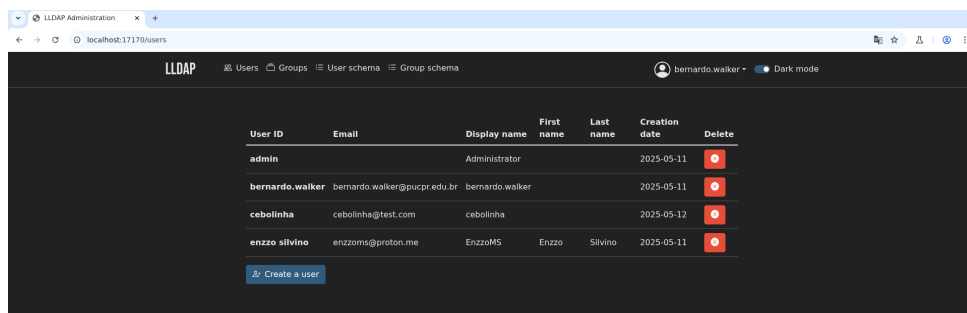


Figure 23: LLDAP Administration Screen

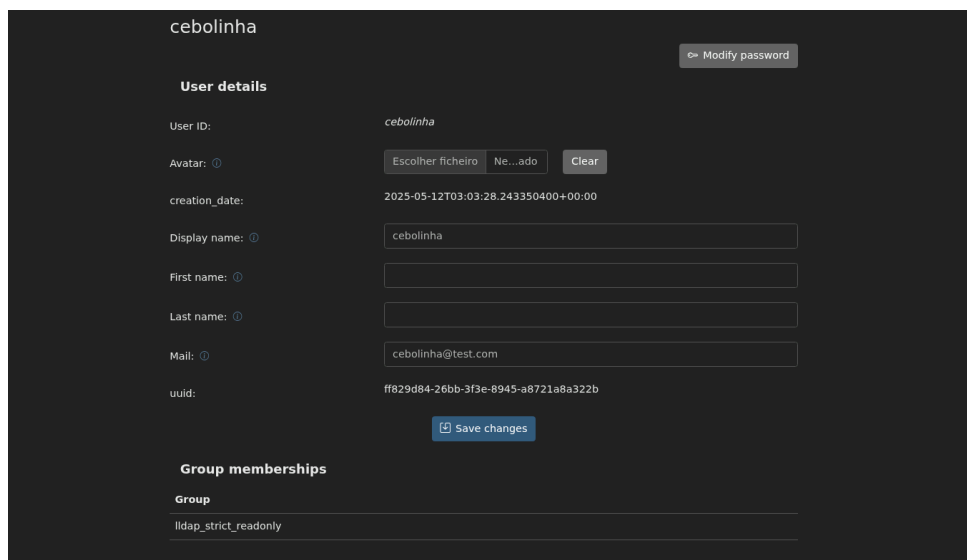


Figure 24: View of Test User (cebolinha)

## LDAPSEARCH Query

The `ldapsearch` command was used to validate the structure and read permissions assigned to the test user. Below are examples of searches performed:

```
1 ldapsearch -x -H ldaps://localhost:6360 -D "uid=cebolinha,ou=
  people,dc=example,dc=com" -W -b "dc=example,dc=com" "(cn=
  cebolinha)"
```

Listing 29: LDAP Query for cebolinha

```
1 ldapsearch -x -H ldaps://localhost:6360 -D "uid=cebolinha,ou=
  people,dc=example,dc=com" -W -b "dc=example,dc=com" "(cn=admin)
  "
```

Listing 30: LDAP Query for admin user

```
cebolinha@ene49-ep-pucpr:~$ ldapsearch -H ldaps://192.168.8.34:6360 -D "uid=cebolinha,ou=people,dc=example,dc=com" -w "c3b0l1nh4" -b "dc=example,dc=com" "(cn=cebolinha)"
# extended LDIF
#
# LDAPv3
# base <dc=example,dc=com> with scope subtree
# filter: (cn=cebolinha)
# requesting: ALL
#
# cebolinha, people, example.com
dn: uid=cebolinha,ou=people,dc=example,dc=com
cn: cebolinha
createTimestamp: 2025-05-12T03:03:28.243350400+00:00
entryUUID: ff829d84-26bb-3f3e-8945-a8721a8a322b
mail: cebolinha@test.com
objectclass: inetOrgPerson
objectclass: posixAccount
objectclass: mailAccount
objectclass: person
uid: cebolinha

# search result
search: 2
result: 0 Success
control: 1.2.840.113556.1.4.319 false MAUCAQEEAA==
pagedresults: estimate=1 cookie=

# numResponses: 2
# numEntries: 1
cebolinha@ene49-ep-pucpr:~$
```

Figure 25: LDAP Query for cebolinha using ldapsearch

```
cebolinha@ene49-ep-pucpr:~$ ldapsearch -H ldaps://192.168.8.34:6360 -D "uid=cebolinha,ou=people,dc=example,dc=com" -w "c3b0l1nh4" -b "dc=example,dc=com" "(cn=admin)"
# extended LDIF
#
# LDAPv3
# base <dc=example,dc=com> with scope subtree
# filter: (cn=admin)
# requesting: ALL
#
# search result
search: 2
result: 0 Success
control: 1.2.840.113556.1.4.319 false MAUCAQAEAA==
pagedresults: cookie=

# numResponses: 1
cebolinha@ene49-ep-pucpr:~$
```

Figure 26: LDAP Query for admin using ldapsearch

## Wazuh (XDR + SIEM)

**Wazuh** is a security platform that functions as a *SIEM* (Security Information and Event Management) and *XDR* (Extended Detection and Response) solution. It enables log collection, analysis, and correlation, intrusion detection, integrity monitoring, incident response, and more.

### Wazuh Manager Installation

The **Wazuh Manager** installation was performed using the official script:

```
1 curl -sO https://packages.wazuh.com/4.11/wazuh-install.sh
2 chmod +x wazuh-install.sh
3 ./wazuh-install.sh
```

Listing 31: Wazuh Manager Installation

*Note:* The **admin** user password is automatically generated during installation and displayed in the terminal.

### Wazuh Agent Installation

On a client machine (such as in the DMZ or internal network), the **Wazuh Agent** was installed with the commands below, using the Wazuh Manager's IP:

```
1 wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/
   wazuh-agent_4.11.2-1_amd64.deb
2 sudo WAZUH_MANAGER='192.168.8.34' WAZUH_AGENT_NAME='DMZ' dpkg -i
   ./wazuh-agent_4.11.2-1_amd64.deb
3 sudo systemctl daemon-reload
4 sudo systemctl enable wazuh-agent
5 sudo systemctl start wazuh-agent
```

Listing 32: Wazuh Agent Installation

### Wazuh Agent Configuration

To ensure audit logs are monitored, the agent configuration file was edited:

```
1 <localfile>
2   <log_format>audit</log_format>
3   <location>/var/log/audit/audit.log</location>
4 </localfile>
```

Listing 33: Audit Log Monitoring Addition

### Audit Rules for Privileged Commands

To ensure visibility into administrative privilege usage, rules were added to **auditd**:

```

1 -a exit,always -F arch=b64 -F euid=0 -S execve -k audit-wazuh
2 -a exit,always -F arch=b32 -F euid=0 -S execve -k audit-wazuh

```

Listing 34: Audit Rules in /etc/audit/rules.d/audit.rules

These rules ensure that all `execve()` system calls performed by users with effective ID 0 (`root`) are logged and processed by Wazuh.

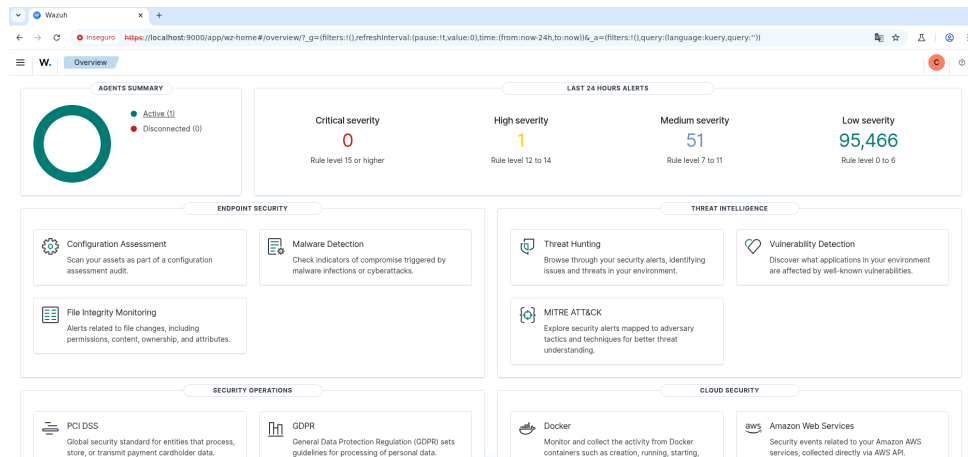


Figure 27: Wazuh Dashboard with Active Agent in DMZ

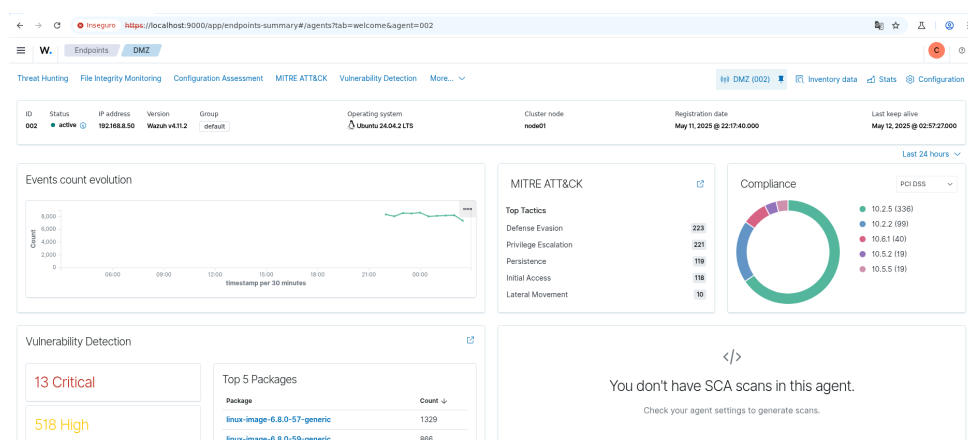


Figure 28: Wazuh Agent Interface Running on DMZ Machine



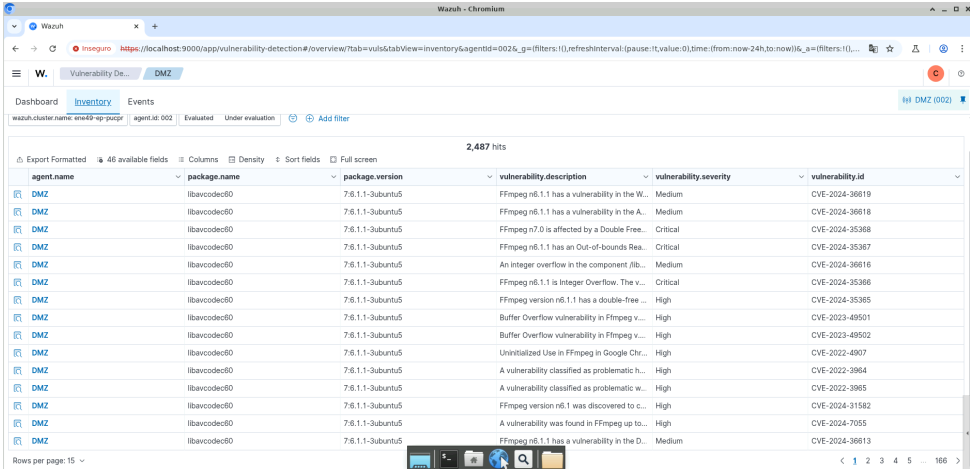


Figure 29: Vulnerability Inventory Detected

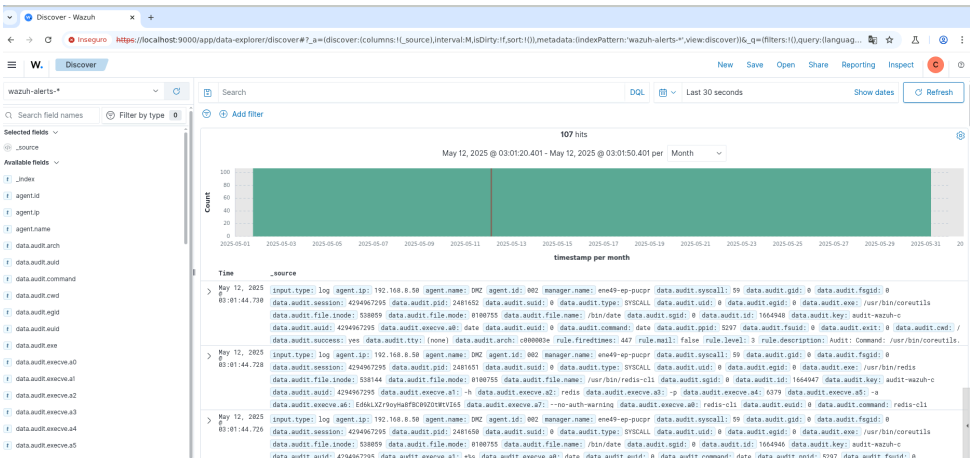


Figure 30: Wazuh Alert Center with Security Events

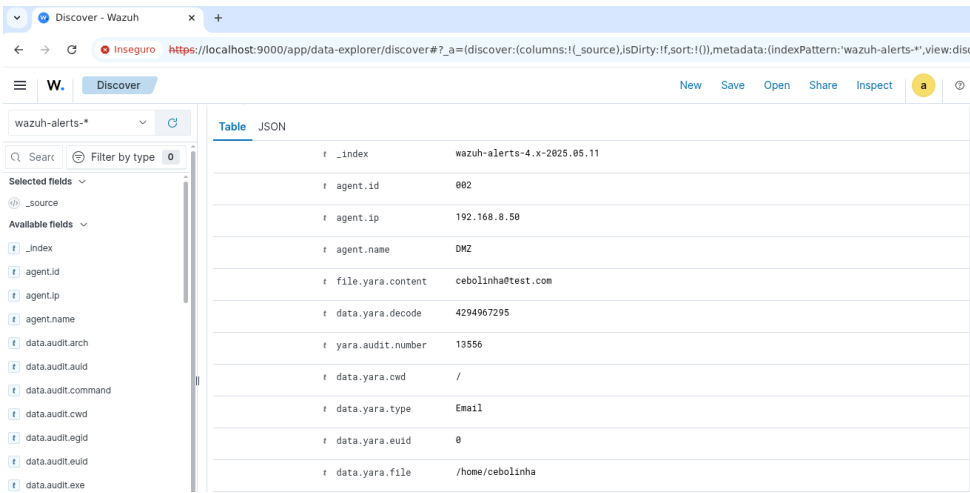


Figure 31: Event of Execution with Root Permissions Captured

## YARA Integration with Wazuh (DLP)

To enhance **Data Loss Prevention (DLP)** capabilities, integration between **YARA** and the **Wazuh** agent was performed. YARA is a powerful tool for identifying patterns and sensitive content in files.

### YARA Rules

Rules were defined in the `/opt/dlp/rules/dlp.yar` file. Below are some of the rules used:

```
1 rule Detect_PrivateKey_PEM {
2   meta:
3     description = "Detects private keys in PEM format"
4   strings:
5     $pem1 = "-----BEGIN PRIVATE KEY-----"
6     $pem2 = "-----END PRIVATE KEY-----"
7   condition:
8     $pem1 and $pem2
9 }
10
11 rule Detect_Email {
12   meta:
13     description = "Detects email addresses"
14   strings:
15     $email = /[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}/
16   condition:
17     $email
18 }
19
20 rule Detect_PhoneNumber_BR {
21   meta:
22     description = "Detects Brazilian phone numbers"
23   strings:
24     $phone = /\((?\d{2}\)?\s?\d{4,5}-\d{4})/
25   condition:
26     $phone
27 }
```

Listing 35: YARA Rules for Sensitive Data Detection

### Scanning Script and Scheduling

A `yara_scan.sh` script was created to scan defined directories, executing YARA rules and logging the output:

```
1 #!/bin/bash
2
3 echo "=== [$(date)] Scan Start ===" >> /var/log/yara-dlp.log
4 yara -r /opt/dlp/rules/dlp.yar /home/* >> /var/log/yara-dlp.log
5 2>&1
```

```

5 echo "=== Scan finished ===" >> /var/log/yara-dlp.log
6 echo >> /var/log/yara-dlp.log

```

Listing 36: Excerpt from yara\_scan.sh Script

This script is automatically executed every 4 hours via a scheduled cron task:

```

1 0 */4 * * * root /usr/local/bin/yara_scan.sh

```

Listing 37: Crontab Scheduling

## Integration with Wazuh Agent

To enable Wazuh to monitor YARA detections, the agent was configured to track the log file:

```

1 <localfile>
2   <log_format>full_command</log_format>
3   <location>/var/log/yara-dlp.log</location>
4 </localfile>

```

Listing 38: YARA Log Monitoring in Wazuh Agent

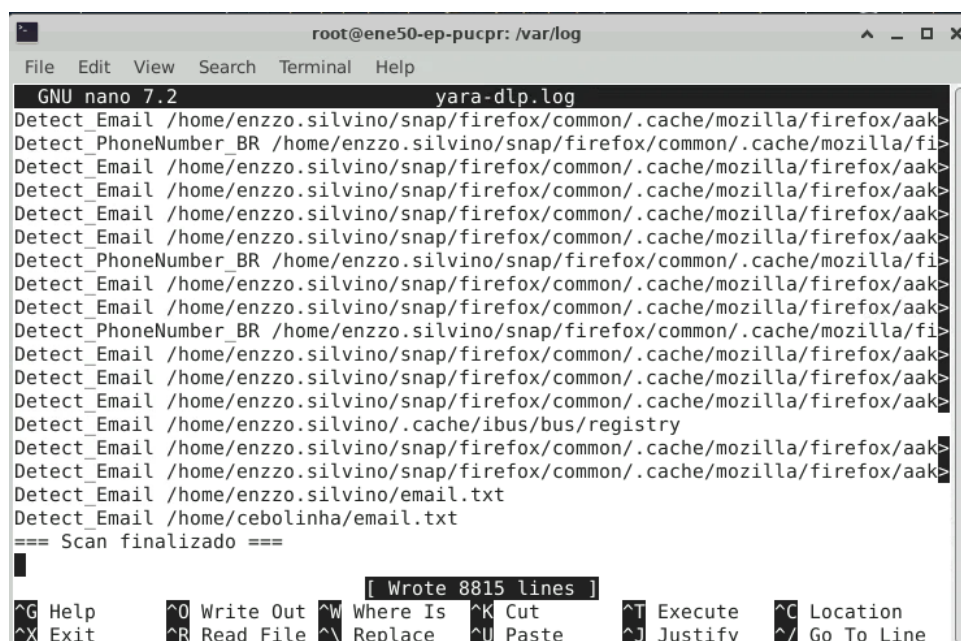


Figure 32: Log Indicating Email Presence in Test User's File

## HashiCorp Vault (PAM)

To ensure secure credential delivery between systems, the **HashiCorp Vault** tool was used, implementing the **Privileged Access Management (PAM)** standard. The solution dynamically distributes MySQL database credentials from the internal network (192.168.8.34) to a PHP application hosted on Apache in the DMZ (192.168.8.50).

### Workflow

- The **Vault** stores credentials in the KV v2 engine.
- DMZ machine authentication is performed via **AppRole** with the **Vault Agent**.
- The agent dynamically generates the `/etc/app/db.env` file with username and password.
- Apache is automatically reloaded when the template is updated.
- Communication between the DMZ and Vault is protected by TLS with a self-signed certificate.

### Vault Configuration (host 192.168.8.34)

File `vault.hcl`:

```
1 ui = true
2 storage "file" {
3   path = "/opt/vault/data"
4 }
5 listener "tcp" {
6   address = "0.0.0.0:8200"
7   tls_cert_file = "/opt/vault/tls/tls.crt"
8   tls_key_file = "/opt/vault/tls/tls.key"
9   tls_disable = false
10 }
11 api_addr = "https://192.168.8.34:8200"
```

Listing 39: vault.hcl

- Certificate with SAN: `vault.interno.local` + IP.
- Initialization: `vault operator init -key-shares=5 -key-threshold=3`.

### Enable KV v2 engine:

```
1 vault secrets enable -path=secret kv-v2
2 vault kv put secret/db-credentials username="remote_user"
   password="#####"
```

### db-read.hcl Policy:

```
1 path "secret/data/db-credentials" {
2   capabilities = ["read"]
3 }
```

```
1 vault policy write app-db-read db-read.hcl
```

### AppRole Configuration:

```
1 vault auth enable approle
2 vault write auth/approle/role/app-role \
3   token_policies="app-db-read" \
4   token_ttl=1h token_max_ttl=4h \
5   token_bound_cidrs="192.168.8.50/32" \
6   secret_id_num_uses=0 secret_id_ttl=0
7
8 vault read -field=role_id auth/approle/role/app-role/role-id > /
   etc/vault/role_id
9 vault write -field=secret_id -f auth/approle/role/app-role/secret
   -id > /etc/vault/secret_id
```

## Vault Agent Configuration (host 192.168.8.50)

### Environment file template: /etc/vault/db-template.tpl

```
1 DB_USER={{ with secret "secret/data/db-credentials" }}{{ .Data.
   data.username }}{{ end }}
2 DB_PASS={{ with secret "secret/data/db-credentials" }}{{ .Data.
   data.password }}{{ end }}
```

### Agent Configuration: /etc/vault/agent.hcl

```
1 vault {
2   address = "https://192.168.8.34:8200"
3   tls_skip_verify = false
4 }
5
6 auto_auth {
7   method "approle" {
8     mount_path = "approle"
9     config = {
10       role_id_file_path = "/etc/vault/role_id"
11       secret_id_file_path = "/etc/vault/secret_id"
12     }
13   }
14   sink "file" {
15     config = {
16       path = "/etc/vault/.vault-token"
17     }
18   }
19 }
20
21 template {
22   source = "/etc/vault/db-template.tpl"
23   destination = "/etc/app/db.env"
24   perms = "0640"
```

```

25     command      = "systemctl reload apache2"
26 }

```

### Systemd Service: vault-agent.service

```

1 [Unit]
2 Description=Vault Agent - AppRole + template
3 After=network-online.target
4
5 [Service]
6 ExecStart=/usr/bin/vault agent -config=/etc/vault/agent.hcl
7 Restart=on-failure
8 RestartSec=5s
9
10 [Install]
11 WantedBy=multi-user.target

```

## Apache/PHP Integration

The PHP code directly reads the file:

```

1 $creds = parse_ini_file('/etc/app/db.env');
2 $mysqli = new mysqli('192.168.8.34', $creds['DB_USER'], $creds['
    DB_PASS'], 'fortune_quotes');

```

### Permissions:

- /etc/app: 750
- db.env: 640
- Group: www-data

```

filho.cassio@ene49-ep-pucpr: ~
File Edit View Search Terminal Help
filho.cassio@ene49-ep-pucpr:~$ vault kv patch -address=https://192.168.8.34:8200 secret/db-credentials password="NOVA_SENHA15H12"
===== Secret Path =====
secret/data/db-credentials

===== Metadata =====
Key          Value
----          -
created_time  2025-05-12T20:06:57.4216248Z
custom_metadata <nil>
deletion_time n/a
destroyed     false
version       20
filho.cassio@ene49-ep-pucpr:~$ vault status -address=https://192.168.8.34:8200
Key          Value
----          -
Seal Type     shamir
Initialized   true
Sealed        false
Total Shares  5
Threshold     3
Version       1.19.3
Build Date    2025-04-29T10:34:52Z
Storage Type   file
Cluster Name  vault-cluster-47422ae6
Cluster ID    07a45341-b096-ae07-d25f-56d920f19366
HA Enabled    false
filho.cassio@ene49-ep-pucpr:~$

```

Figure 33: Password Update via vault kv patch

```

filho.cassio@ene50-ep-pucpr:/tmp$ sudo journalctl -u vault-agent -f
[sudo] password for filho.cassio:
mai 12 16:47:21 ene50-ep-pucpr vault[2938000]: 2025-05-12T16:47:21.147Z [INFO] agent: (runner) starting
mai 12 16:47:21 ene50-ep-pucpr vault[2938000]: 2025-05-12T16:47:21.159Z [INFO] agent.auth.handler: renewed auth token
mai 12 16:51:59 ene50-ep-pucpr vault[2938000]: 2025-05-12T16:51:59.449Z [INFO] agent: (runner) rendered "/etc/vault/db-template.tpl" => "/etc/app/db.env"
mai 12 16:51:59 ene50-ep-pucpr vault[2938000]: 2025-05-12T16:51:59.449Z [INFO] agent: (runner) executing command "[\"systemctl reload apache2\"]" from "/etc/vault/db-template.tpl" => "/etc/app/db.env"
mai 12 16:51:59 ene50-ep-pucpr vault[2938000]: 2025-05-12T16:51:59.449Z [INFO] agent: (child) spawning: sh -c systemctl reload apache2
mai 12 17:29:54 ene50-ep-pucpr vault[2938000]: 2025-05-12T17:29:54.146Z [INFO] agent.auth.handler: renewed auth token
mai 12 18:12:27 ene50-ep-pucpr vault[2938000]: 2025-05-12T18:12:27.133Z [INFO] agent.auth.handler: renewed auth token
mai 12 18:13:22 ene50-ep-pucpr vault[2938000]: 2025-05-12T18:13:22.666Z [INFO] agent: (runner) rendered "/etc/vault/db-template.tpl" => "/etc/app/db.env"
mai 12 18:13:22 ene50-ep-pucpr vault[2938000]: 2025-05-12T18:13:22.666Z [INFO] agent: (runner) executing command "[\"systemctl reload apache2\"]" from "/etc/vault/db-template.tpl" => "/etc/app/db.env"
mai 12 18:13:22 ene50-ep-pucpr vault[2938000]: 2025-05-12T18:13:22.667Z [INFO] agent: (child) spawning: sh -c systemctl reload apache2

```

Figure 34: Vault Agent Logs Indicating Successful Rendering

```

bernardo.walker@ene50-ep-pucpr: ~
File Edit View Search Terminal Help

bernardo.walker@ene50-ep-pucpr:~$ sudo cat /etc/app/db.env
DB_USER=remote_user
DB_PASS=NOVA_SENHA15H12

bernardo.walker@ene50-ep-pucpr:~$

```

Figure 35: Generated Content in db.env File

```

/**
 * Connect to the database
 *
 * @return mysqli Database connection
 */
function connectDB() {
    $host = '192.168.8.34:3307';
    $database = 'fortune_quotes';

    $envFile = '/etc/app/db.env';
    if (is_readable($envFile)) {
        $creds = parse_ini_file($envFile);
        if ($creds !== false && isset($creds['DB_USER'], $creds['DB_PASS'])) {
            $username = $creds['DB_USER'];
            $password = $creds['DB_PASS'];
        }
    }
    error_log("connectDB: using DB_USER={$username} DB_PASS={$password}");

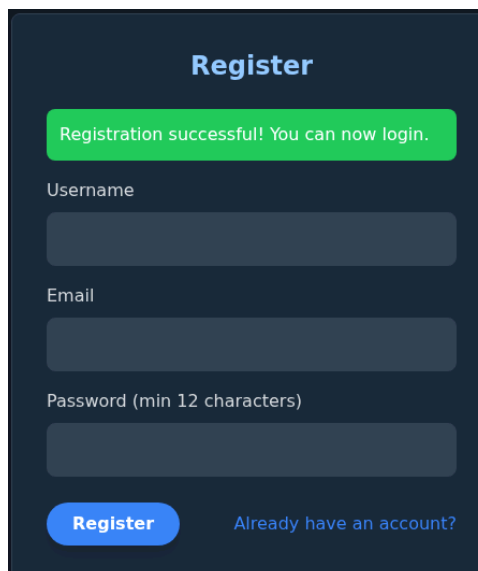
    $db = new mysqli($host, $username, $password, $database);

    if ($db->connect_error) {
        die('Database connection failed: ' . $db->connect_error);
    }

    return $db;
}

```

Figure 36: Web Application Code Accessing Database



**Register**

Registration successful! You can now login.

Username

Email

Password (min 12 characters)

**Register** [Already have an account?](#)

Figure 37: Web Application Registering Normally with Dynamic Credentials



## Firewall Rules

Traffic control between network zones was achieved through restrictive firewall rules, ensuring only necessary flows are allowed, thus minimizing the attack surface. The rules were applied to routers/firewalls separating the zones: **Internal Network**, **DMZ**, and **External**.

### Blocking Internal Network to External Machine

To prevent data theft and unauthorized access from within the network to the external world, the Internal Network was blocked from initiating any direct connections to the external machine.

Firewall / Regras / LAN33											
Flutuante WireGuard WAN LAN33 LAN49 LAN111 LAN1113000 WIREGUARDTUN											
Regras (Arraste para mudar a ordem)											
<input type="checkbox"/>	Estados	Protocolo	Origem	Porta	Destino	Porta	Gateway	FILA	Agenda	Descrição	Ações
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP/UDP	*	*	192.168.111.8	*	*	nenhum		Bloquear a rede interna de acessar o kali	
<input type="checkbox"/>	✓ 11/3,95 GiB	IPv4 *	*	*	*	*	*	nenhum			
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	LAN33 subnets	*	*	*	*	nenhum		Default allow LAN to any rule	

Figure 38: Blocking All Ports from Internal Network to External Machine

### DMZ to Internal Network Access Rules

Communication from the DMZ to the Internal Network is strictly controlled. Only pre-authorized services are allowed, such as MySQL database access (port 3306) and Vault authentication (port 8200).

### External Machine Restrictions

The External Machine is limited to accessing only public services exposed in the DMZ, such as:

- Web Application (HTTP/HTTPS) – Ports 80/443
- Email Server (SMTP/IMAP/Submission) – Ports 25, 587, 993

Direct connection attempts to the Internal Network are completely blocked. Administrative access is performed in two steps:

1. SSH from the external machine to the DMZ (port 22)
2. SSH from the DMZ to the Internal Network (port 2222)

































































Firewall / Regras / LAN49											
Flutuante   WireGuard   WAN   LAN33 <b>LAN49</b> LAN111   LAN1113000   WIREGUARDTUN											
Regras (Arraste para mudar a ordem)											
<input type="checkbox"/>	Estados	Protocolo	Origem	Porta	Destino	Porta	Gateway	FILA	Agenda	Descrição	Ações
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	192.168.8.34	445 (MS DS)	*	nenhum		Compartilhamento de arquivos	   
<input type="checkbox"/>	✓ 0/44 KIB	IPv4 TCP	*	*	192.168.8.34	2222	*	nenhum		SSH rede interna	   
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	192.168.8.34	17170	*	nenhum		Serviço Docker 2	   
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	192.168.8.34	6360	*	nenhum		Serviço Docker 1	   
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	192.168.8.34	55000	*	nenhum		Canal interno Wazuh	   
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	192.168.8.34	1515	*	nenhum		Autenticação Wazuh	   
<input type="checkbox"/>	✓ 0/57,38 MB	IPv4 TCP	*	*	192.168.8.34	1514	*	nenhum		Recepção de logs Wazuh	   
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	*	*	192.168.8.34	9000	*	nenhum		Bloquear wazuh dashboard	   
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	192.168.8.34	8201	*	nenhum		Vault HTTPS API	   
<input type="checkbox"/>	✓ 1/4,77 MB	IPv4 TCP	*	*	192.168.8.34	8200	*	nenhum		Vault HTTP API	   
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	192.168.8.34	3389 (MS RDP)	*	nenhum		Permite RDP via xrdp	   
<input type="checkbox"/>	✓ 0/10,05 MB	IPv4 TCP	*	*	192.168.8.34	3307	*	nenhum		Permitir acesso MySQL 3307	   
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	192.168.8.0/24	*	192.168.8.50	22 (SSH)	*	nenhum			   
<input type="checkbox"/>	✗ 0/10 KIB	IPv4 TCP/UDP	*	*	192.168.8.34	*	*	nenhum		Bloquear acesso a rede interna	   
<input type="checkbox"/>	✓ 16/4,97 GB	IPv4 *	*	*	*	*	*	nenhum			   
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	LAN33 subnets	*	*	*	*	nenhum		Default allow LAN to any rule	   

Figure 39: DMZ to Internal Network Rules: Only Authorized Connections Allowed























































Firewall / Regras / LAN111											
<span>Flutuante</span> <span>WireGuard</span> <span>WAN</span> <span>LAN33</span> <span>LAN49</span> <span>LAN111</span> <span>LAN1113000</span> <span>WIREGUARDTUN</span>											
Regras (Arraste para mudar a ordem)											
<input type="checkbox"/>	Estados	Protocolo	Origem	Porta	Destino	Porta	Gateway	FILA	Agenda	Descrição	Ações
<input type="checkbox"/>	✓ 0/1,22 MB	IPv4 TCP	*	*	192.168.8.50	3390	*	nenhum		RDP alternativo	  
<input type="checkbox"/>	✓ 0/10 KB	IPv4 TCP	*	*	192.168.8.50	8080	*	nenhum		Mailcow	  
<input type="checkbox"/>	✓ 0/84 B	IPv4 TCP	*	*	192.168.8.50	631	*	nenhum		Impressão via CUPS	  
<input type="checkbox"/>	✓ 0/25 KB	IPv4 TCP	*	*	192.168.8.50	993 (IMAP/S)	*	nenhum		IMAP sobre TLS	  
<input type="checkbox"/>	✓ 0/7 KB	IPv4 TCP	*	*	192.168.8.50	143 (IMAP)	*	nenhum		Recebimento IMAP	  
<input type="checkbox"/>	✓ 0/24 KB	IPv4 TCP	*	*	192.168.8.50	995 (POP3/S)	*	nenhum		POP3 sobre TLS	  
<input type="checkbox"/>	✓ 0/9 KB	IPv4 TCP	*	*	192.168.8.50	110 (POP3)	*	nenhum		Recebimento POP3	  
<input type="checkbox"/>	✓ 0/8 KB	IPv4 TCP	*	*	192.168.8.50	587 (SUBMISSION)	*	nenhum		Submissão SMTP	  
<input type="checkbox"/>	✓ 0/32 KB	IPv4 TCP	*	*	192.168.8.50	465 (SMTP/S)	*	nenhum		SMTP sobre TLS	  
<input type="checkbox"/>	✓ 0/10 KB	IPv4 TCP	*	*	192.168.8.50	25 (SMTP)	*	nenhum		Envio de e-mail	  
<input type="checkbox"/>	✓ 0/447 KB	IPv4 TCP	*	*	192.168.8.50	22 (SSH)	*	nenhum		SSH dmz open	  
<input type="checkbox"/>	✓ 0/13,33 MB	IPv4 TCP	*	*	192.168.8.50	8443	*	nenhum		Mailcow - SOGo	  
<input type="checkbox"/>	✓ 0/73,63 MB	IPv4 TCP	*	*	192.168.8.50	443 (HTTPS)	*	nenhum		acesso ao site	  
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP/UDP	*	*	192.168.8.50	*	*	nenhum		Rejeitar qualquer conexão a dmz	  
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP/UDP	*	*	192.168.8.34	*	*	nenhum		Bloquear acesso do kali a rede interna	  
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	*	*	*	22 (SSH)	*	nenhum			  
<input type="checkbox"/>	✓ 1/1,00 GB	IPv4 *	*	*	*	*	*	nenhum			  
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	LAN33 subnets	*	*	*	*	nenhum		Default allow LAN to any rule	  

Figure 40: External Machine Only Accesses Public DMZ Services; Internal Access Requires SSH Proxy