

**ERSC**

ENGENHARIA DE REDES E  
SISTEMAS DE COMPUTADORES  
ESTG-IPVC

# Report: Hackers Activity on the Internet

a report authored by

Célio Pina([celiopina@ipvc.pt](mailto:celiopina@ipvc.pt))

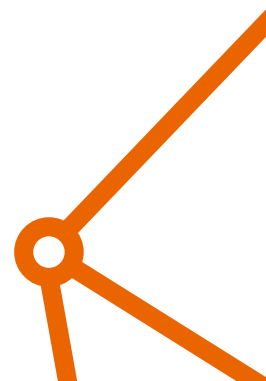
Ruben da Luz([rubenluz@ipvc.pt](mailto:rubenluz@ipvc.pt))

supervised by

Prof. Hugo Almeida and Prof. Pedro Pinto



10 June, 2023



# 1 Introduction

This report provides a comprehensive overview of the installation process of a honeypot in the AWS (Amazon Web Services) cloud environment. The installation and configuration of a honeypot serve as a proactive security measure to attract potential attackers and gather valuable data about their activities. By deploying a honeypot in the AWS cloud, organizations can gain valuable insights into the strategies, techniques, and motives of malicious actors, ultimately enhancing the protection of their systems and networks against cyber threats.

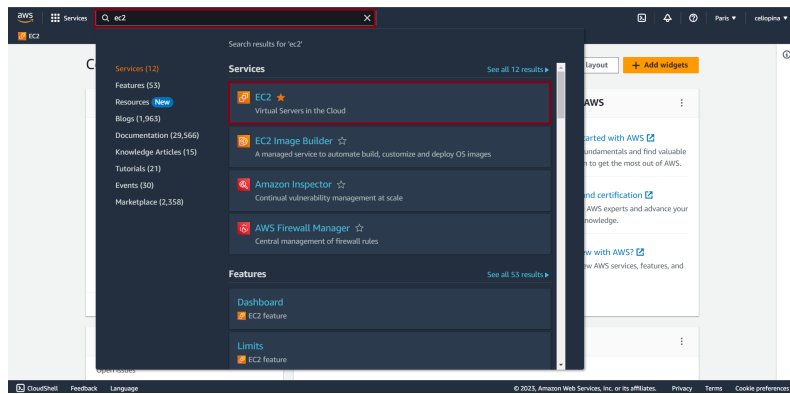
The report outlines the step-by-step process of setting up the honeypot, including the necessary configurations and considerations specific to the AWS environment. It covers aspects such as network setup, instance provisioning, and security configurations, ensuring that the honeypot operates effectively and securely within the AWS cloud. Additionally, the report highlights the importance of honeypots as a valuable tool for understanding hacker activity and developing more robust security solutions. By documenting the installation process, this report serves as a valuable resource for organizations looking to implement honeypots in the AWS cloud, contributing to their overall cybersecurity strategy.

Through the detailed exploration of the installation process, this report aims to provide practical guidance and insights for organizations seeking to enhance their cybersecurity posture and defend against evolving cyber threats.

## 2 Setting up the Honeypot

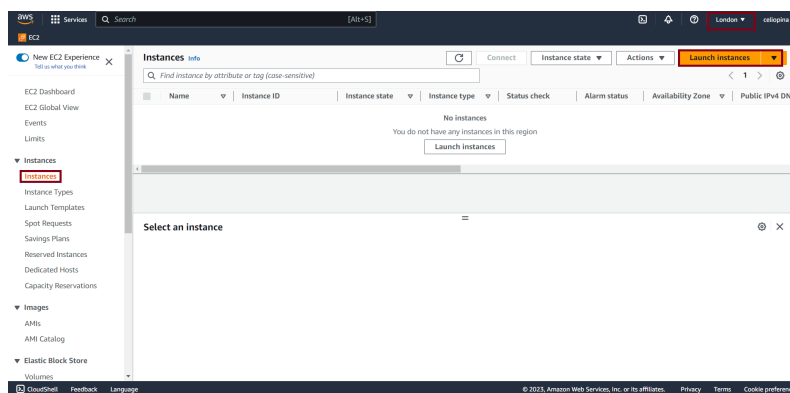
Create an account into Amazon AWS, then log in the search bar navigate to **"EC2 Console"**.

In the top right corner select the region where you want to set your honeypot,

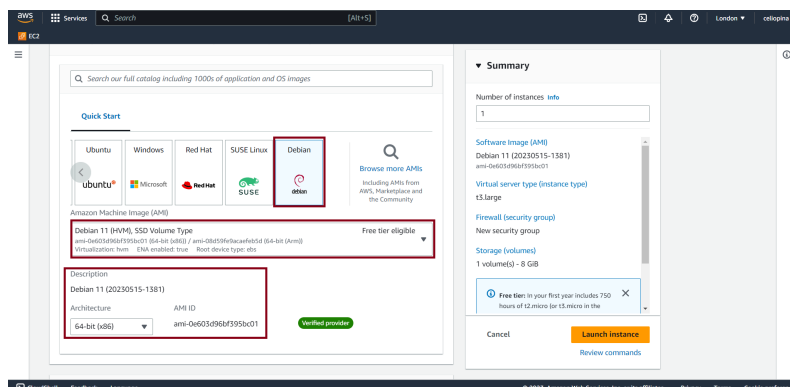


keeping in mind that where you set it up may change where the attacks come from.

With the region selected you need to launch an instance to host the honeypot, you can do that by clicking on instances on the left menu and then on **Launch Instances**.

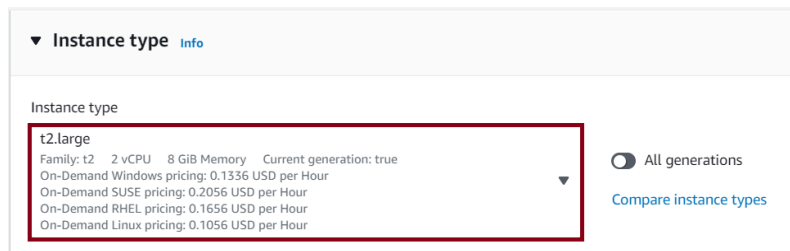


Use the **Debian 11** AMI(Amazon Machine Image) to host the honeypot(you can select others AMIs depending on your machine)

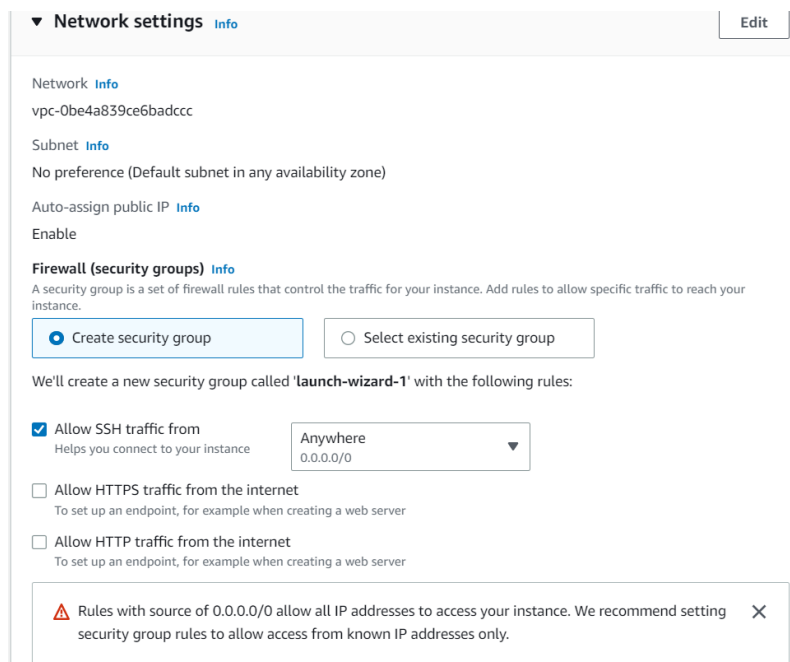


To ensure optimal performance and prevent memory limitations for the honeypot, it is crucial to select a suitable instance type with sufficient processing power and memory capacity. In this case, it is recommended to choose the t2.large instance type. This selection will help maintain the honeypot's efficiency and stability during operation.

*Please note that keeping the honeypot running will incur costs, so it is important to be mindful of this aspect.*



Deploy the instance in the default VPC provided by the region. The only configuration change required is enabling the "Auto-assign Public IP" option to ensure accessibility to potential attackers.



For storage increase the Size from 8 to 128 GiB

▼ **Configure storage** [info](#) Advanced

1x  GiB  Root volume (Not encrypted)

[Free tier eligible customers can get up to 30 GB of EBS General Purpose \(SSD\) or Magnetic storage](#) ×

[Add new volume](#)

0 x File systems [Edit](#)

Generate a key pair that will serve as the means to SSH into the instance. It is crucial to keep the key pair safe as losing it will result in the loss of access to the instance. Assign a name to the key pair, download it, and proceed to launch the instances.

**Create key pair** ×

**Key pair name**  
Key pairs allow you to connect to your instance securely.

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

☒ **RSA**  
RSA encrypted private and public key pair

☐ **ED25519**  
ED25519 encrypted private and public key pair (Not supported for Windows instances)

**Private key file format**

☒ **.pem**  
For use with OpenSSH

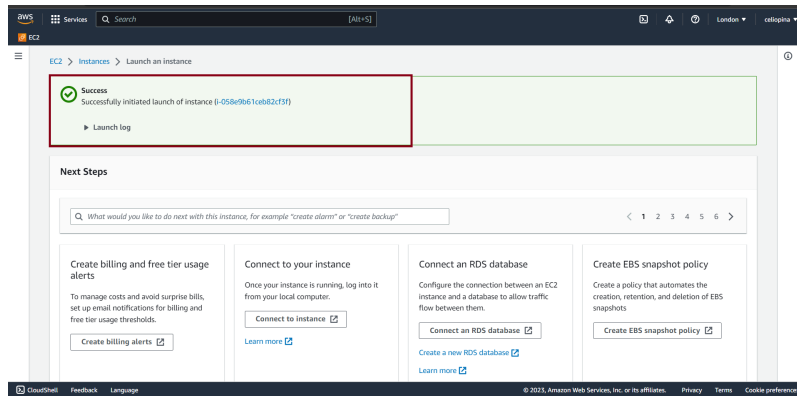
☐ **.ppk**  
For use with PuTTY

**⚠** When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

[Cancel](#)
[Create key pair](#)

After launching the instances you should receive confirmation that the instances were launched successfully.

Finally return to the EC2 console and patiently wait for the instance to initiate.



Initially, the status check will indicate "Initializing," but it is essential to wait until the status changes to "2/2 checks passed." Once the status shows "2/2 checks passed," utilize the previously created key to establish an SSH connection with the instance. To obtain the public IP address of the instance, select the instance and copy the Public IPv4 address.

Now locate the directory where you have saved the downloaded key file. In order to proceed, it is necessary to modify the permissions of the key file to make it readable.

```
chmod 400 honeypot4.pem
```

Access the instance via SSH by executing the provided command below:

```
ssh -i honeypot4.pem admin@13.37.249.179
```

Once you have successfully established an SSH connection to the instance, the next step is to ensure that the instance is up to date before proceeding with the installation of Git. Upgrading the instance is crucial to ensure compatibility and optimal performance. After the upgrade, Git can be installed, which is essential for cloning the honeypot repository from GitHub. To accomplish this, follow the sequence of commands provided below:

```
sudo apt update  
sudo apt upgrade
```

```
sudo apt install git
```

Once git is installed clone the repository using:

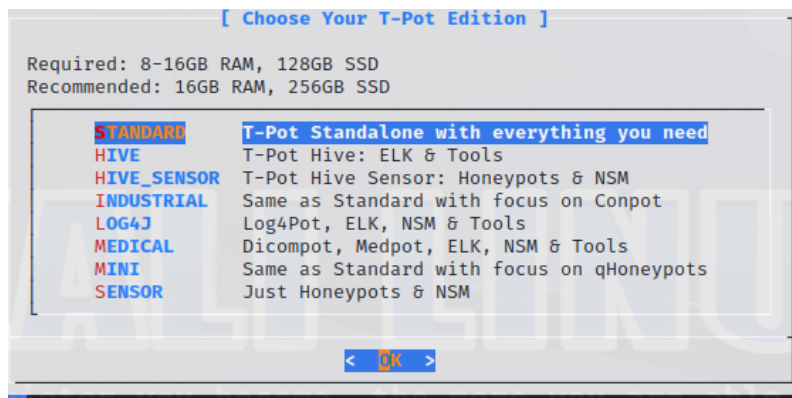
```
git clone https://github.com/telekom-security/tpotce.git
```

Go to the recently generated "tpotce" directory and execute the installation script:

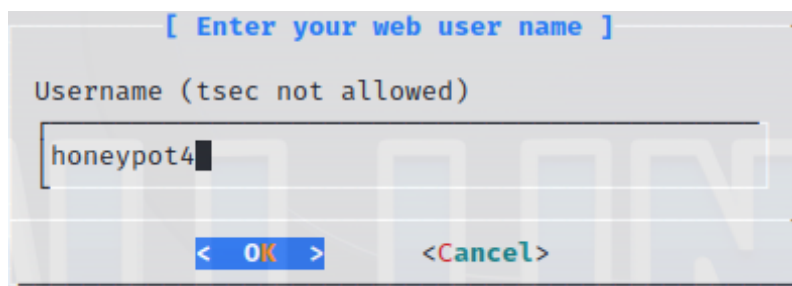
```
cd tpotce
```

```
sudo ./install.sh --type=user
```

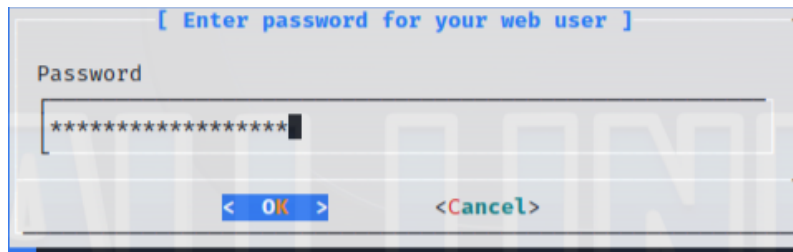
After the installation process is completed, select the Standard option and press the "Enter" key to confirm.



Create a username and a password:



The username and password credentials will grant access to the web administration portal of the honeypot. It is important to note that after this step, the SSH connection may be lost due to the installer remapping various ports, including SSH. Moving



forward, the configuration of security groups is necessary.

Return to the EC2 console, locate and select your instance, and proceed to the Security tab. From there, click on the provided hyperlink under the Security groups section and then edit inbound rules.

Remove the existing rule currently in place, and create three new rules to:

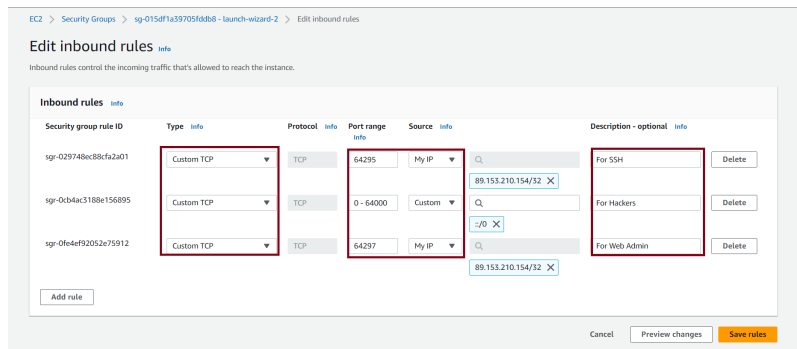
- Enable SSH access to the instance.
- Facilitate login to the web admin portal.
- Allow attackers to access ports 1-64,000.

For all three rules, select "Custom TCP" as the Type. For the first rule, enter "64295" as the Port range. The tptoc installer has changed the SSH port from 22 to 64295. Choose "My IP" from the Source drop-down menu to automatically assign your IP address, granting only you SSH access. Feel free to add a description like "For SSH" for easier identification.

To configure the rule for the web admin portal, follow a similar process as before. Click on "Add rule" and select "Custom TCP" as the Type. Enter "64297" as the Port range. Once again, choose "My IP" for the Source and, if desired, provide a description such as "Web Admin Portal" for better organization.

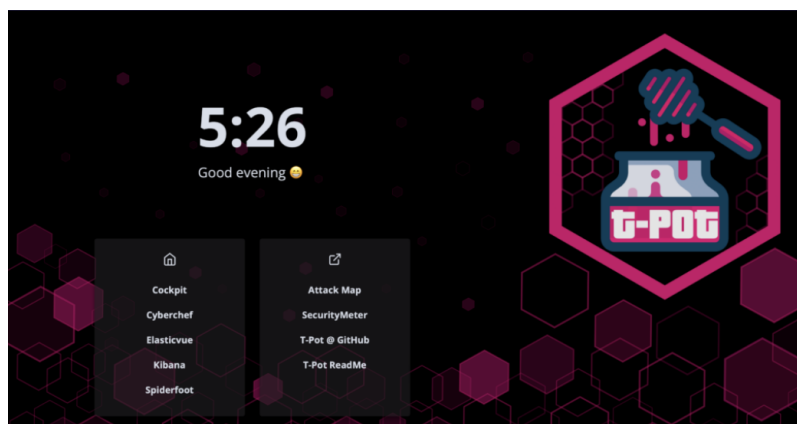
Finally, create a rule to allow all traffic. Click on the "Add rule" button. Specify the port range as "1-64000" and select "Anywhere-IPv4" as the source. Once you have entered the information, click on "Save rules" to apply the changes.





Now, you should be able to access the web admin portal through your web browser using the following URL:

`https://13.37.249.179:64297`



After successfully setting up a research honeypot, you now have the opportunity to explore the web admin portal, which provides access to various features and tools like Kibana, Elasticsearch, Spiderfoot, etc.

## 3 Creating a secure Cloud Service Environment

### 3.1 Set up a Web Server in AWS

Step 1: Create an EC2 instance

1. Access the AWS Management Console at <https://console.aws.amazon.com/>.
2. Sign in to your AWS account.
3. On the Management Console home page, click "EC2" to access the AWS Elastic Compute service.
4. Click "Launch Instance" to create a new EC2 instance.

**Iniciar uma instância** [Informações](#)

O Amazon EC2 permite criar máquinas virtuais, ou instâncias, que são executadas na Nuvem AWS. Comece a usar rapidamente seguindo as etapas simples abaixo.

**Nome e tags** [Informações](#)

Nome

Web\_Server

[Adicionar mais tags](#)

Figure 1: Add a name to the instance

5. In the first step, select the desired region for your EC2 instance.
6. In the second step, choose an Ubuntu Server image as your AMI (Amazon Machine Image).
7. Select the desired instance type based on your resource requirements and click "Next".
8. Configure instance options such as number of instances, networking, and storage options. Click "Next" to continue.



Figure 2: Choosing the Ubuntu 20.04 LTS

- Configure group security settings. Make sure to allow HTTP traffic (port 80) and, if necessary, HTTPS traffic (port 443). Click "Review and Launch".

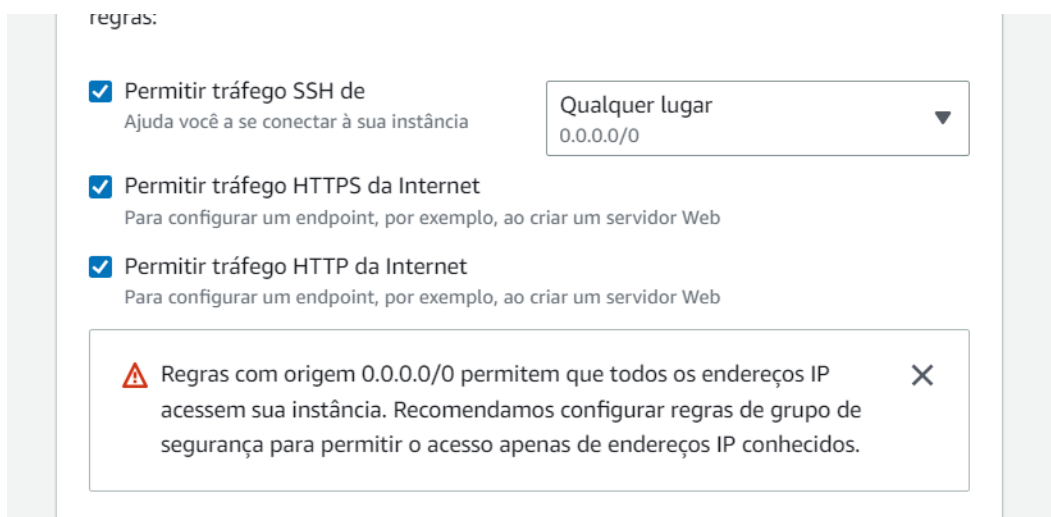


Figure 3: Allow http and https on the instance

- Review your instance settings and click "Launch".

11. In the pop-up window, select an existing key or create a new one to access the EC2 instance. Click "Launch Instances".

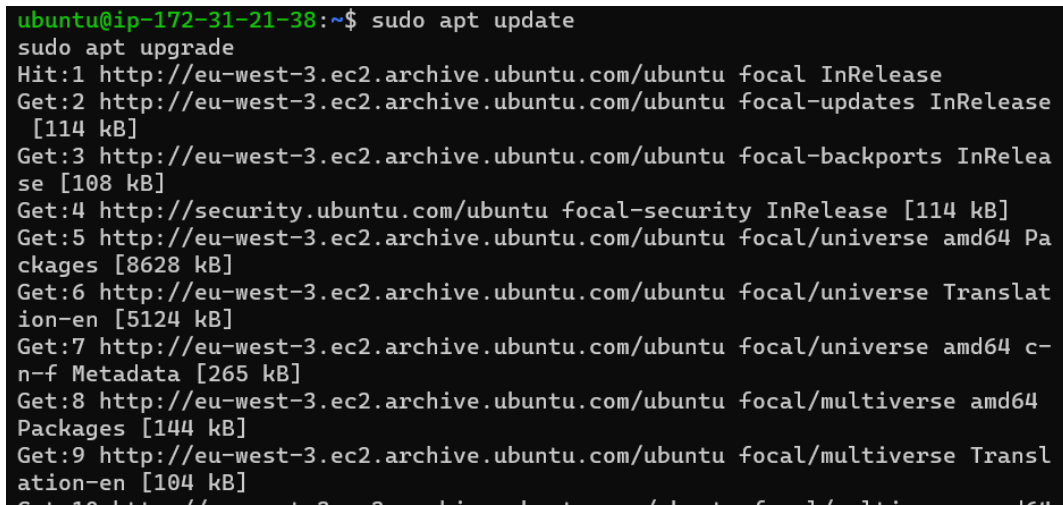
Step 2: Configure access to the EC2 instance

1. Wait for the EC2 instance to launch, then select it from the list of EC2 instances.
2. On the bottom panel, click the "Connect" button and follow the instructions to access the instance using SSH.

Step 3: Configure the web server

1. Connected to the EC2 instance, you will be at the Ubuntu Server terminal. Run the following commands to update the operating system:

```
sudo apt update
sudo apt upgrade
```



```
ubuntu@ip-172-31-21-38:~$ sudo apt update
sudo apt upgrade
Hit:1 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:7 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
```

2. Install the Apache by executing the following comand

```
sudo apt install apache2
```

```

ubuntu@ip-172-31-21-38:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0
  ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0
  ssl-cert
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 1867 kB of archives.
After this operation, 8098 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

3. After installation, you can verify that Apache is running by typing the public IP address of the EC2 instance into a web browser. You will see the default Apache page if everything is configured correctly

## 3.2 Configure company website

Step 1: Transfer the website files to the EC2 instance

1. Connect to your EC2 instance using SSH. You can use an SSH client such as PuTTY (on Windows) or Terminal (on macOS and Linux);

Step 2: Configure site file permissions

1. Navigate to the web server folder by running the command:

```
cd /var/www/html/
```

2. Remove the index.html file

```
sudo rm -r index.html
```

3. Download the files that were previously uploaded to github

```

sudo apt install git

sudo git clone https://github.com/rubendaluz/nisc.git

sudo mv nisc/Site_empresa/* /var/www/html/

sudo rm -r nisc/

```

4. Make sure the file permissions are correct. Run the following command to set the correct permissions:

```

sudo chown -R www-data:www-data *

sudo chmod -R 755 *

```

### Step 3: Access the website

1. Open a web browser and enter the public IP address of your EC2 instance. You should see the website being displayed.

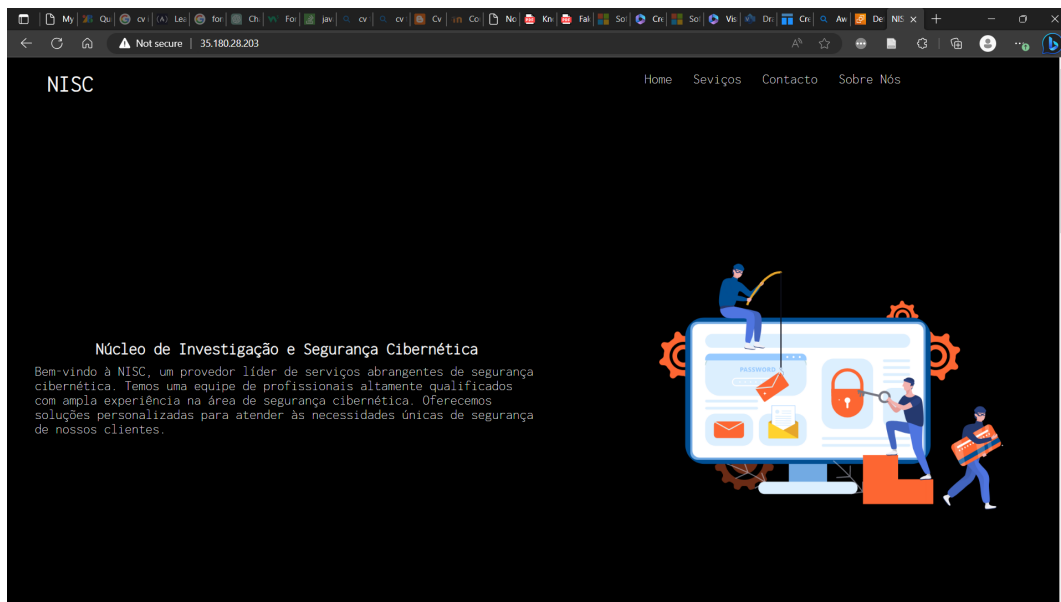


Figure 4: Web Page of the company we created

## 3.3 Implementing Security Good Practices

### 3.3.1 Create script to update software everyday

To create a Bash script that updates the operating system daily on an Ubuntu web server hosted on AWS, follow the steps below:

1. Connect to your EC2 instance using SSH.
2. Create a new script file using your text editor of choice. For example, let's call it

`update_script.sh`:

```
nano update_script.sh
```

3. Add the following content to the script file:

```
#!/bin/bash
# Atualiza o sistema operacional
sudo apt update
sudo apt upgrade -y
```

4. Pressione 'Ctrl + X' para sair do editor e, em seguida, pressione 'Y' e 'Enter' para salvar as alterações.
5. Dê permissão de execução ao arquivo de script:

```
chmod +x update_script.sh
```

6. Now, let's schedule the daily execution of the script using cron. Run the following command to edit cron jobs:

```
crontab -e
```

7. Select the default text editor (usually 'nano' will be displayed).

8. At the end of the file, add the following line to run the script every day at 23:59:

```
59 23 * * * /complete/path/to/update_script.sh
```

9. Press 'Ctrl + X' to exit the editor, then press 'Y' and 'Enter' to save the changes.

Now the update script will run automatically every day at the end of the day. The operating system will be updated without the need for manual intervention. Be sure to regularly review the script execution logs and verify that updates are being applied correctly.

Keep in mind that performing automatic operating system updates in a production environment requires care and proper testing. Be sure to make regular backups of important data and check that updates do not cause compatibility issues with other software or applications running on your web server.

### 3.3.2 Monotoring Logs

1. Install Shodan on the Ubuntu server. In the SSH terminal of the Ubuntu server, run the following commands:

```
sudo apt update
sudo apt install python3-pip
sudo pip3 install shodan
```

2. Configure the Shodan: - Create an account on the Shodan website (<https://www.shodan.io/>) if you don't already have one. - Log in to your Shodan account and get your API key. - In the Ubuntu server's SSH terminal, run the following command to configure your API key:

```
shodan init your_api_key
```



3. Create a monitoring script: - On the Ubuntu server, create a new Python file, for example 'monitorizacao.py', and edit it with your preferred text editor. - Add the following code to the 'monitorizacao.py' file to perform a query in Shodan and print the results:

```
import shodan

# Consultation in Shodan

def shodan_query():

    try:

        api = shodan.Shodan('your_api_key')

        results = api.search('your_query_here')

        for result in results['matches']:

            print(result['ip_str'])

    except shodan.APIError as e:

        print('Error: ', e)

# Run the query

shodan_query()
```

- Replace "your\_query\_here" with your desired query, such as a specific service, country or any other filter. - Save and close the file. And to create an Api Key go to <https://account.shodan.io/>, create an account if you don't have one and create your api key and replace it on the script.

4. Run the monitoring script: - In the SSH terminal of the Ubuntu server, run the following command to run the script:

```
python3 monitoring.py
```

- The script will send the query to Shodan and print the results to the terminal.