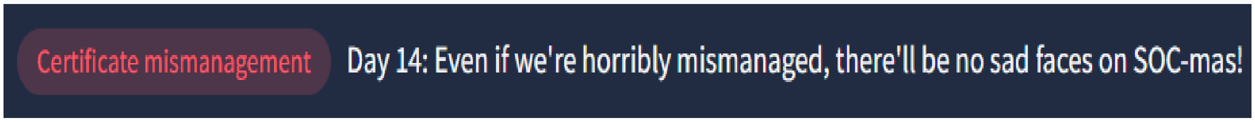


Certificate mismanagement

Day 14: Even if we're horribly mismanaged, there'll be no sad faces on SOC-mas!

24.12.2024

Nikhil Kumar

STORY

It’s a quiet morning in the town of Wareville. A wholesome town where cheer and tech come together. McSkidy is charged to protect the GiftScheduler, the service elves use to schedule all the presents to be delivered in Wareville. She assigned Glitch to the case to make sure the site is secure for G-Day (Gift Day). In the meantime, Mayor Malware works tirelessly, hoping to not only ruin SOC-mas by redirecting presents to the wrong addresses but also to ensure that Glitch is blamed for the attack. After all, Glitch’s warnings about the same vulnerabilities Mayor Malware is exploiting make the hacker an easy scapegoat.

Learning Objectives

In today's task you will learn about:

- Self-signed certificates
- Man-in-the-middle attacks
- Using Burp Suite proxy to intercept traffic

Certified to Sleigh

Public Key:

- A part of a cryptographic key pair (public and private keys).
- Public key is shared with everyone and used to **encrypt data**.

Private Key:

- Kept **secret** by the website or server.
- Used to **decrypt data** encrypted with the public key.

Metadata:

- Contains information about the certificate and its holder (e.g., website details).

- Includes:
  - **Certificate Authority (CA)**: The issuer of the certificate.
  - **Subject**: Details about the certificate holder (e.g., website like www.example.com).
  - **Unique Identifier**: A unique number to identify the certificate.
  - **Validity Period**: Start and end dates of the certificate's validity.
  - **Signature**: Proof of the certificate's authenticity from the CA.
  - **Hashing Algorithm**: Used to ensure data integrity.

Sign Here, Trust Me

1. **Certificate Authority (CA)**:
   - A trusted organization that issues digital certificates (e.g., GlobalSign, Let's Encrypt, DigiCert).
   - Browsers trust these CAs and verify their certificates as valid.
2. **Steps in Certificate Usage**:
   - **Handshake**:
     - Browser requests a secure connection.
     - Website sends its **certificate** containing the **public key** and metadata.
   - **Verification**:
     - Browser checks the certificate's validity by ensuring:
       - The CA is trusted.
       - The certificate hasn't expired.
       - The certificate hasn't been tampered with.
     - If valid, secure communication proceeds.
   - **Key Exchange**:
     - Browser encrypts a **session key** using the website's **public key**.
     - Session key is used for encrypting all communications.
   - **Decryption**:
     - Website (server) decrypts the session key using its **private key**.
     - Both browser and server now share the **session key** for secure communication.
3. **Role of Certificates in HTTPS**:
   - **Authentication**: Verifies the website is legitimate.
   - **Encryption**: Protects data during transmission.

- **Data Integrity**: Ensures the data isn't altered during transmission.

Why HTTPS is Secure:

Certificates enable authentication, encryption, and data integrity, making HTTPS communication safe and secure.

Self-Signed Certificates vs. Trusted CA Certificates

1. **Self-Signed Certificates**:
    - Created and signed by the same entity (e.g., a company).
    - Example: Wareville creates and signs a certificate for its own GiftScheduler site.
    - **Limitations**:
        - Browsers don't trust these certificates due to the lack of third-party verification.
        - Risk of misuse in **man-in-the-middle attacks**.
    - **Use Case**:
        - Best suited for **internal environments** (e.g., testing or development).
2. **Trusted CA Certificates**:
    - Issued and verified by a **Certificate Authority (CA)**.
    - The CA acts as a **trusted third party** to authenticate the website.
    - **Advantages**:
        - Trusted by browsers, ensuring secure communication.
        - Prevents interception or tampering of data during transmission.
    - **Limitations**:
        - The process can be time-consuming (especially without automation).
3. **Key Differences**:
    - **Trust Level**: Trusted CA certificates are globally trusted, while self-signed certificates are not.
    - **Verification**: CA certificates involve third-party verification; self-signed certificates do not.
    - **Purpose**: CA certificates are used for production environments, while self-signed certificates are for internal or test environments.

4. **Security Implications**:
    - Self-signed certificates should only be used in isolated environments (e.g., no Internet connection).
    - Public environments require trusted CA certificates to ensure secure and authentic communication.

How Mayor Malware Disrupts G-Day

There are less than two weeks until G-Day, and Mayor Malware has been planning its disruption ever since Glitch raised the self-signed certificate vulnerability to McSkidy during a security briefing the other day.

His plan is near perfect. He will hack into the Gift Scheduler and mess with the delivery schedule. No one will receive the gift destined for them: G-Day will be ruined! [*evil laugh*]

Preparation!!

First things first: the Glitch spoke about a self-signed certificate, but Mayor Malware can't believe that the townspeople—usually so security-savvy it's maddening to him—would easily disregard such a critical vulnerability. Is it a trap set up by the Glitch and McSkidy to catch him red-handed? He definitely needs to check for himself.

Before that, though, he wants to make sure that his tracks are well covered. To prevent any DNS logs from alerting his enemies, he will resolve the Gift Scheduler's FQDN locally on his machine.

To achieve this, let's add the following line to the `/etc/hosts` file on the AttackBox:
`MACHINE_IP gift-scheduler.thm`

We can use the following command:

Terminal

```
root@attackbox:~# echo "MACHINE_IP gift-scheduler.thm" >>
/etc/hosts
```

To verify that the line above was added to the file, we can execute the following:

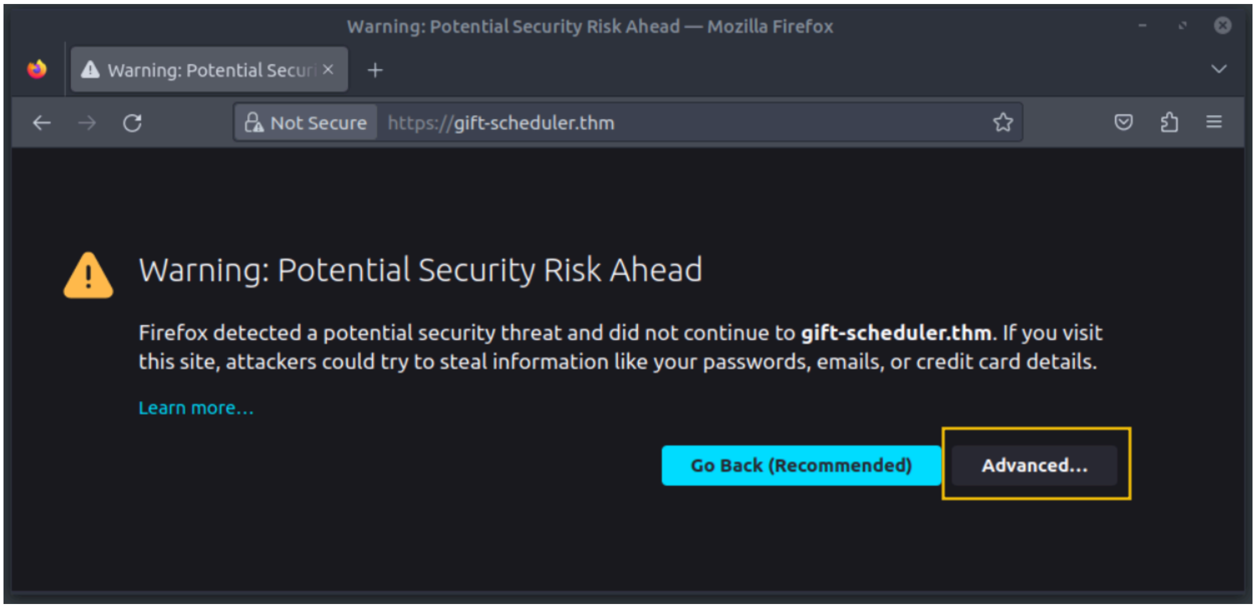
Terminal

```
root@attackbox:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       tryhackme.lan   tryhackme

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
MACHINE_IP gift-scheduler.thm
```

Now, Mayor Malware can navigate to the Gift Scheduler website without leaving a trace on Wareville's DNS logs.

Let's open the Firefox browser and navigate to `https://gift-scheduler.thm`. We'll be presented with the following warning page:

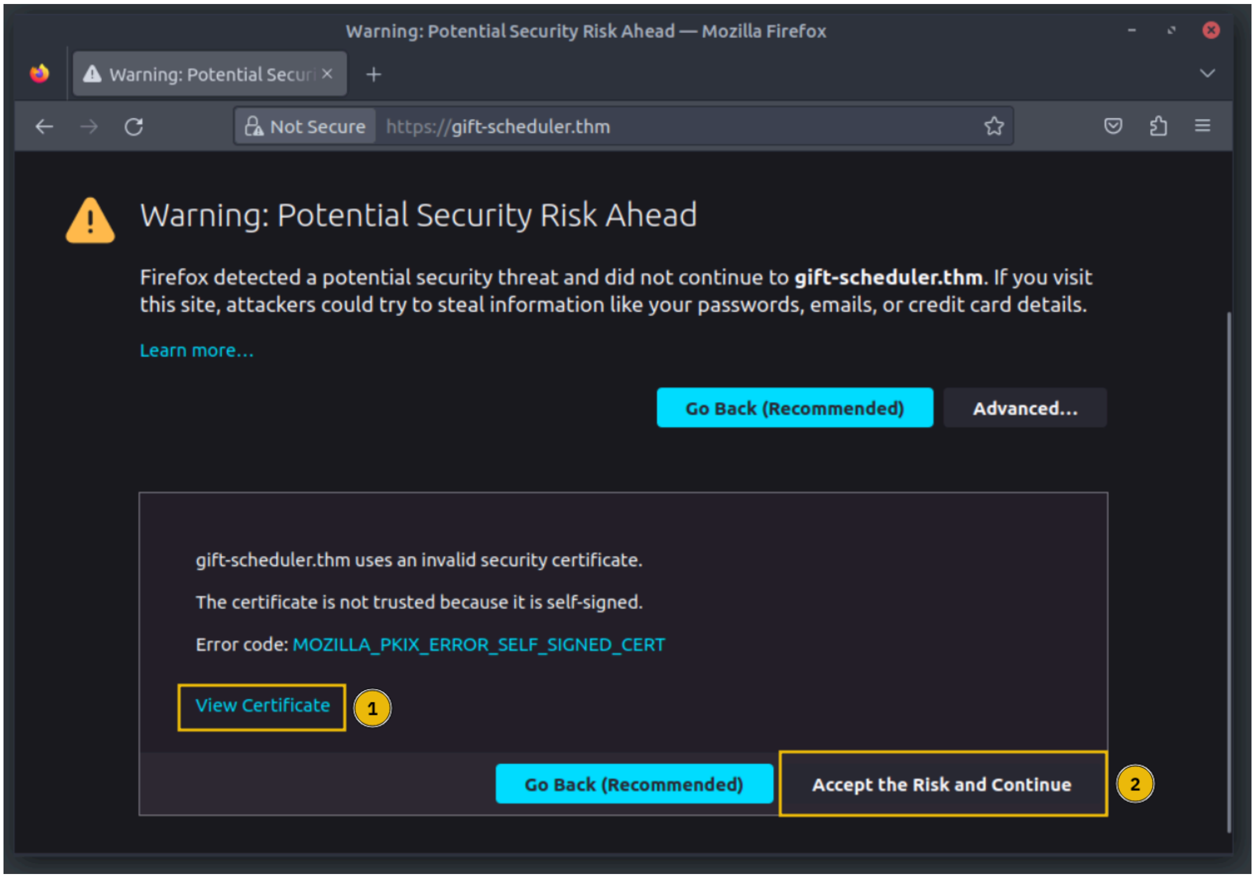


Warning: Potential Security Risk Ahead — Mozilla Firefox

Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **gift-scheduler.thm**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

We can click on the `Advanced` button to expand the warning's details.



Warning: Potential Security Risk Ahead — Mozilla Firefox
Warning: Potential Securi ×
Not Secure
https://gift-scheduler.thm

Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **gift-scheduler.thm**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

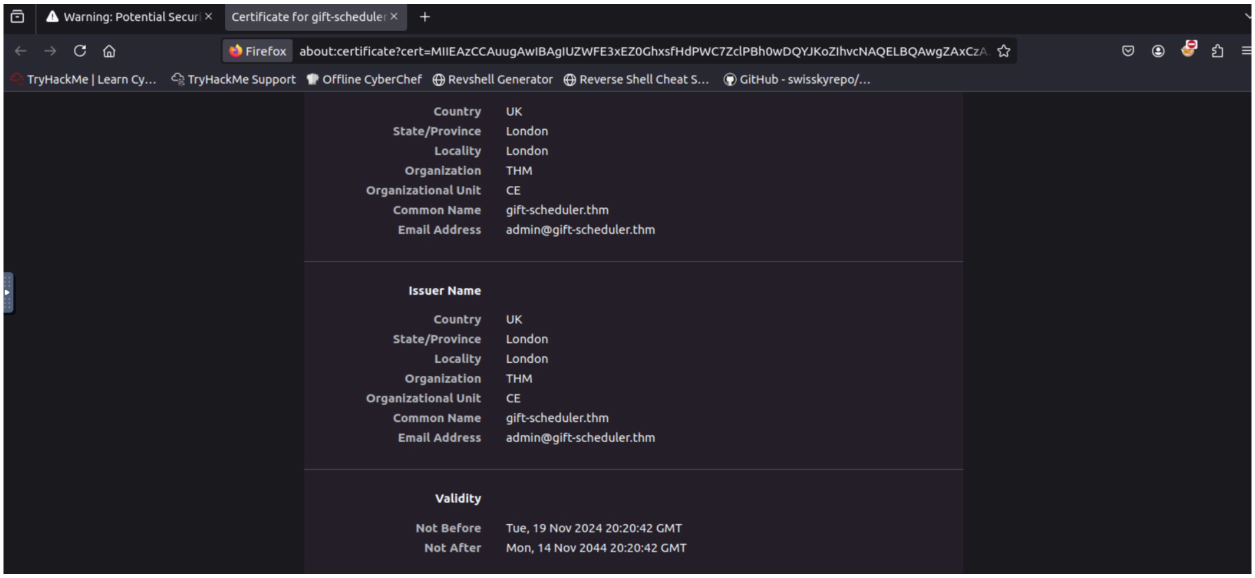
Learn more…
Go Back (Recommended)
Advanced…

gift-scheduler.thm uses an invalid security certificate.

The certificate is not trusted because it is self-signed.

Error code: MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT

When we click on the `View Certificate` link marked with a 1 in the screenshot above, a new tab opens with the certificate details.



about:certificate?cert=MIIEAzCCAuugAwIBAgIUZWFE3xEZ0GhxsfHdPWC7ZclPBh0wDQYJKoZIhvcNAQELBQAwgZAxCzA

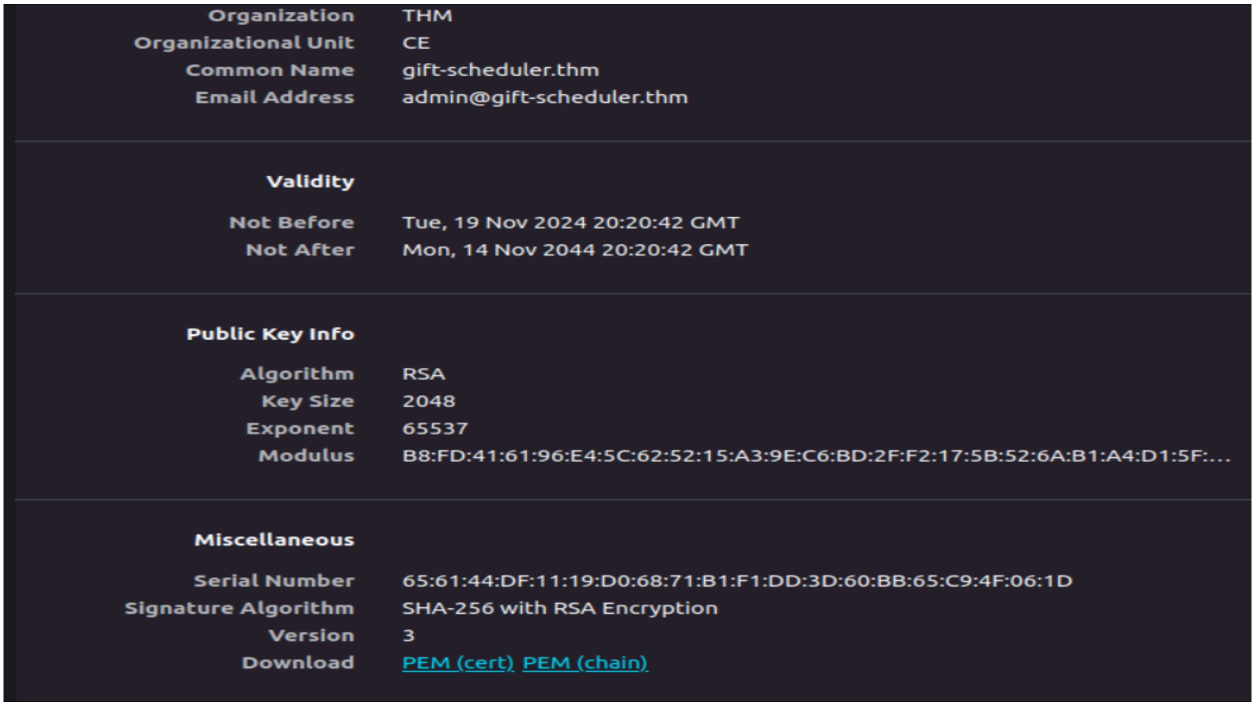
Country	UK
State/Province	London
Locality	London
Organization	THM
Organizational Unit	CE
Common Name	gift-scheduler.thm
Email Address	admin@gift-scheduler.thm

Issuer Name

Country	UK
State/Province	London
Locality	London
Organization	THM
Organizational Unit	CE
Common Name	gift-scheduler.thm
Email Address	admin@gift-scheduler.thm

Validity

Not Before	Tue, 19 Nov 2024 20:20:42 GMT
Not After	Mon, 14 Nov 2044 20:20:42 GMT



Organization	THM
Organizational Unit	CE
Common Name	gift-scheduler.thm
Email Address	admin@gift-scheduler.thm
Validity	
Not Before	Tue, 19 Nov 2024 20:20:42 GMT
Not After	Mon, 14 Nov 2044 20:20:42 GMT
Public Key Info	
Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	B8:FD:41:61:96:E4:5C:62:52:15:A3:9E:C6:BD:2F:F2:17:5B:52:6A:B1:A4:D1:5F:...
Miscellaneous	
Serial Number	65:61:44:DF:11:19:D0:68:71:B1:F1:DD:3D:60:BB:65:C9:4F:06:1D
Signature Algorithm	SHA-256 with RSA Encryption
Version	3
Download	PEM (cert) PEM (chain)

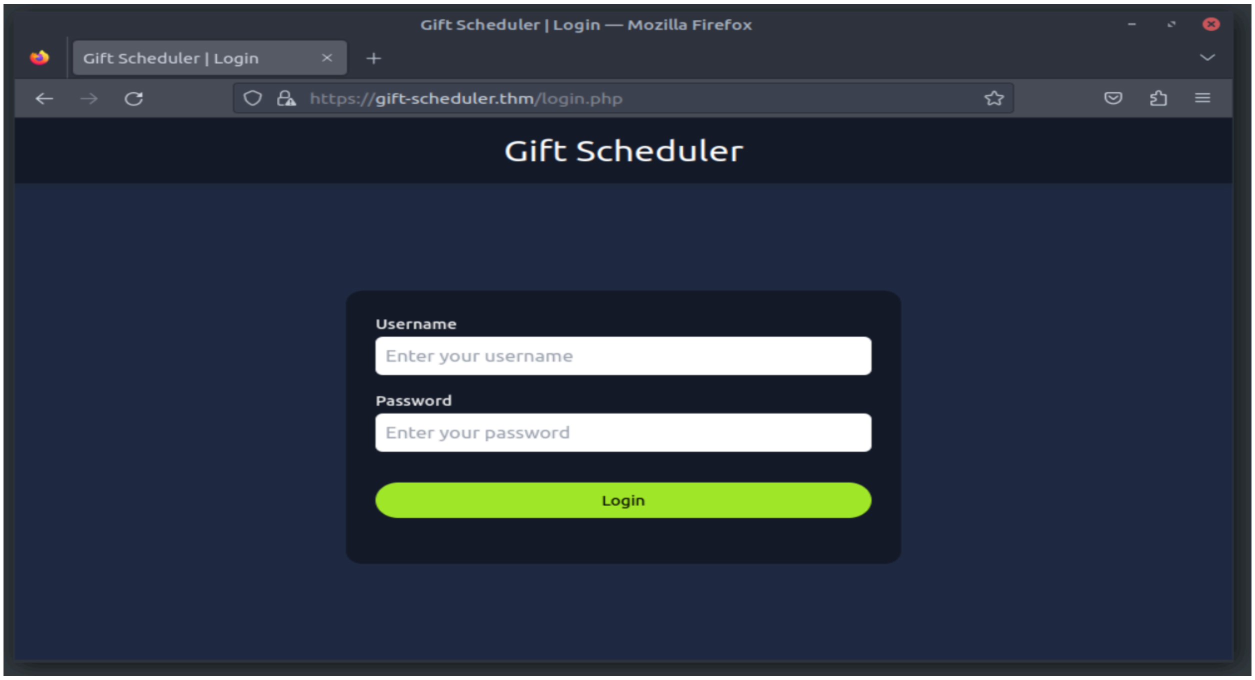
Mayor Malware can't believe his luck! This is evidence that the Glitch was speaking the truth: the Gift Scheduler web server uses a self-signed certificate.

This means that the townspeople and all the elves will be used to clicking on the `Accept the Risk and Continue` button (marked with 2 on the screenshot above) to access the website, to the point it's become a habit.

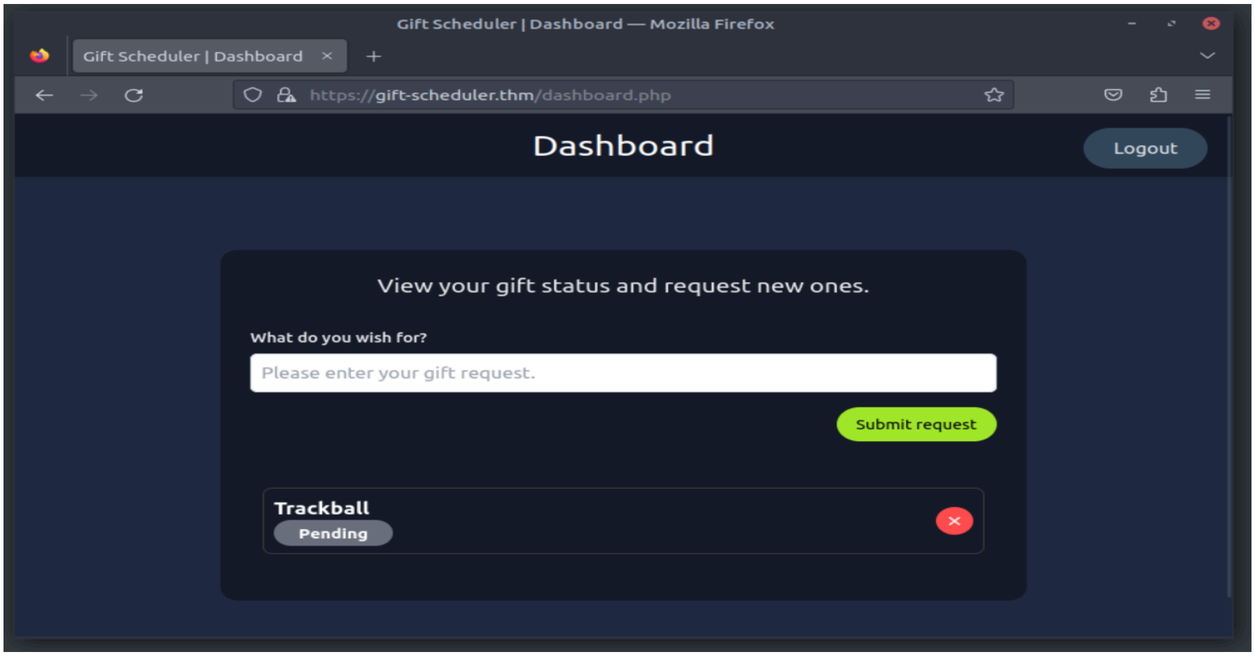
Mayor Malware does just that and inserts his credentials into the login form.

Username: mayor_malware

Password: G4rbag3Day



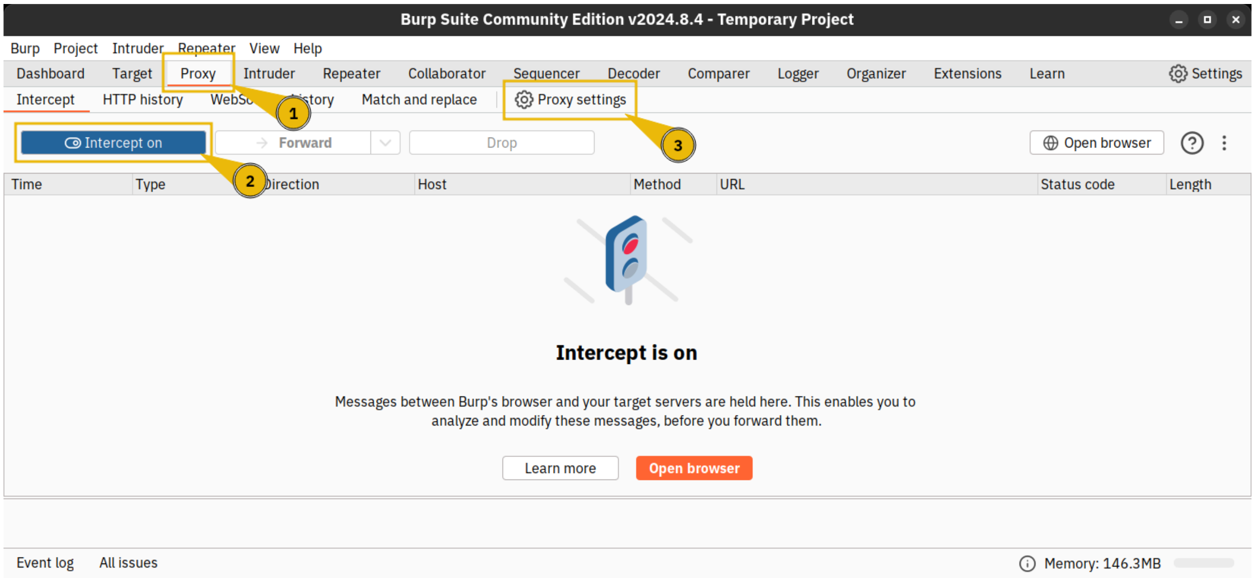
With his credentials, he can't do anything but send a gift request—as if he were to ever do such a sickeningly sweet gesture. To carry out his evil plan, he will need to sniff some admin credentials. Maybe some of the elves' passwords. Or even—if he gets lucky—Marta May Ware's account!



Gift Scheduler | Dashboard — Mozilla Firefox
Gift Scheduler | Dashboard
https://gift-scheduler.thm/dashboard.php
Dashboard
Logout
View your gift status and request new ones.
What do you wish for?
Please enter your gift request.
Submit request
Trackball
Pending

To sniff the elves' traffic, the next step will be to start a proxy on his machine and route all of Wareville's traffic to it. This way, the **Mayor** will be **In The Middle** between the townspeople and the Gift Scheduler. This position will allow him to sniff all requests forwarded to the sickening website.

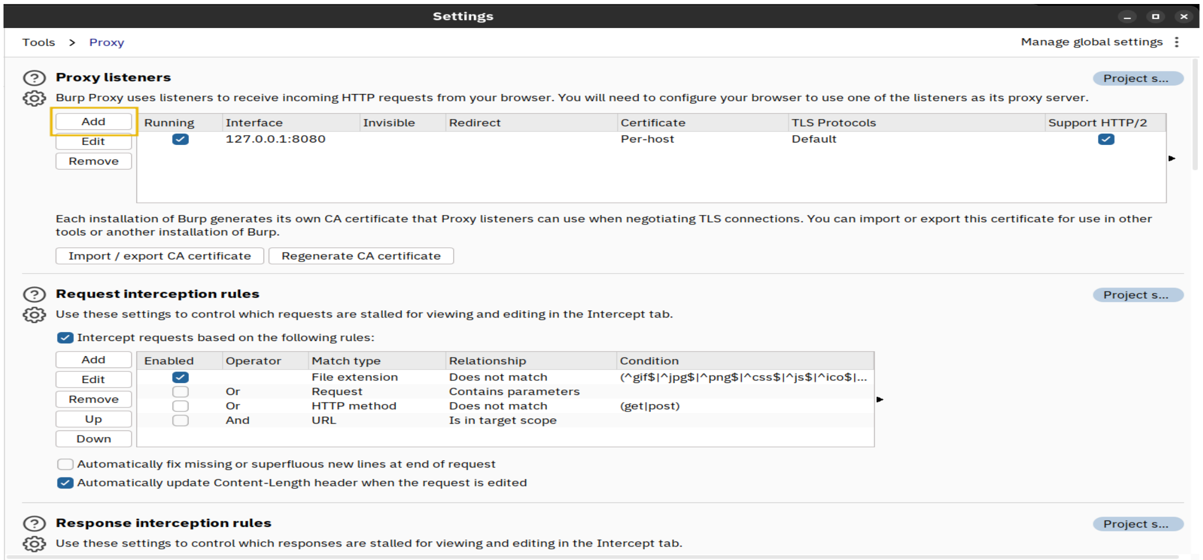
Let's start the Burp Suite proxy by typing `burp` in the terminal. A new window will open. We can accept the default configuration by clicking on `Next`, then `Start Burp` in the next window.


Burp Suite Community Edition v2024.8.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Open browser
Time Type Direction Host Method URL Status code Length
1
2
3
Learn more
Open browser
Event log All issues
Memory: 146.3MB

Intercept is on

Messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.

Once Burp Suite loads, we will select `Proxy` (number 1 in the screenshot above) and then toggle off the `Intercept on` option (number 2) to prevent users from noticing any delays in the website responses. Finally, let's open the `Proxy Settings` (number 3) to set a new listener on our AttackBox IP address.



Proxy listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols	Support HTTP/2
☑	127.0.0.1:8080			Per-host	Default	☑

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools or another installation of Burp.

Request interception rules

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☑ Intercept requests based on the following rules:

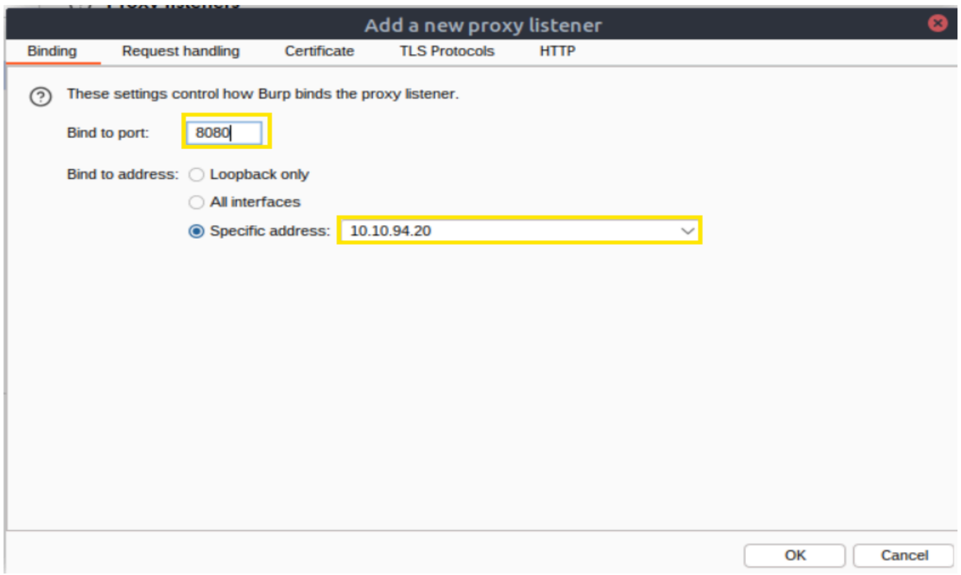
Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$|...
□	Or	Request	Contains parameters	
□	Or	HTTP method	Does not match	(get|post)
□	And	URL	Is in target scope	

□ Automatically fix missing or superfluous new lines at end of request
☑ Automatically update Content-Length header when the request is edited

Response interception rules

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

- We can click on the `Add` button highlighted in the screenshot above. Burp Suite will prompt us for the new listener's configuration.



Add a new proxy listener

These settings control how Burp binds the proxy listener.

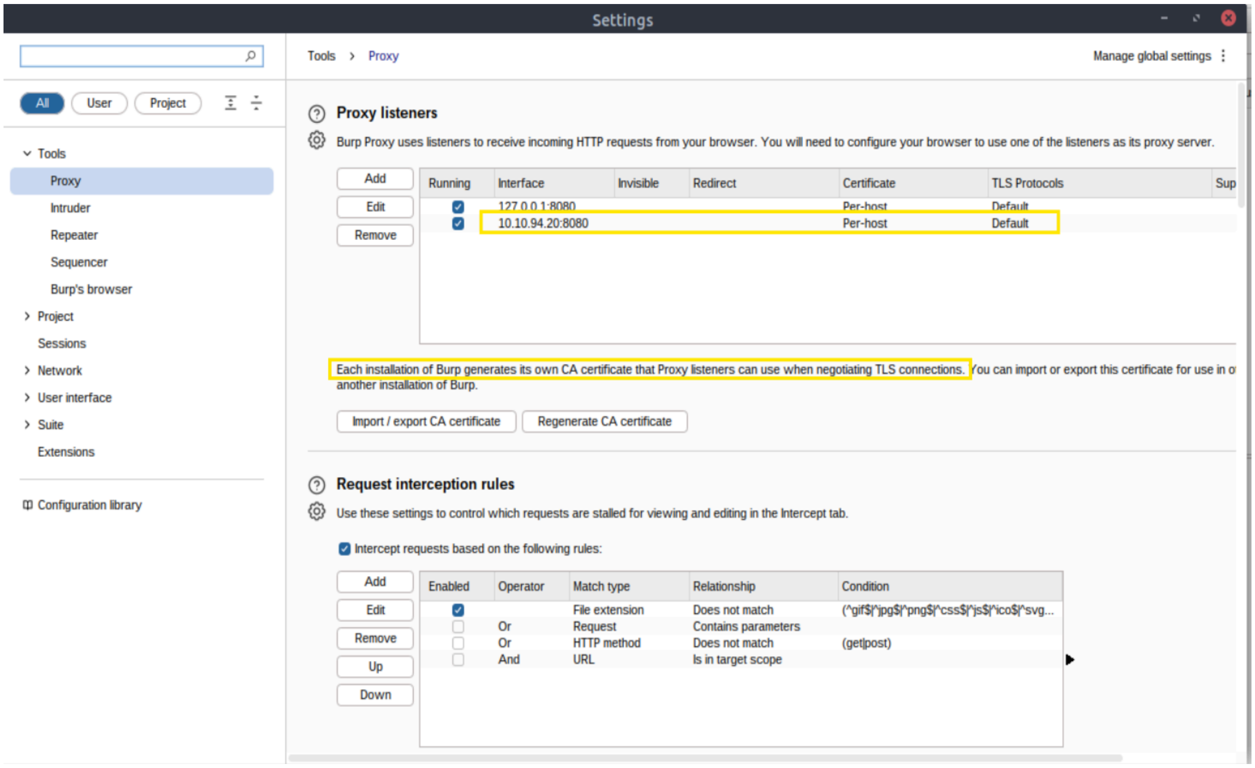
Bind to port: 8080

Bind to address: ○ Loopback only

○ All interfaces

● Specific address: 10.10.94.20

We must set the listening port to `8080` and toggle the `Specific address` option. The box next to it will automatically specify the IP address of our AttackBox, `CONNECTION_IP`. Finally, we can click on `OK` to apply the configuration.



Settings

Proxy listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols	Sup
☑	127.0.0.1:8080			Per-host	Default	
☑	10.10.94.20:8080			Per-host	Default	

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in o another installation of Burp.

Request interception rules

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☑ Intercept requests based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	`(^gif$|^jpg$|^png$|^css$|^js$|^ico$|^svg...`
□	Or	Request	Contains parameters	
□	Or	HTTP method	Does not match	(get|post)
□	And	URL	Is in target scope	

Mayor Malware rubs his hands together gleefully: as we can read in the yellow box in the screenshot above, Burp Suite already comes with a self-signed certificate. The users will be prompted to accept it and continue, and Mayor Malware knows they will do it out of habit, without even thinking of verifying the certificate origin first. The G-Day disruption operation will go off without a hitch!

Sniff From The Middle

Now that our machine is ready to listen, we must reroute all Wareville traffic to our machine.

Mayor Malware has a wonderful idea to achieve this: he will set his own machine as a gateway for all other Wareville's machines!

Let's add another line to the AttackBox's `/etc/hosts` file. **Note:** The `CONNECTION_IP` address in the snippet should reflect the IP of our AttackBox, which can be found at the top of the page.

Terminal

```
root@attackbox:~# echo "CONNECTION_IP wareville-gw" >>
/etc/hosts
```

This will divert all of Wareville's traffic, usually routed through the legitimate Wareville Gateway, to Mayor Malware's machine, effectively putting him "In The Middle" of the requests. **Note:** In practice, the adversary can launch a similar attack if they can control the user's gateway and their attack can easily succeed against websites not using properly signed certificates. This attack requires more than adding an entry into the `/etc/hosts` file; however, this task aims to emulate parts of the attack.

As a last step, we must start a custom script to simulate the users' requests to the Gift Scheduler. On the AttackBox, the script can be found in `/root/Rooms/AoC2024/Day14`. If you are using your own attacking machine connected to our VPN, you can download the script from here. Remember to run `chmod +x route-elf-traffic.sh` to make it executable.

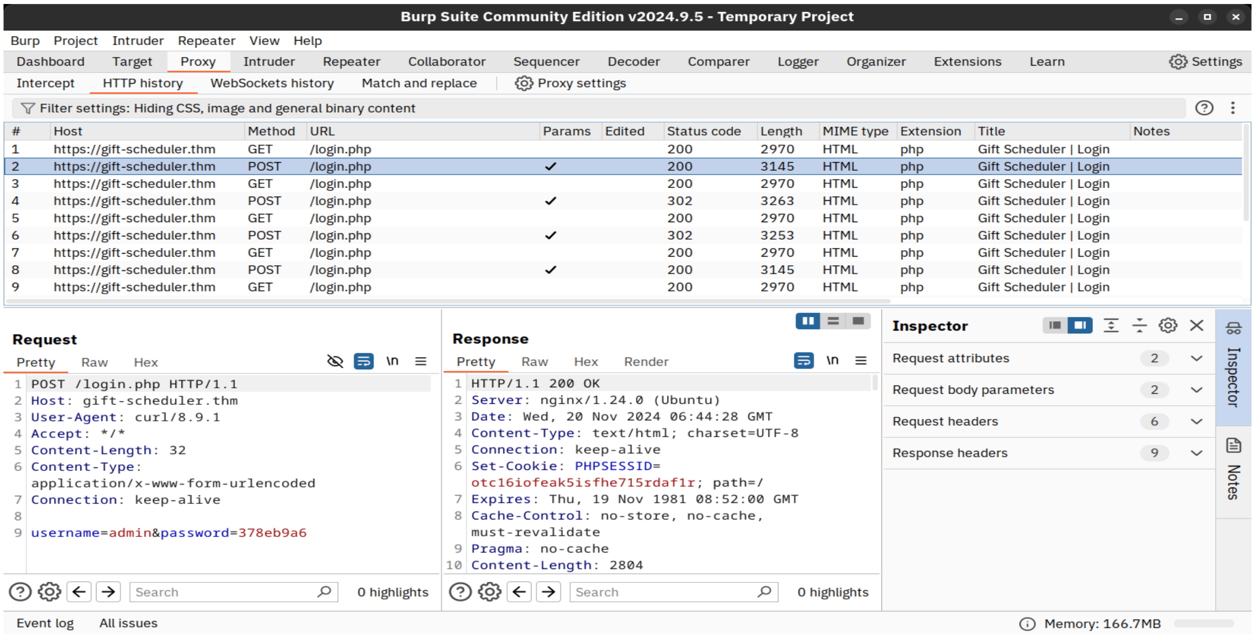
Note: Keep the script running so that new user requests will constantly be captured in Burp Suite.

Terminal

```
root@attackbox:~# cd ~/Rooms/AoC2024/Day14

root@attackbox:~/Rooms/AoC2024/Day14#
./route-elf-traffic.sh

Verifying archive integrity...  100%   MD5 checksums are
OK. All good.

Uncompressing Intercept Traffic  100%

Intercepting user traffic in progress...

 User request intercepted successfully at 2024-12-11
16:05:56

 User request intercepted successfully at 2024-12-11
16:06:23

 User request intercepted successfully at 2024-12-11
16:06:36
[...]
```

Pwn the Scheduler

We can return to the open Burp Suite window and click on the `HTTP History` tab.


Burp Suite Community Edition v2024.9.5 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Intercept HTTP history WebSockets history Match and replace Proxy settings
Filter settings: Hiding CSS, image and general binary content
Request
Pretty Raw Hex \n
Response
Pretty Raw Hex Render \n
Inspector
Request attributes 2
Request body parameters 2
Request headers 6
Response headers 9
Inspector
Notes
Search 0 highlights
Search 0 highlights
Event log All issues
Memory: 166.7MB

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	https://gift-scheduler.thm	GET	/login.php			200	2970	HTML	php	Gift Scheduler | Login	
2	https://gift-scheduler.thm	POST	/login.php	✓		200	3145	HTML	php	Gift Scheduler | Login	
3	https://gift-scheduler.thm	GET	/login.php			200	2970	HTML	php	Gift Scheduler | Login	
4	https://gift-scheduler.thm	POST	/login.php	✓		302	3263	HTML	php	Gift Scheduler | Login	
5	https://gift-scheduler.thm	GET	/login.php			200	2970	HTML	php	Gift Scheduler | Login	
6	https://gift-scheduler.thm	POST	/login.php	✓		302	3253	HTML	php	Gift Scheduler | Login	
7	https://gift-scheduler.thm	GET	/login.php			200	2970	HTML	php	Gift Scheduler | Login	
8	https://gift-scheduler.thm	POST	/login.php	✓		200	3145	HTML	php	Gift Scheduler | Login	
9	https://gift-scheduler.thm	GET	/login.php			200	2970	HTML	php	Gift Scheduler | Login	

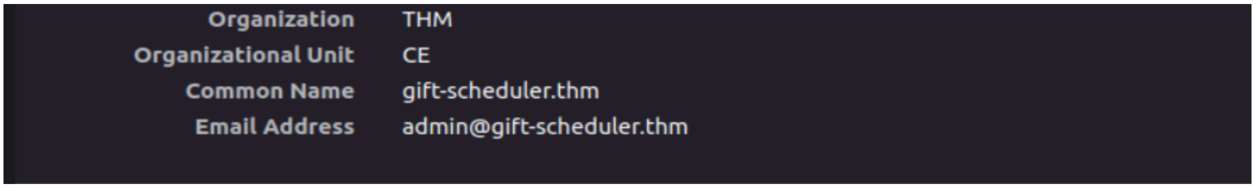
```
POST /login.php HTTP/1.1
Host: gift-scheduler.thm
User-Agent: curl/8.9.1
Accept: */*
Content-Length: 32
Content-Type:
application/x-www-form-urlencoded
Connection: keep-alive

username=admin&password=378eb9a6
```

```
HTTP/1.1 200 OK
Server: nginx/1.24.0 (Ubuntu)
Date: Wed, 20 Nov 2024 06:44:28 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Set-Cookie: PHPSESSID=
otc16iofeak5isfhe715rdaf1r; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache,
must-revalidate
Pragma: no-cache
Content-Length: 2804
```

There is a triumphant gleam in Mayor Malware's eyes while he stares intently at the web requests pouring on his screen. He can finally see them: the POST requests containing clear-text credentials for the Gift Scheduler website! Now, he only needs to wait and find the password to a privileged account.

- ***Now I need to give the following answers to complete the event.***

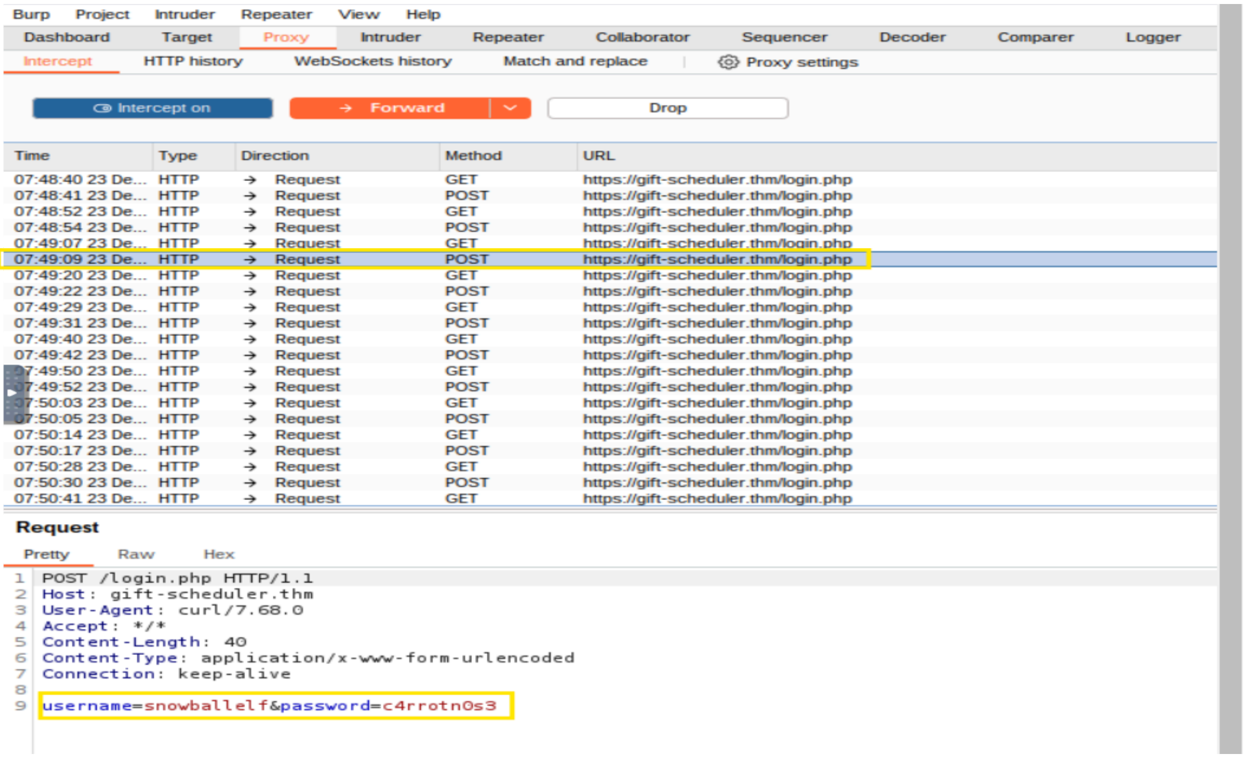
Q1. What is the name of the CA that has signed the Gift Scheduler certificate?
Correct Answer - **THM**



Organization	THM
Organizational Unit	CE
Common Name	gift-scheduler.thm
Email Address	admin@gift-scheduler.thm

- *You can see the organization in the view certificate section gives you your desired answer.*

Q.2 Look inside the POST requests in the HTTP history. What is the password for the `snowballelf` account?
Correct Answer- **c4rrotn0s3**



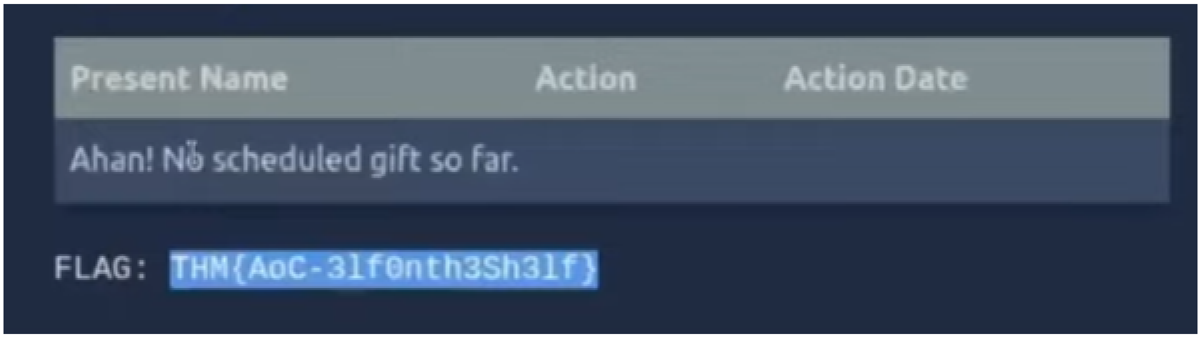
Time	Type	Direction	Method	URL
07:48:40 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:48:41 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:48:52 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:48:54 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:49:07 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:49:09 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:49:20 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:49:22 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:49:29 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:49:31 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:49:40 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:49:42 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:49:50 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:49:52 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:50:03 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:50:05 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:50:14 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:50:17 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:50:28 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php
07:50:30 23 De...	HTTP	→ Request	POST	https://gift-scheduler.thm/login.php
07:50:41 23 De...	HTTP	→ Request	GET	https://gift-scheduler.thm/login.php

Request

Pretty Raw Hex

```
POST /login.php HTTP/1.1
Host: gift-scheduler.thm
User-Agent: curl/7.68.0
Accept: */*
Content-Length: 40
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

username=snowballelf&password=c4rrotn0s3
```

Q3. Use the credentials for any of the elves to authenticate to the Gift Scheduler website. What is the flag shown on the elves' scheduling page?
Correct Answer-**THM{AoC-3lf0nth3Sh3lf}**

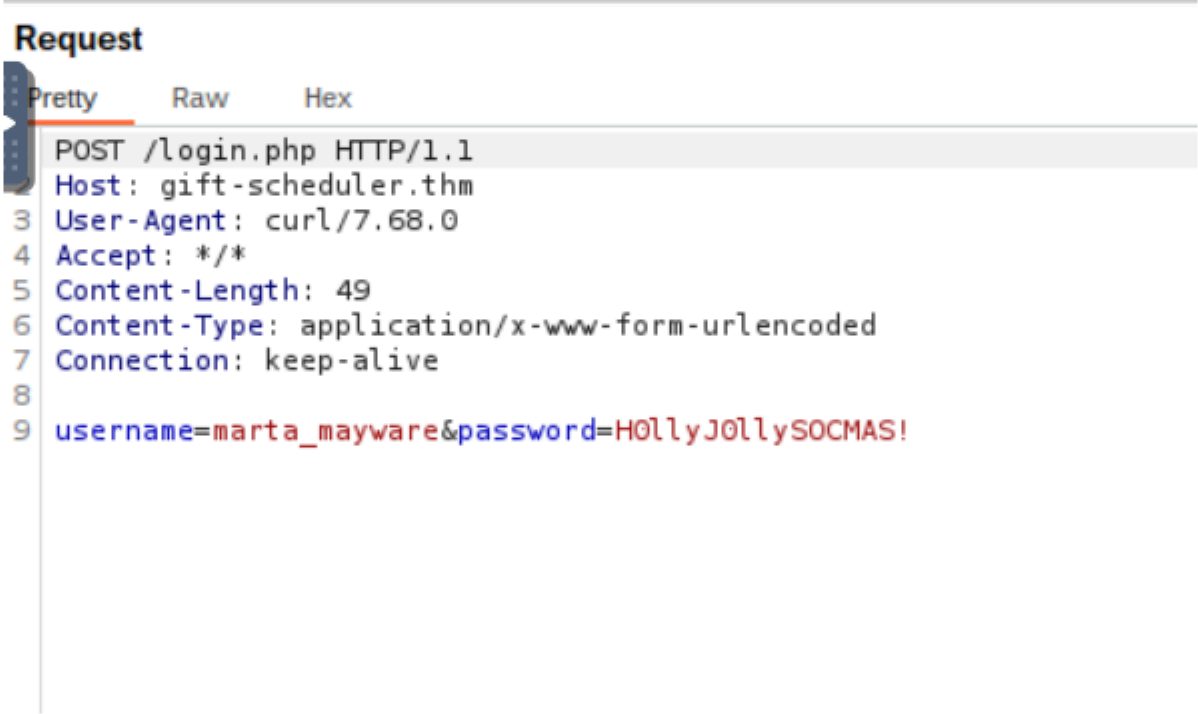


Present Name	Action	Action Date
Ahan! Nŏ scheduled gift so far.		

FLAG: THM{AoC-3lf0nth3Sh3lf}

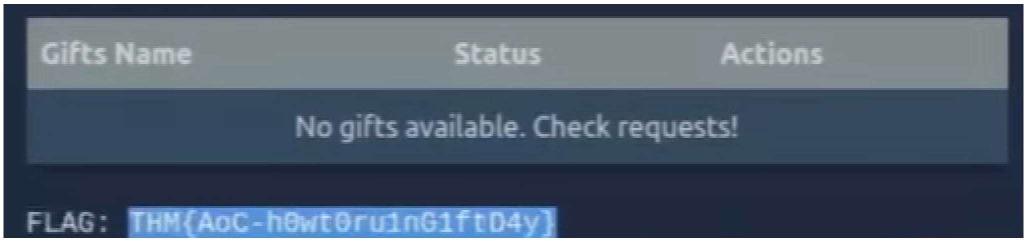
Q4. What is the password for Marta May Ware's account?
Correct Answer- **H0llyJ0llySOCMAS**!

- Just went for checking the POST requests and saw the required username of marta_mayware with intended login password.



```
POST /login.php HTTP/1.1
Host: gift-scheduler.thm
User-Agent: curl/7.68.0
Accept: */*
Content-Length: 49
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

username=marta_mayware&password=H0llyJ0llySOCMAS!
```

Q.5 Mayor Malware finally succeeded in his evil intent: with Marta May Ware's username and password, he can finally access the administrative console for the Gift Scheduler. G-Day is cancelled!
What is the flag shown on the admin page?
Correct Answer- **THM{AoC-h0wt0ru1nG1ftD4y}**



Gifts Name	Status	Actions
No gifts available. Check requests!		

FLAG: THM{AoC-h0wt0ru1nG1ftD4y}