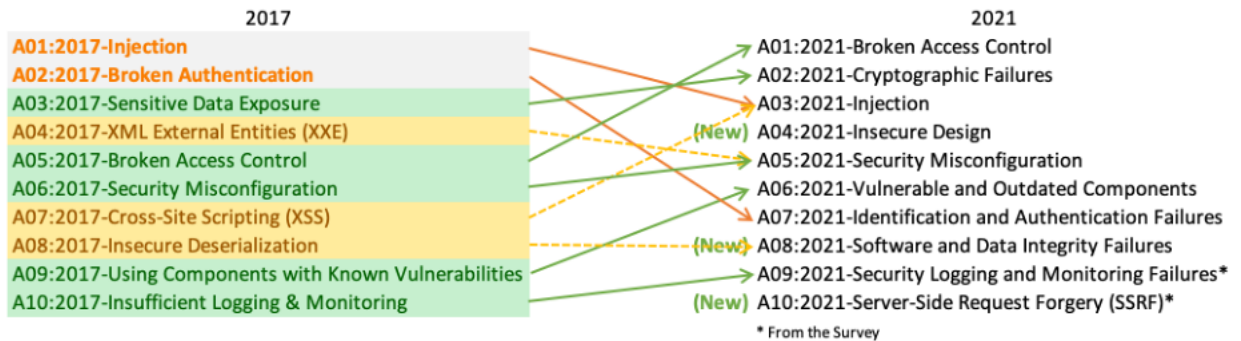


## Task 1: Introduction

- The TryHackMe OWASP Top 10 module provides a practical tour through the most common web vulnerabilities. It's a hands-on learning experience, covering everything from basic setup to advanced security threats. This course is essential for anyone keen on mastering web application security.
- Overview of the OWASP Top 10, a standard awareness document for developers and web application security.
- Importance of understanding common web vulnerabilities and threats.

<https://tryhackme.com/p/Damiano254>



## Task 2: Accessing Machines Through OpenVPN

- Setting up and using OpenVPN to access virtual machines for practical exercises.
- Significance of secure connections while accessing remote systems.

## Task 3: [Severity 1] Injection

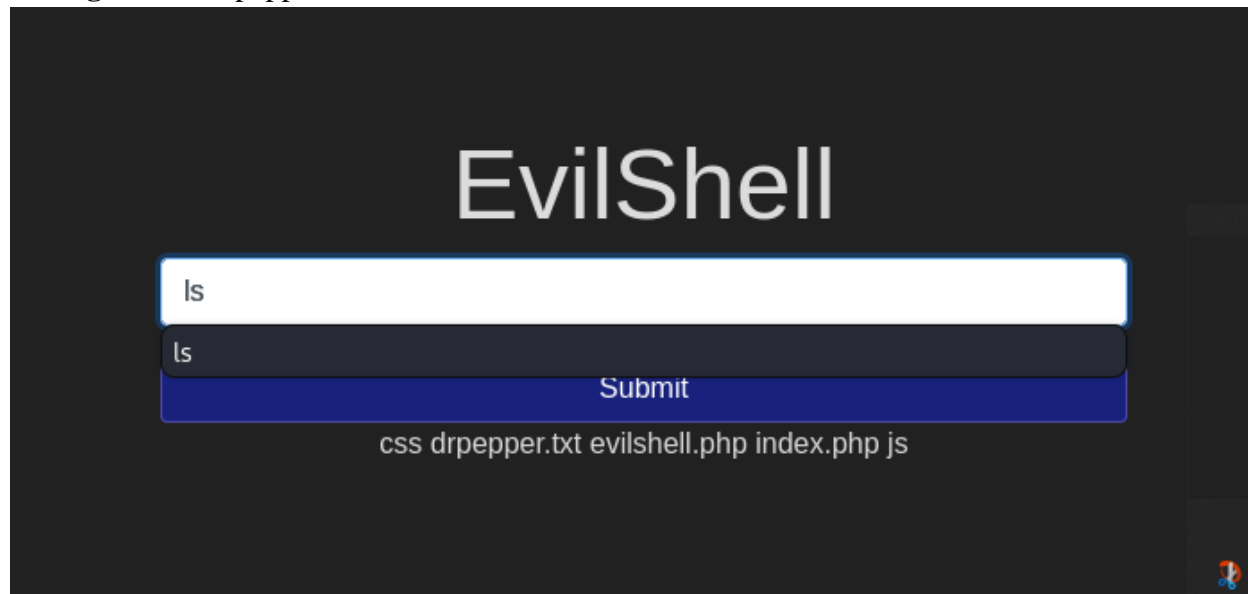
- Understanding injection flaws, especially SQL, NoSQL, OS, and LDAP injection.
- Attackers can use these flaws to execute unauthorized commands or access data.
- Example: SQL injection where malicious SQL statements are inserted into an entry field.

## Task 4: [Severity 1] OS Command Injection

- Understanding how attackers execute arbitrary commands on the host operating system.
- This vulnerability arises due to insufficient input validation.
- Example: Web application calls a system command with user input, which is exploitable.

## Task 5: [Severity 1] Command Injection Practical

- Practical exploration of command injection.
- Identifying and exploiting command injection vulnerabilities.
- Example Answers:
  - **Strange file:** Adrpepper.txt



- **Non-root users:** 0

# EvilShell

```

root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin
/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin
/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr
/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin
/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var
/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin
/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-
Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-
network:x:100:102:systemd Network Management,,,:/run/systemd/netif:
/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver /run

```

- 
- **App running user:** www-data

# EvilShell

www-data

- **User's shell:** /usr/sbin/nologin

# EvilShell

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

- **Ubuntu version:** 18.04.4

# EvilShell

Static hostname: injection Icon name: computer-vm Chassis: vm Machine  
ID: 4851ceb47275439fb91a3e9f1cf7068f Boot ID:  
aa1dfad3e42a4440878526ad82562366 Virtualization: xen Operating  
System: Ubuntu 18.04.4 LTS Kernel: Linux 4.15.0-101-generic Architecture:  
x86-64

- **MOTD beverage:** Dr Pepper

```
cat /etc/update-motd.d/00-header
```

Submit

```
#!/bin/sh # # 00-header - create the header of the MOTD # Copyright (C)
2009-2010 Canonical Ltd. # # Authors: Dustin Kirkland # # This program is
free software; you can redistribute it and/or modify # it under the terms of
the GNU General Public License as published by # the Free Software
Foundation; either version 2 of the License, or # (at your option) any later
version. # # This program is distributed in the hope that it will be useful, #
but WITHOUT ANY WARRANTY; without even the implied warranty of #
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the # GNU General Public License for more details. # # You should have
received a copy of the GNU General Public License along # with this
program; if not, write to the Free Software Foundation, Inc., # 51 Franklin
Street, Fifth Floor, Boston, MA 02110-1301 USA. [ -r /etc/lsb-release ] && .
/etc/lsb-release if [ -z "$DISTRIB_DESCRIPTION" ] && [ -x /usr/bin
/lsb_release ]; then # Fall back to using the very slow lsb_release utility
DISTRIB_DESCRIPTION=$(lsb_release -s -d) fi printf "Welcome to %s (%s
%s %s)\n" "$DISTRIB_DESCRIPTION" "$(uname -o)" "$(uname -r)"
"$(uname -m)" DR PEPPER MAKES THE WORLD TASTE BETTER!
```

#### Task 6: [Severity 2] Broken Authentication

- Exploring issues related to authentication mechanisms.
- Vulnerabilities can allow attackers to compromise passwords, keys, or session tokens.

#### Task 7: [Severity 2] Broken Authentication Practical

- Practical example: exploiting broken authentication.
- Finding flags in user accounts by bypassing authentication.
- **Example Answers:**
  - **Darren's flag:** fe86079416a21a3c99937fea8874b667

Username:

darren

Email:

try@fakeemail.com

Password:

••••

Register

Error: This user is already registered

Username:

darren

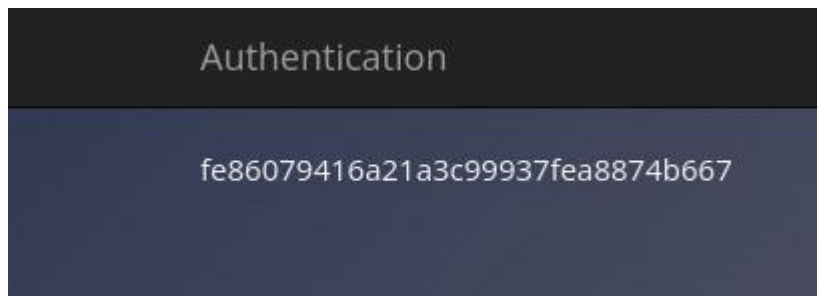
Email:

email@fake.com

Password:

••••

Register



- **Arthur's flag:** d9ac0f7db4fda460ac3edeb75d75e16e

A screenshot of a web application's registration page. The page has a dark background with the word "Register" in large white letters at the top. Below it is a form with three input fields: "Username:" containing "arthur", "Email:" containing "me2@fake.com", and "Password:" containing four black dots. A large teal "Register" button is below the form. At the bottom, there is a dark footer bar. On the left of the footer is the word "Authentication". On the right is the text "Logged in as arthur | Log Out". Below the footer bar, on a dark blue background, is the alphanumeric string "d9ac0f7db4fda460ac3edeb75d75e16e".








### Task 8-11: [Severity 3] Sensitive Data Exposure

- Understanding the risks of exposing sensitive data.
- Inadequate protection of sensitive data like financial, healthcare, or PII.
- Practical challenge: identifying and accessing sensitive data.
- **Example Answers:**
  - **Sensitive directory:** /assets

```
view-source:http://10.10.54.16/login/?response=failure
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Tryhackme Attacktive ...
3 <html>
4 <head>
5 <title>Login</title>
6 <meta name="viewport" content="width=device-width, user-scalable=no">
7 <meta charset="utf-8">
8 <link rel="shortcut icon" type="image/x-icon" href=".../favicon.ico">
9 <link type="text/css" rel="stylesheet" href=".../assets/css/style.css">
10 <link type="text/css" rel="stylesheet" href=".../assets/css/loginStyle.css">
11 <link type="text/css" rel="stylesheet" href=".../assets/css/orkney.css">
12 <link type="text/css" rel="stylesheet" href=".../assets/css/icons.css">
13 <script src=".../assets/js/jquery-3.5.1.min.js"></script>
14 <script src=".../assets/js/loginScript.js"></script>
15 </head>
16 <body>
17 <header>
18 <a id="home" href="/">Sense and Sensitivity</a>
19 <a id="login" href="/login">Login</a>
20 </header>
21 <div class="background"></div>
22 <!-- Must remember to do something better with the database than store it in /assets... -->
23 <main>
24 <div class="content">
25 <form method="POST" action="/api/login">
26 <input type="text" name="username" placeholder="Username"><br>
27 <input type="password" name="password" placeholder="Password"><br>
28 <input id="loginBtnFunc" type="submit" value="Login!">
29 </form>
30 <i id="loginBtnStyle" class="material-icons">arrow_forward</i>
31 <p class="responseMsg" id="errorMsg">Invalid Credentials, please try again</p>
32 </div>
33 </main>
34 <footer><span>©copy; Sense and Sensitivity, 2020</span></footer>
35 </body>
36 </html>
37
```

- Notable file: webapp.db

# Index of /assets

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">css/</a>	2020-07-14 17:52	-	
 <a href="#">fonts/</a>	2020-07-14 15:42	-	
 <a href="#">images/</a>	2020-07-14 15:42	-	
 <a href="#">js/</a>	2020-07-14 15:52	-	
 <a href="#">php/</a>	2020-07-14 15:42	-	
 <a href="#">webapp.db</a>	2020-07-14 17:52	28K	

Apache/2.4.29 (Ubuntu) Server at 10.10.54.16 Port 80

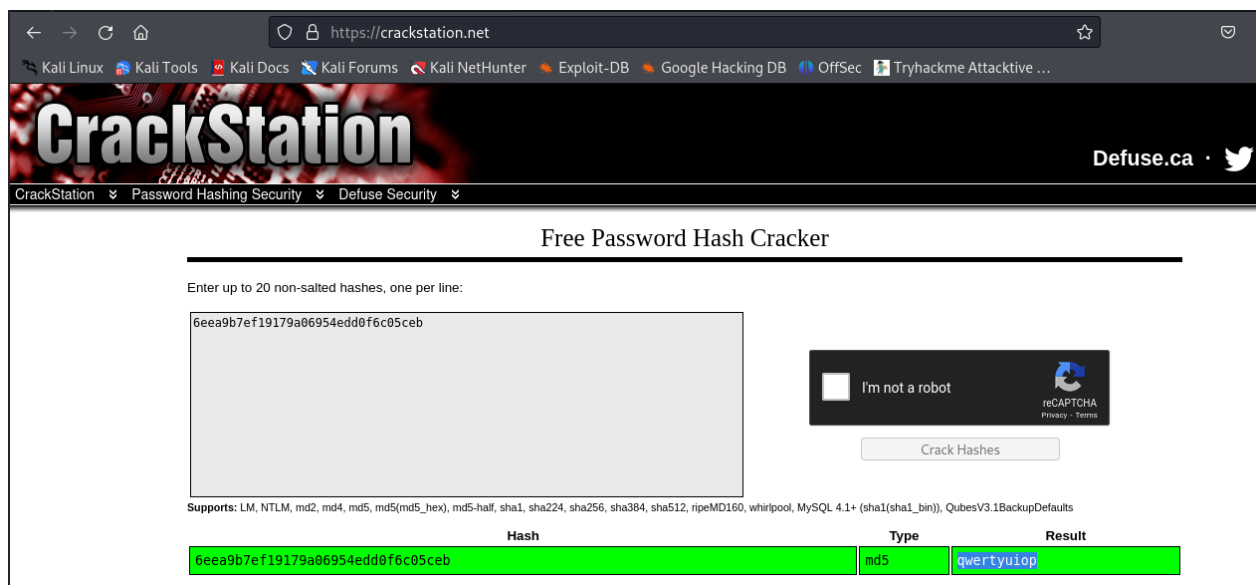
- Admin password hash: 6eea9b7ef19179a06954edd0f6c05ceb

```
kali (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~/Downloads
File Actions Edit View Help
kali@kali: ~/Downloads x kali@kali: ~ x kali@kali: ~/Downloads x
(kali@kali)~/Downloads
$ ls
'academy-regular(1).ovpn' 'academy-regular(3).ovpn' 'Damiano254(1).ovpn' 'Damiano254(3).ovpn' 'Nessus-10.7.0-ubuntu1404_amd64.deb' 'webapp(1).db'
'academy-regular(2).ovpn' 'academy-regular.ovpn' 'Damiano254(2).ovpn' 'Damiano254.ovpn' 'Starting_point_Damiano254.ovpn' 'webapp.db'
(kali@kali)~/Downloads
$ file webapp.db
webapp.db: SQLite 3.x database, last written using SQLite version 3022000, file counter 255, database pages 7, 1st free page 5, free pages 1, cookie 0x6, schema 4, UTF-8, version-valid-for 255
(kali@kali)~/Downloads
$ sqlite3 webapp.db
Command 'sqlite3' not found, did you mean:
command 'sqlite3' from deb sqlite3
Try: sudo apt install <deb name>
(kali@kali)~/Downloads
$ sudo apt install sqlite3
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package sqlite3
(kali@kali)~/Downloads
$ sqlite3 webapp.db
SQLite version 3.44.2 2023-11-24 11:41:44
Enter ".help" for usage hints.
sqlite> tables
...> .tables
...> exit
...> clear
...> ^Z
zsh: suspended sqlite3 webapp.db
```

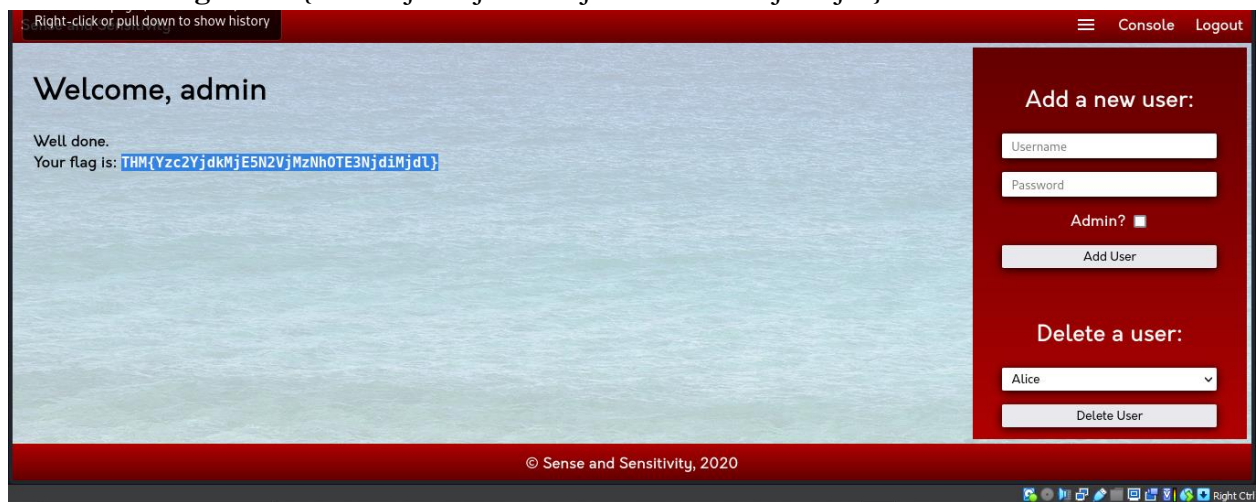
```
kali (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~/Downloads
File Actions Edit View Help
kali@kali: ~/Downloads x kali@kali: ~ x kali@kali: ~/Downloads x
(kali@kali)~/Downloads
$ sudo apt install sqlite3
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package sqlite3
(kali@kali)~/Downloads
$ sqlite3 webapp.db
SQLite version 3.44.2 2023-11-24 11:41:44
Enter ".help" for usage hints.
sqlite> tables
...> .tables
...> exit
...> clear
...> ^Z
zsh: suspended sqlite3 webapp.db
(kali@kali)~/Downloads
$ sqlite3 webapp.db
SQLite version 3.44.2 2023-11-24 11:41:44
Enter ".help" for usage hints.
sqlite> tables
...> .tables
...> exit
...> clear
...> ^Z
zsh: suspended sqlite3 webapp.db
(kali@kali)~/Downloads
$ sqlite3 webapp.db
SQLite version 3.44.2 2023-11-24 11:41:44
Enter ".help" for usage hints.
sqlite> PRAGMA table_info(users);
0|userID|TEXT|1|0
1|username|TEXT|1|0
2|password|TEXT|1|0
3|admin|INT|1|0
sqlite> SELECT * FROM users;
4413096d0c93359b898b6202288a650|admin|6ee9b7ef19179a06954edd0f6c05ceb11
23023b67a32488588db1e28579ced7ec|Bob|ad0234820205b903196ba818f7a872b11
4e8423b514ee575394ff78caed3254d|Alice|268b38ca7b84f44fa0a6cdc86e6301e010
```

- Admin's plaintext password: qwertyuiop





- **Admin flag:** THM{Yzc2YjdkMjE5N2VjMzNhOTE3NjdiMjdl}



## Task 12-16: [Severity 4] XML External Entity (XXE)

- Understanding XXE attacks where applications process XML input maliciously.
- Full form of XML: Extensible Markup Language.

What is XML?

XML ([eXtensible Markup Language](#)) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is a markup language used for storing and transporting data.

- XML prolog is not mandatory in documents.

Above the line is called XML prolog and it specifies the XML version and the encoding used in the XML document. [This line is not compulsory to use but it is considered a 'good practice' to put that line in all your XML documents.](#)

- Validation against a schema and specifying XML version and encoding in the prolog.

Every XML document mostly starts with what is known as XML Prolog.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Above the line is called XML prolog and it specifies the XML version and the encoding used in the XML document. This line is not compulsory to use but it is considered a 'good practice' to put that line in all your XML documents.

- **Understanding DTD in XML**
- DTD stands for Document Type Definition and is crucial for defining how an XML document should be structured.
- It specifies the legal elements and attributes in an XML document
- **Validating XML Document with DTD**
- The XML document provided uses **note.dtd** to ensure it adheres to the defined structure.

- The DTD checks if the XML document contains the specified elements in the correct format and order.
- **Answers to Your Questions**
- **How do you define a new ELEMENT?**  
!ELEMENT

In a DTD, elements are declared with an ELEMENT declaration.

## Declaring Elements

In a DTD, XML elements are declared with the following syntax:

```
<!ELEMENT element-name category>
```

- **How do you define a ROOT element?**

!DOCTYPE

So now let's understand how that DTD validates the XML. Here's what all those terms used in `note.dtd` mean

- !DOCTYPE note - Defines a root element of the document named note
- !ELEMENT note - Defines that the note element must contain the elements: "to, from, heading, body"
- !ELEMENT to - Defines the to element to be of type "#PCDATA"
- !ELEMENT from - Defines the from element to be of type "#PCDATA"
- !ELEMENT heading - Defines the heading element to be of type "#PCDATA"
- !ELEMENT body - Defines the body element to be of type "#PCDATA"

NOTE: #PCDATA means parseable character data.

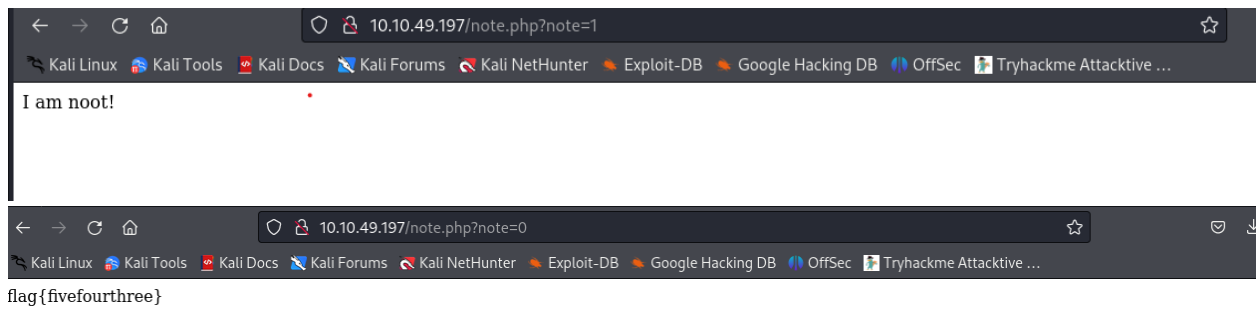
- **How do you define a new ENTITY?**

!ENTITY

The screenshot shows a web application interface for learning XML. On the left, there is a sidebar menu with categories like 'XQuery Syntax', 'XML DTD', 'XSD Schema', and 'XSD How To'. The 'XML DTD' section is expanded, and 'DTD Entities' is selected. The main content area is titled 'An Internal Entity Declaration' and shows the syntax for an internal entity declaration: `<!ENTITY entity-name "entity-value">`. Below the syntax, there is an 'Example' section showing a DTD example: `<!ENTITY writer "Donald Duck.">`. The top of the page has a navigation bar with links to 'Tutorials', 'Exercises', 'Certificates', 'Services', and a search bar.

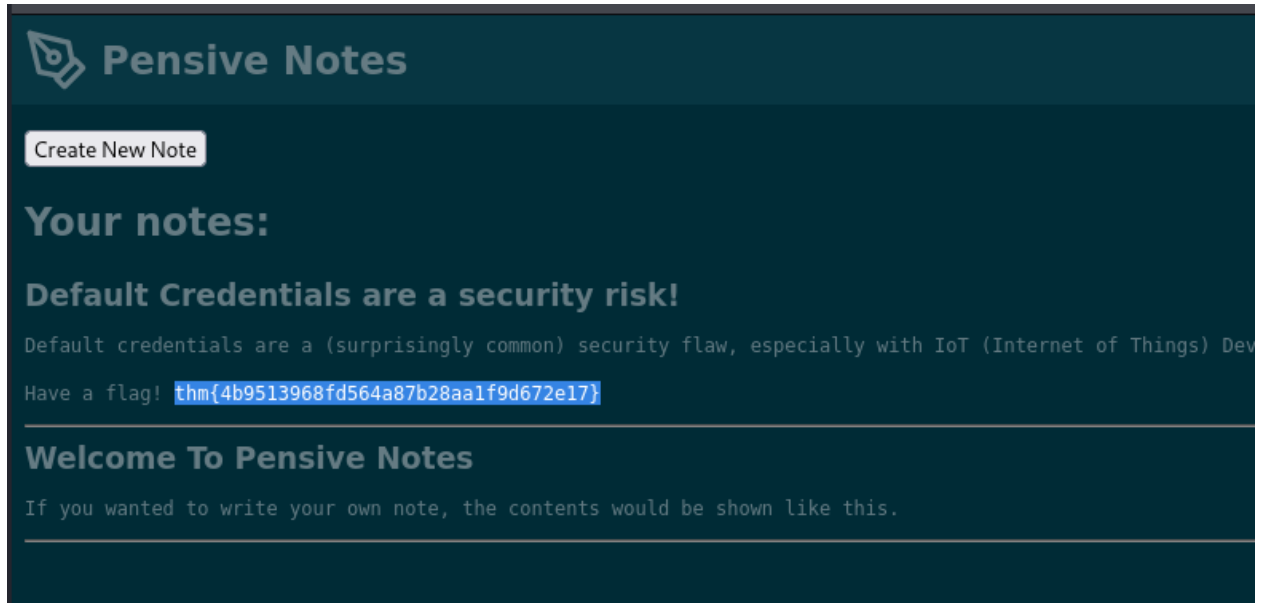
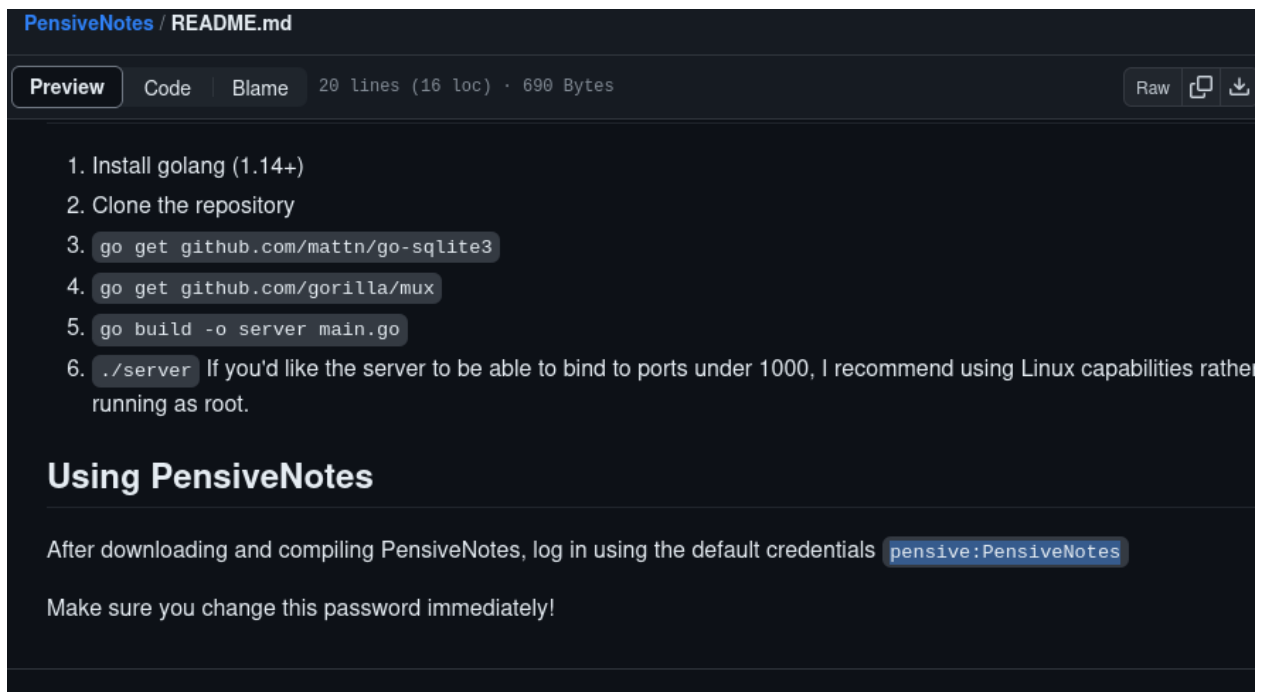
- 
- Practical XXE Payload and Exploiting tasks.
- **Example Answers:**
  - User in `/etc/passwd`: falcon





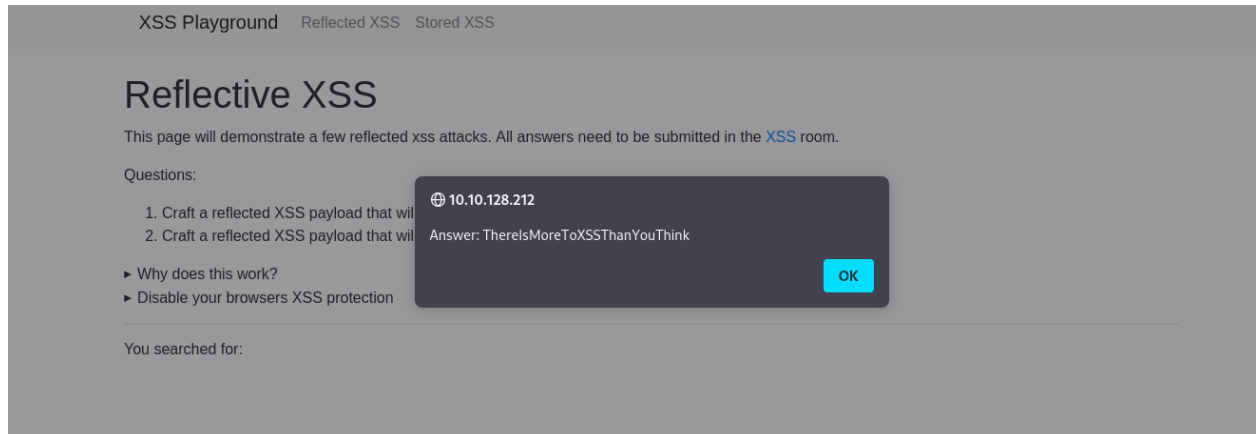
## Task 19: [Severity 6] Security Misconfiguration

- Discussing risks from default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages.
- Practical task: Finding a flag through security misconfiguration.
- **Example Answer:**
  - **Flag:** thm{4b9513968fd564a87b28aa1f9d672e17}

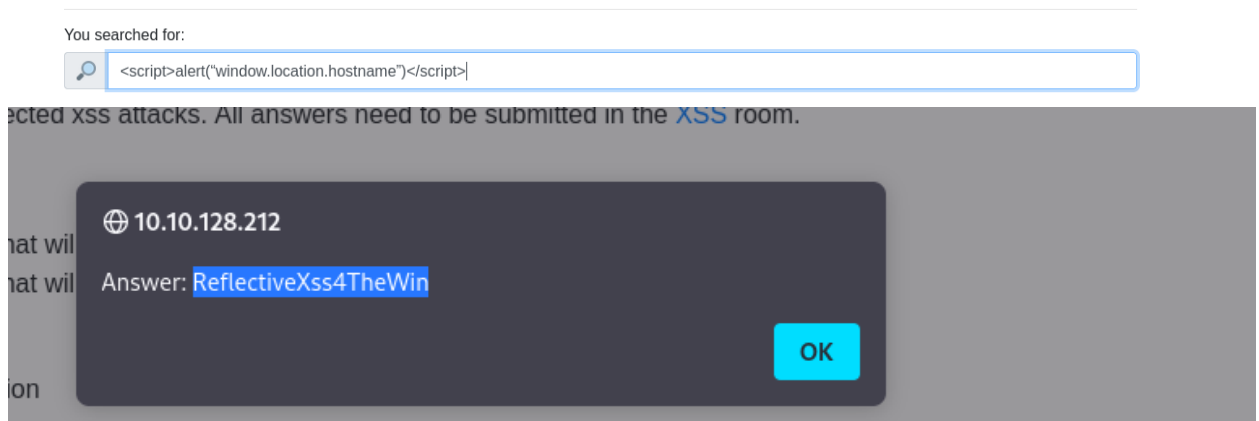


## Task 20: [Severity 7] Cross-site Scripting (XSS)

- Understanding XSS, where attackers inject malicious scripts into content viewed by other users.
- Reflected XSS Practical: Crafting payloads to create popups.
  - **Payload for "Hello" popup: ThereIsMoreToXSSThanYouThink**



- **Payload for IP address popup: ReflectiveXss4TheWin**



- Stored XSS Practical: Inserting HTML and JavaScript into comments.

- **Inserting HTML: HTML\_T4gs**

Comments

Successfully added a HTML comment! Answer for Q1: **HTML\_T4gs**

Jack: Hey Everyone!

Logan: Hey Jack, how're you?

Jack: Yeah good thanks!

damian:

Hello

Add a comment

<h1>hello</h1>

- **Creating alert popup: W3LL\_D0N3\_LVL2**

Successfully added a comment!

Jack: Hey Everyone!

Logan: Hey Jack, how're you?

Jack: Yeah good thanks!

damian:

# Hello

damian:

damian:

Add a comment

```
<script>alert(document.cookie)</script>
```

(stored) cross-site scripting attacks. All answers need to be submitted in the [XSS](#) room.

🌐 10.10.128.212

W3LL\_D0N3\_LVL2

☐ Don't allow 10.10.128.212 to prompt you again

OK

- **Defacing website text:** `websites_can_be_easily_defaced_with_xss`

Add a comment

```
<script>document.querySelector("#thm-title").textContent = 'I am a hacker'</script>
```

Comment

## Task 21: [Severity 8] Insecure Deserialization

- Understanding how deserialized data can be manipulated to carry out attacks like denial of service.
- Tomcat developed by The Apache Software Foundation.

# James Duncan Davidson

Apache Tomcat

Apache Tomcat default page

Original author(s)	James Duncan Davidson
Developer(s)	The Apache Software Foundation
Initial release	1999
Stable release	10.1.19 / 19 February 2024

- Insecure deserialization can lead to Denial-of-Service attacks.

Even in cases where remote code execution is not possible, insecure deserialization can lead to privilege escalation, arbitrary file access, and denial-of-service attacks.



PortSwigger  
https://portswigger.net › web-security › deserialization

Insecure deserialization | Web Security Academy - PortSwigger

## Task 22: [Severity 8] Insecure Deserialization - Objects

- Concept of object states and behaviors.
- **Example Answer:** A dog sleeping is a **Behaviour**.

Objects

A prominent element of object-oriented programming (OOP), objects are made up of two things:

- State
- Behaviour

Simply, objects allow you to create similar lines of code without having to do the leg-work of writing the same lines of code again.

For example, a lamp would be a good object. Lamps can have different types of bulbs, this would be their state, as well as being either on/off - their behaviour.

## Task 23: [Severity 8] Insecure Deserialization - Deserialization

- Understanding deserialization, converting data from a binary format back into an object.
- Base-2 formatting for data sent across networks is called Binary.

base 2 format of data

Images Pdf Calculator In c Videos Shopping News Books Maps

About 2,060,000,000 results (0.29 seconds)

A **binary** number is a number expressed in the base-2 numeral system or binary numeral system, a method of mathematical expression which uses only two symbols: typically "0" (zero) and "1" (one).

$x =$	1100.101110
$x \times 2^6 =$	1100101110.01110...
$x \times 2 =$	11001.01110...
$\times (2^6 - 2) =$	1100010101
$x =$	1100010101/111110
$x =$	$(789/62)_{10}$

## Task 24: [Severity 8] Insecure Deserialization - Cookies

- Exploring how cookies can be insecurely deserialized.
- **Example Answers:**
  - **URL for cookie path webapp.com/login:** webapp.com/login

If a cookie had the path of **webapp.com/login**, what would the URL that the user has to visit be?

- **Secure cookies technology acronym:** HTTPS

A Secure cookie is an HTTP cookie with a Secure attribute set. With the Secure attribute set, the cookie must always be sent over an encrypted **HTTPS** connection (SSL/TLS) and never in plain text.



NordVPN  
https://nordvpn.com › CyberSecurity › Glossary

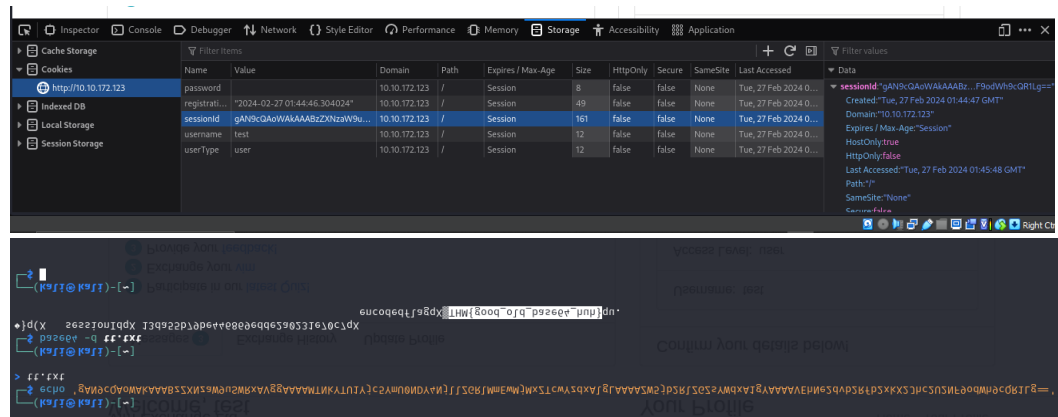
Secure cookie definition - NordVPN

About featured snippets Feedback

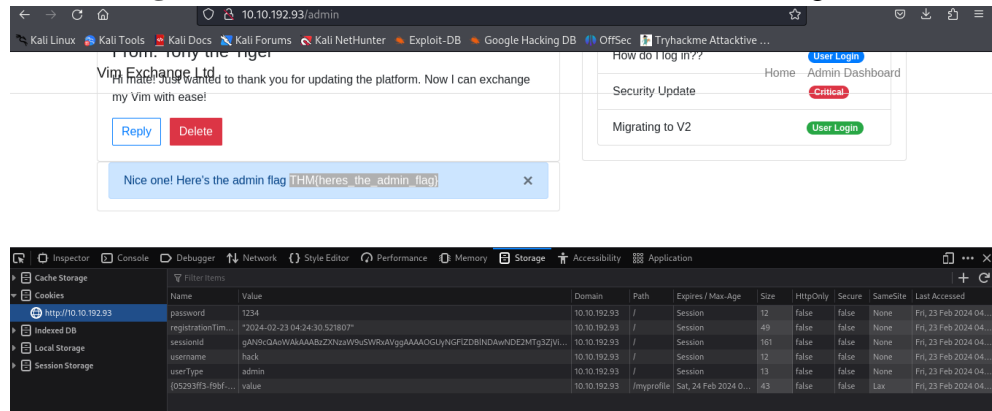
## Task 25: [Severity 8] Insecure Deserialization - Cookies Practical

- Practical challenge with cookie manipulation.
- **Example Flags:**
  - **First flag:** THM{good\_old\_base64\_huh}





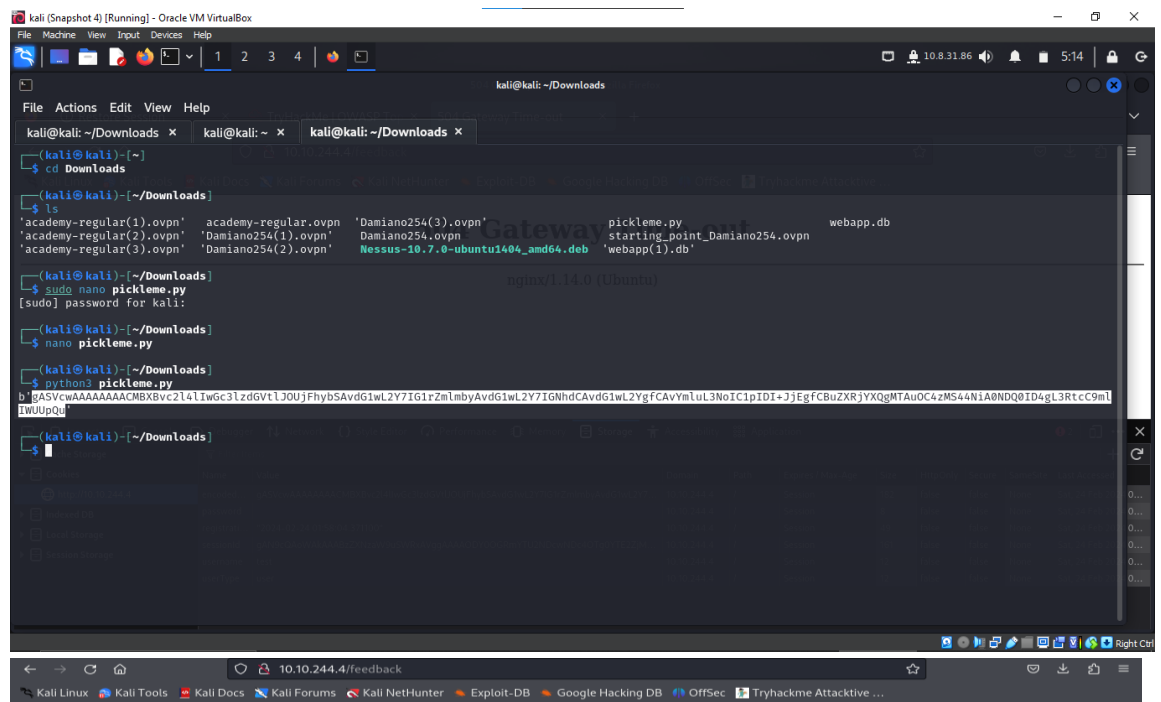
- **Second flag (admin dashboard):** THM{heres\_the\_admin\_flag}



## Task 26: [Severity 8] Insecure Deserialization - Code Execution

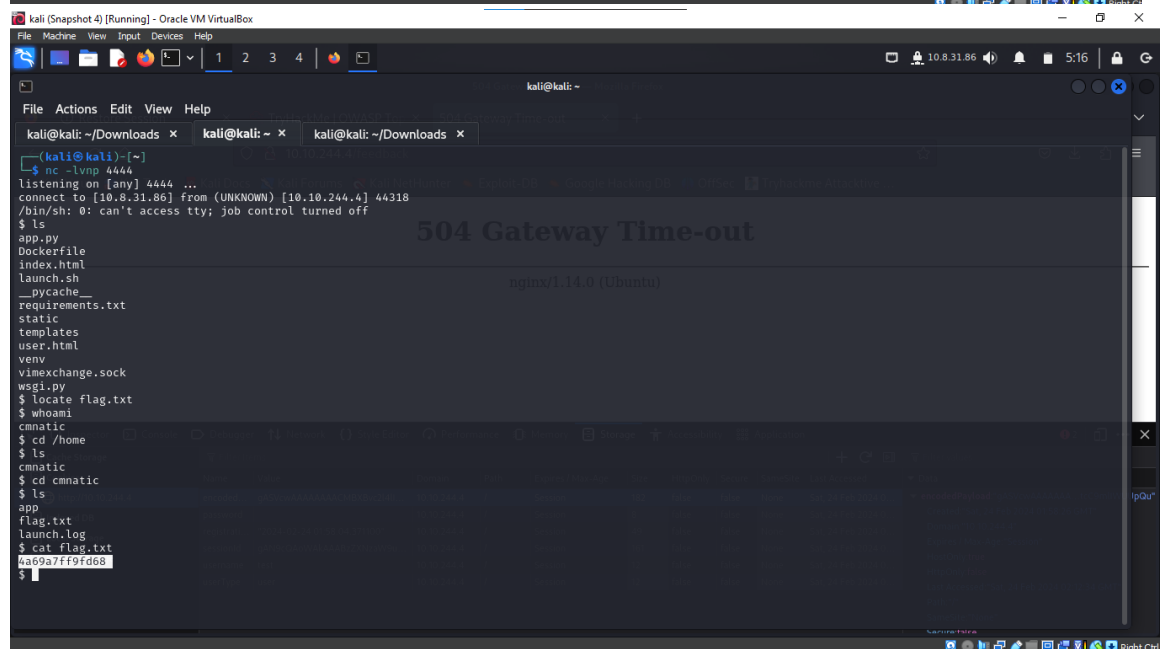
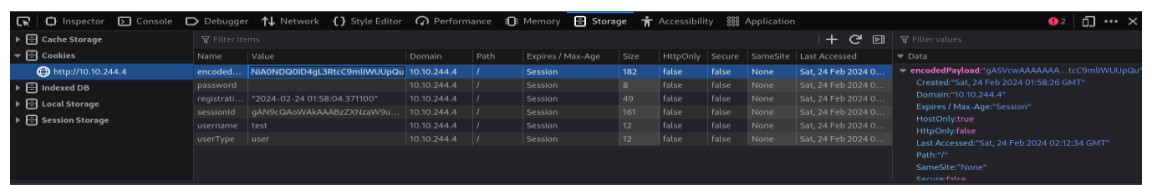
- Exploring how deserialization vulnerabilities can lead to code execution.
- **Example Answer:**
  - **flag.txt:** 4a69a7ff9fd68





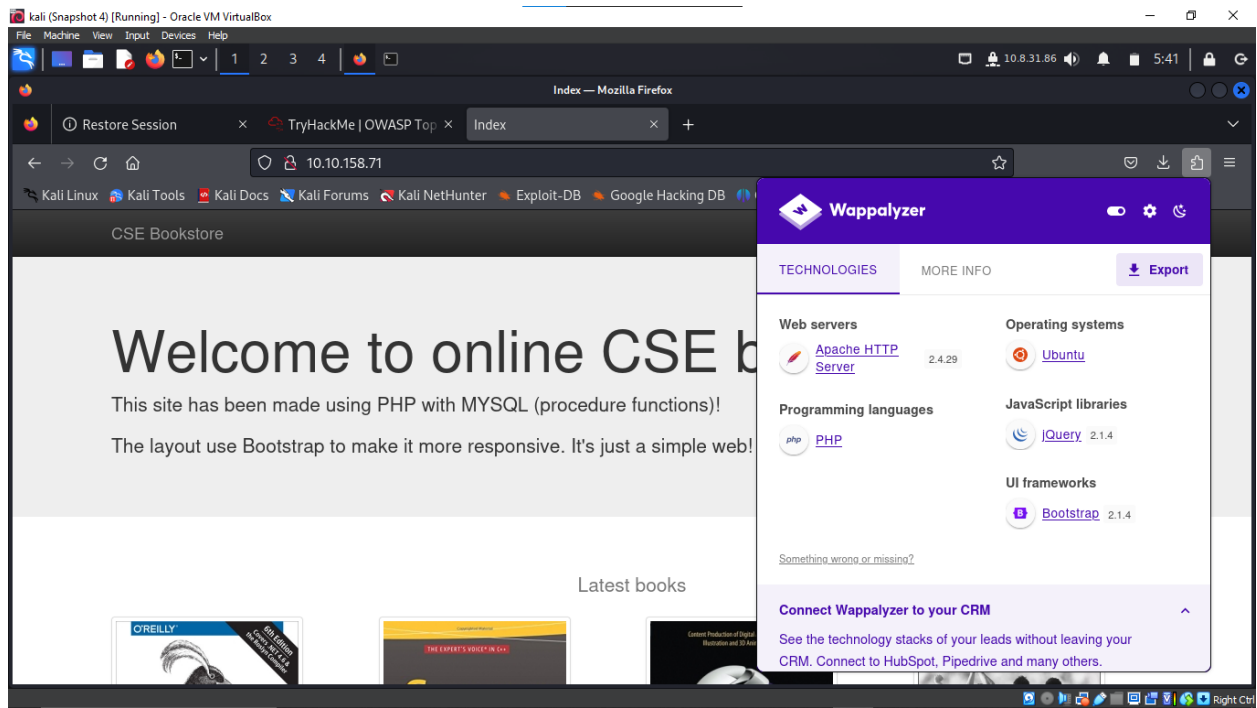
## 504 Gateway Time-out

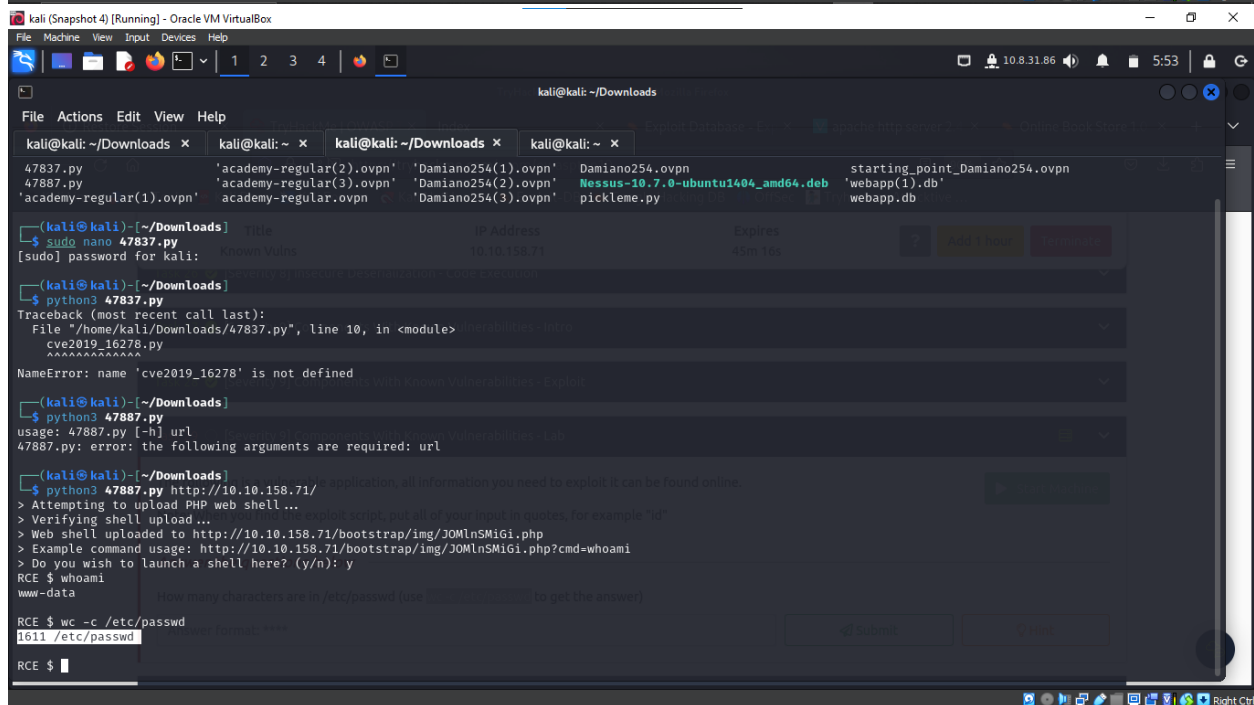
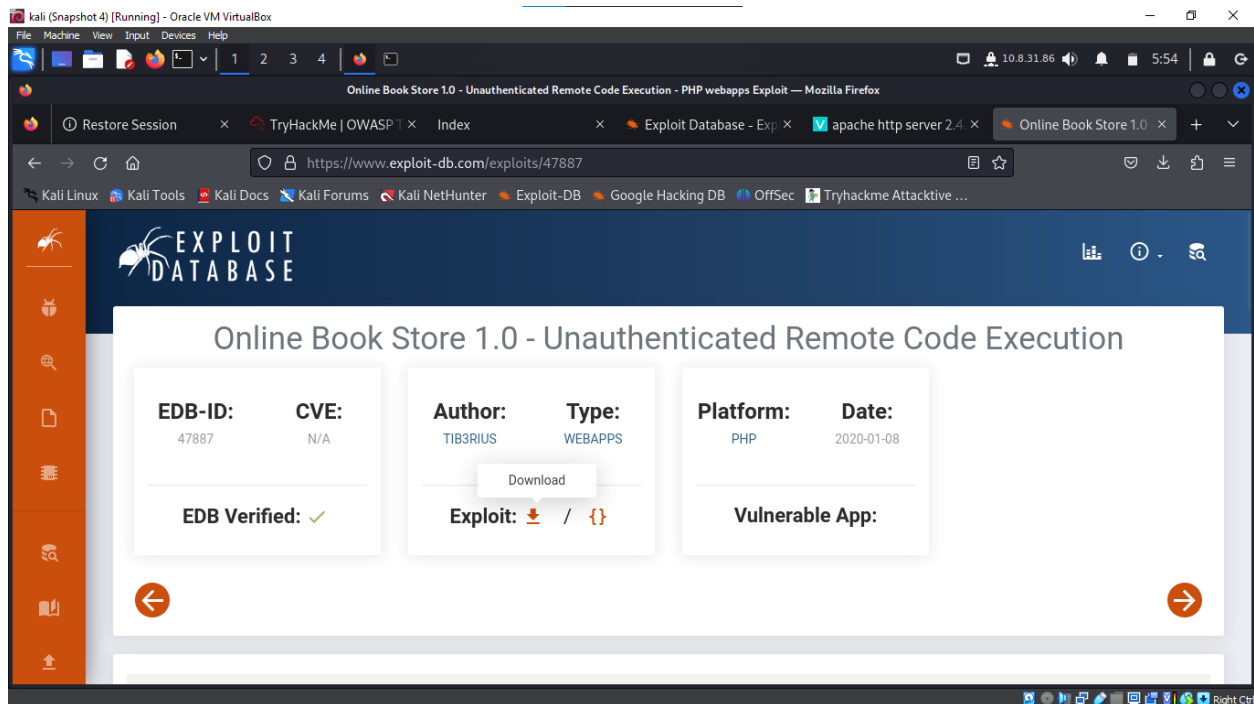
nginx/1.14.0 (Ubuntu)



## Task 27-29: [Severity 9] Components With Known Vulnerabilities

- Discussing risks associated with using software components with known vulnerabilities.
- Practical lab tasks involving exploitation of these vulnerabilities.
- **Example Answer:**
  - Characters in /etc/passwd: 1611





## Task 30: [Severity 10] Insufficient Logging and Monitoring

- Understanding risks due to inadequate logging, monitoring, and alerting.
- Practical analysis of logs to identify attack details.
- **Example Answers:**
  - **Attacker's IP address: 49.99.13.16**

```
kali@kali: ~/Downloads
ls
47837.py 'academy-regular(2).ovpn' 'Damiano254(1).ovpn' Damiano254.ovpn Expires pickleme.py webapp.db
47887.py 'academy-regular(3).ovpn' 'Damiano254(2).ovpn' login-logs.txt starting_point_Damiano254.ovpn
'academy-regular(1).ovpn' academy-regular.ovpn 'Damiano254(3).ovpn' Nessus-10.7.0-ubuntu1404_amd64.deb 'webapp(1).db'

cat login-logs.txt
200 OK 12.55.22.88 jir22 2019-03-18T09:21:17 /login
200 OK 14.56.23.11 rand99 2019-03-18T10:19:22 /login
200 OK 17.33.10.38 after11 2019-03-18T11:11:44 /login
200 OK 99.12.44.20 rad4 2019-03-18T11:55:51 /login
200 OK 67.34.22.10 bfff1 2019-03-18T13:08:59 /login
200 OK 34.55.11.14 hax0r 2019-03-21T21:08:15 /login
401 Unauthorised 49.99.13.16 admin 2019-03-21T21:08:20 /login
401 Unauthorised 49.99.13.16 administrator 2019-03-21T21:08:25 /login
401 Unauthorised 49.99.13.16 anonymous 2019-03-21T21:08:30 /login
401 Unauthorised 49.99.13.16 root 2019-03-21T21:08:30 /login
```

## • Type of attack: Brute Force

```
kali@kali: ~/Downloads
ls
47837.py 'academy-regular(2).ovpn' 'Damiano254(1).ovpn' Damiano254.ovpn Expires pickleme.py webapp.db
47887.py 'academy-regular(3).ovpn' 'Damiano254(2).ovpn' login-logs.txt starting_point_Damiano254.ovpn
'academy-regular(1).ovpn' academy-regular.ovpn 'Damiano254(3).ovpn' Nessus-10.7.0-ubuntu1404_amd64.deb 'webapp(1).db'

cat login-logs.txt
200 OK 12.55.22.88 jir22 2019-03-18T09:21:17 /login
200 OK 14.56.23.11 rand99 2019-03-18T10:19:22 /login
200 OK 17.33.10.38 after11 2019-03-18T11:11:44 /login
200 OK 99.12.44.20 rad4 2019-03-18T11:55:51 /login
200 OK 67.34.22.10 bfff1 2019-03-18T13:08:59 /login
200 OK 34.55.11.14 hax0r 2019-03-21T21:08:15 /login
401 Unauthorised 49.99.13.16 admin 2019-03-21T21:08:20 /login
401 Unauthorised 49.99.13.16 administrator 2019-03-21T21:08:25 /login
401 Unauthorised 49.99.13.16 anonymous 2019-03-21T21:08:30 /login
401 Unauthorised 49.99.13.16 root 2019-03-21T21:08:30 /login
```

## Conclusion

Completing the TryHackMe OWASP Top 10 module equips learners with vital skills in identifying and addressing web vulnerabilities. It's a journey from foundational knowledge to advanced security practices, emphasizing the importance of ongoing learning in the dynamic field of cybersecurity.

