



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

OWASP VITCC

PRESENTS

Official Writeup for the **WASPCON** CTF cum Hands-on session
held on 10th October

LEVEL 1:

The first level was the simplest one of all. All we had to do was write the command “!owasp give me flag” in the OWASP discord server and we get the flag.



Flag for level 1: WASP{G3t_th3_f1r5t_flag}

LEVEL 2:

Find the next flag



Level 2 had a bunch of clickable images, all of them except one rick-rolling us. To find the correct image, instead of trying out every image one by one, we view the source code of the page.

```
Line wrap □
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <meta charset="utf-8">
5     <!-- Fonts -->
6     <link rel="preconnect" href="https://fonts.googleapis.com">
7     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
8     <link href="https://fonts.googleapis.com/css2?family=Kanit:wght@300&display=swap" rel="stylesheet">
9
10
11   <!-- Favicon -->
12   <link rel="icon" href="/images/owasp-logo.png">
13   <title>Image</title>
14   <h1 style="text-align:center; font-family: 'Kanit', sans-serif; font-size: 4rem;">Find the next flag</h1>
15 </head>
16 <body>
17   <a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"></a>
18   <a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"></a>
19   <a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"></a>
20   <a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"></a>
21   <a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"></a>
22   <a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"></a>
23   <a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"></a>
24   <a href=".y.php"></a>
25   <a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"></a>
```

We can see the second last image to not be a youtube link. Clicking on .y.php we are redirected to the flag.



Flag for level 2: WASP{1704S5WY7URM38IZU7HL}

LEVEL 3:



Level 3 starts off with a nicely designed web page. Since there is nothing to directly do here, we again view the source code.

In the source code we find a comment <!--V0FTUHtsRXYzMV8zX3IzQGNoZWQhISF9-->. This looks like a base64 encryption. Going online for a base 64 decoder, we find the flag.

Decode from Base64 format

Simply enter your data then push the decode button.

```
V0FTUHtsRXYzMV8zX3IzQGNoZWQhISF9
```

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

Source character set.

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

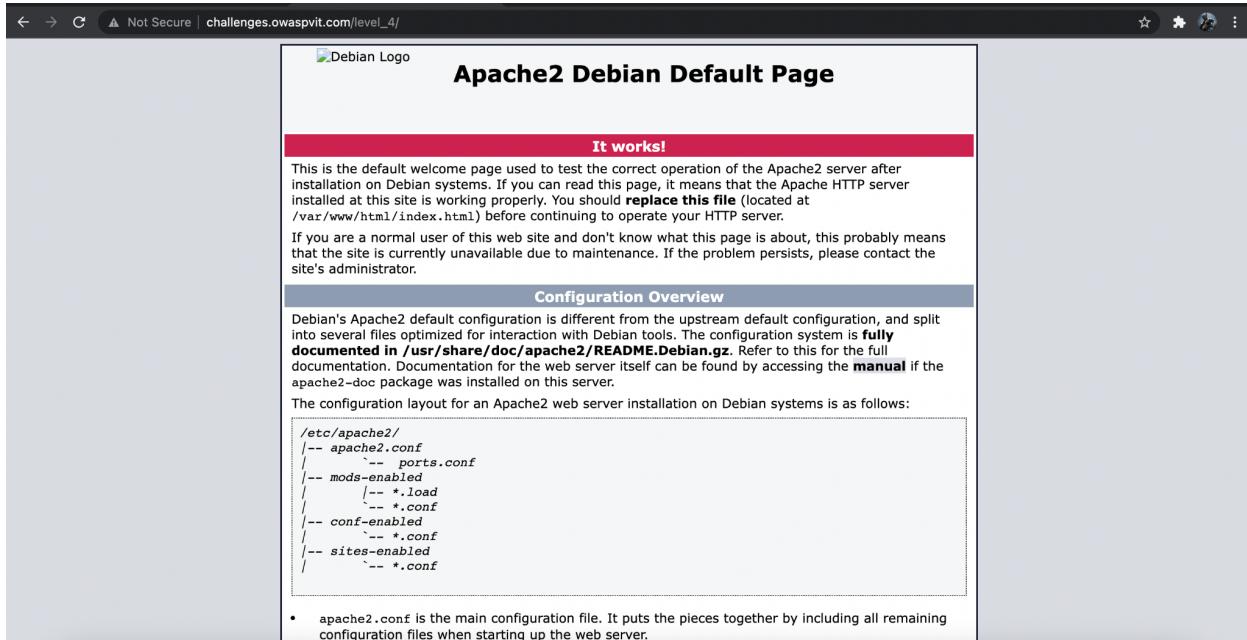
Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

◀ DECODE ▶ Decodes your data into the area below.

```
WASP{IEv31_3_r3@ched!!!}
```

Flag for level 3: WASP{IEv31_3_r3@ched!!!}

LEVEL 4:



Not Secure | challenges.owaspit.com/level_4/

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

We have checked everything in level 4. There's nothing on the index page. Nothing of significance in the source code. So the next thing we do is check the robots.txt file. (robots.txt is a file on every webpage which contains additional setup instructions.)

So redirecting the webpage to http://challenges.owaspit.com/level_4/robots.txt we find

User-agent: *
Disallow: /flag.txt/

We see that there is another directory called flag.txt. So redirecting to http://challenges.owaspit.com/level_4/flag.txt we get the flag.



Not Secure | challenges.owaspit.com/level_4/flag.txt

Nice work!
Flag = WASP{Mr_RoBoT_sAyS_hI}

Flag for level 4: WASP{Mr_RoBoT_sAyS_hI}

LEVEL 5:



Level 5 is a simple html page with an image of a cookie in the middle. The cookie here refers to the cookies stored in the browser. We are expected to check the cookies here.
In chrome -> right click-> inspect -> Application -> Storage - > cookies.

A screenshot of the Chrome DevTools Application tab. The Cookies section is selected, showing a list of stored cookies. One cookie, 'FlagCookie', is highlighted and selected. Its value is displayed as 'WASP{cOoKiEs_yUmM_yUmM}'.

Name	Value	Dom...	Path	Exp...	Size	Http...	Sec...	Sam...	Sam...	Priority
Zair	7843211	chall...	/	2021...	11					Med...
FlagCookie	WASP{cOoKiEs_yUmM_yUmM}	chall...	/	2021...	33					Med...
InCookie	3123wsacs	chall...	/	2021...	16					Med...
crawler	374nccsaq	chall...	/	2021...	15					Med...
capture-cookie	vuhenvdqa	chall...	/	2021...	23					Med...
Cookie1	36194831981	chall...	/	2021...	18					Med...

We can see that the flag for this level is stored in the FlagCookie.

Flag for level 5: WASP{cOoKiEs_yUmM_yUmM}

LEVEL 6:



In the source code of this page we find two pieces of information.

```
25         margin-right: auto;
26         border-radius: 50%;
27     }
28 
```

<!--dvvnhb iru vwhjdqrjudsksb lv vwhjfwi-->

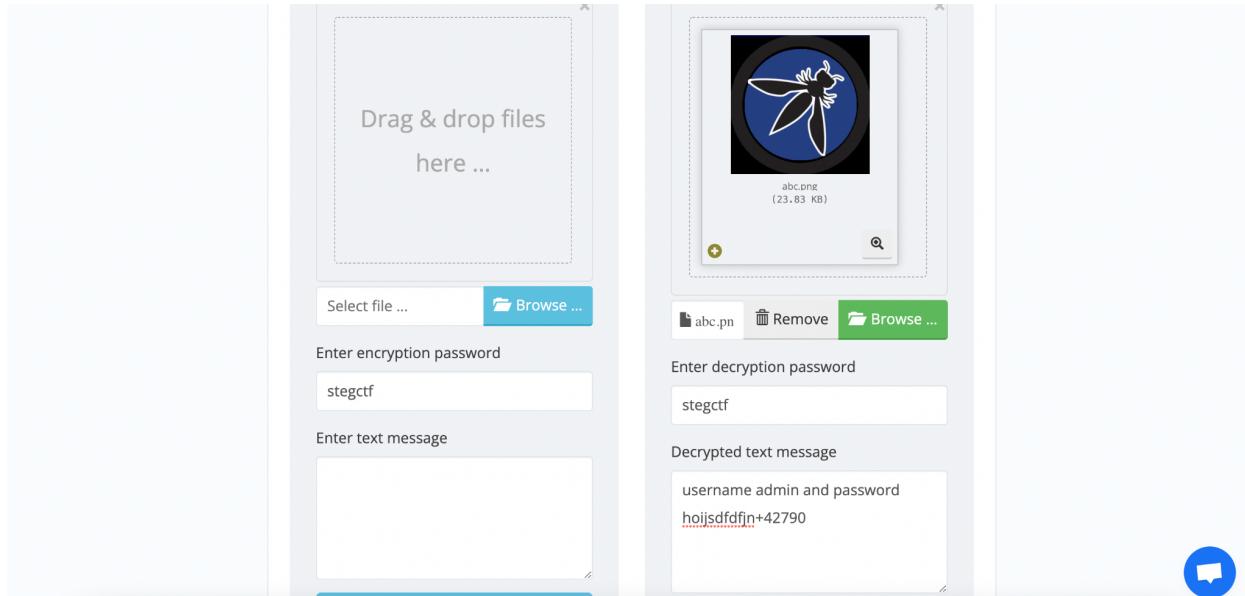
```
29 
```

```
30 </head>
31 <body>
32     <div class="container">
33         <div class="row">
34             <div class="col-lg-3 col-md-2"></div>
35             <div class="col-lg-6 col-md-8 login-box">
36                 <div class="col-lg-12 login-key">
37                     <i class="fa fa-key" aria-hidden="true"></i>
38                 </div>
39                 <section class="steg_ctf">
40                     <p></p>
41                 </section>
42                 <div class="col-lg-12 login-title">
43                     OWASP VITCC
44                 </div>
45 
46                 <div class="col-lg-12 login-form">
47                     <div class="col-lg-12 login-form">
48                         <div method = "GET" action = "./Steg.php">
49                             <form-group>
50                                 <label class="form-control-label" style="color:white;">USERNAME</label>
51                                 <input type="text" required name = "username" class="form-control">
52                             </form-group>
53                                 <label class="form-control-label" style="color:white;">PASSWORD</label>
54                                 <input type="password" required name = "password" class="form-control" i>
55                             </div>
56 
57                         <div class="col-lg-12 loginbtm">
58                             <div class="col-lg-6 login-btm login-text">
59                                 <!-- Error Message -->
60                             </div>
61                             <div class="col-lg-6 login-btm login-button">
62                                 <button type="submit" class="btn btn-outline-primary">LOGIN</button>
63                             </div>
64                         </div>
65                     </form>
66                 </div>
67             </div>
68         </div>
69     </div>
```

The first is a ROT encoded message. Going to rot13.com we can try out each ROT to find that it is ROT-23 encoded. The decoded message is - passkey for steganography is stegctf.

So we have found a passkey. And also steganography is mentioned. Steghide is one such steganography tool which is used to encode/decode hidden messages in photos and audio files.

Downloading the image we found in the source page we move towards an online steghide tool and enter the passkey to get -

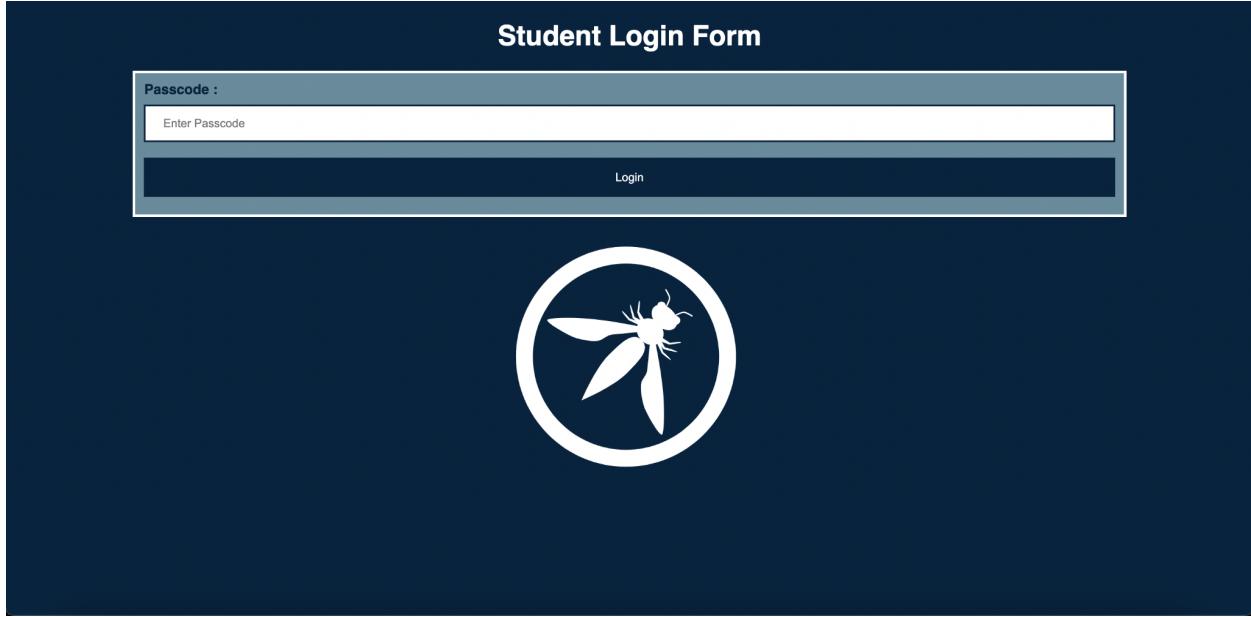


Submitting the username and password in the index page of level 6 we get the flag.



Flag for level 6: WASP{CTF_stg1729}

LEVEL 7:



Here in the source code of the page at the bottom we find a code.

```
100: function generateId(len) {
101:   var arr = new Uint8Array((len || 40) / 2);
102:   window.crypto.getRandomValues(arr);
103:   return Array.from(arr, (dec)=>dec.toString(16).padStart(2, "0")).join("");
104:
105:
106:
107:
108:
109: var caesarShift = function (str, amount) {
110:   if (amount < 0) {
111:     return caesarShift(str, amount + 26);
112:   }
113:
114:   var output = "";
115:
116:   for (var i = 0; i < str.length; i++) {
117:     var c = str[i];
118:
119:     if (c.match(/[a-z]/i)) {
120:       var code = str.charCodeAt(i);
121:
122:       if (code >= 65 && code <= 90) {
123:         c = String.fromCharCode(((code - 65 + amount) % 26) + 65);
124:       } else if ((code >= 97 && code <= 122) {
125:         c = String.fromCharCode(((code - 97 + amount) % 26) + 97);
126:       }
127:     }
128:     output += c;
129:   }
130:
131:   return output;
132: };
133:
134: validate = () => {
135:   var form = document.getElementById("theform");
136:   function handleForm(event) {
137:     event.preventDefault();
138:   }
139:   form.addEventListener("submit", handleForm);
140:   let username = document.getElementById("pc").value;
141:   if (caesarShift(username, 446) == "PIzIpwIzIrJpeK") {
142:
143:     document.getElementById("flag").innerHTML =
144:       "Congratulations! Your flag is your input encoded in format WASP{input}";
145:   }
146: }
```

Looking closely at the code we can understand what it is trying to do. It's obvious that there is caesar shift involved, with the shift number 446, and the encoded word - PIzIpwIzIrJpeK

Any online caesar shift decoder will decode the message and give the output as

LEvElsEvEnFlaG

Submitting this in the login page we are told that LEvElsEvEnFlaG is the flag in the format WASP{flag}

Flag for level 7: WASP{LEvElsEvEnFlaG}

LEVEL 8:

Search...

This level is dedicated to simple XSS vulnerability. Having some basic knowledge of XSS will help.

Since this level requires no filtering a simple command like - <script>alert(0)</script> will result in the flag.



P.S. This is not the only method to solve this level.

Flag for Level 8:WASP{cOnGrATuLaTioNs_Challenge8_cOmPl3ted!}

LEVEL 9:

Not Secure | challenges.owaspvit.com/level_9/

Student Login Form

Username :

Password :



This level is dedicated to Sqli (Sql injection). To solve this level you need to have a very basic understanding of the same. Googling a little of sql will make it easier.

Here the username should be - '`' or '1' = '1' #`

And you can type anything in the password field.

Submitting it will result in the flag.

Not Secure | challenges.owaspvit.com/level_9/login.php

Your flag is WASP{level_9_flag}

Flag for Level 9: WASP{level_9_flag}

LEVEL 10:



In the view-source code link just above the verify button we get a useful piece of information -

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<
<body>
<h1>natas6</h1>
<div id="content">
<?
include "/secret.txt";
if(array_key_exists("submit", $_POST)) {
    if($_POST['secret'] == "F5asdfjanjlhbasdfss") {
        print "Access granted. The password is <censored>";
    } else {
        print "Wrong secret";
    }
}
?>

<form method=post>
Input secret: <input name=secret><br>
<input type=submit name=submit>
</form>

<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

There is a directory called /secret.txt. Redirecting the webpage to http://challenges.owaspvit.com/level_10/secret.txt we find a text, which when submitted in the secret field in the index page, we get the flag.



Flag for level 10: WASP{CTF_256}

LEVEL 11:

The screenshot shows a browser window with the URL challenges.owaspvit.com/level_11/needlesearch.php. The page displays the message "You're very close to the finish line!" above a search form. The form includes a text input field containing "abc" and a "Search" button.

Here typing random words as “flag” or just letters are providing us with a bunch of random words.

You're very close to the finish line!

abc
Find words containing: Search

a abandon abandoned ability able about above abroad absence absent absolute absolutely absorb abuse academic accent accept acceptable access accident accidental accidentally accommodation accompany according to account account for accurate accurately accuse achieve achievement acid acknowledge a couple acquire across act action active activity actor actress actual actually ad adapt add addition additional add on address add up add up to adequate adequately adjust admiration admire admit adopt adult advance advanced advantage adventure advert advertise advertisement advertising advice advise affair affect affection afford afraid after afternoon afterwards again against age aged agency agent aggressive ago agree agreement ahead aid aim air aircraft airport alarm alarmed alarming alcohol alcoholic alive all allied allow allow for all right ally almost alone along alongside aloud alphabet alphabetical alphabetically already also alter alternative alternatively although altogether always am amaze amazed amazing ambition ambulance among amount amount to amuse amused amusing analyse analysis ancient and anger angle angrily angry animal ankle anniversary announce annoy annoyed annoying annual annually another answer anti- anticipate anxiety anxious anxiously any anybody anyone anything anyway anywhere apart apart from apartment apologize apparent apparently appeal appear appearance apple application apply appoint appointment appreciate approach appropriate approval approve approving approximate approximately April area argue argument arise arm armed arms army around arrange arrangement arrest arrival arrive at arrow art article artificial artificially artist artistic artistically as ashamed aside aside from ask asleep aspect assist assistance assistant associate association assume assure at atmosphere atom attach attached attack attempt attempted attend attend to attention attorney attract attraction attractive audience August aunt author authority automatic automatically autumn available average avoid awake award aware away awful awfully awkward awkwardly baby back background back up backward backwards bacteria bad badly bad-tempered bag baggage bake balance ball ban band bandage bank bar bargain barrier base based base on basic basically basis bath bathroom battery battle bay beach break bear beard beat beat up beautiful beautifully beauty because because of behalf behave behaviour beneath birthday black blame blank board boat brain branch brand brave bread break break down breakfast break in break into break off break out break up break breath breathe breathe in breathe out breathing brilliant bring back bring forward broad broadcast broadly businessman cabinet cable cake calculate calculation call call back called call for call off call up calm calm down calmly camera camp campaign camping can 1 can 2 cancel cancer candidate candy cannot cap capable capacity capital captain capture car card cardboard care career care for careful carefully careless carelessly carpet carry carry on carry out case cash cast castle cat catch catch up category cause cease celebrate celebration central certain certainly certificate chain chair chairman chairwoman challenge chamber chance change round round channel chapter character characteristic charge charity chart chase chase away chat cheap cheaply cheat cheat of chemical chocolate cigarette cinema circumstance claim clap class classic classroom clean clean up clear clearly clear out clear up climate coach coal coast coat collapse colleague combination come across comfortable comfortably command commercial communicate communication company compare comparison complain complaint complicate complicated concentrate concentrate on concentration congratulate congratulation conservative considerable considerably consideration constant constantly contact contain container contemporary contract contrast contrast contrast contrasting conventional conversation cottage courage crack cracked craft crash crazy cream create creature credit card criminal critical crucial cultural cupboard curtain cut back dad daily damage damp dance dancer dancing danger dangerous dare dark date data date back daughter day dead deaf deal deal in deal with dear death debate decade decay declare decorate decoration decorative decrease defeat delay deliberate deliberately delicate demand demonstrate department departure desperate desperately detail detailed determination diagram diamond diary dictionary die away digital disabled disadvantage disagree disagreement disagree with doing disappear disappoint disappointed disappointing disapproval disapprove disapproving disaster disease display distance dollar dominate downstairs downward downwards draft drag drama dramatic dramatically

But instead if we look closely at the URL, we can see that this is not named index.php as usual, but needlesearch.php

Simply typing “needle” in the search box yields us the flag.

The screenshot shows a browser window with the URL challenges.owaspvit.com/level_11/needlesearch.php?needle=needle&submit=Search. The page displays the message "You're very close to the finish line!" above a search form. The form includes a text input field containing "WASP{NeEdLe_HuH?}" and a "Search" button.

Flag for level 11: WASP{NeEdLe_HuH?}

LEVEL 12

Not Secure | challenges.owaspvit.com/level_12/

Student Login Form

Username :

Password :



This page is very similar to Level 9. It also focuses on Sqli, *but* with filtering.

The username could be : `'or'1='1/**/union/**/Select/**/flag,1/**/from/**/owasp#`

And you can type anything in the password field.

Submitting it will result in the flag.

Not Secure | challenges.owaspvit.com/level_12/login.php

admin auwdjawoidjawiodj
WASP{You_ReAcheD_LeveL_13} 1

P.S. This is not the only method to solve this level.

Flag for Level 12: WASP{You_ReAcheD_LeveL_13}

LEVEL 13:



As it suggests this is the harder version of level 8 - XSS with filtering.

Something like : will result in the flag for the level.



P.S. Even if this is not the only method to solve the level, the usage of alert(FLAG) is mandatory as mentioned in the level.

Flag for Level 13: WASP{GCHJP2FVWIMSZCG4L8W1}

LEVEL 14: (OSINT)

Find the name of Noelle's sister's dog, and wrap it in WASP{} in small letters to get the flag.

FORMAT: WASP{dog name}

P.S. Noelle's family only uses twitter



So this is an OSINT challenge. Basically meaning we have to search through google and social media to get information.

Looking at the question we can formulate a plan. First we have to find information about noelle. Then about her sister. And finally about her dog.

The image provided to us stores some useful information in its metadata. Using exiftool, we can see through its contents.

Directory	/tmp
File Size	23 KiB
File Modification Date/Time	2021:10:10 05:55:15+00:00
File Access Date/Time	2021:10:10 05:55:15+00:00
File Inode Change Date/Time	2021:10:10 05:55:15+00:00
File Permissions	-rw-----
File Type	JPEG
File Type Extension	jpg
MIME Type	image/jpeg
JFIF Version	1.01
Resolution Unit	inches
X Resolution	72
Y Resolution	72
XMP Toolkit	Image::ExifTool 12.16
Creator	noelle091002
Image Width	800
Image Height	400
Encoding Process	Progressive DCT, Huffman coding
Bits Per Sample	8
Color Components	3

We see a username. We were told in the question that Noelle's family uses only twitter. So searching this username in twitter yields us Noelle's twitter profile.

Noelle Demonia
@noelle091002

Hiiii I am Noelle, a sophomore studying psychology. I love cats. First time on twitter :)

New York Joined October 2021

0 Following 0 Followers

No followed by anyone you're following

Tweets Tweets & replies Media Likes

Noelle Demonia @noelle091002 · Oct 9
Hey guys have a look at my playlist and let me know how it is!!!

open.spotify.com
Vibe ❤️
Noelle - Playlist · 6 songs

Noelle Demonia @noelle091002 · Oct 9
OOH THIS IS MY FIRST TWEET YAYYY

We aren't able to find anything which could help us directly in her profile. But she has shared a spotify playlist link. So let's check that out.

The screenshot shows a Spotify interface. At the top left is the Spotify logo. To the right are 'SIGN UP' and 'LOG IN' buttons. The main area displays a collaborative playlist named 'Vibe ❤️'. Below the title, it says 'Noelle • 1 like • 7 songs, 22 min 50 sec'. The playlist table has columns for '#', 'TITLE', 'ALBUM', 'ADDED BY', 'DATE ADDED', and a small icon. The tracks listed are:

#	TITLE	ALBUM	ADDED BY	DATE ADDED	
1	THATS WHAT I WANT Lil Nas X	MONTERO	8lbf6aqij38l5kzitg18jf79	1 day ago	2:23
2	Down Jay Sean, Lil Wayne	All Or Nothing	8lbf6aqij38l5kzitg18jf79	1 day ago	3:32
3	Replay Iyaz	Replay	8lbf6aqij38l5kzitg18jf79	1 day ago	3:02
4	Stereo Hearts (feat. Adam Levine) Gym Class Heroes, Adam Levine	The Papercut Chronicles II	8lbf6aqij38l5kzitg18jf79	1 day ago	3:30
5	The Nights Avicii	The Days / Nights	8lbf6aqij38l5kzitg18jf79	1 day ago	2:56
6	Wavin' Flag - Coca-Cola® Celebratio... KNAAN	Troubadour (Champion Edition - As...	8lbf6aqij38l5kzitg18jf79	1 day ago	3:32
7	Payphone Maroon 5, Wiz Khalifa	Overexposed Track By Track	1cjbglub0k8sf2xife195bm...	1 day ago	3:51

On the far left of the Spotify interface, there are navigation links: Home, Search, Your Library, Create Playlist, and Liked Songs. At the bottom left are 'Cookies' and 'Privacy' links.

Here we see that it is a collaborative playlist indicating that people other than Noelle have contributed to this. And if we take a look at the "Added by" column, we can see that surely the last song has been added by another user. Taking a look at that user's profile -

The screenshot shows a Spotify user profile page. At the top left is the Spotify logo. To the right are 'SIGN UP' and 'LOG IN' buttons. The main area features a circular profile picture of a puppy. Below the picture, the username 'DemonicTracy04' is displayed in large white text. Underneath the username, there are 'PROFILE' and 'FOLLOW' buttons. On the far left of the Spotify interface, there are navigation links: Home, Search, Your Library, Create Playlist, and Liked Songs.

She shares the same surname as Noelle, and she also has a dog as her pfp, so it's safe to assume that she may be her sister. Searching this exact username on twitter we find -

Twitter

Home

Explore

Notifications

Messages

Bookmarks

Lists

Profile

More

Tweet

← Tweet

Tracy Demonia @DemonicTracy04

Hey all! Welcome the newest addition to our family - NYMERIA! Oh she's so cuteeeee ❤️



10:29 AM · Oct 9, 2021 · Twitter Web App

1 Like

Search Twitter

Relevant people

Tracy Demonia @DemonicTracy04 Follow

Hey Tracy here. I loveeee dogs. Preparing for the SAT'sxD

What's happening

India national news · Last night
मंत्री कोई भी आर्यन खान को जमानत देने से विरोध करता है

Entertainment · Trending
#byjus

7,589 Tweets

Entertainment · Trending
#Boycott_SRK_Related_Brands

56.7K Tweets

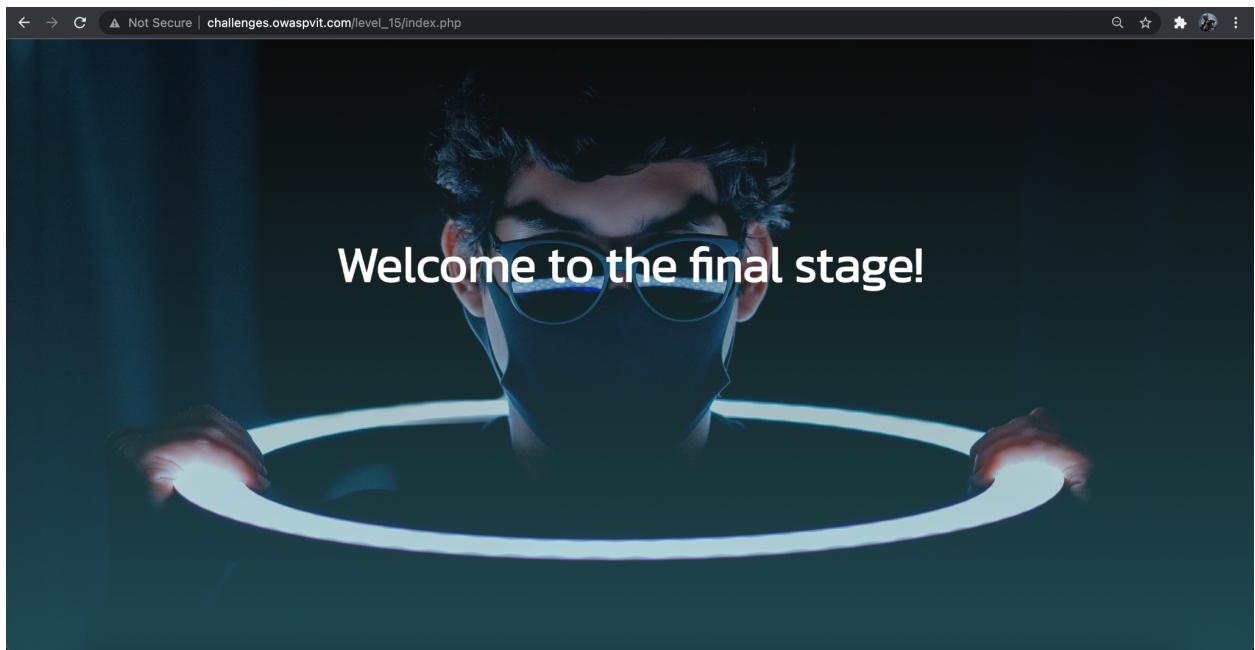
NewsBytes · Yesterday
Aryan Khan's lawyer specifies next step won't come until Monday

NewsBytes · Yesterday
Mukesh Ambani enters the elusive \$100bn club alongside

And voila! We found her dog's name.

Flag for level 14: WASP{nymeria}

LEVEL 15:



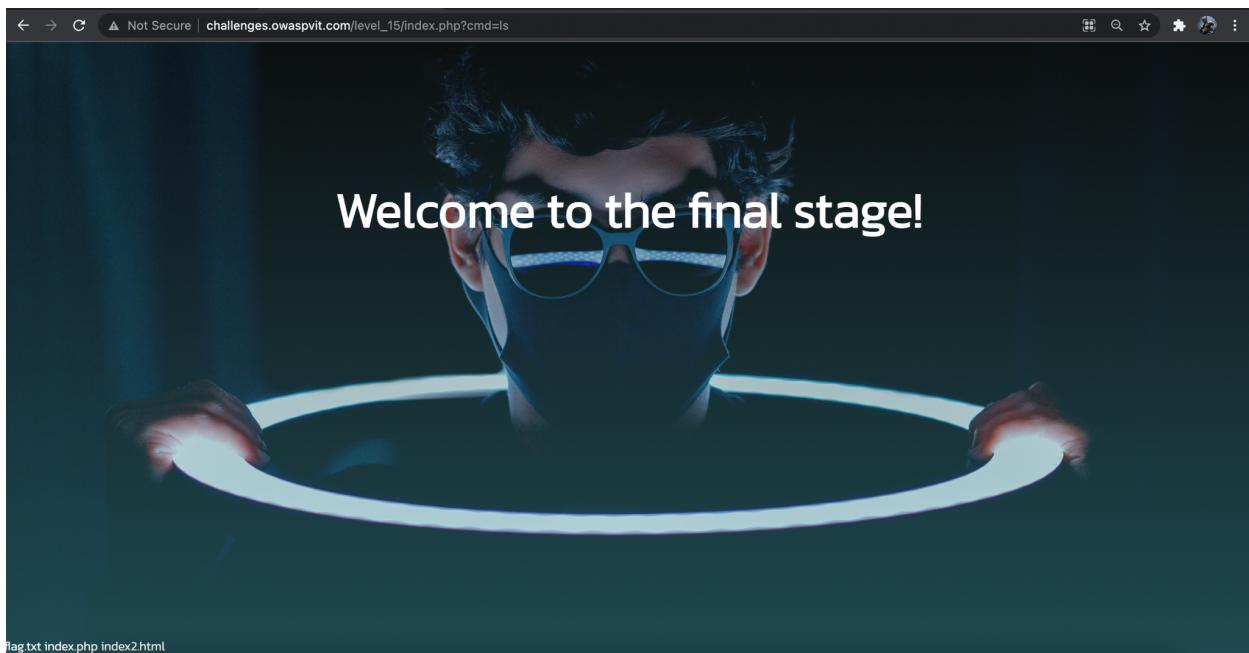
We have reached the final stage. Snooping around in the view page source we find another base64 encoded message - <!--
Q29tbWFuZCBpbmp1Y3RpB24gdXNpbmcgY21kIGlzIGEgZnVuIHRoaW5nIHRvIGRvIQ==
-->

Decoding this we get a clue to solving this round - Command injection using cmd is a fun thing to do.

We can see that level 15 has an index.php page. To learn more about this we can search command injection in php url.

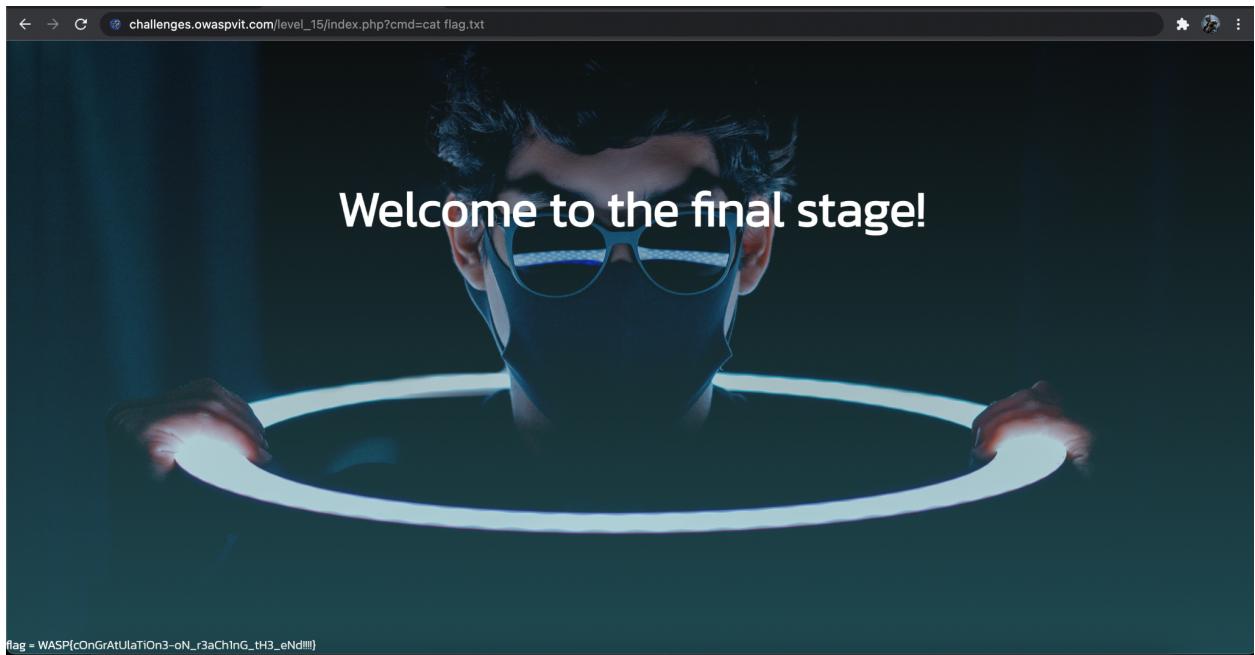
To solve this we will add - **?cmd=ls** to the url. So the new url looks like -
http://challenges.owaspit.com/level_15/index.php?cmd=ls

We have injected the command **ls** in the url, which lists all directories and files stored in this level.



At the bottom of the page we see that this level contains an index.php, and index2.html file, and a flag.txt.

To open the flag.txt we will use the **cat** command. Injecting ?cmd=cat flag.txt in the url - (new url - https://challenges.owaspit.com/level_15/index.php?cmd=cat%20flag.txt)



We get the flag!

Flag for level 15: flag = WASP{cOnGrAtUlaTiOn3-oN_r3aCh1nG_tH3_eNd!!!!}
