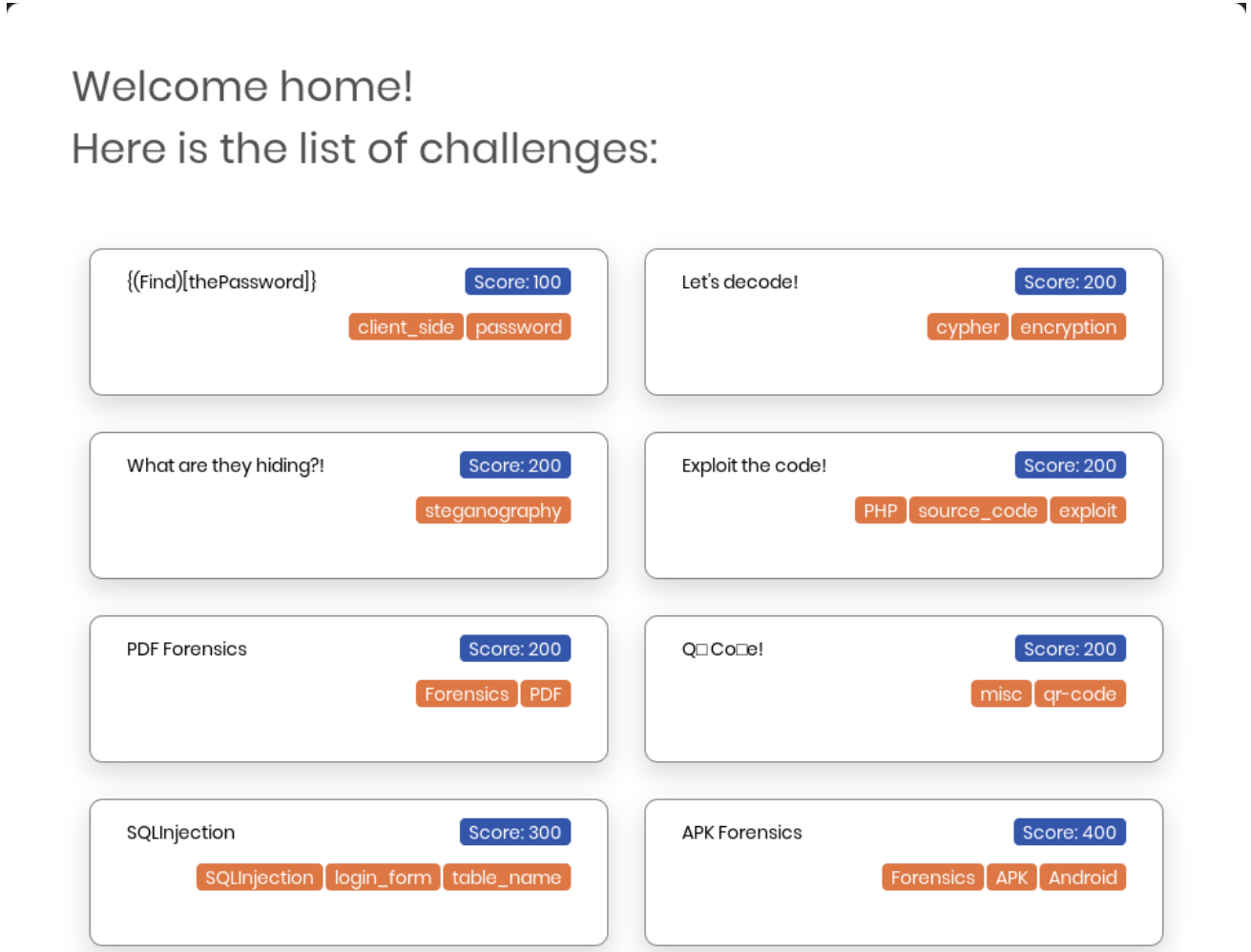
UNB CTF Competition Breakdown

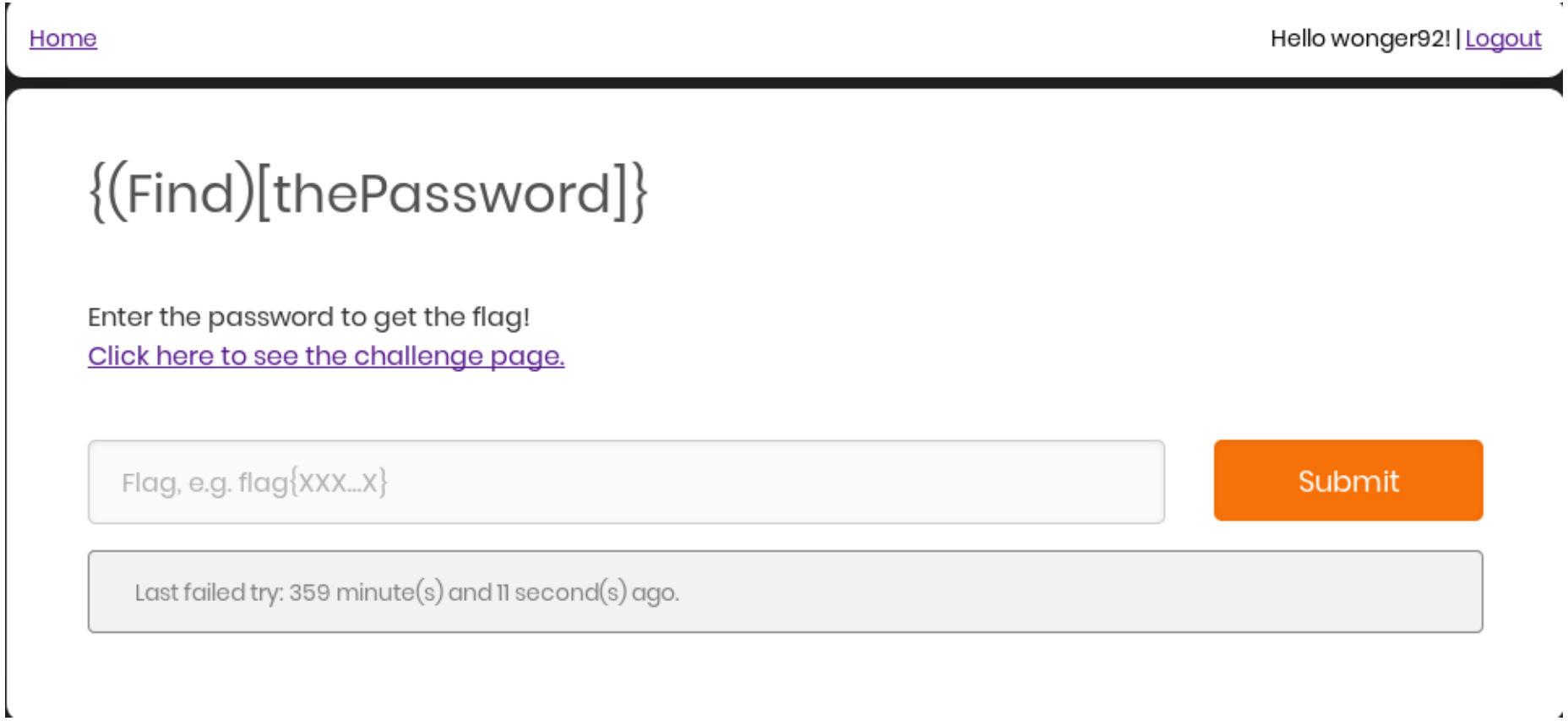
BY KEVIN WONG

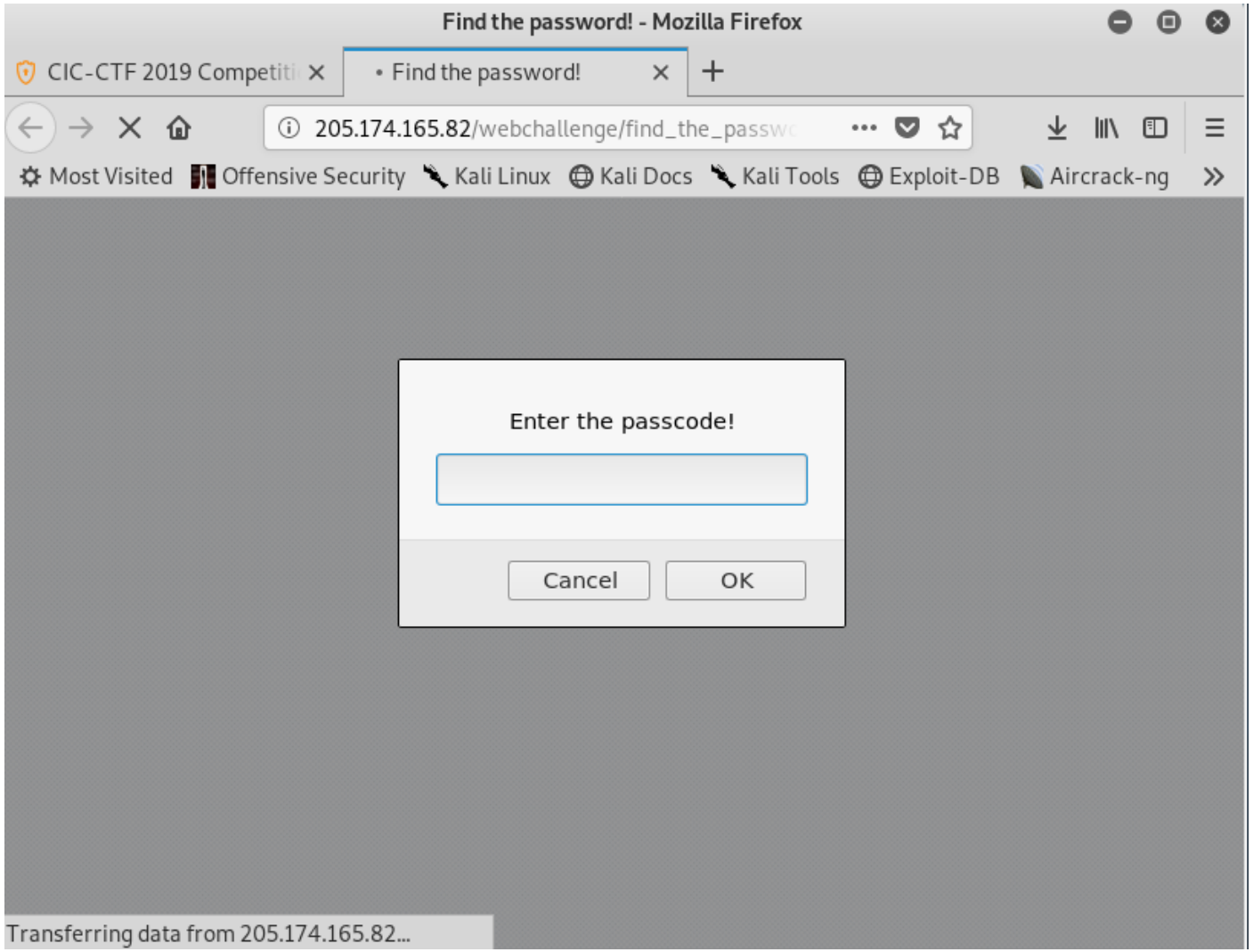


Challenges Page

Following slides have the challenges listed in order from left to right

Password Challenge (Pg 1)





Password Challenge (Pg 2)

Approaches:

- Brute Force via Hydra
- Simple password guesses (i.e: test123, password, etc)

‘Let’s decode’ Challenge

Let's decode!

Just for warming up, let's try to decrypt a simple substitution cipher! Except it's not that simple!
This is an ASCII to ASCII substitution cipher. It means the alphabet is the whole ASCII space (0x00 .. 0x7F).
Fortunately, both the message and the cipher just contain printable characters (not control characters).
However, the cipher (or in other words, the message) is case sensitive, and also contains a limited set of punctuations.
The space character is also a part of the alphabet, meaning the space caracter in cipher text is different from space character in the plain text.
All right! Let's get to work!
(Copy and paste the message below into a text editor and start analyzing it!)

```
HYP =EIBJG=EWYRYPTTTE#YO<PIFGRIFIYOJDE>YGEW NYW WEFAIJE= JJI<

DEXYYWEQYMDECAIJEIJEVYGPE@RI<KS IL>EMGCE@GODZELY= E=YP E<IMM PIJAEF UFDEPGJFEI<OYP EFA =T
```

Flag, e.g. flag{XXX...X}

Submit

No submissions yet.

see decode_challenge text file to solve for this challenge

Welcome home!
Here is the list of challenges:

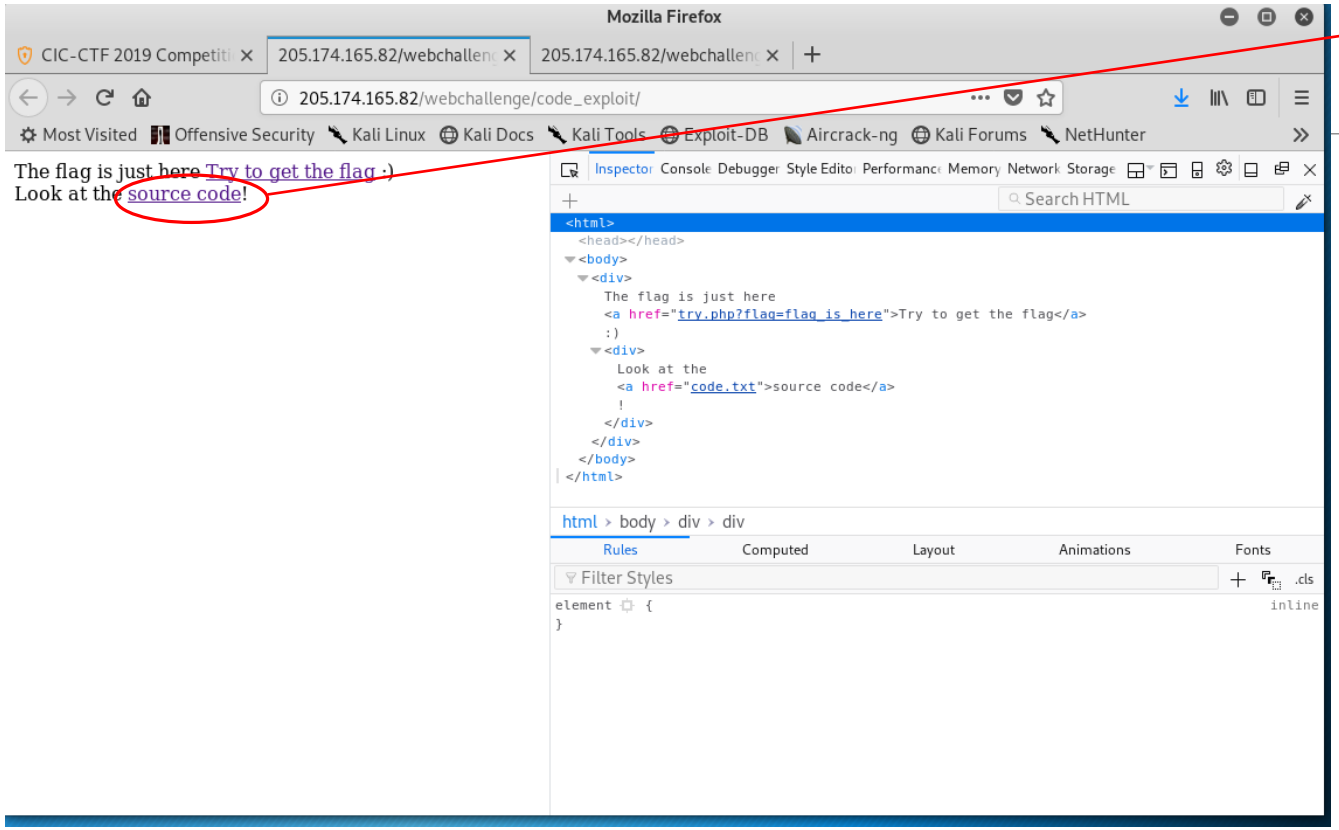
What are they hiding?!

The picture below has been retreived from a secret communication. We beleive the should be a secret message hidden inside of it, maybe a location or something. But we are not sure. Can you help us figure out what is the secret message?

Steganography Challenge

see steg_img file to solve for this challenge

‘Exploit The Code’ Challenge

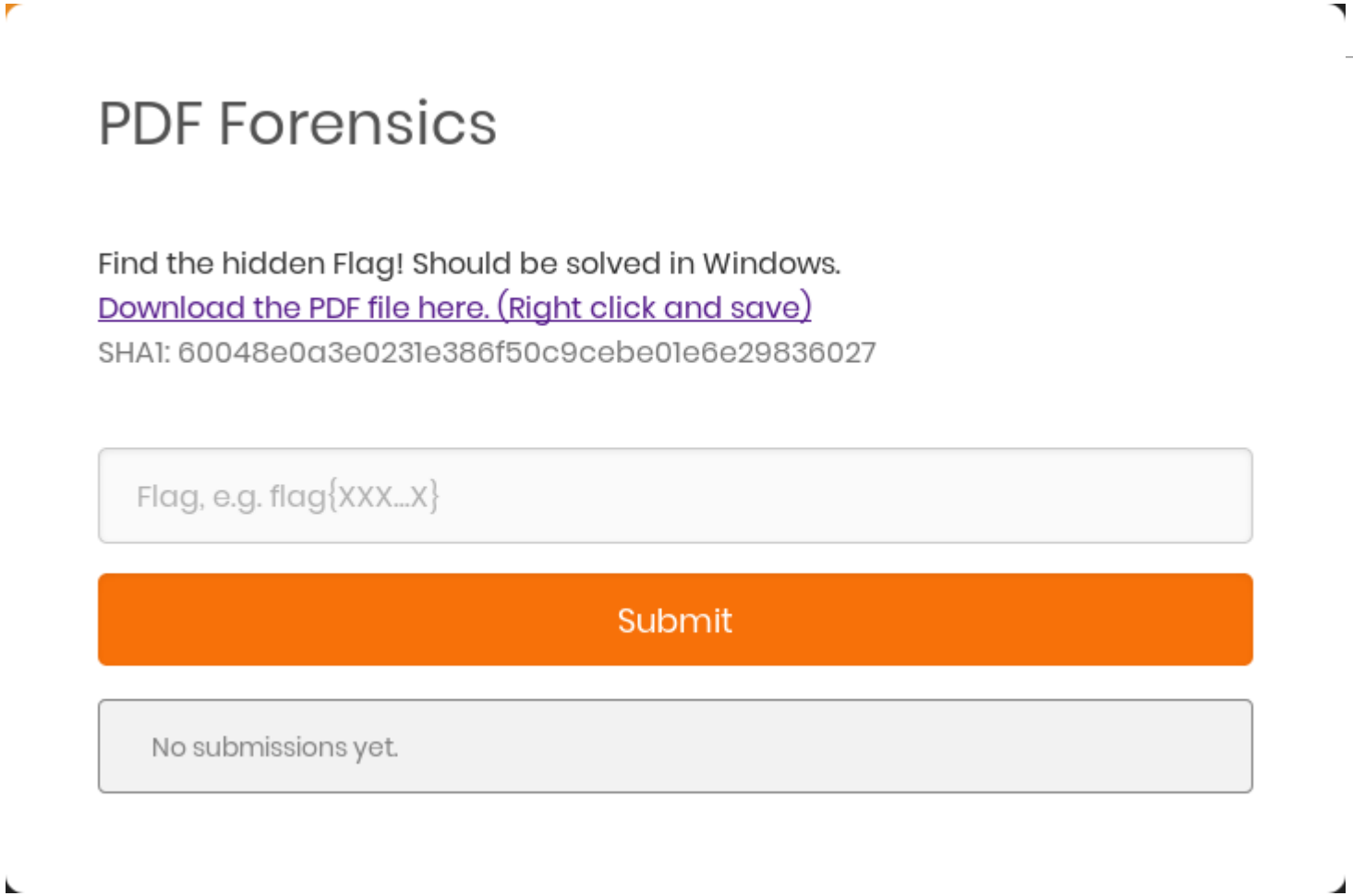



```php
<?php
$flag=$_GET["flag"];
$flag=urldecode($flag);


if(preg_match("/[_]+/", $_GET["flag"]))
{
        die('Try Again :(');
}
if($flag=="flag_is_here")
{
        echo "Flag_is:{**********}";
}
else
{
        die('Try Again :(');
}
?>
```

Welcome home!
Here is the list of challenges:

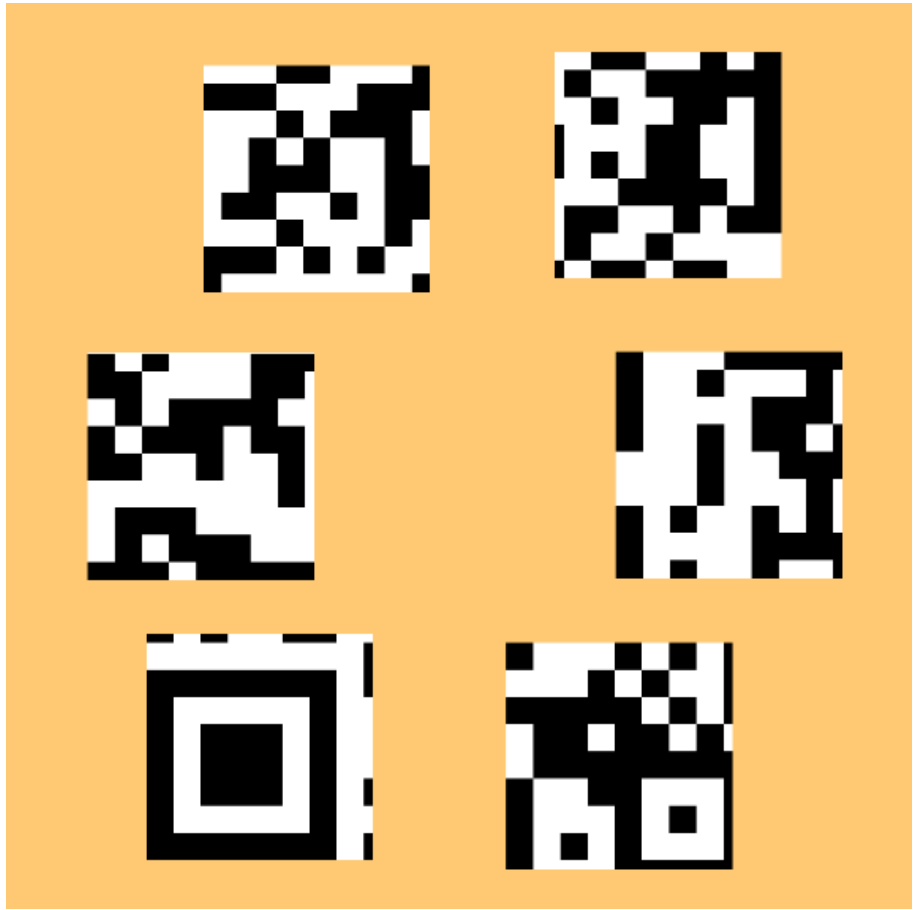
‘PDF Forensics’ Challenge



See pdf_forensics file *

Q□ Co□e!

We have retrieved these pieces of a qr code in an investigation. But it seems some of the pieces are missing.
Also, we don't know their correct orientation. (They might be rotated.)
Could you help us figure are what is the hidden message inside this qr code?



didn’t take individual images of the q-code itself. May have to use ‘Snipping Tool’ on Windows to isolate the six parts of the q-code to solve for this challenge

Welcome home!
Here is the list of challenges:

{(Find)[thePassword]} — Score: 100
client_side password

Let's decode! — Score: 200
cypher encryption

What are they hiding?! — Score: 200
steganography

Exploit the code! — Score: 200
PHP source_code exploit

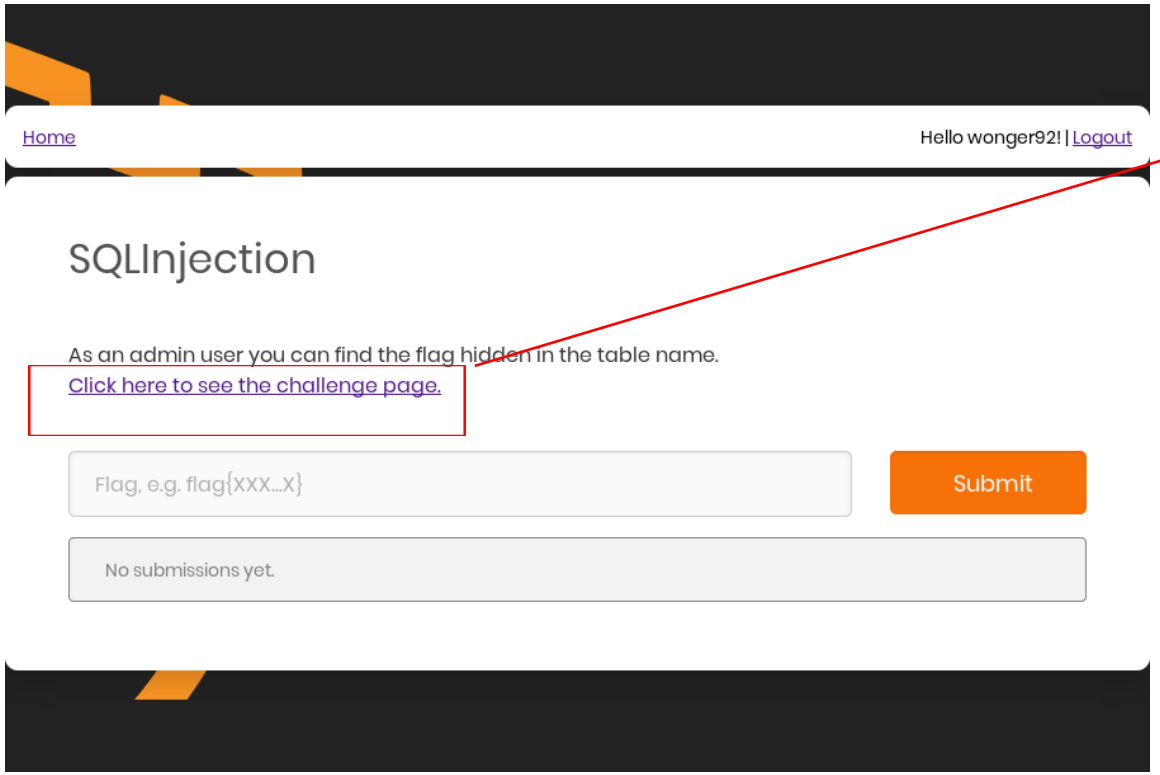
PDF Forensics — Score: 200
Forensics PDF

Q□Co□e! — Score: 200
misc qr-code

SQLInjection — Score: 300
SQLInjection login_form table_name

APK Forensics — Score: 400
Forensics APK Android

‘SQL Injection’ Challenge



As an admin user you can find the table name as a flag.

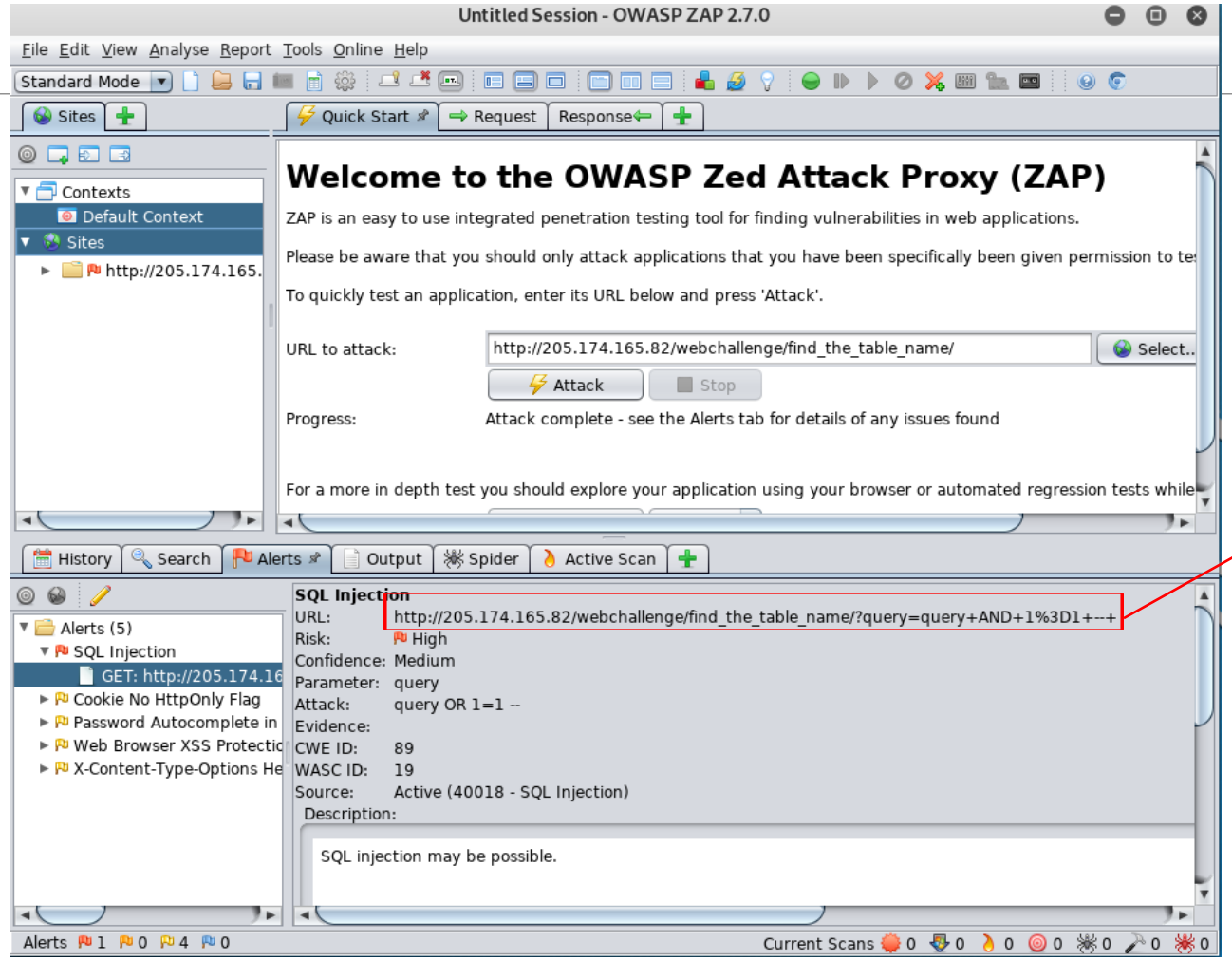
USERNAME:

PASSWORD:

LOGIN

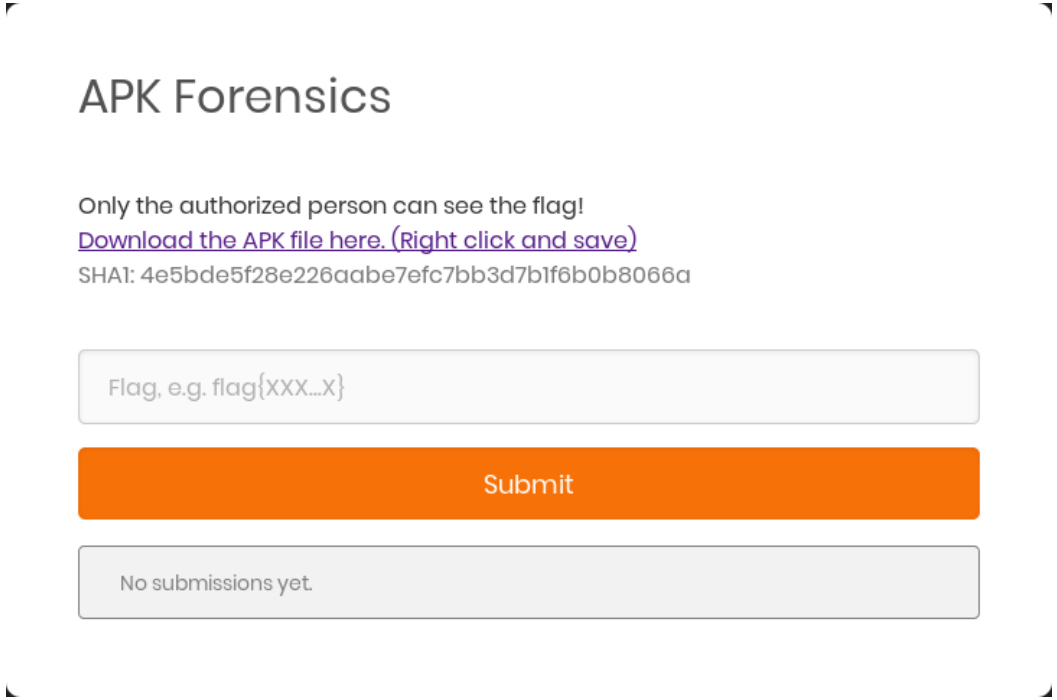
Limitation: Allowed to send one login request per second! If you try to send more requests in one second, they would be blocked!!!
Hint: The Table name's characters are in {a-z_} and it starts with "tiny".

Attack Approach – OWASP ZAP Results



Entered URL – Did not work ☹

Android APK Challenge



See app-release.apk file to do this challenge