A Platform for
Cyber Security Training
and
Holding CTF competitions

COMP 4801 Final Year Project
Department of Computer Science
The University of Hong Kong

Name: Han Yu
UID: 3035141736
Supervisor: Dr. S. M. Yiu
ABSTRACT

Cybersecurity issues is now attracting more and more attention as the rate of cybercrime increases. Meanwhile, security specialists are in high demand and will be playing a necessary role in the industry.

To promote and improve computer security education in the University of Hong Kong, HKU is planning to establish a customized Capture the Flag (CTF) platform used for CTF competition and exercise. There are two main open source CTF platforms, CTFd and FBctf, that can be utilized by adding personalized functionalities and embedded with challenges.

The project is aimed to seek a way for better combing the challenges with CTF platform so as to make it adaptable when holding a CTF competition. In the first part, two CTF platforms, CTFd and FBctf are analyzed carefully based on their installation, functionality and performance, after which FBctf is chosen to be the target CTF platform in this project. In the second part, four selected challenges which are vulnerable web attacking problems without answers will be lined with FBctf and their source code will be modified to be more friendly to new learners. What’s more, some additional functionality, including timer on web page and recording of user input, will be implemented based on the web server.

The category of challenges that are of the concern of this project is web attacks. Illegal modification of cookies, source code and URL of the web server are included, and SQL injection vulnerabilities are also introduced. In future work, more kinds of web attacks, such as XSS and CSRF, will be analyzed and embedded with the platform.
ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. S. M. Yiu for his patient guidance and continuous support, which throws a light on the project objective and helps me to overcome encountered difficulties. I am also very grateful for comments given to my group mates by Dr. K. P. Chow in interim presentation and they are useful for my work in the second part of the project.

Especially, I have to show my sincere thanks to Zhang Yuan, my senior friend, who offered a lot of help for answering my technical questions. Without him, I can’t get such progress during this project.
# Table of Contents

Abstract

Acknowledgments

1. Project Background and Objective 6

2. Analysis of FBctf and CTFd 8
   2.1 Administration Panel 8
   2.2 Team Panel 11
   2.3 Source Code 11

3. Theory: Web Attack Challenges 14
   3.1 Challenge 1: Webpage Code Modification 14
   3.2 Challenge 2: Direct Modification of URL Console 19
   3.3 Challenge 3: Brute Force 20
   3.4 Challenge 4: SQL Injection 23

4. Implementation 27
   4.1 Modification of Vulnerable Web Server for Attacking Challenges 27
      4.1.1 Timer and Challenge 1: Webpage Code Modification 27
      4.1.2 Challenge 2: Direct Modification of URL Console 28
      4.1.3 Challenge 3: Brute force 28
      4.1.4 Input Record and Challenge 4: SQL injection 30
4.2 Combination of FBctf and Web Servers

5. Future Work

6. Conclusion

References
1. Project Background and Objective

Along with the widely use of information technology in both business and daily life, data streams flowing among electronic devices contains valuable private information and any loss or damage may cause huge cause to both individuals and organizations. Ranging from the software platform embedded in everyone’s laptop to the electronic gate people entered every day, as long as it is under the control of the programming logic stored inside, it can be the target of cyber criminals.

Cybercrime is now attracting more attention than traditional crimes because of the serious loss it can cause. In contrast, there is a lack of cyber security professionals all over the world to provide proper solutions for such problems. According to the statistic, there would be a vacancy of 3.5 million cybersecurity jobs in 2021 (Cybersecurity Ventures, 2017). To meet the need of cyber security professionals, many companies and universities are now holding cyber security education programmes as well as competitions to enhance the skills of young people who are planning a career towards this field. Capture the Flag (CTF) is one of these events aimed at selecting skilled students or professionals in different areas in cyber security. It has attracted a lot of young participants and professionals all over the world to take part in since it is first held all around the world in 1990s. (Harmon, 2016)

CTF has evolved into various forms and keeps embracing popular topics in the industry during its history. The narrow meaning of CTF refers to either attack and defense or jeopardy style competitions, while it is more often to be used to refer to all the competitive events including games and other before-competition training materials nowadays. As one of the traditional styles of CTF competition, attack and defense contest requires teams to patch their own vulnerable server as well as seeking for ways to hack other servers, while jeopardy style allows teams to get marks by solving a set of challenges provided within limited time.

As one of the oldest CTF, Defcon adapts the traditional form of CTF competition, including both attack and defense style and jeopardy style contests. In Defcon CTF teams are first required to solve a set of challenges online and then the selected ones are invited to take the onsite attack and defense contest. (Vulcan) Attack and defense style competitions are much more difficult to hold as building available vulnerable virtual machine for participants is highly time consuming.
In addition, it is often arranged as the onsite competition considering the coherency of the whole contest system which consists of multiple virtual machines and other components.

This project aims to design and implement an online CTF platform supporting both of the traditional styles competitions. The whole system would be allocated in the Cyber Security lab in the University of Hong Kong and contribute to the cybersecurity training on campus. It should be able to hold the onsite attack and defense style competitions for multiple users on campus.

Furthermore, based on the education purpose, some CTFs are accompanied with related lectures and tutorials for being a qualified teaching tool for cybersecurity knowledge. (Werther, Zhivich, Leek and Zeldovich, 2011) However, most of the cybersecurity trainings are held temporarily or even consists of only a few workshops in weeks. Other training programmes, like picoCTF, are in the form of electronic games, facing only high school students or even lower age. There is a lack of the stable online platform that provides comprehensive cyber security education for self-taught young people as well as competitions for exercising practical skills.

Our project aims to develop a CTF platform which holds both of the competitions of two traditional styles and cooperates with online and offline training materials to assist the students in self-taught learning process. The training materials will be presented both in the form of online knowledge sheets and in offline workshop by tutors. Such workshop may hold several times per year depending on the actual situation to help students get familiar with the CTF platform and get a quick start for online self-taught learning. Additionally, in order to provide students with hand-on exercises about what theories they have learnt, the competitive contests provided by CTF platform will be highly matched with the training materials and thus encourage students to apply theories to practical problems.

With the close cooperation of online contests and learning materials, students are expected to establish solid cybersecurity theory foundation and develop a high level of practical skills towards real life challenges. Additionally, it could also be used by the university to evaluate the skills of students through CTF competitions supported by the platform.
2. Analysis of FBctf and CTFd

By now, the default functionalities of platforms FBctf and CTFd have been analyzed and this section will present the comparison of game interface design, administrator control and source code.

2.1 Administration panel:

1. Registration control:

They both provide options for free registration or reserving the rights for administrators to manually register new teams. In addition, FBctf also can approve new teams by registration tokens sent by administrators. CTFd discloses some of the functionalities, including certain pages, challenges, to anonymous users while FBctf only serves registered users.

2. Game control:

Both FBctf and CTFd keep the records of the game start and end time, but only FBctf includes timer in game panel and can hold events in competitive way. During the game, FBctf supports in time announcements and game pause or termination. On the other hand, CTFd is more suitable for a training platform as it sets no time limit for solving challenges and provide a interface while clearly divide challenges into different challenges.

3. Database refresh and game storage

Both FBctf and CTFd support game import and export. The content could be full game or particular part of settings, such as team logos and challenges. Each time administrators make challenges to CTFd, it will be automatically saved with the account and when come back to CTFd, these changes will remain. However, changes made to FBctf will lose after restart of the server and recompilation of the source code. So it is necessary for administrators to import necessary part of the game before exit and import it again after next login.

4. Challenge Setting

FBctf support different categories of challenges by default, including quiz level, flag level and base level with different setting layout, while CTFd use the same template for all challenges. For FBctf, Level Quiz is the basic level designed for challenges with simple question and simple answer format. The basic layout includes title, description, flag, keep point, capture point fields, and hint, hint penalty, countries (a symbol for challenges) fields are optional. Level Flags and Bases are built based on this layout but added attachment and link fields.
Level Flags challenges are mostly associated with additional files or links to provide interactive problems during the competition. Teams may be directed to vulnerable servers for exploitation or manipulate certain files.

Unlike jeopardy style challenges involved in Level Flags and Quiz, a special King of the Hill game type is introduced in Level Bases. The link will direct teams to a target server and each team compete to take control of the server. The scores depend on the time that they can hold the target server.

In the contrast, CTFd use one default template for creating challenges, which includes name, category, value(points), static/regex key (flags), file attachments and hints. It supports two different type of answers, static key which is the simple answer in text format and regex keys which is compared with regular expressions. As shown in figure 1, Regex keys is checked by match() method in python and administrators have to write specific regular expression for answers of each challenge.

```python
class CTFdRegexKey(BaseKey):
    id = 1
    name = "regex"

    @staticmethod
    def compare(saved, provided):
        res = re.match(saved, provided, re.IGNORECASE)
        return res and res.group() == provided
```

*Figure 1: Part of the source code of CTFd, which shows the correctness check for regex keys*

But FBctf only support simple text answers in Level Quiz and Level Flag.

5. Challenge Setting

As introduced in the above, FBctf uses different levels to differentiate challenges through their supplement materials and format. In addition, both FBctf and CTFd use field tag to indicate challenges in different knowledge area, including programming, cryptography and web hacking.

*Figure 2: Challenges page of CTFd.*
CTFd directly group challenges based on tags and present them in groups. In contrast, FBctf uses countries in World Map to represent each challenge and allow teams to use filter to select challenges in particular category as shown in figure 3.

Also as shown in the figure 2, solved challenges are colored in green in CTFd while FBctf use status filter to select solved and unsolved challenges.

6. Grading Mechanism

CTFd simply assign points to each challenge and add up those of solved problems for final score, while FBctf employs a more complicated grading mechanism for different levels of challenges.

For levels, FBctf sets two fields Default Bonus and Default Bonus Dec. Default Bonus represents the value received when the first team reaches particular level and scores received by following teams will be Default Bonus minus corresponding Bonus Dec. The later a team arrives at a new level, the less bonus it is awarded.

Additionally, FBctf also uses similar mechanism for each challenge. Bonus field represents scores for the first team to capture this challenge and -Dec field is used to calculate scores for following teams. And both -Dec and Default Bonus Dec can be set to zero, that is all teams arrive at the same stage can get equal scores regardless of their precedence.

As discussed in the above, FBctf needs to assign score to teams based on the length of time they hold the control of the target system in Level Bases. So it needs some mechanism to keep track of the duration time. In control panel, it includes a field called Base Cycle, which represents the number of seconds between two Base check. Each time a Base check is made, team holding the target system will be awarded with certain amount of scores.
2.2 Team Panel:

1. Game board:

Both FBctf and CTFd includes basic information fields, like team account, scoreboard and challenges. As shown in figure 4, CTFd separates these fields to different pages and allow administrators to add new pages with HTML and CSS.

![Figure 4: page options in CTFd](image)

But FBctf includes all fields on the same page and add additional timer to show the remaining time of the game, announcements for administrators to communicate with teams and activity block to update information, as shown in figure 5.

![Figure 5: gameboard for competitions](image)

2.3 Source Code

1. CTFd

CTFd is developed with python, which is easy to read and understand. It is also very friendly to customization as it leaves a plugin folder in source code for developer to add new functionalities. Plugins are supposed to be implemented as Python modules under the instructions in CTFd website.
In figure 6, there are two folders challenges and keys in plugin folder now. And the file _init_.py is the main code that is to be compiled. To add a new plugin, developers just need to add additional folder containing the new plugin and then update the _init_.py file to make sure new plugin will be included in load() method.

Similarly, CTFd also supports adding and modifying routes, replacing templates and registering assets.

What’s more, when the platform is running, information about commands submitted to web server will show in the terminal window and it is easier for developers and administrator to keep track of what is happening.
FBctf is developed with Hack, a dialect of PHP language, which is not very familiar for most developers. It doesn’t provide plugin folder for developer to directly add new functionalities. But if one gets familiar with Hack and PHP language, the source code is rather well-structured and easy to modify. As shown in the below, each controller class is in separate files and modification will have little influence on other classes.
3. Theory
In this section, four chosen challenges will be introduced and theories behind them will be discussed in details. In particular, some solutions to the problems will be shown.

These chosen challenges include direct modification of web source code and cache, simple cryptography function, SQL injection and python script programming.

3.1 Challenge: Webpage Code Modification
In this challenge, three different stage are involved and participants have to conquer all the three problems to get the flag.

Stage 1: Inspect the page source code
First to analyze question description in stage 1.

Figure 9: Stage 1 question description
As shown in figure 9, there is a button with the text Proceed to Step 2 but the button is unable to be clicked, which directs the user to look into source code for more details.

Figure 10: right-click options
There are two options related to webpage source code as shown in figure 10, but view page source option can only show users the source code with read access while inspect option allows modification of source code.
Figure 11: source code of current webpage

Figure 11 shows the source code of current webpage, as highlighted, the code represented Proceed to Step 2 button is as the following:

```html
<input type='submit' name='are you ready?' value='Proceed to Step 2' disabled>==$0
```

In this line of code, name attribute is needed to be changed from ‘are you ready?’ to ‘ready’, with hint from the question description. At the same time, ‘disabled’ attribute should be deleted to enable the button to be clicked.

After that, click the button to move to Stage 2.

### Stage 2: Decode and change session value

**Figure 12: Question description of Stage 2**

#### Step 1: decode the string

First there is a cipher string to be decoded to finish this question.

```
UE9XRVJGVUwgQ1RGRVI=
```

To decode this cipher string, base64 encode and decode method needs to be introduced.

#### Base64 encode and decode Method:

Base64 is a kind of encoding method which transforms text consisted of characters in ASC II format into binary numbers and then into characters of base 64 table.
As shown in the ASCII table in figure 12, the red characters are corresponding to an index number and they can be interpreted as the binary string based on this index number.

Take characters CTF as an example:

C is of index number 67, which can be interpreted in bit pattern: 01000011
T is of index number 84, which can be interpreted in bit pattern: 01010100
F is of index number 70, which can be interpreted in bit pattern: 01000110

Then to transfer to base64 string,

```
<table>
<thead>
<tr>
<th>Value</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Q</td>
</tr>
<tr>
<td>53</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>R</td>
</tr>
<tr>
<td>6</td>
<td>G</td>
</tr>
</tbody>
</table>
```

Figure 13 shows the encode process of string CTF. After converted into binary string, every six bits are grouped together and then calculated the corresponding value.

Then as shown in figure 14, based on the values and character matching mechanism, string CTF is finally encoded as string Q1RG.
The decoding process will be the same but just reverse the process of encoding. Characters are first interpreted based on base64 then group binary bits of size 8 and then match them with the help of ASCII table.

When dealing with related problems, there are some characteristics that can help to analyze whether the string is after base64 encoded, as shown in the following:

   a. The size of cipher string is always a multiple of four
   b. The string will probably end with ‘=’

When the size of plaintext is a multiple of three, its characters can be divided into groups of 3 and each group corresponds to four characters in cipher string. But when there are some characters left after these groups, it will be grouped with bit 0 and then be translated into group of four cipher characters. In this case, for cipher bits transferred from all 0 bits, they are represented by ‘=’. This is why in the end of some cipher strings, there are one or multiple equal signs.

Then back to the stage 2, the cipher string is of size 20 and it ends with equal sign. Then it is very likely that it is the cipher string after encoded by base64 encoding method. Then users can either decode it by themselves following the above steps or use online base64 decoder to decode it.

After decoding, the value is POWERFUL CTFER.

**Step 3: Set cookie value**

After getting the plaintext, this step is going to change the cookie value stored to proceed.

![Figure 15: Cookie stored: PHPSESSID and mental](image)

As shown in figure 15, a cookie named mental is stored in local browser. And same as what is said in question description, its value is CONFIDENT. Then modify its value to POWERFUL CTFER.

Then take a look at the source code.
Figure 16: source code of stage 3.

As shown in figure 16, when the button is clicked, a form is submitted if `onsubmit` attribute is true. Then we are required to look into the JavaScript code and modify the value of this function to `true`. After that, click the button and proceed to stage 3.

Stage 3: User Agent

As shown in figure 17, in this stage, user is required to find the agent name and then modify it.

Figure 17: question description of stage 3

In this part, basic information about user agent and how to modify it will be shown.

User Agent in HTTP

User agent is a string which consists of information about current browser and sent to the server in HTTP requests. It is used by the web server to choose the response that can be compatible with the browser.

The following is the response message sent from the server:

```
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36" -H "
```

It can be found in the Network window after clicking inspect as shown in figure 18.

Figure 18: find the user agent message
The cURL message copied:

curl "http://localhost/test/1/" -X POST -H "Cookie: mental=POWERFUL CTFER; PHPSESSID=fitjkf0ag6f0737f6eaut6m494" -H "Origin: http://localhost" -H "Accept-Encoding: gzip, deflate, br" -H "Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7,ja;q=0.6" -H "Upgrade-Insecure-Requests: 1" -H "User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36" -H "Content-Type: application/x-www-form-urlencoded" -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8" -H "Cache-Control: max-age=0" -H "Referer: http://localhost/test/1/" -H "Connection: keep-alive" -H "Content-Length: 0" --compressed

Find and modify the user agent value

In this stage, there is no specific obvious hint about the user agent. So there is a suppose that the function used to find it is programmed in JavaScript file. After trying several times, running decode_agent() function in the console reveals the user agent information as shown in figure 19.

![figure 19: find user agent](image)

Then cURL command is used to change the user agent information.

cURL command is used to send requests or some part of the requests to the web server.

In this case, modify the blue part of above copied cURL message and send it back to the terminal will receive the response message, where the flag is contained, as shown in figure 20.

![Figure 20: flag contained in the response message.](image)

Challenge 2: Direct Modification of URL Console

As shown in figure 21, challenge 2 requires users to collect certain amount of different categories things to get the flags.
After trying many times, participant can find that is seems impossible to get the required dye and bricks to get flag part two. So there must be some other ways to get the required things.

And participant may observe that the collected item always appear in the URL console as shwon in figure 22.

Then it is reasonable to suppose that the mechanism behind this challenge is to use HTTP GET method to craft items.

Then to get certain amount of items, we can directly modify the URL console to collect.

After collecting enough amount of the four materials, the flag will be shown in the webpage.

**Challenge 3: Brute force**

In the challenge, users are needed to choose the best monkey, with ideal name, IQ and body strength as shown in figure 22. And there is a file named *monkey name list* provided and users are expected to try all the name inside this file.
Each time user refreshes the browser or clicks on the superman picture, a new set of IQ and Body strength parameters will be shown. At the same time, the entered monkey name and other parameters will be checked by the server. Only if the ideal name and full marks of IQ and body strength are meet, the monkey is chosen and the flag will be revealed.

In this case, there are a lot of names listed in the monkey name list file and very probably also many sets of possible body conditions. So it will takes plenty of time to try all the possibilities.

What’s more, based on the performance of the website, it can be supposed that the selection of body parameters are all of random choices, which adds to the difficulty of finding the ideal monkey manually.

Based on the analysis in the above, a python script which check all the possibilities will be appropriate to solve this question.

The solution of this challenge is shown as in figure 23.
It will be explained in details in below:

```python
flag_re=re.compile("flag")
```

In this step, it is reasonable to assume that when succeed, the string flag will appear in the HTML response so that the user can get the flag. re.compile() method converts a regular expression pattern to a regular expression object such it can be used for comparison in re.search() or match() method.

```python
client=requests.Session()
url="http://localhost/test/3/"
data=urlopen(url+"monkey%20name%20list.txt").read()
```

This opens the file named “monkey name list.txt” in the specified url. One special point to note that, the stored monkey name list.txt file in the server contains space and when it interpreted as the URL, these spaces are decoded to %20. So when constructing requests containing spaces, those spaces should be replaced by %20.

Note that the result data is in the format of bytes object.

```python
text=data.decode('utf-8')
```

This line is special for python 3. Python3 treats bytes object and string very differently. So if a bytes object is operated as a string, it will encounter an error. So python 3 needs the decode() method to convert bytes object to string. Especially, the decode method is utf-8 by default, so if used another decode method, it needs to be specified.

```python
line=text.splitlines()
```

This code split the string into a list of strings based on newline operator.
html=client.post(url,data={'name':item}).text

This line of code send the HTTP POST request to the specific url with data. The response will be stored in html object.

match=re.search(flag_re,html)

search() method is used to check whether flag_re object, which contains flag, is included in html.

There are typically two functions used to match objects in python, search() and match(), which actually behave differently. match() method focus the anchor at the beginning of the string. If no match is found starting from the first character, it will fail. But search() method will check for the target over the whole string to find the match position.

If a match is found, the html object is printed, where the flag will be disclosed.

**Challenge 4: SQL injection**

In the challenge, users are required to answer some questions to catch the flag as shown in figure 24.

![HKU CTF Training Platform](image)

*Figure 24: question description of challenge 4*

It is impossible for users to answer the questions directly, but there is a FAQ page available for users to find the answers as shown in figure 25.
The search engine will match the questions descriptions stored in FAQ part with user input and reveal the answers. A null user input will reveal figure 26.

Clearly, these answers are not helpful for the solving questions on first page. But from FAQ page, there is an access to the database where the answers are stored and perform SQL injection may helps to disclose the answers needed.

It is reasonable to guess that this search engine use the Like SQL syntax to match the search keywords.

**Step 1: Differentiate Single Quotation or Double Quotation**

To enclose the LIKE closure for further injection needs to know which kind of quotation is used in this SQL syntax. There are two possibilities:

LIKE ‘%[user input]%' or LIKE “%[user input]%”

To check for this quotation, input:

‘ AND 1=0 which reveals SQL error message, shows that the whole syntax is not closed by single quotation and it is very possible that LIKE syntax is the first case

Step 2: Check the number of columns in the database

Now suppose the syntax is

SELECT [unknown] FROM [table_name] WHERE [question] LIKE ‘%[user_input]%'
Since only the last part can be modified by user, it is reasonable to consider using UNION syntax for injection. For using UNION, the number of columns need to be known.

The following input is used to check column numbers:

`order 6 ; --[one space]`

It tries to order the selected data by the sixth column attributes value. But it returns the error: `unknown column 6 in order clause`, which shows that there is not sixth column in this table.

One special thing to note is, for using -- to comment following SQL syntax, a space is needed to be left at the end.

Then use the input:

`order 5 ; --[one space]`

which is executed without error and this shows that there are five columns in the table.

**Step 3: Find the database name**

This step is going to find the name of the database. And the method database() is used to show the name of database in mysql.

The following syntax is used in this step:

`and 1=0 union select 1, 2, 3, 4, 5 ; --`

`and 1=0` is used to disable the default search SQL syntax

`select 1, 2, 3, 4, 5` will return the data 1, 2, 3, 4, 5 in corresponding columns.

The execution result is as shown in figure 27

![Figure 27 Execution result of ' and 1=0 union select 1, 2, 3, 4, 5 ; --](image)

In search engine, although all five columns are selected but only two of them are displayed in the webpage, representing the question description and answer. Since only data 4 and 5 are displayed, it can be deduced that column four stores question description while column five stores answers.
So if we put database() method inside the syntax it can be executed and its value can be revealed as the following:

`' and 1=0 union select 1, 1, database(), 1 ; --`

Then the database name is found to be **hkuctftp_db**.

**Step 4: find the table that stores questions**

The following syntax is used to find the table names inside this database:

`' and 1=0 union select 1, 1, 1, table_name,1 from information_schema.tables where table_schema = 'hkuctftp_db' ; --`

information_schemas.tables is a default syntax to show all the table names inside the database.

After execution, we can get the table names:

intro, solution_table, user_account, user_log.bean

The second table stores the answers required.

Step 5: Retrieve the answers from table

Similarly, use the following input to get the column names of the table

`' and 1=0 union select 1,1, 1, column_name,1 from information_schema.columns where table_name = 'solution_table' ; --`

And use this to retrieve questions and answers:

`' and 1=0 union select 1,1, 1,question, answer from solution_table ; --`

The final result is shown as figure 28:

![Figure 28: questions and answers of challenges 4](image-url)
4. Implementation

This section, the main logic of the source code of the challenges and the modification part will be discussed. What’s more, details about linking fbctf and the web servers in competitions will be included.

4.1 Modification of Vulnerable Web Server for Attacking Challenges

4.11 Timer and Challenge 1: Webpage Code Modification

a. Decode and Encode function in javascript

In original source, there is no function used for encode or decode in the javascript. So users can only rely on manually decoding or some online decode tools. This may not be appropriate for the situation when users are not allowed to connect to world wide network (WAN) to use online decode too. So the decode function is added in the javascript file such that user can run the function in console to get the decode result as shown in figure 29.

![Figure 29: Run base64decode() function in console to decode strings](image)

b. Add timer for users’ reference

FBctf platform supports count-down timer in the challenges panel by default. But when follow the link to jump to another outside webpage to answer questions, it is not easy to get the clear picture of the time spent. So a timer is added to the web server to show the time user spent on this webpage.

As an example, the timer is embedded with the webpage of challenge 1 as shown in figure 30.
4.12 Challenge 2: Direct Modification of URL console

In this challenge, the required amount of craft item 2 can never be satisfied so the users are required to direct modify the URL in console to obtain enough items.

In the original source code, it forces the last two items, dye and bricks, never to be collected as shown in figure 31.

Figure 31: when Walk is clicked, only the first two items may be chosen.

However, the problem is, users are not sure the representation format of other items in the URL if they never appears.

So this part of code is modified as shown in figure 32 to ensure that all four items have have the opportunities to be collected but the second craft requirement can never be matched.

Figure 32: Modified code for Walk action

Challenge 3: Brute force

a. Random names and body conditions

This challenge selects different body conditions each time the page is reloaded or the picture is clicked. In original code, the ideal name is always set to a specific value, such as Jordan.
However, this is not flexible when multiple teams are working on this question at the same time, the name should be different for each player.

So the following code in figure 33 shows the way to generate a new ideal name for each time a session starts.

![Code for generating different ideal names for each session](image)

**Figure 33: Generate different ideal names for each session**

When users connect the web server, each will be assigned a different ideal name. And this name remains unchanged when the page is refreshed or the superman picture is clicked.

The body conditions are also generated randomly as shown in figure 34.

![Code for random generated body conditions](image)

**Figure 34: Random generated body conditions**

Each time, a new body condition set will be created with value between 60 and 100. Then the pick time, which represent the number of times a new set is assigned, will increase. When pick time is a multiple of 88, the ideal body conditions will be found.

b. Object match in python script to find the flag
In original answer, it uses *ideal name* and *ideal character* to carry out object match to find the flag page. But it is not specified in the question description. So in the answer provided in this project, direct matching with *flag* will be utilized to find the flag page.

**Challenge 4: Input Record and Challenge 4: SQL injection**

In this challenge, a database is set up based on the requirements of the question description. To set up this database, administrator needs to run the file *setup.sh* first, which contains the commands for setting the database required each time for this challenge.

What’s more, additional tables are added to the database for recording the inputs of the students.

![figure 35 SQL syntax for creating tables for recording](image)

Administrator needs to manually add team names and passwords in this file. Then after users enter input in both question page and FAQ page, the input will be stored into the database.

### 4.2 Combination of FBctf and Web servers

This section, it will introduces the steps to link the FBctf and web servers in real situation and utilize the built-in functionalities of FBctf.

One prerequisite for this set is that they cannot be hold on the same machines. Facebook can only be hold on Ubuntu 16.04 platform. And in this project, XAMPP us used to hold the web servers. Installing Xammp on linux will create the Lampp folder under /opt, however, FBctf can not be installed with this Lamp folder exiting. So it is recommended to hold the web servers and FBctf platform on two different machines.

a. **Start Web Servers**

In this project, Xammp is used to hold the web server. The original files of web servers are wrapped in Docker, a software used to hold apache and mysql applications. But utilizing docker
requires proper setting of specific docker commands, which can be very confusing for most of the users.

It is developed by Apache friends to mainly hold web related applications, such as Apache server and MySQL database. It provides a GUI interface, which is very user-friendly for all users to control the start and stop of all the servers, as shown in figure 36. The commands used for Xampp is nothing special compared with those for linux system.

![Figure 26: XAMPP Control Panel v3.2.2](image)

What’s more, it also provides a GUI interface, phpmyadmin, for visualizing mysql syntax and makes it easier to debug.

In this project, the web servers are hold by Xampp on windows platform.

b. Link it with FBctf

In flag level of FBctf, administrator can add the link of the web servers and then participants can click on the link or copy it into browsers to enter the webpage for solving problems.

And flag value will be entered in the question set page in advance so that when user submit the flag, it can be matched with the stored answer.

There are some grading mechanism to be used in fbctf as discussed in the above. For the first player solving this question, some initial mark will be given but for after comers, the mark reward will be deduced by a certain amount each time.
5. Future Work

This project only covers a small portion of web attack categories, more questions, such as XSS and CSRF, are remained as future work.

The software, Xampp, can’t co-exist with FBctf platform. But docker, which is a more complicated tool, can configure web servers at different port number of the localhost. So, if these web servers can be compiled within docker, they can be hold as within the same machine of FBctf. However, as discussed in the above, there is a tradeoff between the easy operation and less machine required.

In the future, the above two points are needed to be explored such that more completed web attack exercise server can be provided for users and it is also convenient for administrators to set the questions.
6. Conclusion

This project consists of two parts: analysis of current CTF platform, including CTFd and FBctf, and construction of four web servers for introductory exercises for web attacks.

The two platforms are compared from different perspective, administration panel, team panel and source code. CTFd is more suitable for exercise, which FBctf supports more functionalities in competition mode. They can both be linked with different web servers to set the questions.

In the second part of this project, four vulnerable servers are built from online open source code and are provided with detailed answers. Session and cookie modification, URL modification and SQL injection are covered in this server. In the future work, it is expected to enhance the content of the web server and add more categories into them. What’s more, a timer and input record function are also established to help users get better control of the competition status.

In this report, both literal description of CTF platforms and technical knowledge of web attacks are covered and future work is also proposed for later comers. If is of great honor if this project could provide a little help for the beginning of HKU CTF training.
Reference:
Cybersecurity Research and Market Intelligence, (2017), Cybersecurity Ventures
Retrieved from https://cybersecurityventures.com/jobs/


Vulcan of DDTek, A history of Capture the Flag at DEF CON

Werther J, Zhivich M, Leek T and Zeldovich N, (2011), Experiences In Cyber Security Education: The MIT Lincoln Laboratory Capture-the-Flag Exercise