

BRAINNEST



---

# CTF

---

Conducted by: Z1Hr3pAScXbjSdSVJT0YUkzCHAq2

## Fantom

*Team Members*

*Email*

Nikolay Pavlyuchkov    nikolay.pavlyuchkov@gmail.com



February 7, 2023

NOTICE: The information provided in this document is **CONFIDENTIAL** and is intended only for CTF

# Table of Contents

|          |  |                |
|----------|--|----------------|
| <b>1</b> | <b>Report Overview</b>                               | <b>II</b>      |
| 1.1      | Executive Summary . . . . .                          | II             |
| 1.2      | Challenge Overview . . . . .                         | III            |
| 1.3      | Scope of the Challenge . . . . .                     | III            |
| <b>2</b> | <b>Observations</b>                                  | <b>IV</b>      |
| <b>3</b> | <b>Trail and Fail approach</b>                       | <b>V</b>       |
| 3.1      | Creative Thought Process - Problem Finding . . . . . | V              |
| 3.2      | CVE Program . . . . .                                | V              |
| 3.3      | Vulnerability Scoring System . . . . .               | V              |
| 3.4      | NIST . . . . .                                       | V              |
| <b>4</b> | <b>Technical Findings</b>                            | <b>VII</b>     |
| 4.1      | High . . . . .                                       | VIII           |
| 4.1.1    | File Type . . . . .                                  | VIII           |
| 4.1.2    | Basic-Mod-1 . . . . .                                | XI             |
| 4.1.3    | H4K3R . . . . .                                      | XII            |
| 4.1.4    | Cyb1 . . . . .                                       | XVI            |
| 4.1.5    | Find Me V5 . . . . .                                 | XVIII          |
| 4.2      | Medium . . . . .                                     | XXI            |
| 4.2.1    | SearchSource . . . . .                               | XXI            |
| 4.2.2    | Power Cookie . . . . .                               | XXII           |
| 4.2.3    | Local Authority . . . . .                            | XXIII          |
| 4.2.4    | Forbidden Paths . . . . .                            | XXIV           |
| 4.2.5    | Fresh Java . . . . .                                 | XXV            |
| 4.2.6    | SQLiLite . . . . .                                   | XXVII          |
| 4.2.7    | Redaction Gone Wrong . . . . .                       | XXVIII         |
| 4.3      | Low . . . . .  | XXX            |
| 4.3.1    | Enhance! . . . . .                                   | XXX            |
| 4.3.2    | BasicFileExploit . . . . .                           | XXXI           |
| 4.3.3    | Inspect HTML . . . . .                               | XXXII          |
| 4.3.4    | File-Run (Fantom) . . . . .                          | XXXIII         |
| 4.3.5    | CVE-XXXX-XXXXX (Fantom) . . . . .                    | XXXIV          |
| 4.4      | Informational . . . . .                              | XXXVI          |
| 4.4.1    | Opinion . . . . .                                    | XXXVI          |
| <b>5</b> | <b>Conclusion</b>                                    | <b>XXXVII</b>  |
|          | <b>Appendices</b>                                    | <b>XXXVIII</b> |
| <b>A</b> | <b>Mind Map</b>                                      | <b>XXXVIII</b> |
| <b>B</b> | <b>References</b>                                    | <b>XXXIX</b>   |

“Freedom is obedience to self-formulated rules.” cit.  
Aristotle

# 1 Report Overview

## 1.1 Executive Summary

Fantom has started the training at <sup>1</sup>**Brainnest** and so <sup>2</sup>**Fantom** has asked to solve a couple of challenges that could be access from the platform training.

The approach to these challenges can have a serious impact on the acquisition of new skills in correlation to the preparation conducted on similar activities where Fantom participate and sometimes win.

This report was written initially on January 15th and submitted on January 17th at 1:00PM. This CTF is in the interest of Fantom to understand and explain the beauty of the exploration in digital world with focus Cyber Security against the adversaries of the digital world.

The Report Overview section contains an outlined summary of Fantom’s resolutions, including the problem solving the ethical hacking activities and the in-depth reasoning that wants to be the answer of the ethical point of view about the technology the freedom of the information and so understand the role of Fantom. With this objectives on the road the training on the picoCTF challenges are aspected to be hard and so preparation and supervision are the two most important points of the trainer from Brainnest all along the way. The Technical Findings section expands upon the report overview by including each discovered vulnerability’s evaluated risk, exploitation details, and recommended remediation steps.

---

<sup>1</sup>Brainnest provides comprehensible analyses of complex economic issues to assist in understanding the issues and opportunities that companies face. Our Economic Consulting practice is involved in a wide range of engagements related to economics, finance, and accounting. We provide critical insight and expert testimony in legal and regulatory proceedings.

<sup>2</sup>Fantom is focused to understand vulnerabilities. Is a new and young cybersecurity team that is militant in Europe and above some standard cybersecurity brands has still the riority on research without the cybersecurity’s trend searching to avoid all the not understandable and avoid the inhattendable in the way to persist on protection of values.

## 1.2 Challenge Overview

Fantom has focus on the following goals:

- Understand the uses of the penetration tester's tools for web application information gathering.
- Conduct google dorking and data scraping or in depth web application active reconnaissance.
- Try and fail approach to solve the problems when is needed a more indepth research.
- Investigate wathever is an obstacole and have to be revealed at time of the engagement.
- Create mind maps of the process and find new way of think about the problem or imagine the new chunks of the information for make connections through the cybersecurity and the other aspect related in the web application life cycle.

## 1.3 Scope of the Challenge

The experience of this whole Capture The Flag is obtained with fun and hand's on training developing skills and confidence starting from the beginning level and reach advanced level of CTF's difficulty

Every challenge is a mix of different vulnerabilities and to solve them some strategy or concept is needed to be understood and applied with careful attention to the cybersecurity's related field when is come to apply in practice to the CTF challenge.

These objectives and the point of view of the attacker are the target which focus on improve confidence using technology and discover vulnerabilities.

The malicious actor abuse the vulnerability and exploit humans. Some times is possible to avoid the incident and conduct penetration testing on the systems and find vulnerabilities before the real malicious actor.

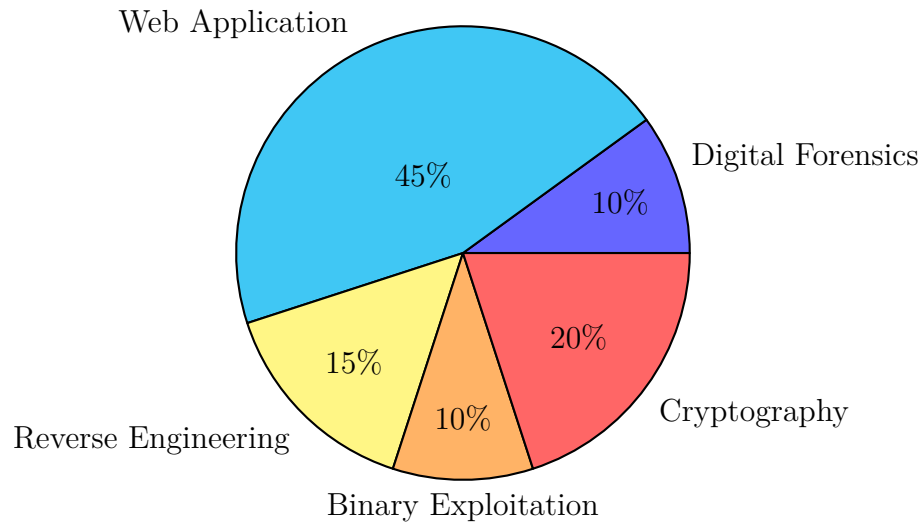
This challenging training points to conduct practical decision to solve problem quickly and so to achieve the right solutions is good to watch to cybersecurity heroes that are facing now the cyber war the hackers and the cybersecurity vulnerabilities.

What are the uncover of methodologies adopted for years by hackers? Where are the foundation of an increasing number of malicious actors that is going to scare the entire cyber world? Who is going to fight against them and why there is some kind of interest to keep all this away from the civils?

When a new vulnerability is discovered many questions are answered and the ethical hacking in many cases is the only solution to the cyber threats.

## 2 Observations

This section serves as an overview of the Cyber Security exercises. A detailed list of all technical findings can be found in Section 4. Any challenge can be an example of the vulnerabilities found in real world and the relative exploit is to consider a potential vector attack used by hackers in these examples.



## Cyber Security Layers

In these observations Fantom identifies the vulnerabilities and adopts the hacker's point of view to solve the challenges. Every CTF come with one more Cyber Security vulnerability. And so this pie is a graphical representation of the global ethical activities conducted.

### 3 Trail and Fail approach

#### 3.1 Creative Thought Process - Problem Finding

The Fantom adopts the general problem solving approach which begin with the problem finding some times the hardest part of the whole picture

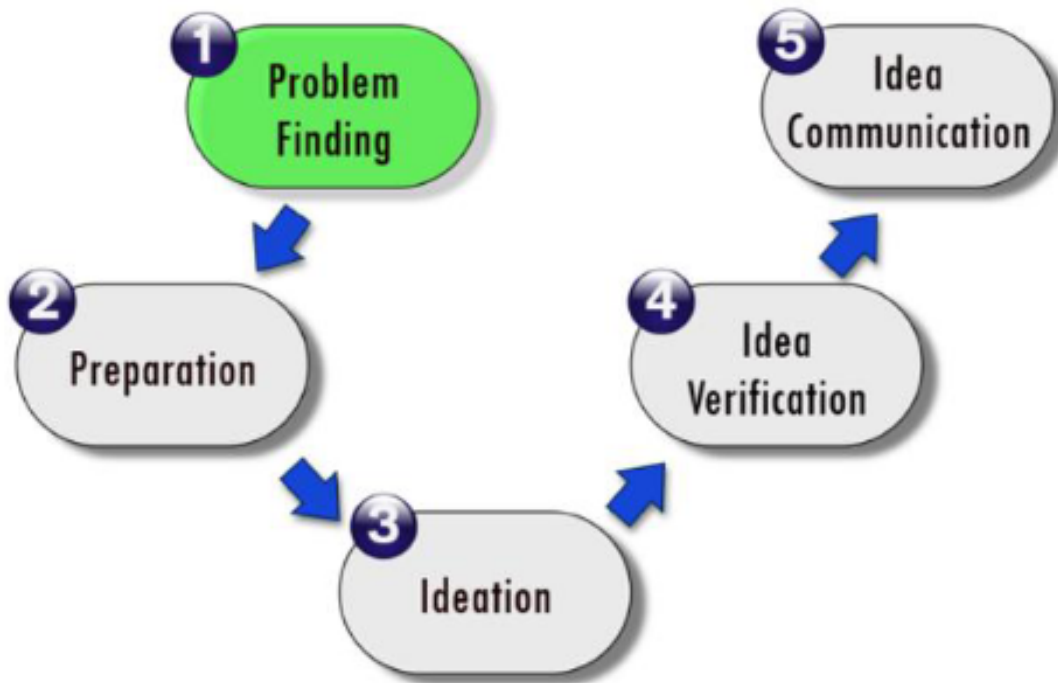


Figure 1: Problem Solving

#### 3.2 CVE Program

The Fantom adopt the CVE Program to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities. The CVE Record is what Fantom search and consult for each vulnerability they found. Information technology and cybersecurity professionals use CVE Records to communicate and to collaborate and in the end to coordinate their efforts to prioritize and address the vulnerabilities.

#### 3.3 Vulnerability Scoring System

The Fantom adopts and gives numerical score to a qualitative representation (such as low, medium, high, and critical) to identify management of vulnerability measures.

#### 3.4 NIST

The National Institute of Standards and Technology Special Publications (NIST SP) primarily comprise recommendations and best practices for information security. Federal agencies

| CVSS v2.0 Ratings |                  | CVSS v3.0 Ratings |                  |
|-------------------|------------------|-------------------|------------------|
| Severity          | Base Score Range | Severity          | Base Score Range |
|                   |                  | None              | 0.0              |
| Low               | 0.0-3.9          | Low               | 0.1-3.9          |
| Medium            | 4.0-6.9          | Medium            | 4.0-6.9          |
| High              | 7.0-10.0         | High              | 7.0-8.9          |
|                   |                  | Critical          | 9.0-10.0         |

Figure 2: NVD particular vulnerability score system.

are required to follow any NIST SP outlined in the Federal Information Processing Standard (FIPS). Table 1 provides security and privacy control methodology which are organized into 20 families. These control families are referenced throughout the document and are used to constitute common terminology. Additionally, referenced in NIST 800-53 is control families enhancements to help provide guidance to aid in securing CTF's information systems [nist80053].

Table 1: Risk Management Framework.

| ID | Process    | Description                                      |
|----|------------|--|
| 1  | Prepare    | Essential activities to prepare the organization |
| 2  | Categorize | Categorize the system and information management |
| 3  | Select     | Select the set of NIST SP 800-53 controls        |
| 4  | Implement  | Implement the controls                           |
| 5  | Assess     | Assess to determine if the controls are working  |
| 6  | Authorize  | Senior official evaluate the system              |
| 7  | Monitor    | Continuously monitor control implementation      |

## 4 Technical Findings

This table shows the total number of the CTF's achieved during the Brainnest's training. The CTFs are categorized based on the difficulty level. The difficulty levels were calculated using the picoCTF platform scoring system and the Brainnest's supervision. All the calculations are then found in the CVE's and scored in the following table.

**CTF score board**

| Difficulty          | Low | Medium | High | Critical |
|---------------------|-----|--------|------|----------|
| Vulnerability Count | 8   | 6      | 2    | 1        |

The following table breaks down the challenge of the CTF by overall simplicity, understability, and deepness. The scores were calculated Fantom exploring measurements.

**Difficulty of CTF**

| CTF                  | Research | Learn | Deepness |
|----------------------|----------|-------|----------|
| Enhance!             | 1        | 3     | 2        |
| File Extensions      | 6        | 10    | 8        |
| File-Run             | 2        | 2     | 2        |
| CVE                  | 3        | 3     | 3        |
| Power Cookie         | 3        | 4     | 4        |
| Forbidden Paths      | 4        | 4     | 4        |
| Local Authority      | 4        | 4     | 4        |
| BasicFileExploit     | 2        | 2     | 2        |
| InspectHTML          | 3        | 3     | 2        |
| SearchSource         | 4        | 3     | 5        |
| Fresh Java           | 5        | 5     | 5        |
| SQLiLite             | 5        | 5     | 5        |
| Redaction gone wrong | 6        | 5     | 5        |
| Basic-Mod1           | 6        | 7     | 7        |
| H4K3R                | 7        | 7     | 8        |
| Cyb1                 | 8        | 9     | 9        |
| FindMeV5             | 9        | 9     | 9        |



## 4.1 High

### 4.1.1 File Type

CTF level: **High**

Description:

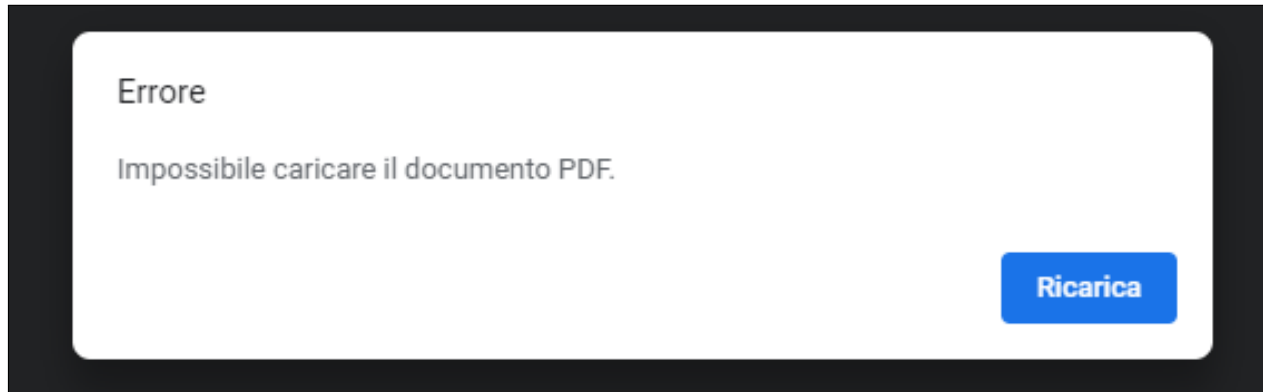


Figure 3: The file has a wrong extension

It's right there is some code and the file has been archived with a Linux program that creates auto-extracting archive. And so once I see the threat actor's vision is to obfuscate

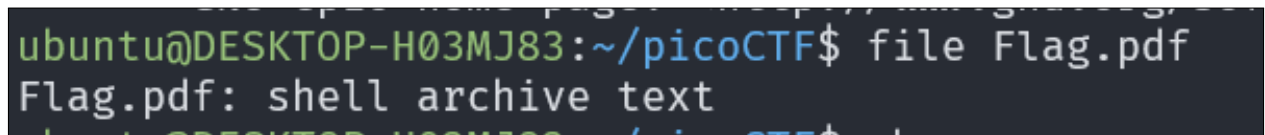


Figure 4: Shell Archive

the information using the *ID: T1027 Obfuscated Files or Information technique* I need to understand why and which tool of Linux are involved by the way the extraction is complete but the flag is not readable.

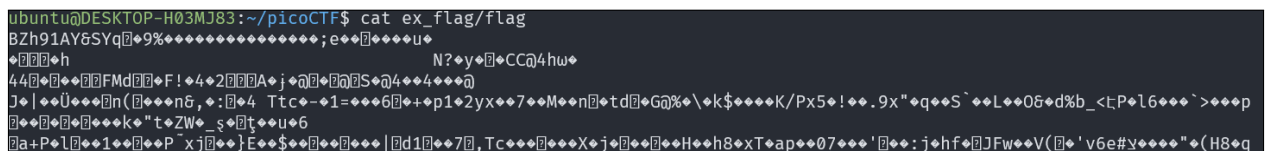


Figure 5: Ghibberish Text Not Human Readable

Searching on internet I found a website is very useful and it has a set of tool that can be useful and identify the file type similar to the `file` in CLI from Linux but it gives some information available on the web. After analyzing the file I see the what is going on and so the flag



**Online TrID File Identifier**

Identification results:

File size: 5KB

**Warning:**  
The file seems to be plain text. TrID is best suited to analyze binary files!

| Match  | Ext | File type               | MIME type | Related URL  | Def's author  |
|--------|-----|-------------------------|-----------|--|---|
| 82.50% | SHR | SHAR                    | SHA       | shar SHell self-extracting aRchive   | text/plain <a href="https://en.wikipedia.org/wiki/Shar">https://en.wikipedia.org/wiki/Shar</a> Marco Pontello |
| 17.50% | SH  | Linux/UNIX shell script |           | <a href="http://www.lysator.liu.se/~forsberg/linux/shell-scripts.html">http://www.lysator.liu.se/~forsberg/linux/shell-scripts.html</a> Marco Pontello |   |

Figure 6: Online TRID

that I've obtained with the first Linux command is an archive itself wich is obfuscated by extension removal.

```
ubuntu@DESKTOP-H03MJ83:~/picoCTF$ ar x flag
ubuntu@DESKTOP-H03MJ83:~/picoCTF$ ls
Flag.pdf  flag
ubuntu@DESKTOP-H03MJ83:~/picoCTF$ file flag
flag: cpio archive
```

Figure 7: ar Linux archive

The TRiD online gives many information about the file type the algorithm method of encryption used  
it explains the differences of extensions providing some resources it has many different warnings about the content of the file  
information about where to find that type of file when you need to use it if in the CLI of Linux or in GUI with Windows  
it gives information about the author and the probability of real match and so if the utility isn't on our system we can update and install the package and proceed to decoding the file.

```
ubuntu@DESKTOP-H03MJ83:~/picoCTF$ unxz flag.xz
ubuntu@DESKTOP-H03MJ83:~/picoCTF$ ls
ex_flag  flag  flag.lz4  flag.lzo
ubuntu@DESKTOP-H03MJ83:~/picoCTF$ file flag
flag: ASCII text
ubuntu@DESKTOP-H03MJ83:~/picoCTF$ cat flag
7069636f4354467b66316c656e406d335f6d406e3170756c407431306e5f
6630725f3062326375723137795f39353063346665657d0a
```

Figure 8: The Flag Encrypted.

Fantom have to repeat the analysis of the file until is able to recognize a text file. In the original flag file text there are two lines encrypted in ASCII code.

```
file types  X
1  picoCTF{f1len@m3_m@n1pul@t10n_f0r_0b2cur17y_950c4fee}
2  💡
```

Figure 9: The Flag Decrypted.

### 4.1.2 Basic-Mod-1

CTF level: **High**

Description:

```

1  202 137 390 235 114 369 198 110 350 396 390 383 225 258 38 291
2  | 17  26  20  13   3  36  13  36  17  26  20  13   3  36  1  32
3
4  75 324 401 142 288 397
5  1  28  31  31  29  27
6
7
8
9  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
10 A B C D E F G H I J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X
11
12 24 25 26 27 28 29 30 31
13 Y  Z  0  1  2  3  4  5
14
15 32 33 34 35 36
16 | 6  7  8  9  _
17
18 R 0 U N D _ N _ R 0 U N D _ B 6 1 2 5 5 3 1
19 💡
20 picoCTF{R0UND_N_R0UND_B6B25531}

```

Figure 10: Basic Mod Flag

So complete this CTF has been not so linear and for obtain the flag there were many steps to do to pass at the following phase of the competition. And so first was clear to understand what was that strange sequence of numbers and why there were isolated with spaces. Of course the meaning was that there were some connection to the alphabet and so once you try to decode the string you initially don't see any result. Looking well at the explanation of the CTF the number meaning is going to be an encoded modulo operator that the resolution will give the position of the letter in English alphabet. So resolving all the steps and paying attention at the formula that is going to show you the flag.

### 4.1.3 H4K3R

CTF level: **Critical**

Description:

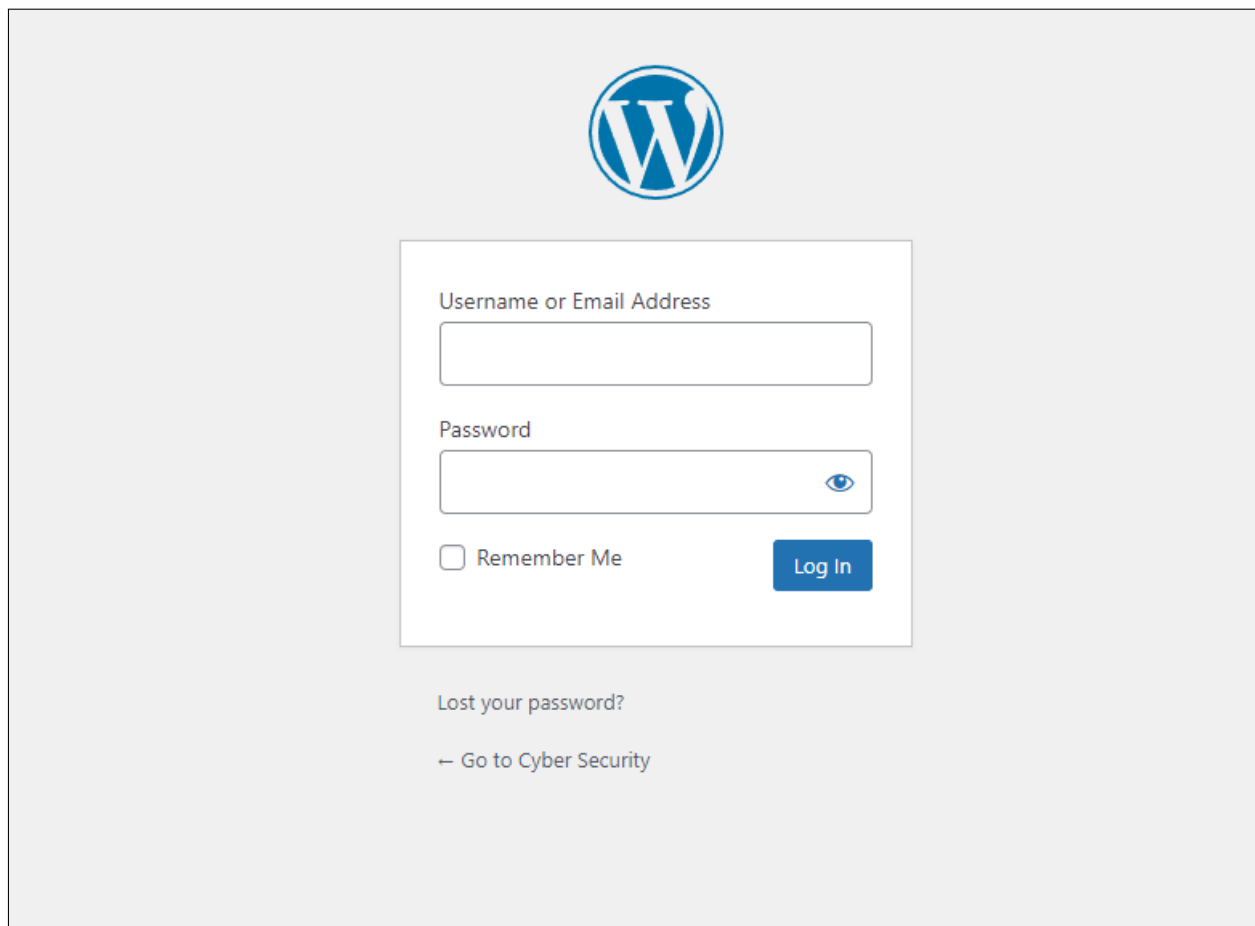


Figure 11: Word Press

Web site is made with Wordpress not accessible login page is on the path and we can look at the standard web browser configuration and understand why some web sittings are so vulnerable.

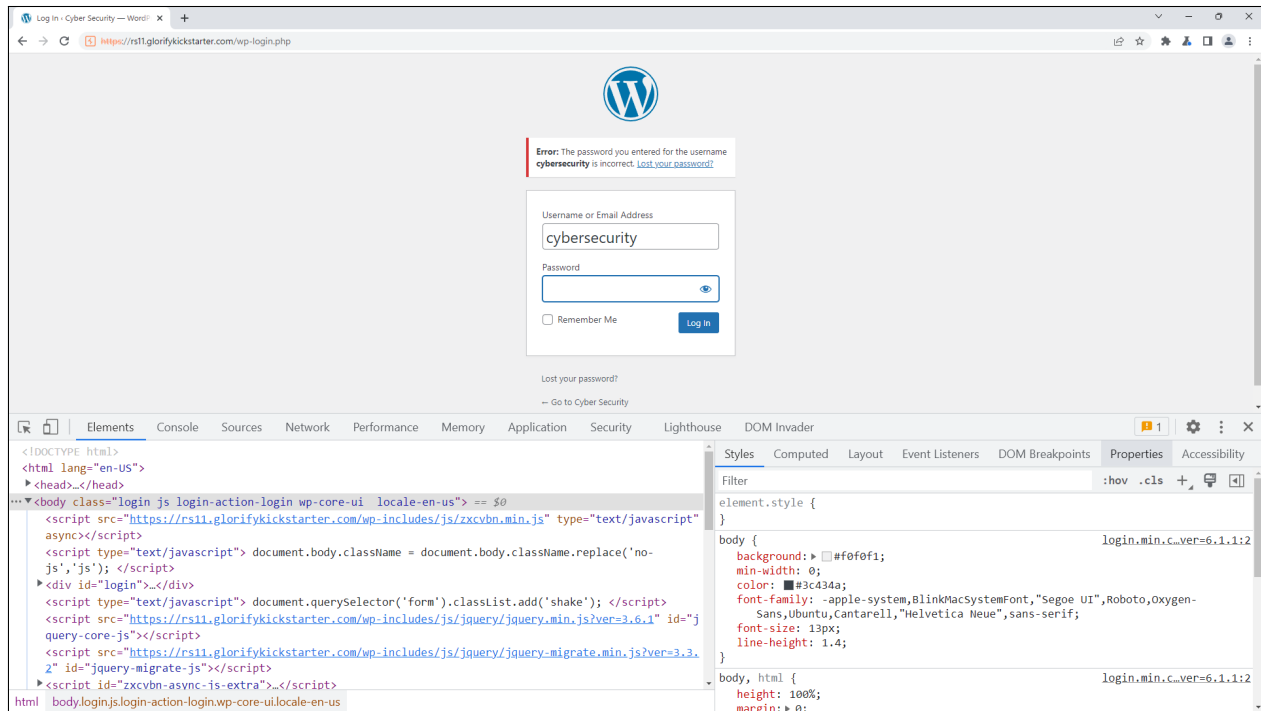


Figure 12: Login form exposes the users.

If someone access the login psge and tries is possible to gain advantage of the defences of the Wordpress authentication method and so like this image the user cyberasecurity is now exposed and now only with the brute forcing method of the password we try to exploit the login authorization form.

The image shows a web form with three input fields and a submit button. The first field is labeled 'Your Name' with a red asterisk and contains the text 'Neo Anderson'. The second field is labeled 'Email' with a red asterisk and contains the placeholder text 'Email'. The third field is labeled 'Group' with a red asterisk and contains the placeholder text 'Select'. Below these fields is a green 'Submit' button.

Figure 13: Neo Anderson.

But before proceeding we inspect inside the web application and try to find something useful information. And so looking at the filesystem and through the obscurities of the Wordpress authentic web application we try to access to other hidden directories. And so executing some jumps from literally different links and looking at the robots.txt configuration and encounter other configurations we find what we were searching for and hidden form were the user can complete the challenge and insert the username the email and the group course.

```
(root@DESKTOP-H03MJ83)-[/home/kali]
# nmap -sV --script http-wordpress-brute.nse --script-args http-wordpress-brute.uservar=cybersecurity,http-wordpress-brute.uri="/wp-login.php",http-wordpress-brute.hostname="rs11.glorifykickstarter.com",http-wordpress-brute.passdb=/home/kali/nmap_password_list.txt -p443 192.254.234.32
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 23:58 CET
NSE: [http-wordpress-brute] usernames: Time limit 10m00s exceeded.
NSE: [http-wordpress-brute] usernames: Time limit 10m00s exceeded.
NSE: [http-wordpress-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 192-254-234-32.unifiedlayer.com (192.254.234.32)
Host is up (0.25s latency).

PORT      STATE SERVICE VERSION
443/tcp   open  ssl/http Apache httpd
| http-wordpress-brute:
|   Accounts: No valid accounts found
|_ Statistics: Performed 3046 guesses in 597 seconds, average tps: 5.1
|_http-server-header: Apache

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 616.80 seconds
```

Figure 14: Fantom cracks the user.

The other part of the challenge was to access the user account and still we have his username we are going to perform some attempts to bruteforcing the account with Kali Linux. There are many instruments that can be used to guess the password and many are specialized on the wordpress 'wp-login.php' form. We try a shot with Nmap that has many scripts that are accessible in Kali and the after a few attempts the web site begins to respond in a some kind strange way and begins to give errors on the output and so we think that is not a good response from the site and try to analyze the web site with many other tools like Burp, Zap, Wpsca and Metasploit many other attempts could be more sophisticated but this response

of the web site is too strange and is probably configured with many tools that don't give the chance of brute forcing with automatic tools and is more likely to find more useful information in the inspector of the web application.



#### 4.1.4 Cyb1

CTF level: **Critical**

##### Description:

The trainer of Brainnest has designed an application vulnerable in the login page so conduct some inspections and obtain the informations about the user's credentials.

The vulnerability are not given at the beginning and the indication is that there is some weak design in the login form.

```
105
106 <!-- the password for the user "think" is "123" -->
107
108 <!------>
109 <!------>
110 <!------>
111 <!------>
112 <!------>
113 <!------>
```

Figure 15: The user's username and password.

The credentials are correct and Fantom has access to the account. After a while what we can see is limited and so we suppose the weakness is in the type of the credentials. Anything the user's account has is inside the definition of the credentials.

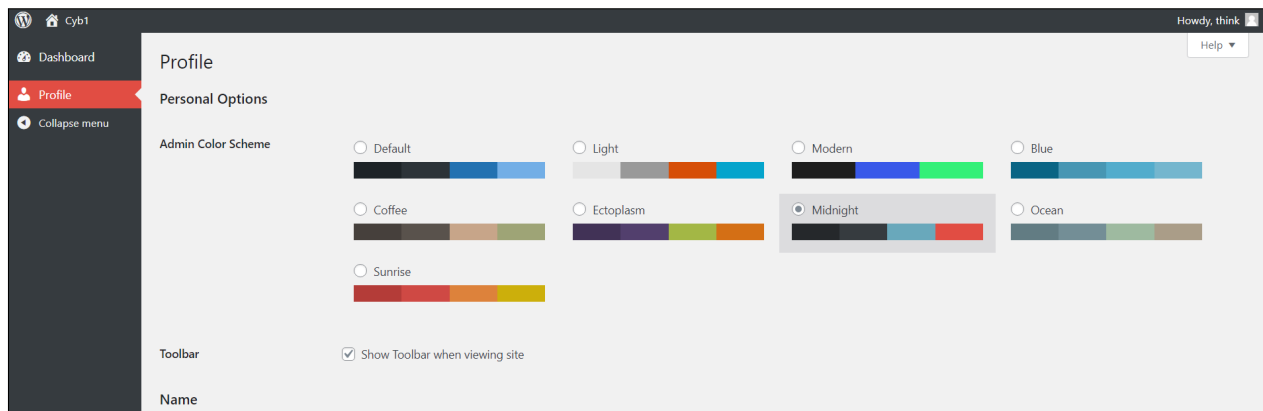


Figure 16: Access to the user's account.

The first try is guess the default administrator credentials. Wordpress responds to the guesses and so confirms if user is correct. And so this means to know that the username is admin the default username. And so found the username is going to be a brute force against the password and so the brute force technique is still time consuming and achive the guess will

take some hours. Looking better at the list of password only two are in some way different so try that one first and obtain the admin's credentials.

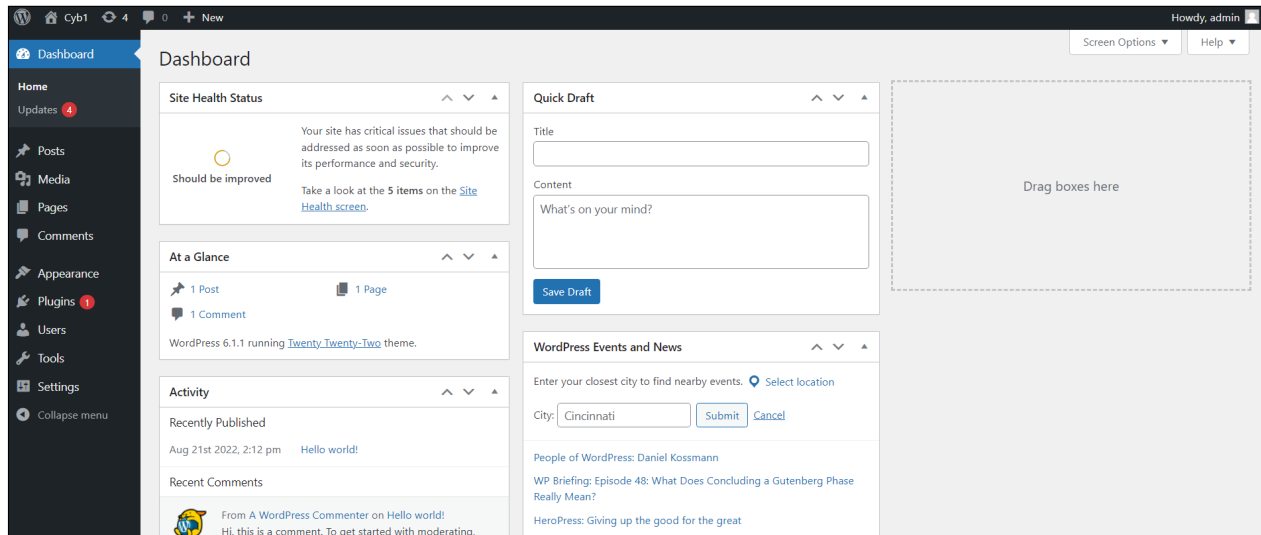


Figure 17: Access to the admin's account.

### Potential Business Impact:

The password can be stored everywhere and keep the ability to put the in the comments is not a good habit and so the login form is easily hackable the other vulnerability is the fault of the design of the wordpress blog that still reveals to users if they have misspelled the password giving them hints about what is an attacker looking for a proper username valid to bruteforce the password and even the admin username for the administrator account is not a good habit that many hackers can find in the websites. So is very easy to guess the popular passwords and the username admin is the default for Wordpress.

### Exploitation Details:

An external hacker may obtain the credentials stored in the comments very easy and once playing with the login form his finding about the username that remain a default have for the hacker a lucky factor for implementing a brute force attack after the username finding.

### Recommended Remediation:

Change the default credentials and change the way the credential are stored on the web application for example not put in the comment even the password is still strong is easy to look at it if the password is in the comment.

### References:

[https://owasp.org/www-project-web-security-testing-guide/latest/4-Web\\_Application\\_Security\\_Testing/04-Authentication\\_Testing/02-Testing\\_for\\_Default\\_Credentials](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/02-Testing_for_Default_Credentials)

### 4.1.5 Find Me V5

CTF level: **Critical**

#### Description:

The file is manipulated by an actor that has changed the extension of the file and substitute it with the PDF signature. So the file isn't readable and the first thing is to go a look at the hex of the file when we see this we understand that the file is not a PDF, and so we need to check the real type of the file. There are many tools, and we easily obtain the file type that is hidden from the view. Many tools confirm us that is an archive file and has an original zip signature, and so we search for the signature of the zip format, and we find the information that can be now replaced in the HxD editor to reveal the archive and the archive functions.

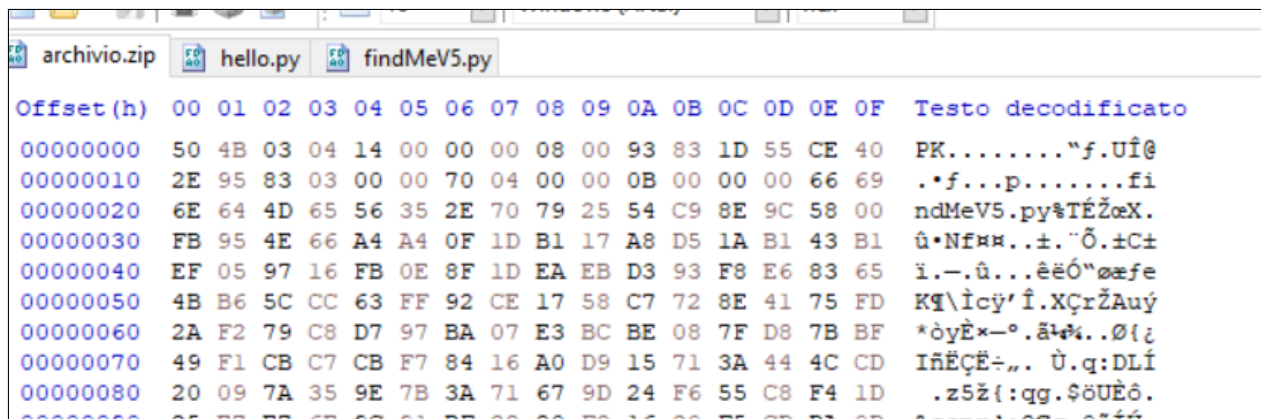


Figure 18: HxD file signature.

After that using an extractor we obtain a directory and inside the directory there is a python program. If we run the program we will be asked for a password and in some way if we put the password we obtain an error, or we obtain a good guess and so the password is like the flag that we are looking for and so the only thing that we have is the python program itself and probably the flag is hidden there. But we don't know nothing about because of that there are some hashes and python functions. After a while over Google we research for a cryptography technique that is used inside the file. And so this kind of encryption requires some key and responds with an encrypted message, and so we find a cryptographer and type the information that are asking for decrypt the bod of the file.

Surprisingly we understand that all the chiper text is a series of instructions that the program evaluates at run time but the variables are taken by the instructions from that particular variable that stores information in encrypted form that we cannot decrypt and so in Visual studio code we write a function that prints any of the information that are in use of the program to check the correct password and print the value of the password then we give it to the original python program and the program tells us that the answer is correct.

```
Decoded:
dYrxjtyjxfgn = input(muHa[501]+muHa[503]+muHa[55]+muHa[57]+muHa[511]+' ')

if dYrxjtyjxfgn == muHa[100]+muHa[200]+muHa[300]+muHa[400]+muHa[500]+muHa[250]+muHa[125]+muHa[75]+muHa[50]+muHa[25]:
    print(muHa[401]+muHa[403]+muHa[45]+muHa[47])
else:
    print(muHa[201]+muHa[203]+muHa[65]+muHa[67]+muHa[97])

Date created: Mon Aug 29 12:26:48 2022
Current time: Sat Feb 4 20:11:15 2023
```

Figure 19: Decode Fernet.

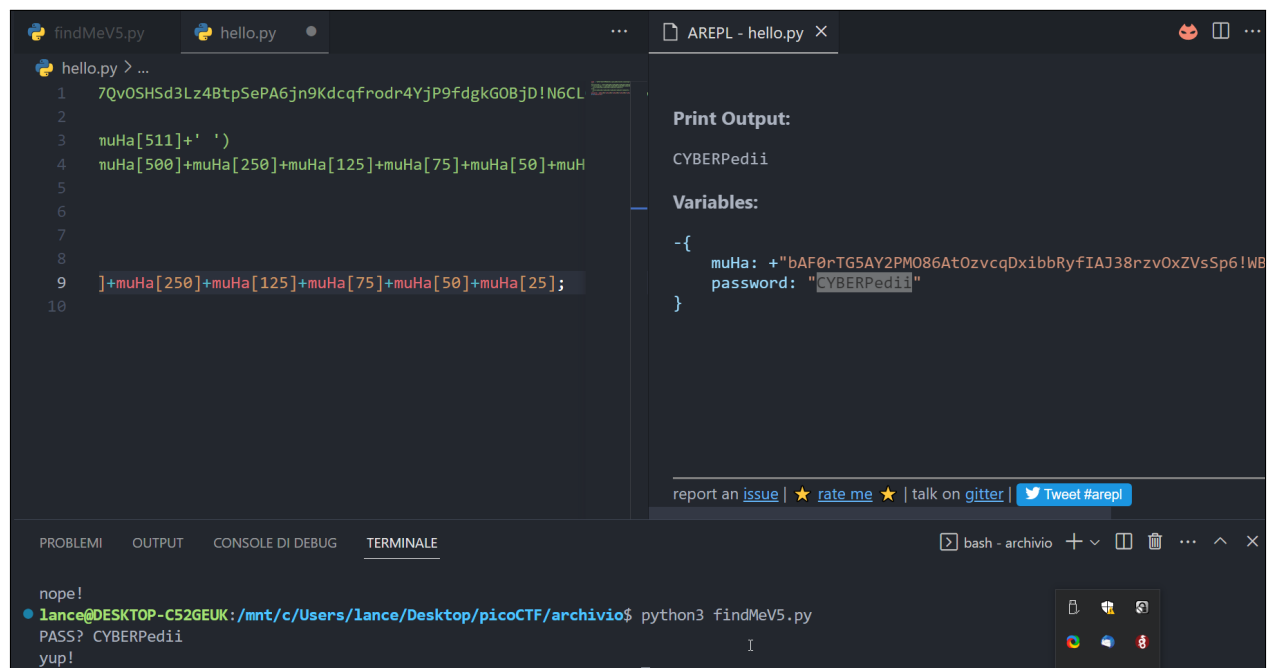


Figure 20: Reveal password in Python.

**Potential Business Impact:**

This CTF is very hard because can be not easily identifiable the technology involved and is needed a cross thinking and knowledge to solve it even so there is the necessity to persist on the evidence and give the value retrieving new information that gives the old information new meaning. This kind of CTF is time-consuming if you not go deeper on findings. So is possible to reach "give up" status easily but off course all is achievable and understandable with the consciousness and going over the evidence acquisition at this stage. And what is not seen is dectecting from the new informations.

**Exploitation Details:**

A user can hide programs in an encrypted code and change the signature so that the is unrecognizable and not usable to Fantom has complete the challenge and thinks about the abuse of actions that can made this kind of technique when some code is sent across the web in way that no signature and behavior system can alert so this kind of attack can evade and bypass common defensive countermeasures and is interesting to see how an encrypted program maintain the same functionalities at the level of a completely untraceable state and is probably not only be able to hide information to the examiner but it can hide inside himself more components in encrypted form too and this goes to damage the system that not adopt proper monitorings on the download.

**Recommended Remediation:** This kind of program could lead to a zero-day attack and so is very difficult to prevent once the download is completed and in this case if some have found the file and register the signature on the antivirus list an update antivirus is able to detect it from the signature but if the attack is heavy and so the first of the attacked systems are brought down then this kind of identification is not going to arrive sufficiently in time and so the resolution of this kind of attack has to be identified with a SIEM control system that gives the chance to trigger an alert to suspicious file if the attacker has done a good job he has the advantage of surprisingly blazing attack and so if the attack is dormant this technique be mitigated by a security specialist's operations.

**References:**

<https://asecuritysite.com/forensics/magic>

<https://asecuritysite.com/encryption/ferdecode>

## 4.2 Medium

### 4.2.1 SearchSource

CTF level: **Medium**

**Description:** The flag is inside the source code of the web page. So the best way to find it is with the inspector and not anything else.

Using the inspector inside the web browser has different possibilities and of course the deepest of the knowledge of this tool is the fact that the totum of the web application penetration testing and so diving in the various uses of this tool's functionalities gives you the mind map that could be not so evident at first place. And this is not basic when you need to inspect some bugs in the Web Browser with this handy and quick tool very easy to understand.

```
font-size: 18px;
color: #000;
line-height: 18px;
}
/** banner_main picoCTF{1nsp3ti0n_0f_w3bpag3s_8de925a7} */
.carousel-indicators li {
  width: 20px;
  height: 20px;
  border-radius: 11px;
  background-color: #070000;
}
.carousel-indicators li.active {
  background-color: #35a30a;
}
.carousel-indicators {
```

Figure 21: The missing information

### 4.2.2 Power Cookie

CTF level: **Medium**

**Description:**

When the developer uses web technologies if he don't pay attention is very easy to someone to look at the code and understand the logic and is so easy to see vulnerabilities that can be exploited using the same logic and technology to fake the weak defences of the web application.

(see Figure 22).

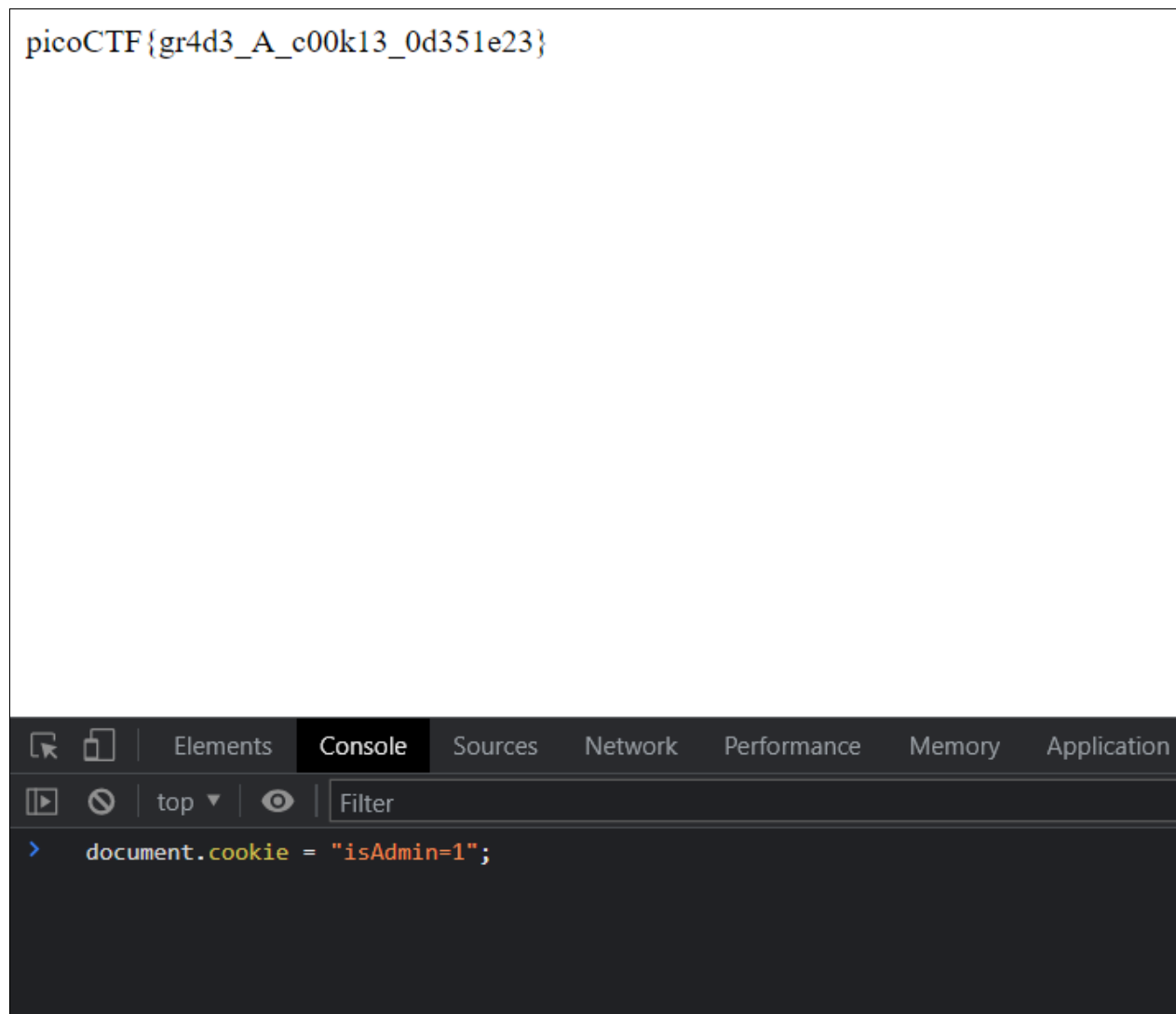


Figure 22: Grade a cookie.

Send fake information or change values is easy in the web browser and reveal some useful information make simple the exploit for an attacker.

**Potential Business Impact:** This kind of vulnerability must be avoided because of the fact that even if there are some kind of defences the whole system is fooled by this weakness. Avoid or mitigate is necessary in the program logic? No sometimes using the same technology with some extra layer of privacy can do the job.

### Exploitation Details:

Exploiting the cookie is a way that many web attacks make use of. The browser sends and requests information and it expects to an answer from the client and then it accepts the session and establishes the authentication of the user so if the user is not legitimate the cookie keeps this information for the browser the host which can be accessible from an attacker and momentarily fake the session sending the incorrect information and gain unauthorized access to the host.

### Recommended Remediation:

Proper validation of the cookies disables the possibility to change parameters of cookies make private the logic and not accessible to the users verify what can be seen about the application in the external world gain better defences using more cybersecurity focused decision at the design stage of application

### References:

<https://hackcommander.github.io/pentesting-article-1/#>

## 4.2.3 Local Authority

CTF level: **Medium**

### Description:

Users are curious and many are going to use your application in a not proper way and so they will try to look inside the logic of the application and will find usernames and passwords when they are hard coded and not encrypted

(see Figure 24).

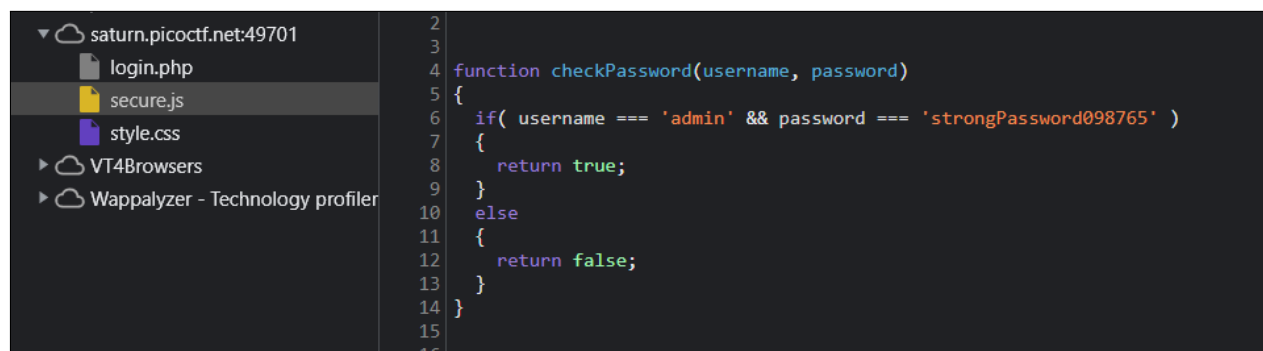
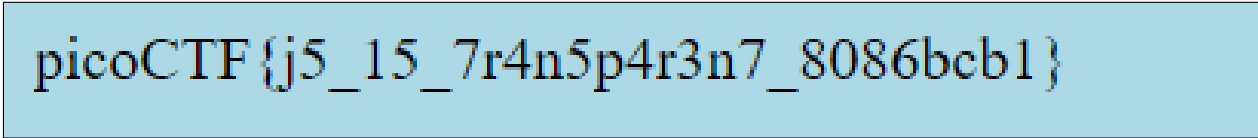


Figure 23: Hard coded username and password in the javascript file.



Is possible to look at the code and obtain informations about any thing and if this information is too much simple to reach and attacker will gain and use it for illicit purposes.



picoCTF{j5\_15\_7r4n5p4r3n7\_8086bcb1}

Figure 24: All that is on the web is nearly accessible to everyone.

### Potential Business Impact:

Not disclosing information on the web application is a good method to avoid unauthorized access. It reveal personal informations and expose the victim at the attacker bad intentions.

### Exploitation Details:

Navigate the web and use tools like inspector or more specialized hacking tools can reveal the code that is behind the application and so information gathering about the user's password have an easy way to the attacker to find them

### Recommended Remediation:

Any time you need to match and perform authentication of the user is better to use encrypted passwords and store them on the remote server and not in the body of the application

### References:

<https://www.techtarget.com/searchsecurity/tip/How-hard-coded-credentials-threaten-industrial-control-systems#:~:text=Hard%2Dcoding%20credentials%20is%20the,or%20generating%20them%20at%20runtime.>

## 4.2.4 Forbidden Paths

CTF level: **Medium**

### Description:

Find the way to navigate through the directories within the path can be very simple if we know the technology and also the location of the files we can simply guess which path is accessible trying to look at the response of the web application. Sometimes when there is no response is what exactly we expect to see from the behavior of an application. When we understand what the application is supposed to do we can ask the application something that the programmer did not mean at the time of implementation and so the program that he wrote is going to reveal part of the structure for example typing "../" in an URL we ask the program to output what is inside the directory up of one in the tree list directories and this is a technique for listing all of them even if the user interaction is not permitted at this level of application or is not supposed to access to these directories. And so the path of the servers that are storing the applications sometimes is more complex and sometimes are similar and

even if they are complex navigate is a technique that exposes the information in front of this kind of attack knowing this gives the ability to look at the informations that are disclosed in the server through the web browser and the path traversing

picoCTF{7h3\_p47h\_70\_5ucc355\_e5a6fcbbc}

### Potential Business Impact:

The path traversal can lead to an attacker the full structure of the web application and so when this happen is possible if the system permits this kind of interaction from the user and once arrived is about to be that there are not file or informations forgotten in the directories or in the tree path because this is an easy target for the attacker to obtain the information.

### Exploitation Details:

The attacker has access to the folder's tree through a simple method wich is exhaustive to list the files on the web application using a relative path.

**Recommended Remediation:** Disable this kind of attack with the code sanitization even more not allowing relative or absolute paths to be executed by any part of the program.

#### 4.2.5 Fresh Java

CTF level: **Medium**

### Description:

There is a java class file that has been obtain from compiling the java program and so online is founded the java decompiler and the flag is inside the program

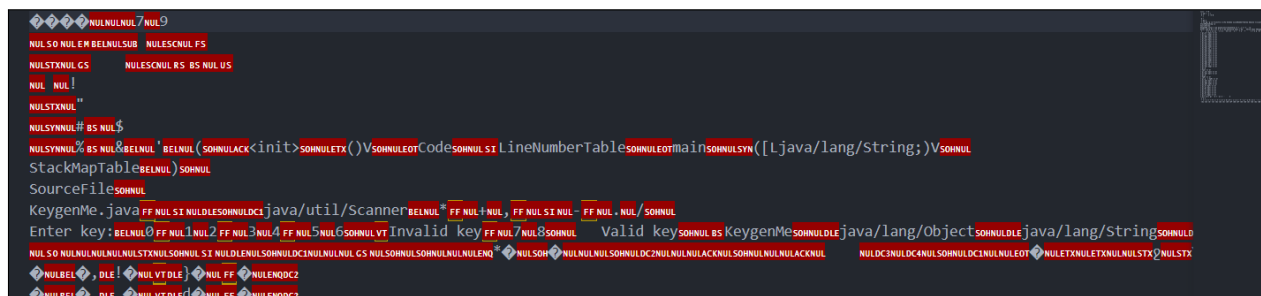


Figure 25: Java compiled class file.

**Potential Business Impact:** Any file can be a threat if you not know enought about it. Java is programming language that has to be compiled and thne after that is run with the Java Virtual Machine the file instructions are translated to close machine language code and this is a reversible process and is possible to look at the original structure of the file.

From the instructions in the code we obtain the flag.

picoCTF{70011ng\_r3qu1r3d\_2bfē1a0d}

```
}  
if (nextLine.charAt(6) != 'F') {  
    System.out.println("Invalid key");  
    return;  
}  
if (nextLine.charAt(5) != 'T') {  
    System.out.println("Invalid key");  
    return;  
}  
if (nextLine.charAt(4) != 'C') {  
    System.out.println("Invalid key");  
    return;  
}  
if (nextLine.charAt(3) != 'o') {  
    System.out.println("Invalid key");  
    return;  
}  
if (nextLine.charAt(2) != 'c') {  
    System.out.println("Invalid key");  
    return;  
}  
if (nextLine.charAt(1) != 'i') {  
    System.out.println("Invalid key");  
    return;  
}  
if (nextLine.charAt(0) != 'p') {  
    System.out.println("Invalid key");  
    return;  
}
```

Figure 26: Java class in readable format.

### Exploitation Details:

Retrieve artifacts from files is common in the cybersecurity. Many times the files aren't in a common form and so is needed to invert the process to obtain a readable file.

### Recommended Remediation:

Research the file types.

### References:

<https://www.javatpoint.com/how-to-create-a-class-file-in-java>

#### 4.2.6 SQLiLite

CTF level: Medium

##### Description:

For this CTF Fantom has to use SQL injection to login in the form of a web site login page. So try with default user name and with classic SQL injection technique. This works and from the login page there is a message that encourage us to look to the HTML to find the flag. (see Figure ??).

The inspector reveals the flag hidden in the HTML but in plain text.

Your flag is:

```
picoCTF{L00k5_l1k3_y0u_solv3d_it_ec8a64c7}
```

##### Potential Business Impact:

Is easy for potential attacker to try to inject SQL code. And is not so difficult to find long list of the SQL examples of attacks cheatsheets (see Figure ??).

**Exploitation Details:** Is the most popular of the injection attempt to bypass login and obtain credentials. When a vulnerability is found in the login form to the SQL injection this will compromise the security of the system and any information are compromised. So this kind of attack is going to exploit the input validation system when user input is trusted and any combination of input are accepted this permit to inject SQL code from any injectable element of the website.

##### Recommended Remediation:

The only technique against the SQL injection is to sanitize the user inputs and to monitor the query to the website database that can contain improper input and that contain SQL code. Because of that the attacker needs to use something strictly related to the input user is not easy to bypass the surveillance of SIEM and SOAR and so he will try to attack the web site in late hours and so to avoid this threat is more correct to contrast the hacker on the input side of the defence and sanitize the input.

##### References:

[https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL_Injection)

<https://www.sqlshack.com/sanitizing-inputs-avoiding-security-usability-disasters/>

### 4.2.7 Redaction Gone Wrong

CTF level: **Medium**

#### Description:

This file in PDF format contains many black stripes that cover some sensible information in the text and so you can't see what is hidden in that part of text. (see Figure 27).

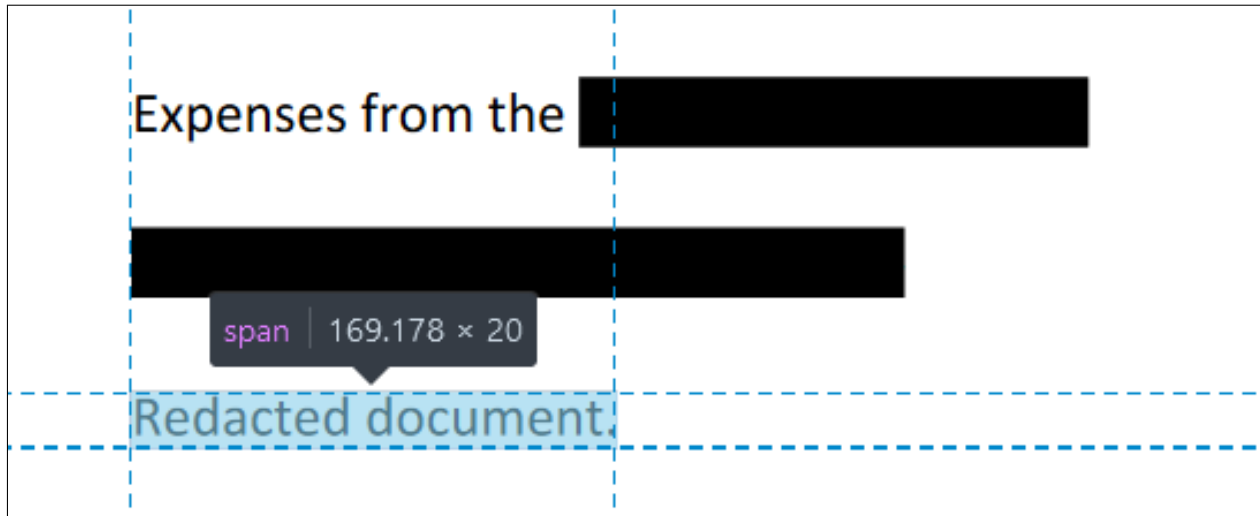


Figure 27: FrontEnd of the file.

Any thing is hidden in the PDF file but a closer look at the code in the inspector makes evident the mistake. Any thing can be seen from the HTML that hides not so well the sensitive information and leaves the secrets exposed on the backend of the file.

```
<span style="left: 11.76%; top: 37.18%; font-size: calc(var(--scale-facto...00px); font-family: sans-
transform: scaleX(0.834248);" role="presentation" dir="ltr">picoCTF{C</span>
<span style="left: 19.67%; top: 37.18%; font-size: calc(var(--scale-facto...00px); font-family: sans-
transform: scaleX(0.927878);" role="presentation" dir="ltr">4n</span>
<span style="left: 21.7%; top: 37.18%; font-size: calc(var(--scale-factor...00px); font-family: sans-
transform: scaleX(0.872403);" role="presentation" dir="ltr">_Y0u_S33_m</span>
<span style="left: 32.06%; top: 37.18%; font-size: calc(var(--scale-facto...00px); font-family: sans-
transform: scaleX(0.949442);" role="presentation" dir="ltr">3_fully}</span>
<br role="presentation"></br>
```

Figure 28: BackEnd of the file.

picoCTF{C4n\_Y0u\_S33\_m3\_fully}

#### Potential Business Impact:

Malicious actor is interested to look at some useful information and sensitive information are that kind of information that must be protected from unauthorized eyes. If the information is not secret the malicious actor is going to destroy reputation and image of the business with more destructive power that this new informaton can provide. And so this new threat opens ports to a various attempts of attack and the mostt common are phishing and identity thief

or unauthorized access if the secret information is a password or many more correlated to the loss of the confidence of the information and so it means to a leak of information to some external threat that this kind of obfuscation is trying to avoid putting the enemy under the black stripes.

**Exploitation Details:**

If for the actor looking at the secret information give him a meaningful reason about the information disclosure he will proceed with an more advanced attempt to produce damage at the victim and so if the information have any relationship with other activities of the business the attacker will proceed to implement focused attacks.

**Recommended Remediation:**

Be careful to what kind of information you are passing in the web and if is really necessary an implementation of secure communication of information these must be implemented in the original document and avoid to use some technology that is easily reversible or in which case is better to look if every piece of information has been covered by the technique. If is not so use hashing or encryption or code that is not understandable outside the perimeter.

**References:**

<https://blog.avepdf.com/how-to-redact-pdf-online-remove-sensitive-personal-data/>

## 4.3 Low

### 4.3.1 Enhance!

CTF level: **Low**

**Description:** The flag is inside an image file. Exist many type of image file this one is called SVG they are a particular type of image and can be opened or modified directly in the web browser.

Fantom obviously find the flag inside the VSCode.

So is possible to find all the pieces of the flag and obtain the full flag like this.

Or in the web browser once changed the extension of the file or if we understand before that there are some line of text that are been enhanced in which manner the text is not visible on the image so Fantom resize the properties of the `<span>` tag and can see the secret message.

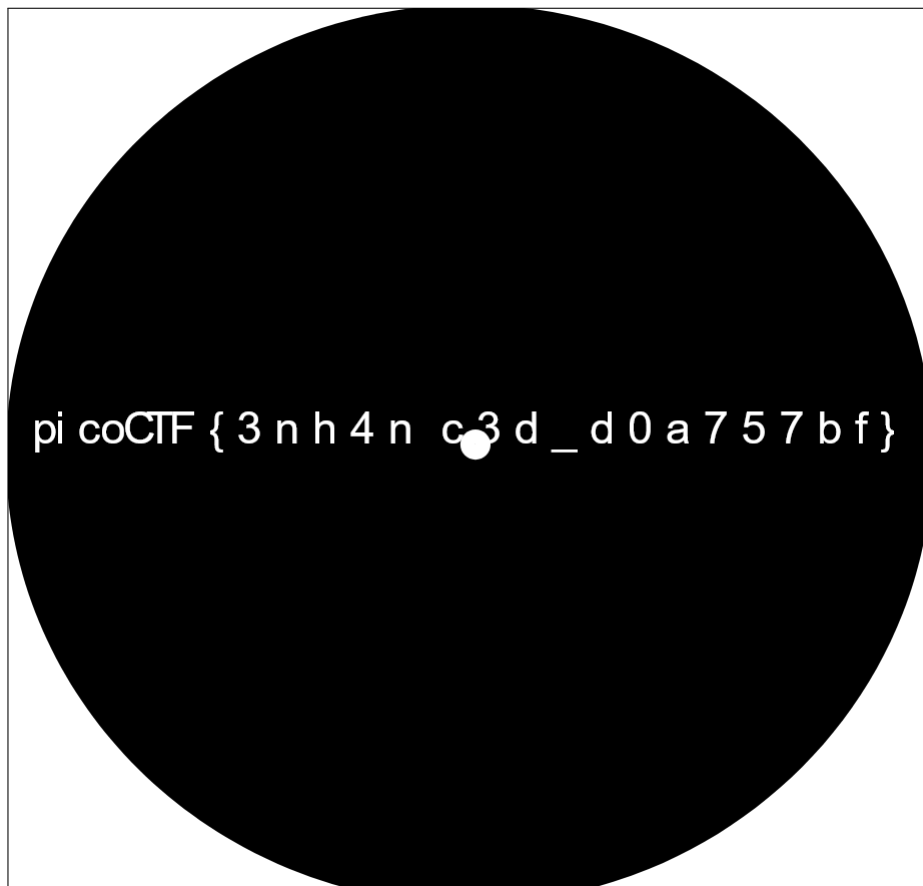


Figure 29: The Flag.

### 4.3.2 BasicFileExploit

CTF level: **Low**

**Description:** The Flag is in the file that is in the terminal that can be executed with some options given to the user that can enter a string of text and see it appear on the screen.

```
ubuntu@DESKTOP-H03MJ83:~/picoCTF/basic_file_exploit$ nc saturn.picoctf.net 55825
Hi, welcome to my echo chamber!
Type '1' to enter a phrase into our database
Type '2' to echo a phrase in our database
Type '3' to exit the program
```

Figure 30: Welcome to my echo chamber

After a little we try to inject some instructions in a way that the program not aspects and so the result of this action is the injection of code and the execution of code from the program can reveal the Flag hidden from the user.

```
No data given.
Please put in a valid number
2 '/ls'/
2 '/ls'/
Please enter the entry number of your data:
picoCTF{M4K3_5UR3_70_CH3CK_Y0UR_1NPU75_68466E2F}
```

Figure 31: Fantom reveal the Flag.



### 4.3.3 Inspect HTML

**CTF level:** Low

**Description:** The flag is inside the HTML. After reading the CTF description that the picoCTF is telling us is to look inside the HTML. And so with inspector we will find the flag.

In the HTML is possible to put some lines that will not been interpeted by the web browser and so some developer can use this functionality to post comment's related to the web page and can be seen only with developer tools.

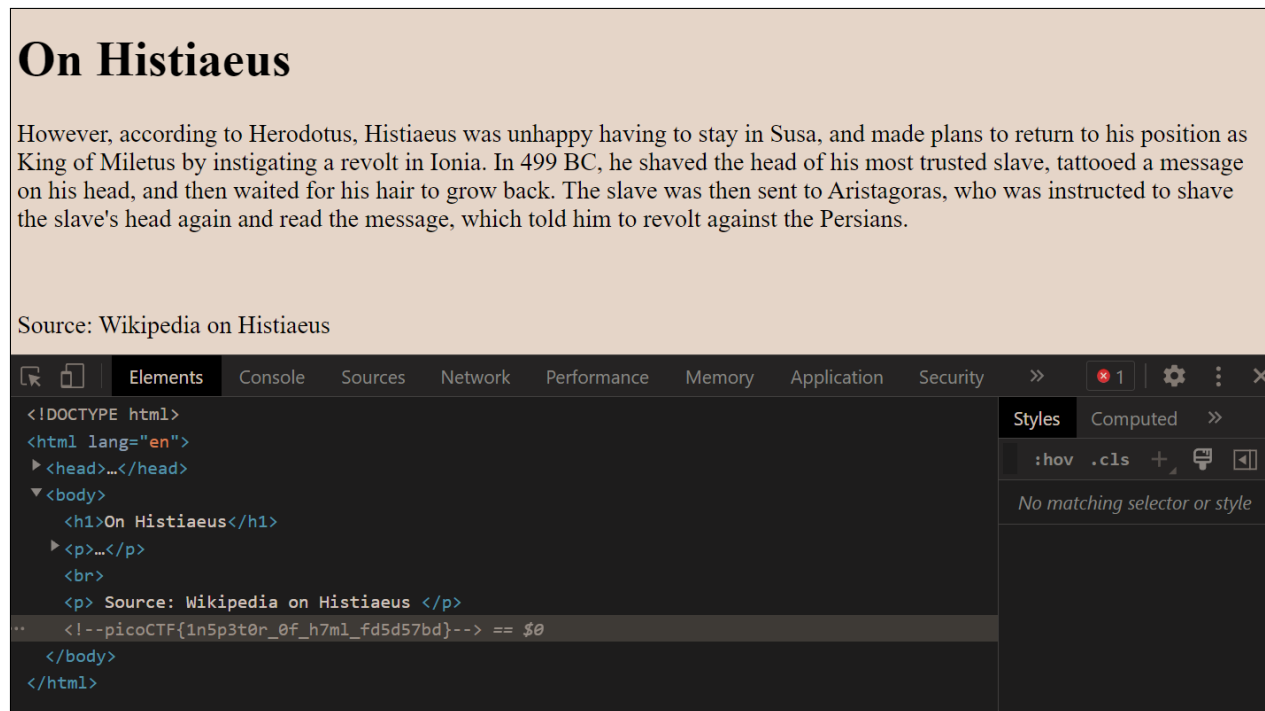


Figure 32: Inspect HTML picoCTF

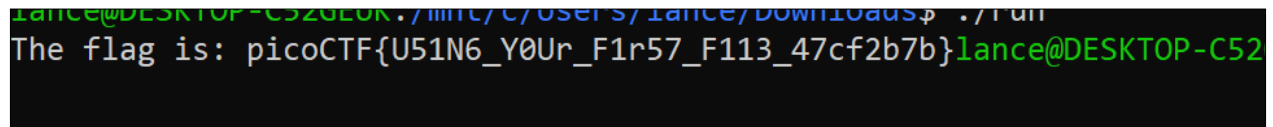
#### 4.3.4 File-Run (Fantom)

CTF level: **Low**

##### Description:

This CTF is simple to understand and so Fantom is able to give the proper solutions.

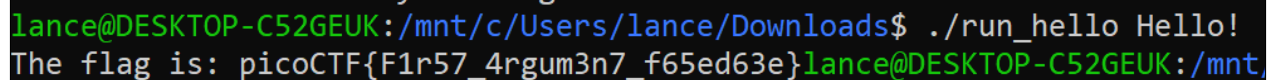
Using your first file means that some programs can do useful things and can reveal the flag



```
lance@DESKTOP-C52GEUK:/mnt/c/Users/lance/Downloads$ ./run
The flag is: picoCTF{U51N6_Y0Ur_F1r57_F113_47cf2b7b}lance@DESKTOP-C52
```

Figure 33: Using your first flag

First argument means a lot of programs can perform some input and give back some result in front of the parameter satisfaction request by the program



```
lance@DESKTOP-C52GEUK:/mnt/c/Users/lance/Downloads$ ./run_hello Hello!
The flag is: picoCTF{F1r57_4rgum3n7_f65ed63e}lance@DESKTOP-C52GEUK:/mnt,
```

Figure 34: First argument

##### Potential Business Impact:

If removing some instruction to the user these could be an interesting and simple technique to store and pass secret messages.

#### 4.3.5 CVE-XXXX-XXXXX (Fantom)

CTF level: **Low**

##### Description:

CVE are vulnerabilities discovered and recorded or vulnerabilities exploited and then recorded by MITRE in a database

Any CVE is a record of different types of vulnerability existent in the vendor's technology. Describing the threat is possible to avoid the risk and mitigate some of the vulnerability and the subsequent cases accordingly to the informations provided by the vendor of the product that is responsible of the patch.

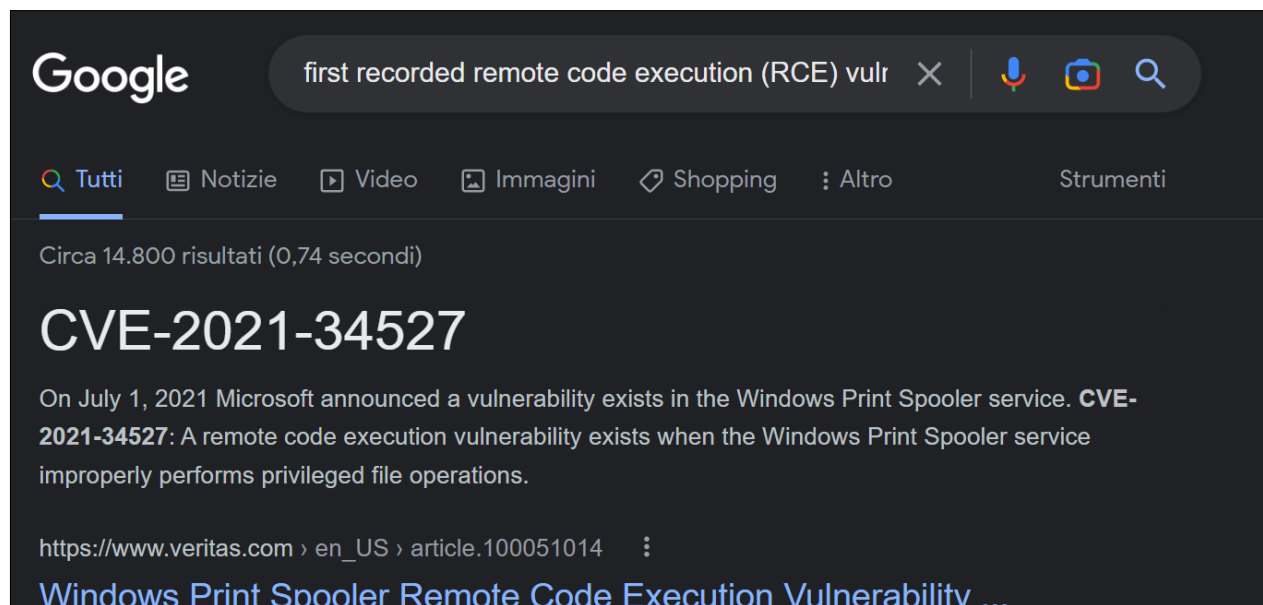


Figure 35: CVE-2021-34527

##### Potential Business Impact:

Because CVE-2021-34527 records the remote code execution exploit focused on the print spooler on Windows systems today is a vulnerability to any Windows system where spooler is operative.

##### Exploitation Details:

The spooler is a simple service which prints the input that receives on the host machine which leaves too much space to any attacker who wants to send files or programs or code and fake the spooler and abuse their privileges from a remote position.

##### Recommended Remediation:

If not used on the system it is better to disable it and if necessary to the operations the service has to be mitigated with the implementations of rules or have been seized from the extension of the spooler service at least possible.

**References:**

<https://www.cvedetails.com/cve/CVE-2021-34527/>

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-34527>

## 4.4 Informational

### 4.4.1 Opinion

#### Description:

In the first week of training at Brainnest Fantom encounter two different type of CTF challenge that could be example of real world threat

**Obfuscation by Obscurity:** This kind of technique is one of mine favorites one because of the simplicity in wich case it permit to not need the understanding of particular technology but for the sake of reason about the world when you communicate and transmit information if they are relevant is virtually impossible to prevent that some could intercept that information because of that the control about the type of information that you can transmit maybe can evade some very specialized tool or the monitoring of the government.

With this kind of technique is possible if you know the defences or monitoring control that persist in real world. It's a kind of evasion from censorship very intellectual technique.

So this extreme but simply technique has the objective to fake who is observing at the communication even in more sophisticated and monitored system remain undetectable for the most accurate adversary and persist on the system for a brief amount of time.

**Enhanced!:** Enhanced that mean something that has been improved is another example of obfuscation of information through the code ofcourse the flag is easier to find even if you don't know how to use the HTML or CSS tools and you can get the flag directly from an IDLE and just looking at the source code. Ofcourse is neither mentioned from the Mitre Att&ck matrix

but in my opinion to be able to pass some file on the internet in plain and modify some parameter in wich way rendering the text very small and then someone else can get the message after rendering again is interesting.

## 5 Conclusion

picoCTF provides many interesting CTF challenges and indepth knowledge of how the various technologies are correlated is essential when you need to explore the digital infrastructure.

After this first week Fantom can apply this knowledge and see how an adversary can find vulnerabilities in technology in programmes and in human mind.

Observe these exploit from the attacker point of view can give Fantom a hope to begin a new exiting career in the cybersecurity and to collaborate with agencies after the end of this month of training with Brainnest.

## A Mind Map



NOTICE: CONFIDENTIAL FOR CTF ONLY Page XXXVIII

## B References

| Name       | Description                                    | Link  |
|------------|--|---|
| file       | determine file type                            | <a href="https://linux.die.net/man/1/file">https://linux.die.net/man/1/file</a>   |
| onlineTRID | Online TrID File Identifier                    | <a href="https://mark0.net/onlinetrid.html">https://mark0.net/onlinetrid.html</a>   |
| dcode      | decripter                                      | <a href="https://www.dcode.fr/">https://www.dcode.fr/</a>   |
| vscode     | IDLE   | <a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>   |
| sh         | Bourne shell                                   | <a href="https://linux.die.net/man/1/sh">https://linux.die.net/man/1/sh</a>   |
| shar       | Shar Files                                     | <a href="https://linux.die.net/man/1/shar">https://linux.die.net/man/1/shar</a>   |
| ar         | ar - create, modify, and extract from archives | <a href="https://manpages.ubuntu.com/manpages/bionic/man1/ar.1.html">https://manpages.ubuntu.com/manpages/bionic/man1/ar.1.html</a> |