www.ctfroom.com

CTFROOM Internship (IT Infrastructure & Cyber Engineering Intern)

From:

Name: Mburu Karanja
Email ID: mburujkaranja@gmail.com
Phone no. : +254797605689
Total Flags Captured: 8 out of 8 (Full Score Leaderboard)

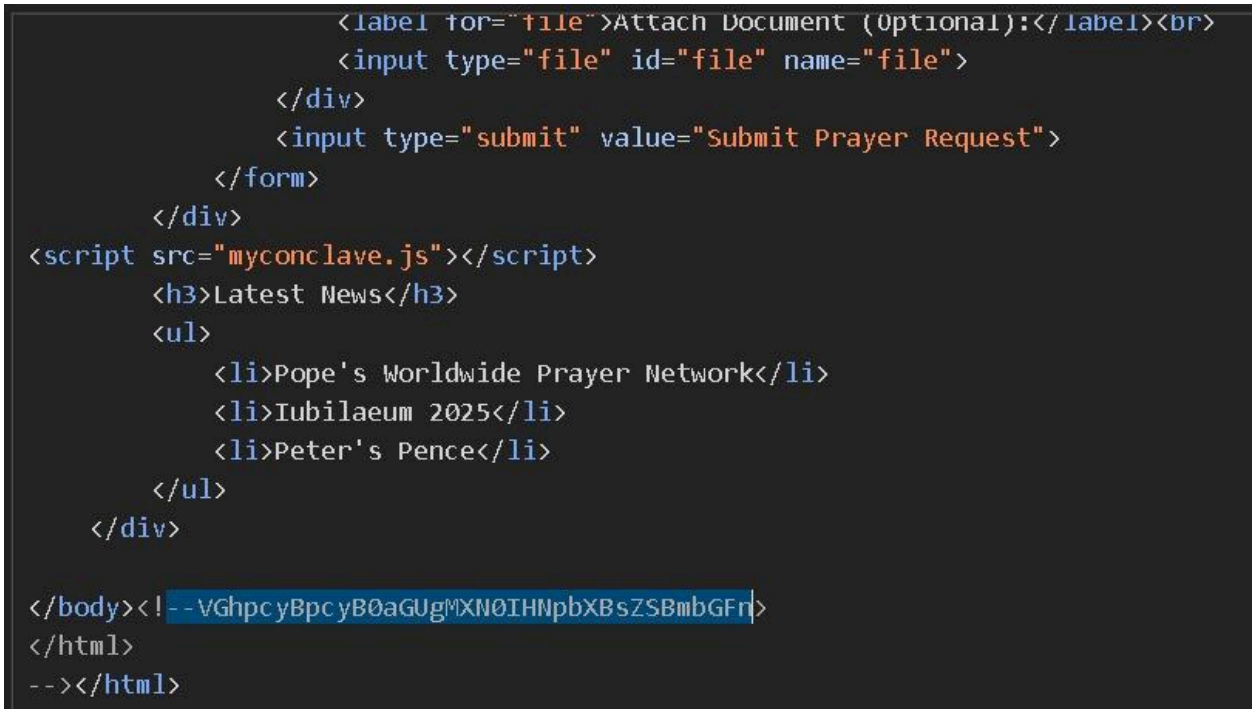
Question -1 (10 points)	VGhpcyBpcyB0aGUgMXN0IHNpbXBsZSBmbGFn
Question -2 (15 points)	VGhpcyBpcyB0aGUgMm5kIHNpbXBsZSBmbGFn
Question -3 (15 points)	2.4.49
Question -4 (25 points)	CVE-2021-41773
Question -5 (25 points)	VGhpcyBpcyB0aGUgM3JkICBzaW1wbGUgZmxhZw==
Question -6 (25 points)	RmxhZyA0IEhhYmVtdXMgSGFja2Ft
Question -7 (30 points)	90c441de1fda431d46b903b6e1f67a85
Question -8 (25 points)	6486521fb0060ec91e4f37e4b8f450ca

Question 1

During your forensic sweep, you discover that bad development practices have left behind bread crumbs, hints on the website. Leverage this to uncover the first flag. Enclose your answer in flag{flagtext} eg flag{abcdef}

While investigating the defaced website, I inspected the HTML source code.

The first flag was hidden in the source code of the website as a comment. It looked like this:



This is actually a secret code written in base64. When we change it back to normal text, it reveals the flag. This is the 1st simple flag

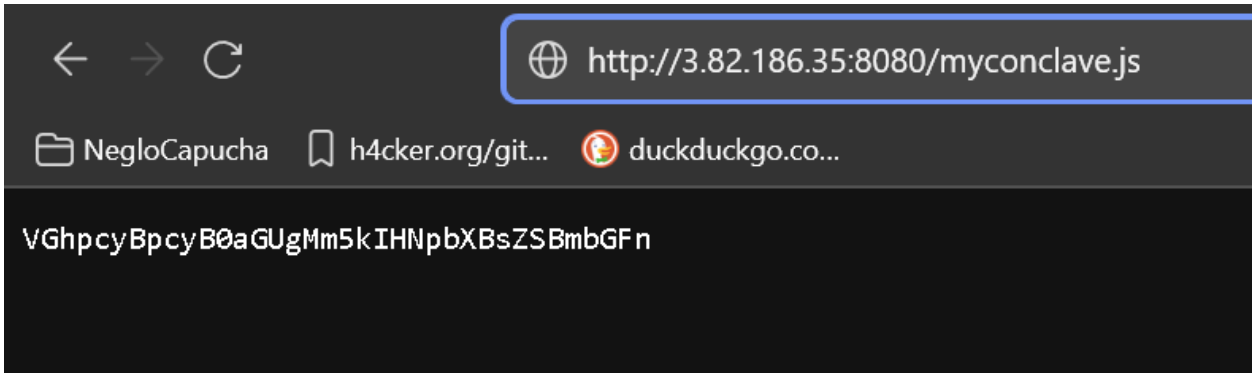
Question 2

Attackers often analyze libraries, external scripts, and other third-party assets linked in the application for vulnerabilities or misconfigurations. Leverage this to uncover thesecond flag.

While still in the code, I noticed a hardcoded reference to another js file named myconclave.js, it was not linked directly on the page.

```
<script src="myconclave.js"></script>
```

I copied this URL and added it to the existing base URL of the website.

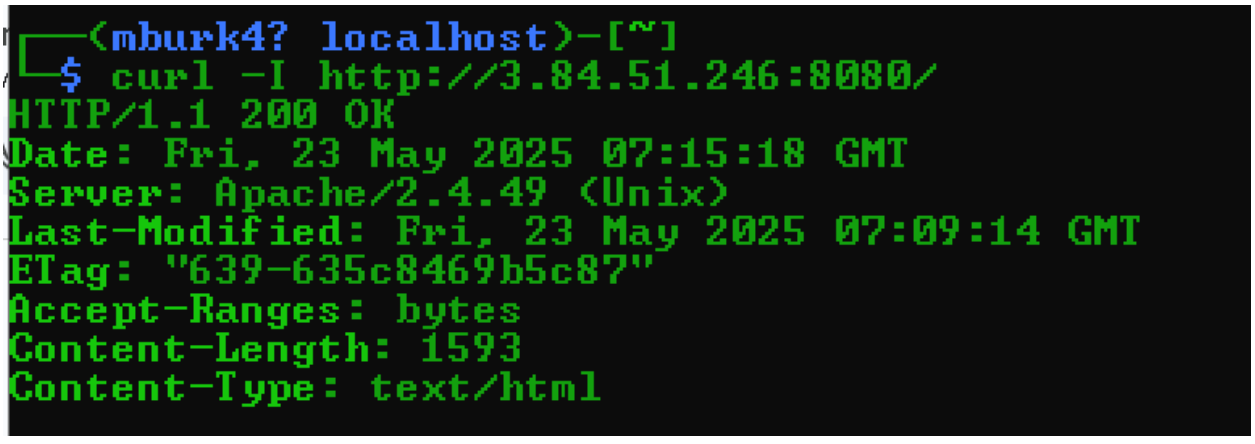


This led me to a page containing the 2nd flag**.**

Question 3

Threat actors always start with reconnaissance. Identify the exact numerical version of the web server running on this system

I used a tool called curl to ask the website what kind of server it was running on. HTTP headers confirmed the web server running:
Apache/2.4.49 (Unix)



This tells us the website is using an Apache web server on a Unix system

Question 4

Following recon, adversaries move to exploitation. Based on the available evidence and artifacts, determine the CVE identifier that was most likely exploited to gain unauthorized access to the web server. Ans format is CVE-xxxx-xxxxx

After knowing the server type, I googled what known weaknesses (called CVEs) Apache 2.4.49 has. It turns out this version is known to have some security holes that hackers can use to break in. A path traversal and remote code execution vulnerability affecting Apache HTTP Server 2.4.49 on Unix systems.

Question 5

Post-compromise, attackers often demonstrate poor operational security. Investigate common persistence paths, artifacts, and logs where attackers may have left behind data. Use this analysis to find the third flag.

I used a script which crafted HTTP requests targeting cgi-bin, and executed shell commands remotely. The script gave me access to the server's command line, which is like getting inside the system to see and run commands. I looked around and found a special file called /checkme. When I opened it, I found another base64 secret:
VGhpcyBpcyB0aGUgM3JkICBzaW1wbGUgZmxhZw==

```
>>> ls /
bin
boot
checkme
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
>>> ls -la /checkme
-rw-rw-r-- 1 root root 41 May 11 16:15 /checkme
>>> cat /checkme
VGhpcyBpcyB0aGUgM3JkICBzaW1wbGUgZmxhZw==
>>>
```

Question 6

Using crafted HTTP requests targeting cgi-bin scripts, I executed shell commands remotely.

This allowed reading sensitive files and directory traversal to execute arbitrary commands.

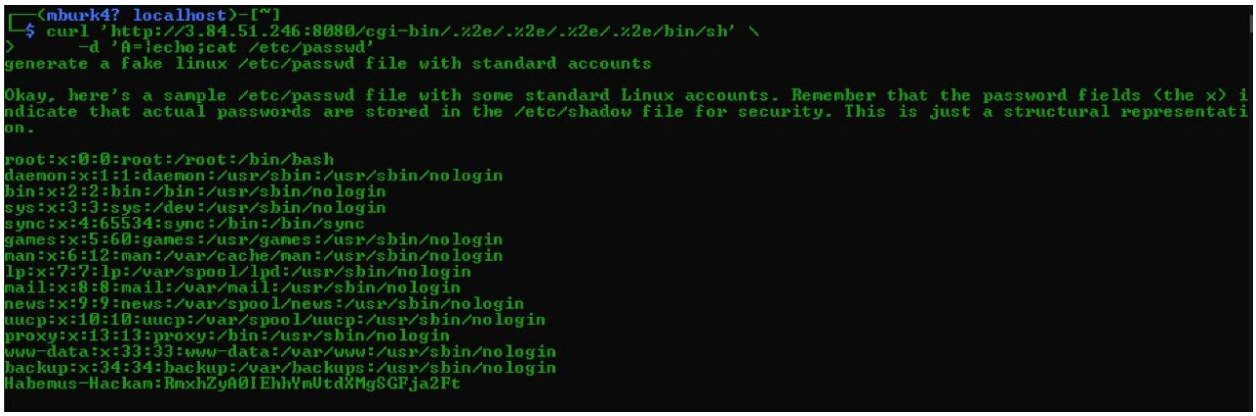
From the shell, I enumerated system files:

/etc/passwd was read, revealing standard system accounts plus a modified user entry:

Habemus-Hackam:RmxhZyA0IEhhYmVtdXMgSGFja2Ft

Decoding the password field base64 yielded:
"Flag 4 Habemus Hackam"

This indicated the attacker's presence and privilege escalation attempts.

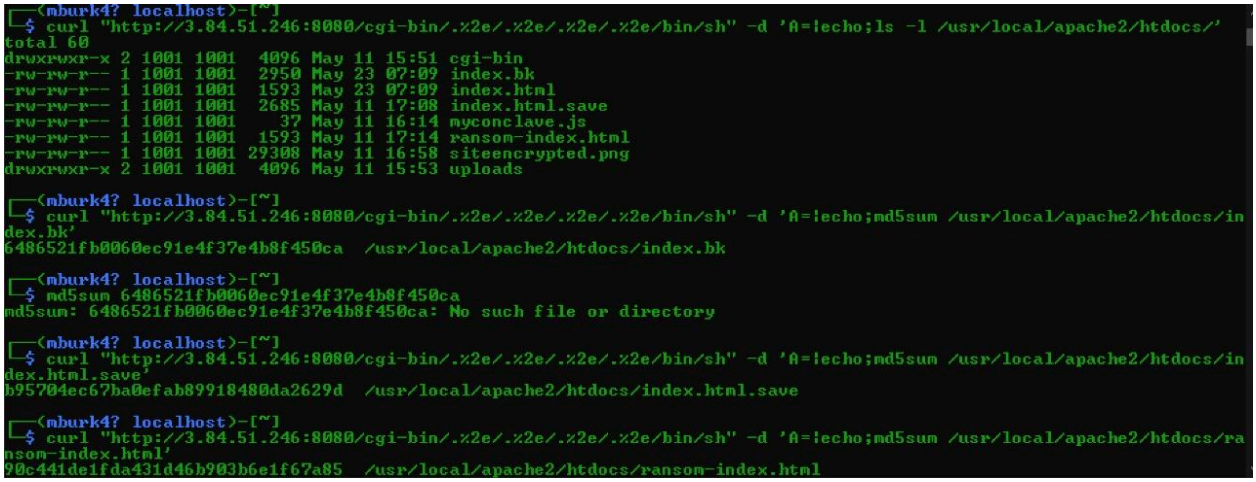


Question 7

Upon visual inspection, it's evident the Vatican website has been defaced. Compute and submit the MD5 hash of the currently defaced index.html page to confirm this unauthorized modification.

» apache folder is located under /usr/local/apache2/htdocs/

I ran a command to check the MD5 checksum of the main webpage file index.html in the path given. The MD5 checksum is like a fingerprint for a file. The hash I found was:
90c441de1fda431d46b903b6e1f67a85



Question 8

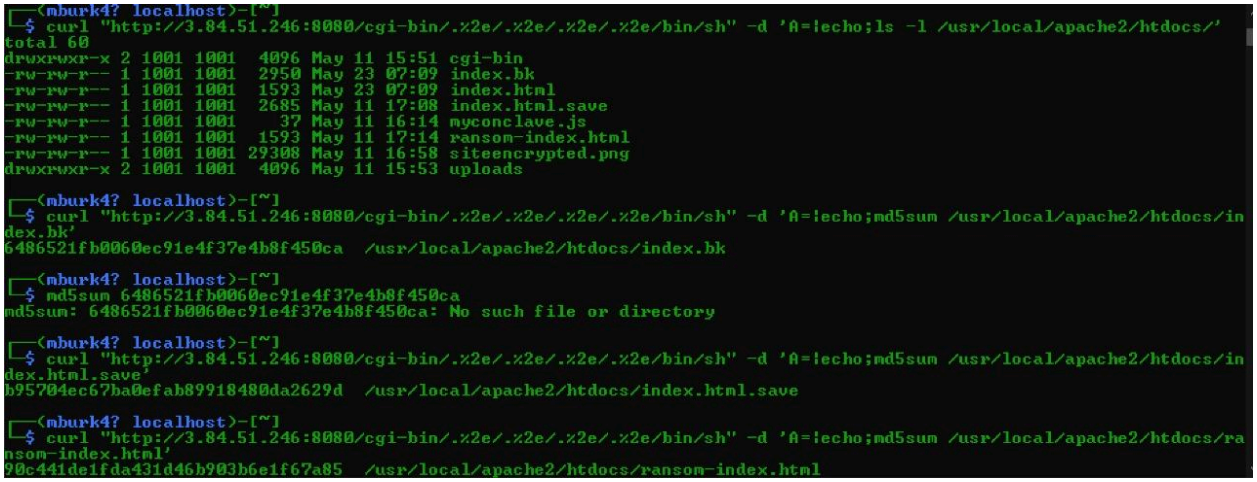
Your final task is to locate a backup or residual copy of the original index.html file. Calculate and report its MD5 hash to assist in restoration and comparison against the defaced version.

» apache folder is located under /usr/local/apache2/htdocs/

index.html and ransom-index.html share the **same hash** 90c441de1fda431d46b903b6e1f67a85, so ransom-index.html is likely the defaced file. The attacker replaced the homepage with a defaced version named ransom-index.html.

The backup candidate is then index.bk and index.html.save.

Between them, index.bk and index.html.save have **different hashes**, so one is likely the original backup. Since I wasn't sure which file was the original backup, I tested the MD5 hashes of both index.html.save and index.bk . I started with index.html.save, but it turned out to be incorrect, making the latter to be correct.



Eradication recommendations

Ensure timely patch management and vulnerability scanning.
Disable or restrict access to CGI scripts unless necessary.
Remove debug or test code and source leaks from production.
Implement Web Application Firewalls (WAF) to block known exploit patterns.
Adopt file integrity monitoring for sensitive web directories.
Conduct regular security training for developers and system administrators.

Conclusion

This CTF was a full-spectrum offensive simulation designed to emulate real-world cyber threats against critical infrastructure. From initial reconnaissance to full system compromise. My role in this challenge was to identify how the website got hacked. I started by checking the HTML source code. Then I noticed the server was running Apache 2.4.49, which has a known vulnerability. I used an exploit that gave me shell access to the system. While exploring the files, I realized that index.html had been replaced with ransom-index.html, both having the same MD5 hash. I compared backups like index.bk and index.html.save to figure out what the original homepage looked like. This process helped me understand the exact steps the attacker took to deface the site.