M&Ms CTF Writeup

Task descriptions

- Scan the web server for a backup of the application. Download and extract the file to get the first flag. *
- The system hosts a second web server which listens on localhost:12322. This service hosts a second flag ("flag.txt") in the web server's root directory. How can you access the flag on this service remotely?
- Compromise the system. A third flag can be found in the root directory ("/") of the system. Describe your actions.

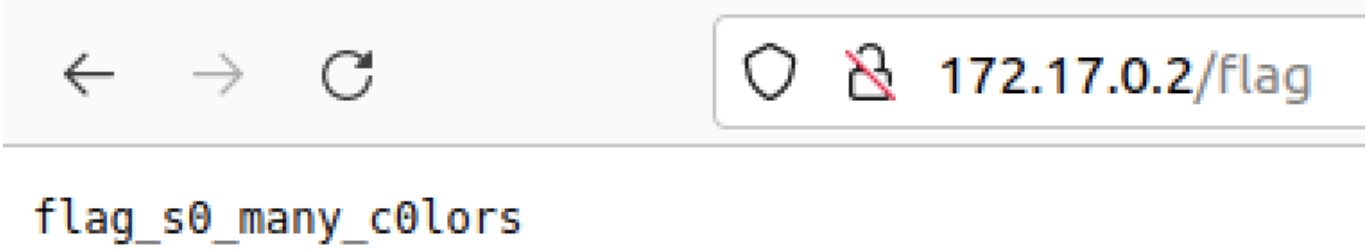
Let's go to the main page of the application first and act as a user would.

This is the webpage of our challenge. it's empty but we can notice that those guys sure are wild.

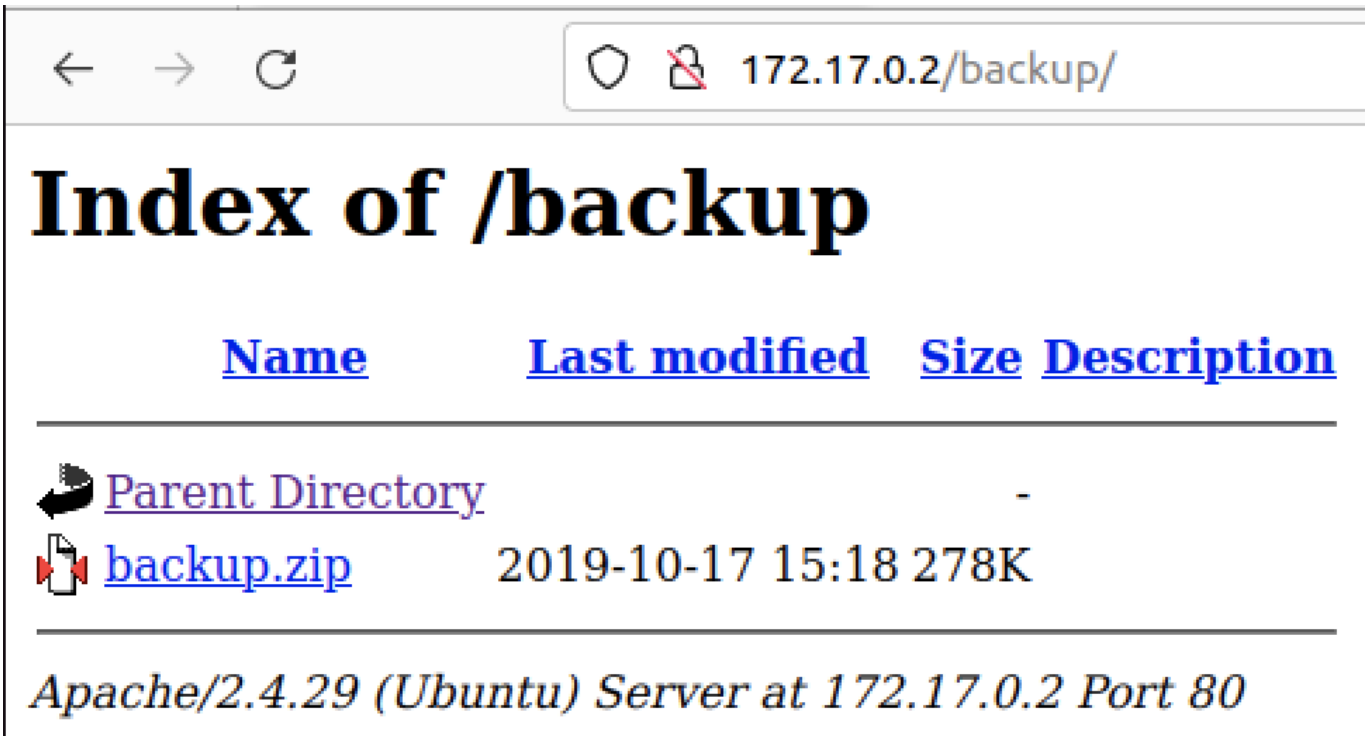
Let's start by checking potential directories with dirb command

```
klesov@klesov:~/Documents/pentesting-thu-2022/containers/mandms$ dirb http://172.17.0.2

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Jan 28 20:45:51 2023
URL_BASE: http://172.17.0.2/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://172.17.0.2/ ----
==> DIRECTORY: http://172.17.0.2/backup/
+ http://172.17.0.2/flag (CODE:200|SIZE:20)
+ http://172.17.0.2/index (CODE:200|SIZE:100)
+ http://172.17.0.2/index.html (CODE:200|SIZE:100)
+ http://172.17.0.2/phpinfo.php (CODE:200|SIZE:68963)
+ http://172.17.0.2/server-status (CODE:403|SIZE:275)

---- Entering directory: http://172.17.0.2/backup/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Sat Jan 28 20:45:52 2023
DOWNLOADED: 4612 - FOUND: 5
```

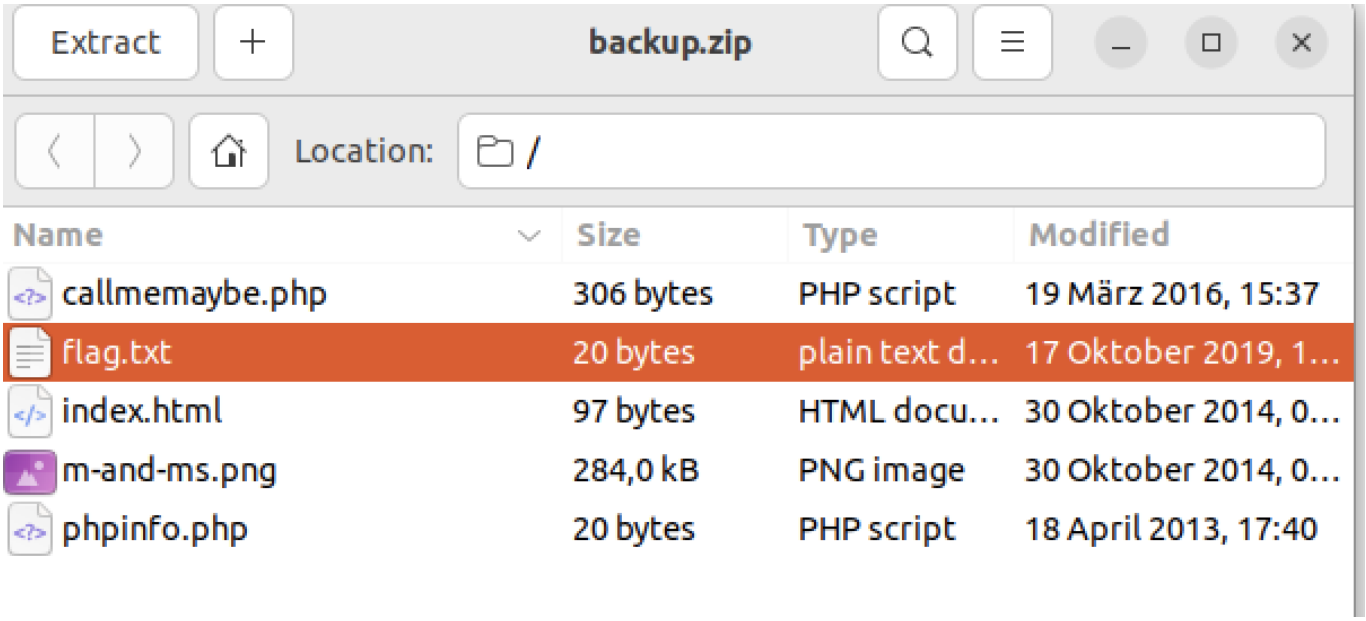
Fitst thing coming to my mind is that we get a /flag directory



Which is an ez flag, the writeup id done at first thought, but we need another flag. Let's explore other directories.



After exploring directories a little bit we notice that the first flag can also be found inside the backup directory in backup.zip file.



If we open the file with one of the greatest song's name, we can see a way to potentially acquire another flag.

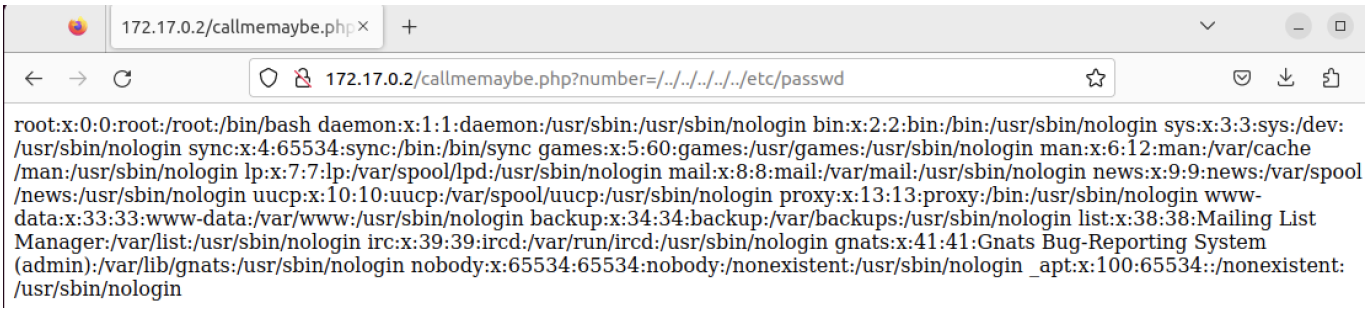
alt text

This script is a **PHP script** that appears to be vulnerable to a **file inclusion vulnerability**. The script uses the file_get_contents function to read the contents of a file specified in the $_GET['number'] parameter. The script then checks if the contents of the file contain the string "/i_am_on_the_guestlist/" and if it does, it uses the eval() function to execute the contents of the file as PHP code. The use of the eval() function with user-supplied data is a security vulnerability because it allows an attacker to **execute arbitrary code on the server**. An attacker could use this vulnerability to execute malicious code, gain access to sensitive information, or perform other actions that could compromise the security of the system. In this case, the

script is checking whether the contents of the file has the string "i_am_on_the_guestlist/" in the contents, if it does, it evaluates the file, which is a way to check if the file is authorized to be executed. But this check is not secure, it could be bypassed.

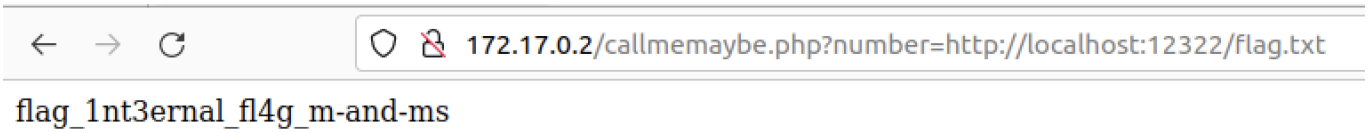
Thank you ChatGPT for explaining this basic script for my writeup, let's try to exploit **path traversal vulnerability** to access /etc/passwd directory.

Why do I think path traversal is the exploitable vulnerability here? Because The $_GET['number'] variable is used to specify the file to be included, but it is not properly sanitized, which allows us use "../../" to traverse the file system and access files outside the intended directory.



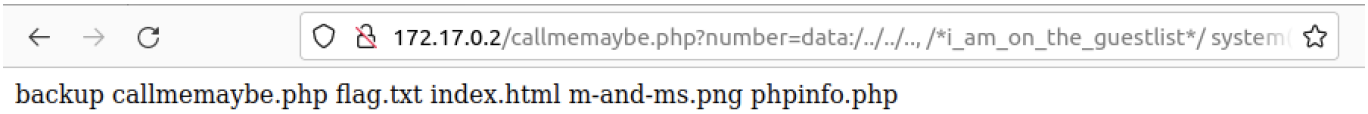
/etc/passwd file is typically readable by all users, so it is not a sensitive file itself, but it can be used to gather information about the system and potentially identify other vulnerabilities.

Since in **second task** we need to access the localhost remotely we could have used metasploit to create a reverse shell, but based on the information we already gathered + we know about localhost:12322 host, we can access web server with the following link:

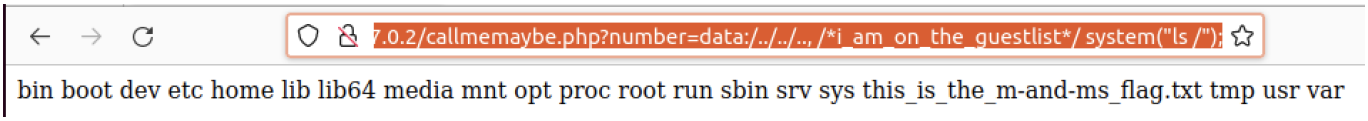


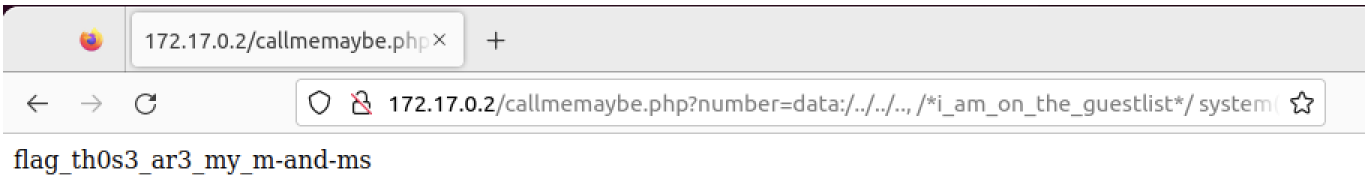
Great success!

Let's go to **task 3**, we need to access **/root** Going back to our php script from earlier we can pass parameter "number" to the function "system" from the script, which should allow us to execute arbitrary commands



The output shows us we already have access to the files in the directory By editing our path a little bit more we get the following:





flag_th0s3_ar3_my_m-and-ms

After we just cat the flag and get the output.