

HacksCTF Writeups

Cryptography

Q) oh-no - 50 points

This includes an auto-generated script when the flag was entered on this [site](#).

Flag: hacks{that_was_easY}

Q) credit-stealer - 60 points

Reading the question, the first line says WW2. Let's keep it in mind. The machine that was developed during WW2 was the Enigma machine. Now let's have a look at the inputs taken in the Enigma machine.

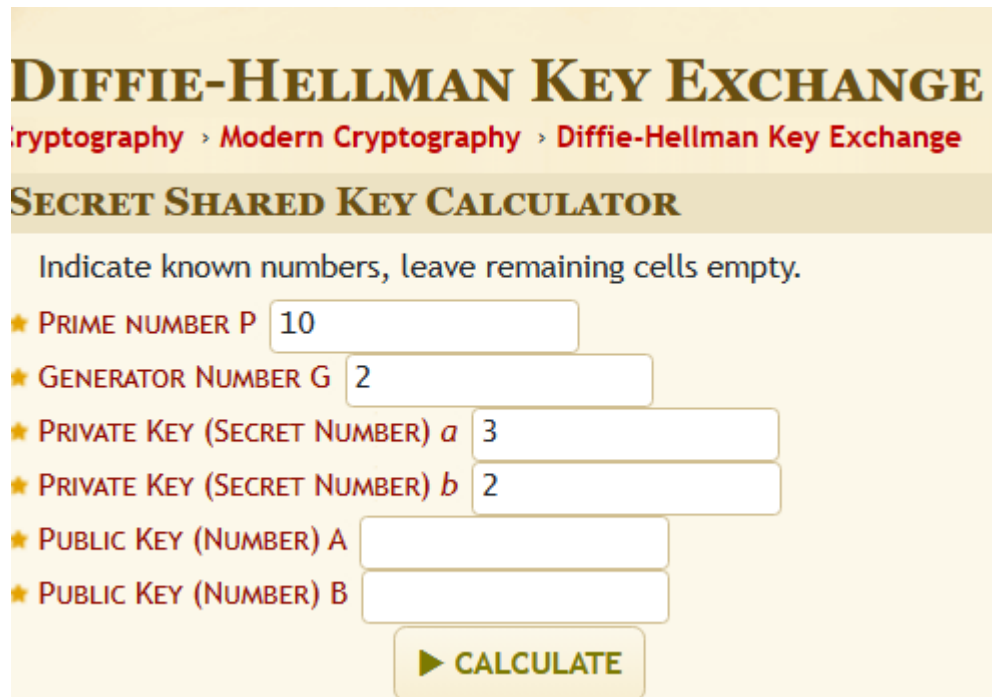
- Walzenlage - I, II, III (3 sons: 1, 2, 3). Synonymously taken as A, B, C
- Umkehrwalze - C (Third one was the most notorious)
- Grundstellung - B, B, B (2=B solved the first 3)
- Ringstellung - A, B, B (next 3 solved by 1, 2, 2 respectively)
- Steckerverbindungen - A-B, U-W (A-B solved this one together, U-W took the credit)
- Message to type on the enigma machine - GDPYMXBKRYG (Final lock was set to this)

The screenshot shows a web-based Enigma machine configuration interface. At the top, a yellow box contains the text "★ MESSAGE TO TYPE ON THE ENIGMA MACHINE" and a text input field with the value "GDPYMXBKRYG". Below this, several configuration options are listed, each with a yellow star icon and a corresponding input field or dropdown menu. The options are: "★ MACHINE TYPE" with a dropdown menu showing "Wehrmacht/Luftwaffe 3 rotors"; "★ SELECT ROTORS TO MOUNT (WALZENLAGE)" with a text input field showing "I, II, III"; "★ SELECT REFLECTOR TO MOUNT (UMKEHRWALZE)" with a dropdown menu showing "C"; "★ INITIAL POSITIONS OF ROTORS (1 PER ROTOR) (GRUNDSTELLUNG)" with a text input field showing "B, B, B"; "★ POSITIONS OF THE ALPHABET RING (1 PER ROTOR) (RINGSTELLUNG)" with a text input field showing "A, B, B"; and "★ PLUG BOARD CONFIGURATION (STECKERVERBINDUNGEN)" with a text input field showing "A-B, U-W". At the bottom right, there is a green button with a right-pointing triangle and the text "ENCRYPT/DECRYPT".

Flag: hacks{THEOGCIPHER}

Q) story-2 - 60 points

In the starting lines, P, G, a b were the characters introduced. These are the inputs for the diffie-hellman exchange. Using these settings we get the answer: 4.



The image shows a web-based calculator titled "DIFFIE-HELLMAN KEY EXCHANGE". Below the title is a breadcrumb trail: "Cryptography > Modern Cryptography > Diffie-Hellman Key Exchange". The main heading is "SECRET SHARED KEY CALCULATOR". A instruction says "Indicate known numbers, leave remaining cells empty." The form contains the following fields:

- ★ PRIME NUMBER P: 10
- ★ GENERATOR NUMBER G: 2
- ★ PRIVATE KEY (SECRET NUMBER) a : 3
- ★ PRIVATE KEY (SECRET NUMBER) b : 2
- ★ PUBLIC KEY (NUMBER) A: (empty)
- ★ PUBLIC KEY (NUMBER) B: (empty)

At the bottom is a green button with a right-pointing triangle and the text "CALCULATE".

Using 1010, 1010 as a prompt or this hint below. We google the coordinates 44,44. Getting the address: Kirovsky District, Stavropol Krai, Russia.

▼ View Hint

lets say you found out about the answer x, then you search the coordinates xx,xx you will find that district

The question asked for the district and the first line confirmed it was Russia. Hence, the flag was hacks{kirovsky}.

Q) enigma

There was no such musical piece by Beethoven which is named enigma. Using steghide, you are prompted to enter a password which is Beethoven. It could be achieved by trial and error as the hint said the password is visible in the file. Doing this, you will get the flag = Hacks{Notes_look_fishy}

Q) Figurines

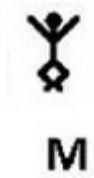
The part of a URL is given:

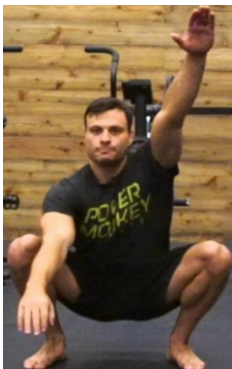
"1RdzSmugPMttCMYEMHnGjAy77EfX9WLb9LIBX6j-awSo/edit"

This URL directs to google Docs file link :

<https://docs.google.com/document/d/1RdzSmugPMttCMyEMHnGjAy77EfX9WLb9LIBX6j-awSo/edit>

The document has, what look like, various unrelated images. The page is littered with various "Sherlock" hints (I AM _ _ _ _ LOCKED and file author is "baker"). Googling Sherlock ciphers gives a "Dancing Man cipher". Comparing the images with any reference of the Dancing man Cipher gives the flag, "MORI4RITY". Cross-check: Moriarity is also the main antagonist of Sherlock Holmes.





4



R



I



T



Y

flag: hacks{MORI4RITY}

Q) BreakingBad

The file Breaking_bad.pdf is a protected file, the password of the file is 10.63.255.254 which can be found by tracking the last host of the IP address given in the question.

Once the file is open you will encounter the ciphered text 3AM^33\$:+=+@/C<2)?@81*B9.

This text must be deciphered using **base 85 decryption** which will result in the following text
92 18 68 AD 53 92 M.

This text has to be deciphered using **periodic table cipher** which will finally result in the required flag which is **UARERADIUM**

Q) Dinner With Salsa (Dance)

A phone book is given, in which all but four have different country codes. These are also the only four Mexican names along with other hints like “Salsa” leads to the **Mexican Army Cipher Wheel**.

The phone numbers of these four (except the trailing 001, 002, 003, 004) together when decoded with the first names of the people and country codes of their phone numbers

A : 01, J: 44, G: 73, O: 91 for wheel positions:

And cipher (concatenating first 7 digits of all phone numbers) :

8415352052585801984446117776

Gives “**TORTILLAHACKED**”

Flag : hacks{TORTILLAHACKED}

Q) perfect-square

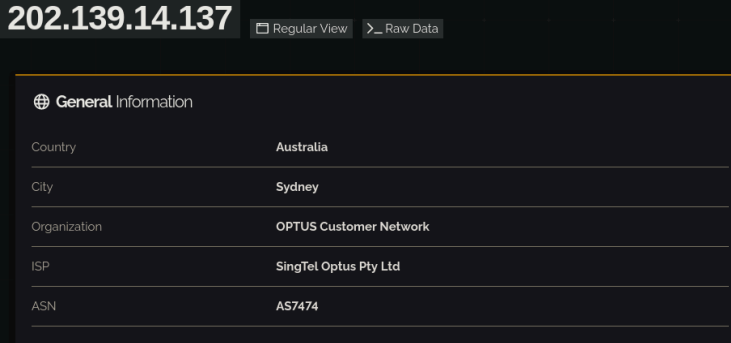
This can be solved after converting from base 64 till you get a readable text(48 times) on cyber chef. Or you can write a python script to detect a “hacks” string while decoding in an infinite loop.

Flag: hacks{s0_many_64s}

OSINT

Q: guesswhereami

An IP is given, whose location needs to be found. Simply use shodan.io to search for the IP. Location is Sydney

A screenshot of the Shodan search results page for the IP address 202.139.14.137. The page has a dark theme. At the top, the IP address is displayed in a large font, followed by buttons for 'Regular View' and 'Raw Data'. Below this is a section titled 'General Information' with a globe icon. It contains a table with the following data: Country: Australia, City: Sydney, Organization: OPTUS Customer Network, ISP: SingTel Optus Pty Ltd, and ASN: AS7474.

General Information	
Country	Australia
City	Sydney
Organization	OPTUS Customer Network
ISP	SingTel Optus Pty Ltd
ASN	AS7474

Flag: hacks{Sydney}

Q: wheredidiarrive

An image of a container is given in the question, it can be tracked using the tracking website of Hapag-lloyd :

<https://www.hapag-lloyd.com/en/online-business/track/track-by-container-solution.html?container=HLXU++6652250>

Last arrival was in Chicago.

Flag: hacks{Chicago}

Q: india

An image “chips.jpeg” is given, Using the brand of the chips “Krachitos” the location can be traced to Argentina. Brute forcing river names according to hints gives the flag.

Flag: hacks{coloradoriver}

Pwn

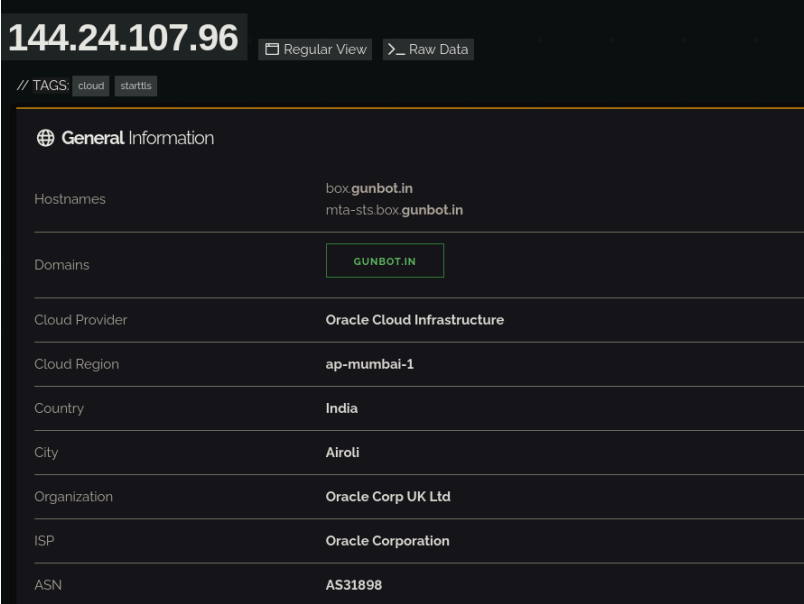
Q) jarrrrrr

Question presents a complex.jar file. After running the file with **java -jar complex.jar**, we are prompted for a registration code. The field can be overflowed, and then prompts for the option to Breach the Archive Vault or Enter the Developer's Sanctum.

Choosing Option 1 prompts for cloud region of the fingerprint :
20:e5:ba:71:ef:bf:97:82:a2:ce:50:33:6c:74:55:f0

Wrong answer gives the prompt : "Incorrect. Come On.Its Not That Hard.SHOW it tDAN"

Now searching the given fingerprint on shodan.io gives:



144.24.107.96 Regular View Raw Data

// TAGS: cloud starttis

General Information

Hostnames	box.gunbot.in mta-sts.box.gunbot.in
Domains	GUNBOT.IN
Cloud Provider	Oracle Cloud Infrastructure
Cloud Region	ap-mumbai-1
Country	India
City	Airoli
Organization	Oracle Corp UK Ltd
ISP	Oracle Corporation
ASN	AS31898

Entering the cloud-region : ap-mumbai-1 in the prompt, gives the flag.

Flag: hacks{1nt3g3rs_G0ne_Wr0ng}

Rabbit Hole:

Name	Date modified	Type	Size
META-INF	20-01-2024 19:41	File folder	
asset	20-01-2024 19:41	Text Document	1 KB
corruption_state	20-01-2024 19:41	Text Document	0 KB
hint	20-01-2024 19:41	Text Document	0 KB
Main.class	20-01-2024 19:41	CLASS File	5 KB

Q)quick-math

Here is the [script](#).

Forensics

Q) gif gif - 170 points

`convert gi.gif %d.png`

You get the frames into different images. Aligning them as such u observe that 7 different QR codes are present.

Now you can either run a command right here as they are in a multiple of 8.

or move the those 8 images to a different directory and run



`convert -append *.png out.`

You get 8 qrs after that, scanning 7 of them will lead you to a youtube video and one will lead to you this site:

<http://l.ead.me/hacksctf>. Going to this site, you get the flag: hacks{P4YTm_KAR0}

Q)Missile-inbound - 700 points

Firstly, File extension was .zip, running bruteforce on it. Going deeper, you have 3 more zips, audio, img and not_audio. Bruteforcing not_audio, you get the password: 02469. Now we have 4 more zips. Doing md5 sum *, we notice that two files are copies of each other. Opening one of them, you get conversation.wav. Listening to it, you see that the dials are DTMF tones.

Go to this site: [Detect DTMF](#) and attach the cropped wav file. You will get the below output.

Detect DTMF Tones

Detect DTMF Tones

no graphic available at this time (child process exited abnormally)

Sample Format RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 2400

Sample Size 1,364,426 bytes
approximately 682,008 usable samples
28.4 seconds

Tones Found	Tone	Start Offset [ms]	End Offset [ms]	Length [ms]
	0	22,082 ± 15	22,413 ± 15	331 ± 30
	1	22,444 ± 15	22,745 ± 15	301 ± 30
	#	22,775 ± 15	23,107 ± 15	331 ± 30
	7	23,137 ± 15	23,439 ± 15	301 ± 30
	3	23,469 ± 15	23,801 ± 15	331 ± 30
	A	23,831 ± 15	24,163 ± 15	331 ± 30
	5	24,193 ± 15	24,495 ± 15	301 ± 30
	1	24,525 ± 15	24,857 ± 15	331 ± 30
	*	24,887 ± 15	25,189 ± 15	301 ± 30
	3	25,219 ± 15	25,551 ± 15	331 ± 30
	6	25,581 ± 15	25,913 ± 15	331 ± 30
	9	25,943 ± 15	26,245 ± 15	301 ± 30

After seeing this you get final flag: hacks{01#7-3A51-*369}

Q) October 25, 2001 - 230 points

The file was hidden in the img folder of b00m.zip.

That folder too had to be brute forced.

After opening that file, you find a windows xp wallpaper. The question named helped us deduce it.

```
binwalk -e output.jpg
```

It will create a folder containing 3 jpg which contain parts of the flag on the top right corner

Append them and you can get the flag: hacks{png5_k1nd4_b0r1n9_6gh9f7b}.

Misc

Q) Simple-Crypto - 40 points

stehigde password was 8, infinity rotate = 8

You get a zip link. Unlocking it with hostisangry as given in question

pbbxa://lzqdm.owwotm.kwu/lzqdm/nwtlmza/1Tle3vvWzCzY7LV49fvLe6IMA8ydYID0A?cax=apizqvo

Use rot13 to get the drive link

<https://drive.google.com/drive/folders/1LAW3nnOrUrQ7DN49xnDw6AES8qvQAV0S?usp=sharing>

Q) Aussie - 40 points

[Aussie Language](#). This was used to write a script in aussie++ language and using the down under method the question file was generated. You could have either gone here or simply just had a look at the script and figured the flag: hacks{chuckasickie}

Q) magik-magik - 100 points

The given file was in .cad format. Running file command, one can see it is a 7z file. When that file is extracted, you get another file which has a wrong file extension. Changing it to .txt format, you get the flag: hacks{M3t@dAt&_man!pu(aTi*n)}

Both these file conversions can be achieved by using **mv** command.

Q)Trans-dimensional Message - 150 points

Skew on ms paint :)

Q) Survey - 200 points

After completing the survey held on google forms, you get the flag: hacks{thank_ya_see_you_next_time}

Q) freq analysis - 230 points

[Frequency Script](#). Using this script, you can write the frequency of the hashes by the side. After observing carefully, you see that 2 hashes are only occurring once. These are { } respectively. Marking those in the file, the above 5 letters would be hacks. Replacing all the hashes with those 5 letters. Next, the most recurring hash is a space. The next is e and doing so you will observe some words forming. Replacing by time, you get the flag:

```
hacks{freq_anal_using_aes}
```

This can also be done by converting the hashes to text but being a beginner level CTF, this sounds fine.

Reverse

Q) Armageddon Sequence

The challenge starts out with a prompt for a “Sequence”. Using `strings` on this binary will result in some basic outputs like “Bzzt!” and the success string “Sickkk...”.

If we decompile the binary with something like Ghidra, we see the success condition where the success string is used:

```
if (local_c0 == &DAT_13398000) {
    puts("Sickkk. Wrap that in hacks{}.");
}
else {
    bzzt(0);
}
```

So now we know our goal, to get `local_c0` = to that amount.

Looking at the code right above the the stub mentioned, we find the meat of our problem:

```
puts("Enter the password:");
read(0,buffer,0x10);

// Check input for correctness.
local_c0 = &DAT_13386000;
for (i = 0; i < 0x40; i += 1) {
    j = i;
    if (i < 0) {
        j = i + 3;
    }
    switch((int)(char)(buffer[j >> 2] ^ mangle_buf[j >> 2]) >>
        (((char)i - ((byte)j & 0xfc)) * 2 & 0x1f) & 3) {
    case 0:
        local_c0 = local_c0 + -0x15000;
        break;
    case 1:
```

```

        local_c0 = local_c0 + -0x1000;
        break;
    case 2:
        local_c0 = local_c0 + 0x1000;
        break;
    case 3:
        local_c0 = local_c0 + 0x15000;
    }
}

```

This is the driving logic for our problem. From this logic, it should be easy to see that we simply need to go from 13386000 to 1339800 through a combination of 4 provided paths. One way to do this would be to through triggering the sequence of cases 3, 1, 1, and 1:

Case 3	$0x13386000 + 0x15000$	$0x1339B000$
Case 1	$0x1339B000 - 0x1000$	$0x1339A000$
Case 1	$0x1339A000 - 0x1000$	$0x13399000$
Case 1	$0x13399000 - 0x1000$	$0x13398000$

But if we actually tried this, it would result in a (handled) segfault. But why does this happen?

As it turns out, some of the operations in assembly required to evaluate this sequence:

```

MOV     RAX,qword ptr [RBP + local_c0] ; read from 0x4012fe
MOV     AL,byte ptr [RAX]=>DAT_13386000 ; read from 0x401305

```

cause a read to sections of memory which are made completely **inaccessible** to the program (no **read**, **write** or **execute** perms), causing a segfault when reading from them. Debugging with the segfaulting input and stepping through the instructions one by one would make it clear where the illegal read is happening.

The ultimate goal of this challenge is to find (one of) the sequences of inputs that don't trigger any of the illegal reads.

Flag: hacks{kEmCYi8OxTO?PAE8}

THE END