

UIUCTF 2023 – Team Hacktoria



MISC

vimjail1

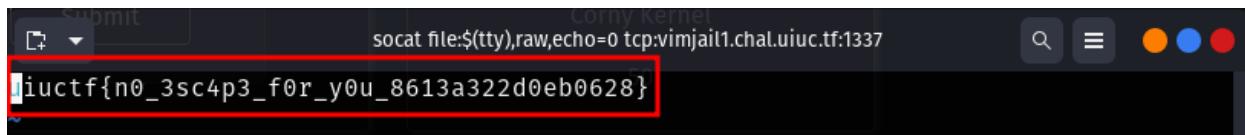
The screenshot shows a challenge card for 'vimjail1'. At the top left is a 'Challenge' button and a '114 Solves' counter. A close button 'X' is at the top right. The challenge title 'vimjail1' is centered above a large '50' representing the score. Below the score is a 'medium' difficulty rating. The challenge description reads: 'Connect with socat file:\$(tty),raw,echo=0 tcp:vimjail1.chal.uiuc.tf:1337. You may need to install socat.' The author is listed as 'richard'. Below the description are four download buttons: 'Dockerfile', 'entry.sh', 'nsjail.cfg', and 'vimrc' (which is highlighted in blue). At the bottom are 'Flag' and 'Submit' buttons.

For this problem, we started off by looking at the vimrc file to see what our bounds are:

```
set nocompatible
set insertmode

inoremap <c-o> nope
inoremap <c-l> nope
inoremap <c-z> nope
inoremap <c-\><c-n> nope|
```

We can see the remaps of some of the well-known ways to exit the insert mode in vim. After trying multiple things, we also see that it blocks `<c-l><c-n>`, but we can bypass this by typing <c-l> 2 times followed by <c-n> which lets us leave the insert mode. After this, we typed :e /flag.txt which allows us to edit the flag.txt and we get the flag:

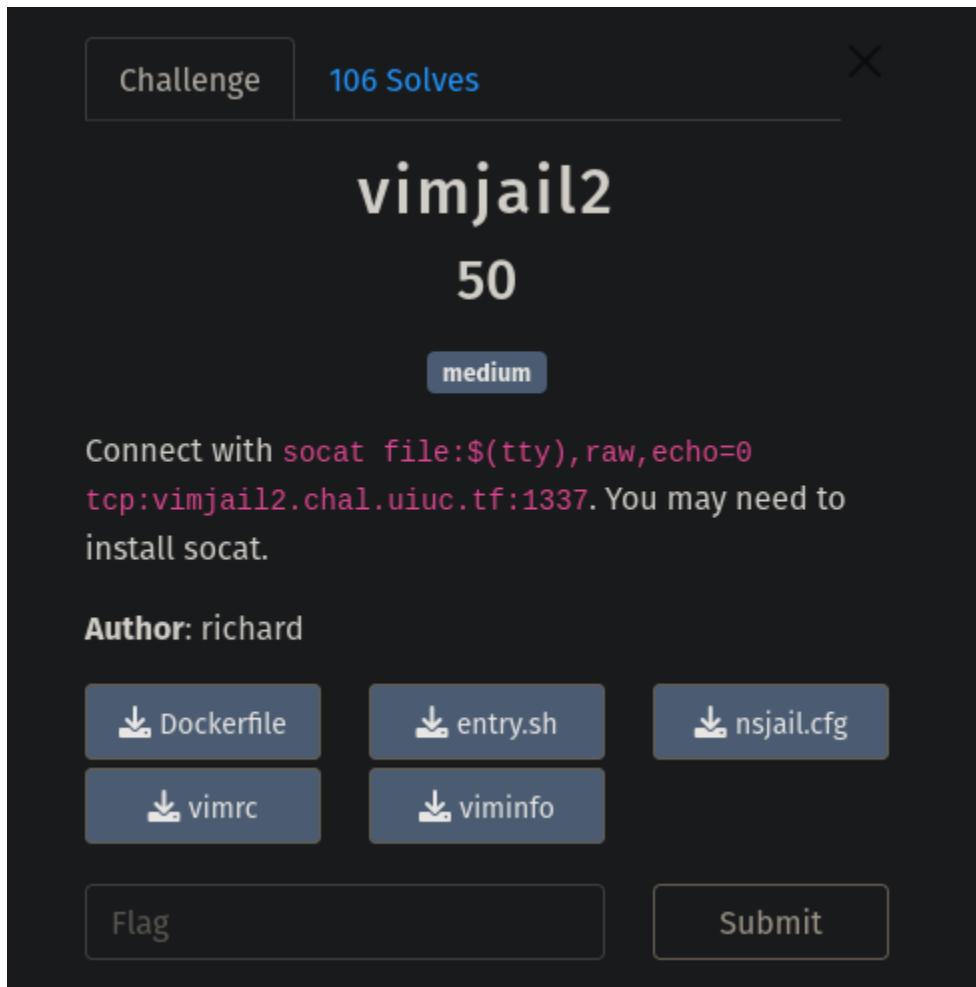


```
Copy Kernel
```

```
socat file:$(tty),raw,echo=0 tcp:vimjail1.chal.uiuc.tf:1337
```

```
uiuctf{n0_3sc4p3_f0r_y0u_8613a322d0eb0628}
```

vimjail2



Challenge 106 Solves X

vimjail2

50

medium

Connect with `socat file:$(tty),raw,echo=0`
`tcp:vimjail2.chal.uiuc.tf:1337`. You may need to
install socat.

Author: richard

[Dockerfile](#) [entry.sh](#) [nsjail.cfg](#)

[vimrc](#) [viminfo](#)

[Flag](#) [Submit](#)

We followed a similar mindset to the previous vimjail question as well. We looked at what the bounds were:


```
1 set nocompatible
2 set insertmode
3
4 inoremap <c-o> nope
5 inoremap <c-l> nope
6 inoremap <c-z> nope
7 inoremap <c-\><c-n> nope
8
9 cnoremap a -
10 cnoremap b -
11 cnoremap c -
12 cnoremap d -
13 cnoremap e -
14 cnoremap f -
15 cnoremap g -
16 cnoremap h -
17 cnoremap i -
18 cnoremap j -
19 cnoremap k -
20 cnoremap l -
21 cnoremap m -
22 cnoremap n -
23 cnoremap o -
24 cnoremap p -
25 cnoremap r -
26 cnoremap s -
27 cnoremap t -
28 cnoremap u -
29 cnoremap v -
30 cnoremap w -
31 cnoremap x -
32 cnoremap y -
33 cnoremap z -
34 cnoremap ! -
35 cnoremap @ -
36 cnoremap # -
37 cnoremap $ -
38 cnoremap % -
39 cnoremap ^ -
40 cnoremap & -
41 cnoremap * -
42 cnoremap - -
43 cnoremap + -
44 cnoremap = -
45 cnoremap ` -
46 cnoremap ~ -
47 cnoremap { -
48 cnoremap } -
49 cnoremap [ -
50 cnoremap ] -
51 cnoremap \|
52 cnoremap \_ -
53 cnoremap ; -
```

Every character seemed to be remapped to a `_` except the character q. First we need to leave the insert mode and to do so we need to follow the same approach as vimjail1, which was typing <c-\> 2 times followed by <c-n>. Then after running `:q` to quit from vim, we are able to see this message:

```
VIM - Vi IMproved
Copyright © 2019 Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable

      Become a registered Vim user!
type :help register<Enter>   for information

117 type :q<Enter>           to exit
type :help<Enter> or <F1> for on-line help
type :help version8<Enter>   for version info

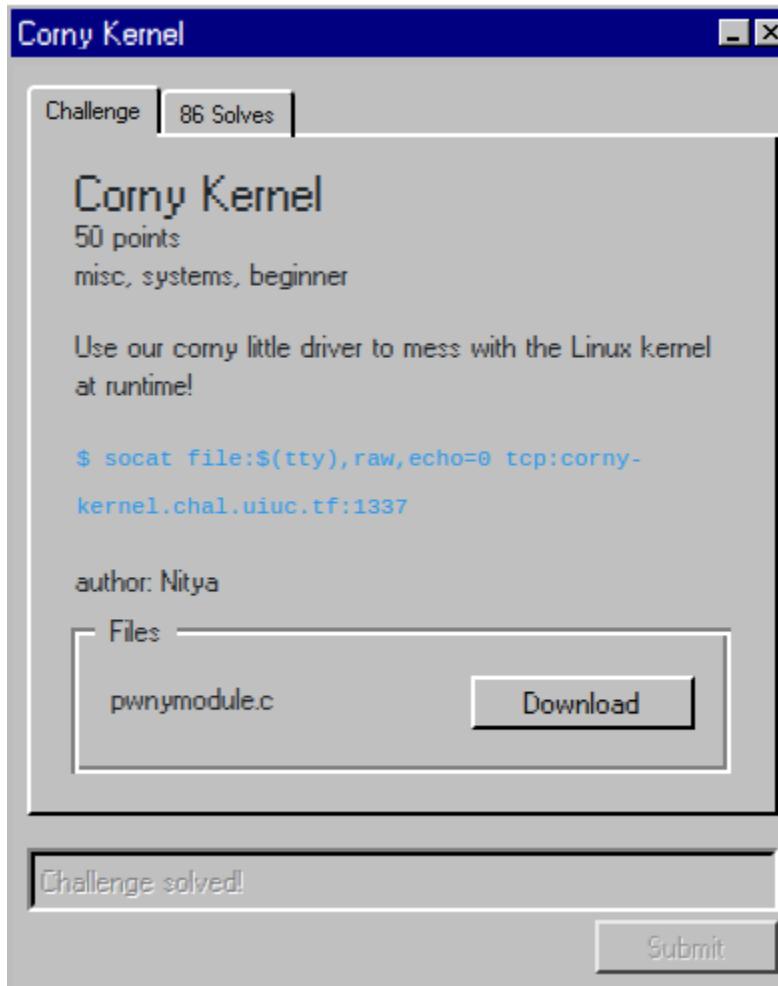
E137: Viminfo file is not writable: /home/user/viminfo
Press ENTER or type command to continue
```

After we hit enter, we are able to see the flag:

```
E137: Viminfo file is not writable: /home/user/viminfo
Press ENTER or type command to continue
uiuctf{<left><left><left><left>_c364201e0d86171b}
  | ~
```

Flag: uiuctf{<left><left><left><left>_c364201e0d86171b}

Corny Kernel



I started off by running the command provided:

```
(kali㉿kali)-[~/Downloads]
└─$ socat file:$(tty),raw,echo=0 tcp:corny-kernel.chal.uiuc.tf:1337
= proof-of-work: disabled =
+ mount -n -t proc -o nosuid,noexec,nodev proc /proc/
+ mkdir -p /dev /sys /etc
+ mount -n -t devtmpfs -o 'mode=0755,nosuid,noexec' devtmpfs /dev
+ mount -n -t sysfs -o nosuid,noexec,nodev sys /sys
+ cd /root
+ exec setsid cttyhack ash -l
/root #
```

I noticed that I was logged in as root. I see that we are given the kernel module, but it seemed to be zipped. I then ran `gunzip` on the file, and was able to see the kernel module:

```
(kali㉿kali)-[~/Downloads]
$ socat file:${tty},raw,echo=0 tcp:corny-kernel.chal.uiuc.tf:1337
= proof-of-work: disabled =
+ mount -n -t proc -o nosuid,noexec,nodev proc /proc/
+ mkdir -p /dev /sys /etc
+ mount -n -t devtmpfs -o 'mode=0755,nosuid,noexec' devtmpfs /dev
+ mount -n -t sysfs -o nosuid,noexec,nodev sys /sys
+ cd /root
+ exec setsid ctythack ash -l
/root # ls
pwnymodule.ko.gz
/root # gunzip pwnymodule.ko.gz
/root # ls
pwnymodule.ko
/root #
```

From there, I loaded the kernel module using `insmod`, which let me see a partial flag:

```
/root # insmod pwnymodule.ko
[ 202.694291] pwnymodule: uiuctf{m4ster_
```

I then removed the module by running `rmmod`. I then ran `dmesg` to see the other debugging messages provided, and was able to get the flag:

```
[ 11.606739] pwnymodule: uiuctf{m4ster_
[ 119.125544] pwnymodule: k3rNE1_haCk3r}
```

Flag: uiuctf{m4ster_k3rNE1_haCk3r}

OSINT

Finding Artifacts 1:

Challenge 370 Solves X

Finding Artifacts 1

50

osint museum

David is on a trip to collect and document some of the world's greatest artifacts. He is looking for a coveted bronze statue of the "Excellent One" in New York City. What museum is this located at? The flag format is the location name in lowercase, separated by underscores. For example: uiuctf{statue_of_liberty}

► View Hint
► View Hint

Flag Submit

For OSINT challenges, I like to follow the gap analysis ideology which means you ask the following questions:

1. What do I know?
2. What does this mean?
3. So what do I need to know?
4. How do I find out?

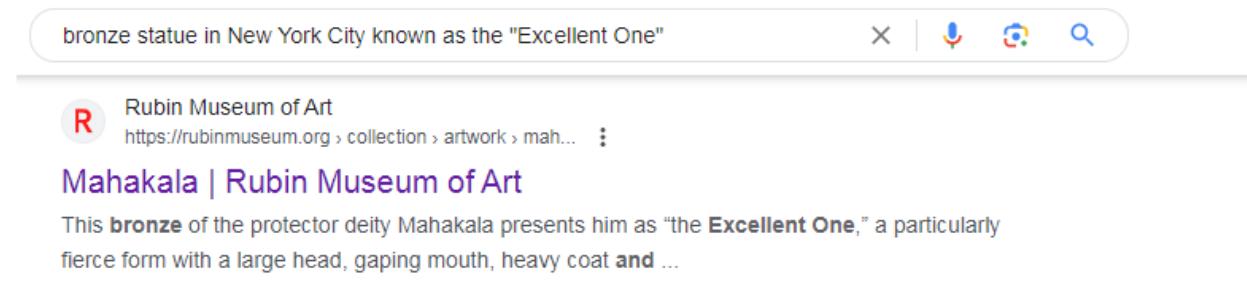
So let's start with **What do I know?**: By reading the instructions, we can directly know that we are looking for the location of one of the world's greatest artifacts, which is a coveted bronze statue of the "excellent one" in New York city.

So what does this mean?: Well, for this challenge it was obvious what that means. It means this is one of the world's greatest artifacts, which is a coveted bronze statue located in New York known as the “excellent one”.

So what do I need to know?: Before finding the location, we have to find the statue.

How do I find out? Searching about any bronze statue that is known as the “excellent one” in New York.

So let's start searching - First, I thought that maybe the statue is named the “excellent one” but of course , that was gonna be too easy. Seeing “excellent one” between quotes made me realize that the name of the statue means “excellent one” in another language or the thing it represents is known as the “excellent one”, so without overthinking too much I went to google and searched for “bronze statue in new york city known as “excellent one”” and one of my results was this:



A screenshot of a Google search results page. The search query in the bar is "bronze statue in New York City known as the \"Excellent One\"". Below the search bar, the first result is from the Rubin Museum of Art, featuring a small red 'R' logo and the text "Rubin Museum of Art" followed by a link to <https://rubinmuseum.org>. The result title is "Mahakala | Rubin Museum of Art". A snippet below the title reads: "This **bronze** of the protector deity Mahakala presents him as “the **Excellent One**,” a particularly fierce form with a large head, gaping mouth, heavy coat and ...".

So this is the flag: uiuctf{rubin_museum_of_art}

Finding Artifacts 2:

Challenge 389 Solves X

Finding Artifacts 2

50

osint **museum**

New York City is known for its sprawling subway system. However, none of that would have been possible without modern earth-moving equipment. Find where the first ever shovel was used to start digging the subway. Flag format should be in uiuctf{name_of_museum}

► View Hint

Flag Submit

Let's use a different path here, as for this challenge it is obvious that we are looking for the location of the first ever shovel used to start digging a subway in New York. First we search for the shovel: By typing "shovel of first subway in new york" we get the answer in the first result:

shovel of first subway in new york



Images

News

Route

Videos

Shopping

Books

Maps

Flights

Finance

About 16,000,000 results (0.37 seconds) « Add Grepper Answer (a)

On March 24, 1900 it was "Tunnel Day" in New York City, for that was the day Mayor Robert Van Wyck used a **sterling silver Tiffany shovel** to ceremonially launch construction of the city's first subway. Mar 24, 2019



CBS News

<https://www.cbsnews.com> › Sunday Morning



[Almanac: Building the New York City subway - CBS News](#)

[About featured snippets](#) • [Feedback](#)

So it was a "sterling silver Tiffany shovel". Next we search for the museum by searching the following: "sterling silver Tiffany shovel" AND museum" and here we got the answer:

"sterling silver Tiffany shovel" AND museum



issuu

<https://issuu.com> › greengale-publishing › docs › gotha...



[Gotham - 2016 - Issue 6 - Winter - Chris Weidman & Stephen ...](#)

A 1900 **sterling silver Tiffany shovel**—just one of the many objects whose story comes to life in the first-ever exhibition on NYC's past, present, ...



The New York Times

<https://www.nytimes.com> › 1994/01/14 › arts › a-gallery...



[A Gallery of Rogues and Reformers](#)

Jan 14, 1994 — ... appropriately, by the **Museum** of the City of New York. ... Rapid Transit subway (the **sterling silver Tiffany shovel** is on display).

So the flag is: uiuctf{museum_of_the_city_of_new_york}

What's for Dinner?

Challenge

287 Solves



What's for Dinner?

50

osint food

Jonah Explorer, world renowned, recently landed in the city of Chicago, so you fly there to try and catch him. He was spotted at a joyful Italian restaurant in West Loop. You miss him narrowly but find out that he uses a well known social network and loves documenting his travels and reviewing his food. Find his online profile.

► View Hint

Flag

Submit

This challenge was the easiest one. You can see in the first line the name “Jonah Explorer” - it is a social media challenge, and as the most used social media in CTFs is Twitter we went to twitter searching for “Jonah Explorer”.

In the first result we find the account and the flag:

[←](#) **Tweet**

 **Jonah Explorer**
@jonahexplorer [...](#)

uiuctf{i_like_spaghetti}

3:46 PM · Jun 16, 2023 · 1,227 Views

2 Retweets **16** Likes

Flag: **uiuctf{i_like_spaghetti}**

Finding Jonah?

[Challenge](#) [244 Solves](#) [X](#)

Finding Jonah?

50

[osint](#) [zip-code](#)

Jonah offered a reward to whoever can find out what hotel he is staying in. Based on the past information (chals), can you find out what the hotel he stayed at was?
Flag should be uiuctf{hotel_name_inn}

► View Hint

 [chicago.jpeg](#)

[Flag](#)

[Submit](#)

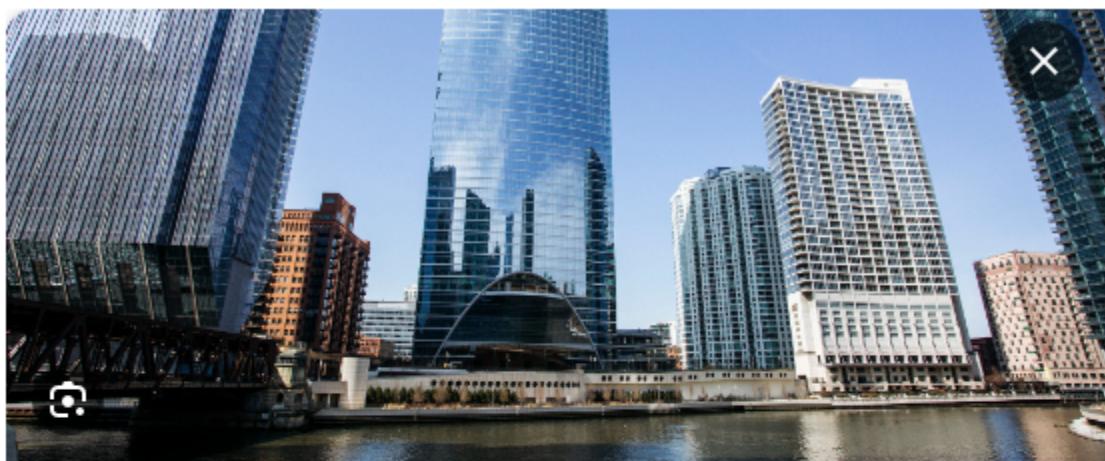
This challenge is a geolocation challenge which gave us this picture:



In this challenge the name of the picture says it is in Chicago, which made it easier. By analyzing the picture, we can see a rectangular building:



So I went to google and searched “rectangular building in chicago”. This search didn’t give me the picture of the building I was looking for, but it gave me another building in the picture:



If Rectangular Buildings Are So Terrible, Why Do Constructors Prefer Them? - Arch2O.com

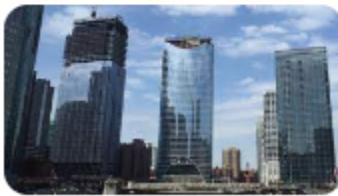
[Visit](#)

Images may be subject to copyright. [Learn More](#)

Related content



 [Chicago Architecture Ce...](#)
[Chicago River building ...](#)

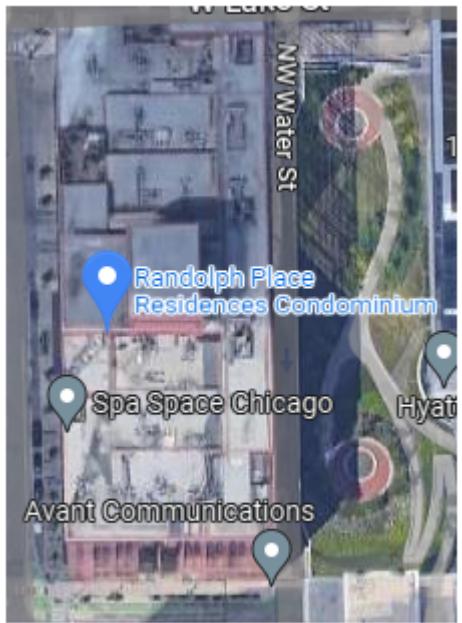


 [Chicago Architecture Ce...](#)
[150 North Riverside | B...](#)



 [Pexels](#)
[Person Showing Buildi...](#)

So now I know this building is called River Point in Chicago. For this challenge google earth pro would be the best choice, but for my connection it is slow, so I decided to go out with google maps. It was harder, but I got the same result. So this was the building i was searching for:



And by searching for hotels nearby, I was able to identify the hotel which is hampton inn:
So the flag: uiuctf{hampton_inn}

Jonah's journal

Challenge 240 Solves X

Jonah's Journal

50

osint worldwide

After dinner, Jonah took notes into an online notebook and pushed his changes there. His usernames have been relatively consistent but what country is he going to next? Flag should be in format uiuctf{country_name}

▶ View Hint

Flag

Submit

So for this challenge, I was able to find answers in 2 ways. The first way is a bit of guessing and the second way I believe is the right one. So let's start with the guessy way:

First i searched about his username "jonahexplorer" in duckduckgo which gave me this link:

<https://www.yelp.com/biz/gioia-ristorante-pastificio-chicago>

It is an italian restaurant in chicago:



Jonah E.
San Francisco, CA
0 1 1

...



6/16/2023

1 photo

I came here during a trip to chicago and it was absolutely amazing. I loved the food here and the chocolate cake was really good. I loved posting the food I had onto Twitter (@jonaheXplorer) where I talk about different restaurants I go to. I would give the food 10/10. Definitely worth coming to again!

The server was really nice as well and service was unmatched. he constantly made sure that we felt comfortable as well as getting us water and other little small things. Finally, the setting of the food and the ambiance made the whole night.



So after that, I just guessed that since it's an italian restaurant, the next destination is Italy, lol, and it was the right answer.

The intended solution was after I searched for his username in duckduckgo I also got his github account <https://github.com/JonahExplorer>. We can see on repository called adventurecodes:

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'main' (selected), 'branches' (2), and 'tags' (0). To the right are 'Go to file', 'Add file', and a green 'Code' button. Below this, a list of commits is shown:

- JonahExplorer Update README.md** (b10066d) 3 weeks ago 2 commits
- README.md** (Update README.md) 3 weeks ago

The 'README.md' file content is displayed in a large box:

```
adventurecodes
storing all my adventure codes
I think that the next place I wanna visit is the great wall of china, but not until I get off of flight UA5040
```

He says that he wants to visit China but that is not the answer. Then we can see in the picture there are 2 branches. Navigating to the second branch led to the following:

The screenshot shows the 'branches' page on GitHub. A search bar at the top has 'Search branches...' entered. Below it, the 'Default branch' section shows the 'main' branch. The 'Active branches' section shows the 'entry-2' branch.

Press compare, then the hash ee05e1f and here the answer:

The screenshot shows a GitHub commit page for the 'entry-2' branch. The commit hash is ee05e1f. It was made by 'JonahExplorer' 3 weeks ago and has 1 parent commit 3dacc60. The commit message is 'Update README.md'. The diff shows:

```
@@ -3,4 +3,4 @@ storing all my adventure codes
3 3
4 4 I think that the next place I wanna visit is the great wall of china, but not until I get off of flight UA5040
5 5
6 - I dont know how these things work but my next destination is not China but actually italy. After I check out from my hotel in the west loop, I'll be heading there.
6 + Sometimes i forgot how these "branches" work, kinda like a tree right?
```

So the flag: uiuctf{italy}

First class mail

Challenge

173 Solves



First class mail

50

osint zip-code

Jonah posted a picture online with random things on a table. Can you find out what zip code he is located in?
Flag format should be uiuctf{zipcode}, ex: uiuctf{12345}.

► View Hint

chal.jpg

Flag

Submit

For this challenge we also have a given picture which is this:



By analyzing the picture nothing seemed useful other than this:



It looked like a barcode to me, and the instruction of challenge says we are looking for a zip code so I searched in google “bar code for zip code” and this was the result:

bar code for zip code

X |

Images Shopping Maps Videos News Books Flights Finance Near oran

About 633,000,000 results (0.51 seconds) « Add Grepper Answer (a)

POSTNET (Postal Numeric Encoding Technique) is a barcode symbology used by the United States Postal Service to assist in directing mail. The ZIP Code or ZIP+4 code is encoded in half- and full-height bars. Most often, the delivery point is added, usually being the last two digits of the address or PO box number.

Wikipedia
https://en.wikipedia.org › wiki › POSTNET

[POSTNET - Wikipedia](#)

About featured snippets • Feedback

So using a POSTNET online decoder

<https://www.dynamsoft.com/barcode-reader/barcode-types/postnet/> gave us this result:

6066111234

By reading a bit of the wikipedia page we conclude that the sum of all digits should be divisible by 10, so $6+0+6+6+1+1+1+2+3+4=30$ means it is written in this format: 60661-1123 which lets us know that the zip code is 60661.

The flag: uiuctf{60661}

CRYPTO

Three-Time Pad

You are given 3 cipher texts and 1 plain text (for the second cipher text), so you have to XOR c2 with p2 to get the key. After getting the key you can use the key to XOR it again with cipher message 1 and cipher message 3. After this, you will get the flag.

I used this script:

```
def xor_decrypt(ciphertext, key):
    return bytes([a ^ b for a, b in zip(ciphertext, key)])Hunter Exploit-DB

def main():
    # Read ciphertexts and known plaintext from files
    with open('c1', 'rb') as file:
        C1 = file.read()

    with open('c2', 'rb') as file:
        C2 = file.read()

    with open('c3', 'rb') as file:
        C3 = file.read()

    with open('p2', 'rb') as file:
        P2 = file.read()

    # Step 1: Obtain the key (K2)
    K2 = xor_decrypt(C2, P2)

    # Step 2: Decrypt the other messages
    P1 = xor_decrypt(C1, K2)
    P3 = xor_decrypt(C3, K2)

    # Print the recovered plaintexts
    print("Plaintext of Message 1 (P1):", P1)
    print("Plaintext of Message 3 (P3):", P3)

if __name__ == '__main__':
    main()
```

```
[root@qaisqupti]# python ez
Plaintext of Message 1 (P1): b'before computers, one-time pads were sometimes'
Plaintext of Message 3 (P3): b'uinctf{burn_3ach_k3y_aft3r_us1ng_1t}'
```

Flag: uiuctf{burn_3ach_k3y_aft3r_us1ng_1t}

At Home

Writeup by voxelbugged

I started by downloading the files. I noticed that chal.txt contained three large integer values - e, n, c. It had nothing else, so without the python script I could not do anything with this file just yet. Of course, I opened the script next. It detailed the exact steps of how the values in chal.txt were calculated, as expected.

However, the most interesting part was that the flag was converted into an int and then used in this equation:

$$c = (\text{flag} * e) \% n$$

This is a modular equation, and the value of flag could be calculated with a simple python script to solve for x as I already knew the values of the other numbers:

```
import math

def modular_inverse(e, n):
    g, x, _ = extended_gcd(e, n)
    if g == 1:
        return x % n

def extended_gcd(a, b):
    if a == 0:
        return b, 0, 1
    else:
        g, y, x = extended_gcd(b % a, a)
        return g, x - (b // a) * y, y

def solve_for_x(e, n, c):
    d = modular_inverse(e, n)
    if d is not None:
        x = (c * d) % n
        return x

e =
359050381528215534161395815035053470579525085604518644266341003311656042231363926028398149682492008978949781520105189684311823250795520058111763310428202654439351923617227315577436407
n =
26866112476805080406600829998667337295621683371086008990123843295238481171468440400188535405203911234020955722625665066118684372692958125334974412114712446241957729405174414181741151229
c =
677433744624485821074401685136875204345945293318217407373961164679281110438150846650621041967540205304693605392533237389357084143630053734587820419554502789543483064015423743097889372065

x = solve_for_x(e, n, c)
byte_x = int.to_bytes(x, 69, "big")
print(byte_x)
```

Once I got the int value, I converted it back to bytes and received the flag:

Flag: uiuctf{W3_hav3_R5A_@_h0m3}

Group Project

Writeup by voxelbugged

I started by simply connecting to the challenge. I saw a message containing three numbers: “g”, “p”, “A”, along with a prompt asking me to “Choose k”. The first thing I did was type in a letter to test what would happen. I got met with a message saying “I said a number...” and the program hung up.

```
voxelbugged@voxelbugged-Lenovo-Legion-Y530-15ICH:~$ nc group.chal.uiuc.tf 1337
== proof-of-work: disabled ==
[$] Did no one ever tell you to mind your own business??
[$] Public:
[$]   g = 2
[$]   p = 11606670944211456728949633835620048147060861407370337823000056905498648643446593536535805710773465574405633628693755223594961872798367172206444941446960573252996930431432451
[$]   A = 10109135570995186584136967550659916282253760222582310956281976992033264793866962872950762502342422697417414758703914868659494299499462266454845123067801399728640167347356
[$]   Choose k = a
[$] I said a number...
```

I connected to the challenge again, this time putting in k as 0. A few interesting things happened:

```
voxelbugged@voxelbugged-Lenovo-Legion-Y530-15ICH:~$ nc group.chal.uiuc.tf 1337
== proof-of-work: disabled ==
[$] Did no one ever tell you to mind your own business??
[$] Public:
[$]   g = 2
[$]   p = 11606670944211456728949633835620048147060861407370337823000056905498648643446593536535805710773465574405633628693755223594961872798367172206444941446960573252996930431432451
[$]   A = 10109135570995186584136967550659916282253760222582310956281976992033264793866962872950762502342422697417414758703914868659494299499462266454845123067801399728640167347356
[$]   Choose k = 0
[$] I said a number...
[$] Ciphertext using shared 'secret' ;;
[$]   c = 31383420538805400549021388799532797474095834602121474716358265812491198185235485912863164473747446452579209175051706
```

The first thing is that the “p” and “A” numbers have changed, suggesting random number generation is involved. The second and more important part is that the challenge actually let it through, giving me both the error message AND the actual cipher!

At this point I downloaded the python script for the challenge to check how it works, and this is when I noticed the exploitable part:

```
Ak = pow(A, k, p)
b = randint(2, p - 1)
B = pow(g, b, p)
Bk = pow(B, k, p)
S = pow(Bk, a, p)

key = hashlib.md5(long_to_bytes(S)).digest()
cipher = AES.new(key, AES.MODE_ECB)
c = int.from_bytes(cipher.encrypt(pad(flag, 16)), "big")|
```

It turns out that setting “k” to 0 forces “S” to always equal 1, which means the resulting cipher will always equal the same value, no matter what numbers are generated for “p” and “A”!

After learning this, I wrote a quick python script to reverse this process and get the flag based on the “S” and “c” values:

```
import hashlib
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
from Crypto.Util.number import bytes_to_long

def reverse_process(S, c):
    key = hashlib.md5(S.to_bytes((S.bit_length() + 7) // 8, "big")).digest()
    cipher = AES.new(key, AES.MODE_ECB)
    decrypted_data = unpad(cipher.decrypt(c.to_bytes((c.bit_length() + 7) // 8, "big")), 16)
    flag = decrypted_data.decode()
    return flag

S = 1
c = 31383420538805400549021388790532797474095834602121474716358265812491198185235485912863164473747446452579209175051706

flag = reverse_process(S, c)
print("Original flag:", flag)
```

After running this, the original flag is outputted:

```
voxelbugged@voxelbugged-Lenovo-Legion-Y530-15ICH:~/Downloads$ python reverser.py
Original flag: uiuctf{brut3f0rc3_a1n't_s0_b4d_aft3r_all!!11!!}
```

Flag: uiuctf{brut3f0rc3_a1n't_s0_b4d_aft3r_all!!11!!}

PWN

REV

WEB