Bsides Vancouver 2025
Cybersecurity Skills Crash Course

Class:
Special Topics

Workshop Number:
AS-SPC-1337_25

Document Version:
1.00

Special Requirements:
Registered account at picoctf.org

Welcome to the Workshop!

In this interactive workshop, we'll be learning some fundamental skills used in cybersecurity through task-based exercises

The workshop will be divided into the following sections:

Workshop Topics and Schedule

10:30 – 11:15	Pt1 Linux OS Operations
11:15 – 11:45	Pt2 Python Programming
11:45 – 12:15	Pt3 Cryptography
12:15 – 1:00	Lunch Break
1:00 – 1:30	Pt4 Web App Hacking
1:30 – 2:00	Pt5 Digital Forensics
2:00 – 2:45	Pt6 Binary Hacking
2:45 – 3:00	Wrap-Up + Q & A

Information Overload

We'll be solving around 2 dozen exercises across 6 categories today, as well as learning many, many, new skills and concepts

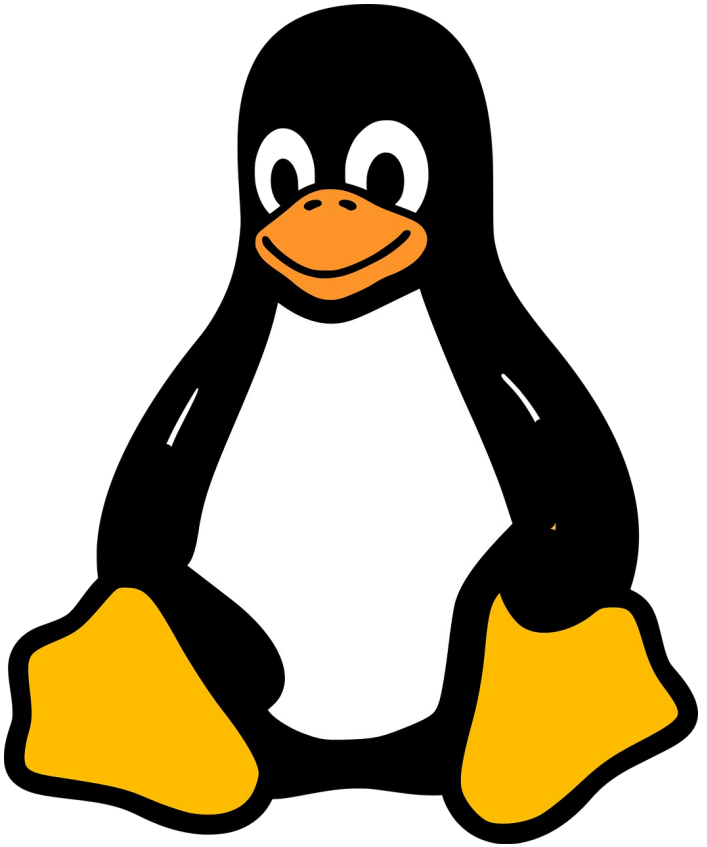
Information Overload

The goal is to introduce as many cybersecurity-related topics as possible, so you all can discover which topic(s) you want to explore further in the future

Information Overload

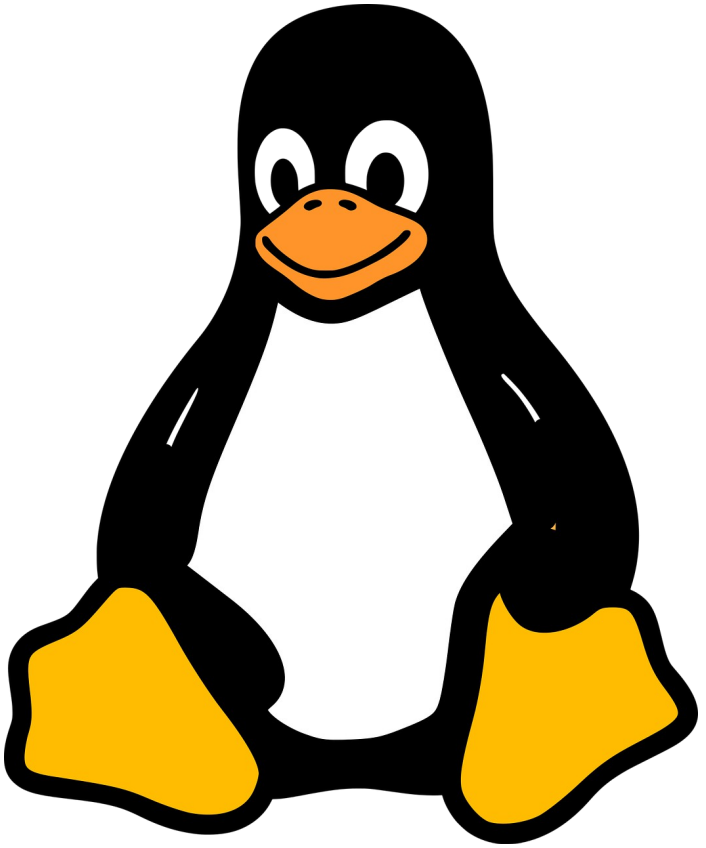
During the workshop, if you don't quite understand an exercise, and get stuck, don't worry! We'll on to a new topic in a few minutes!

Part 1 – Linux OS Operations



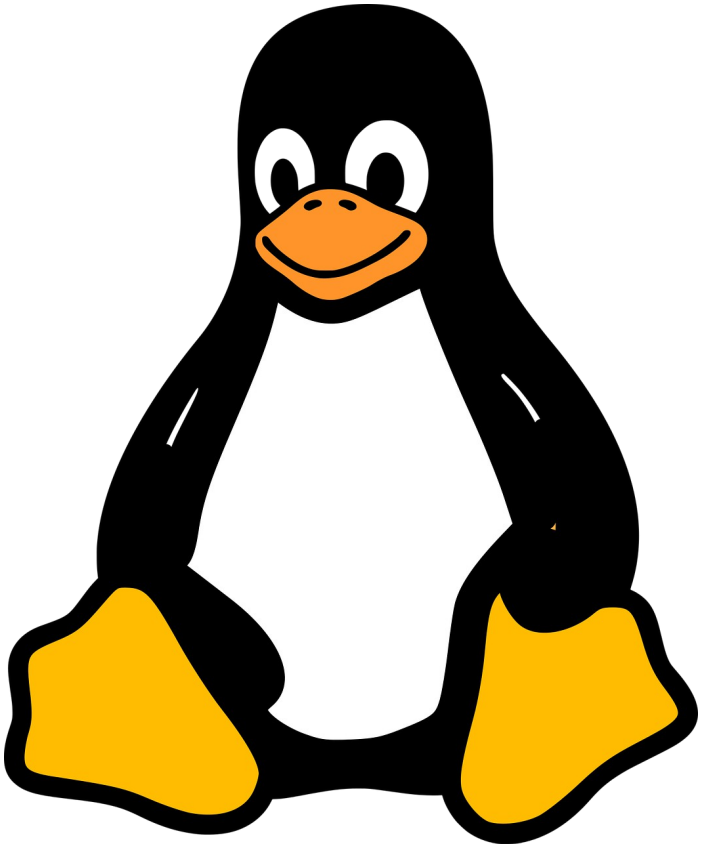
What is Linux?

Like Windows and MacOS, Linux is a computer operating system (OS). However, unlike Windows and MacOS, Linux is free, and open-source, which means its source code can be viewed (and potentially modified) by anybody.



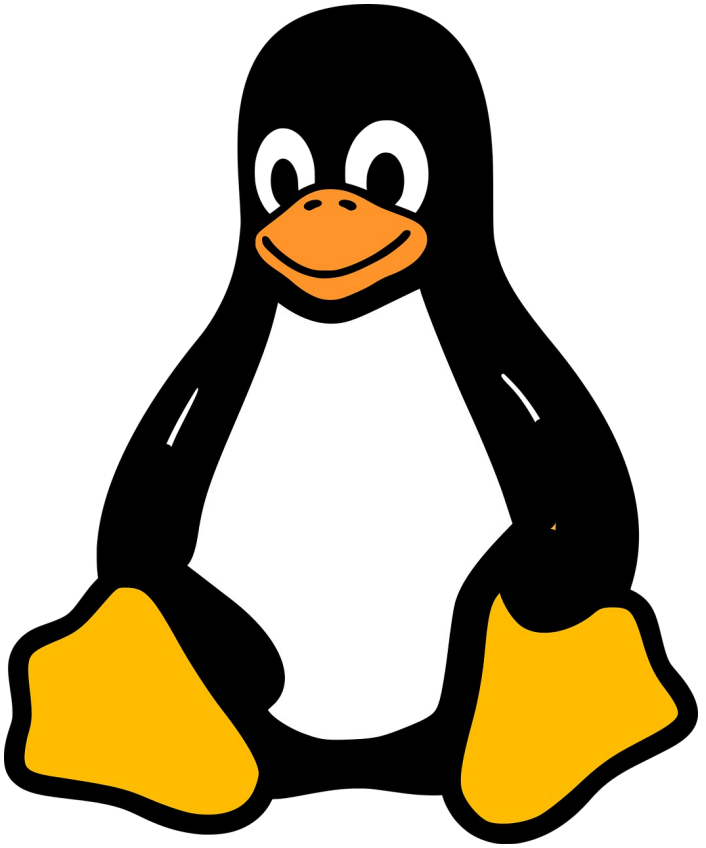
What is Linux?

Linux is a modular OS, which means that software components can be very easily added, updated or removed to fit the user's needs.



What is Linux?

This has led to the development of different distributions (variations) of Linux that cater to different groups of users.



What is Linux?

Specifically, Cybersecurity-oriented distributions of Linux exist, which are ideal for cybersecurity student use.

Why Learn Linux for Cybersecurity?

- Linux is easily accessible to students, since it's free
- Many cybersecurity tools / software are available to Linux users, most of which are free
- Cybersecurity-oriented distributions of Linux are readily available

What are CTFs?

My classes prefer to incorporate CTF games into classes for a more interactive experience, but what are CTFs?

What are CTFs?

Cybersecurity Capture The Flag (CTF) games are training exercises where the goal of the exercise is to “capture the flag” through use of cybersecurity skills.

What are CTFs?

In this context, “capture” means to gain access to a file or other resource, and “flag” refers to a secret phrase or password.

PicoCTF

The CTF game we will be playing to learn basic Linux skills is called PicoCTF. It's a well-known education platform with challenges across many different cybersecurity disciplines.

PicoCTF

In order to participate in picoCTF challenges, we will need a registered account on the website. We can signup at the following link:

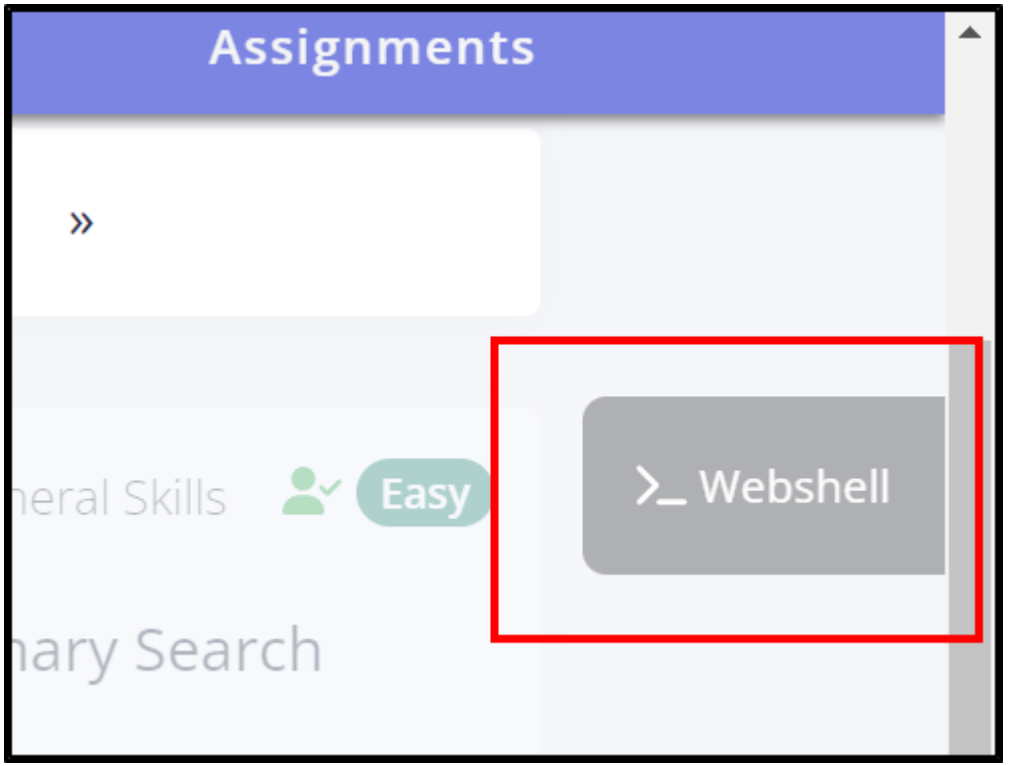
https://play.picoctf.org/register

PicoCTF

Once we have a registered account, login to the PicoCTF website and we'll be ready to begin!

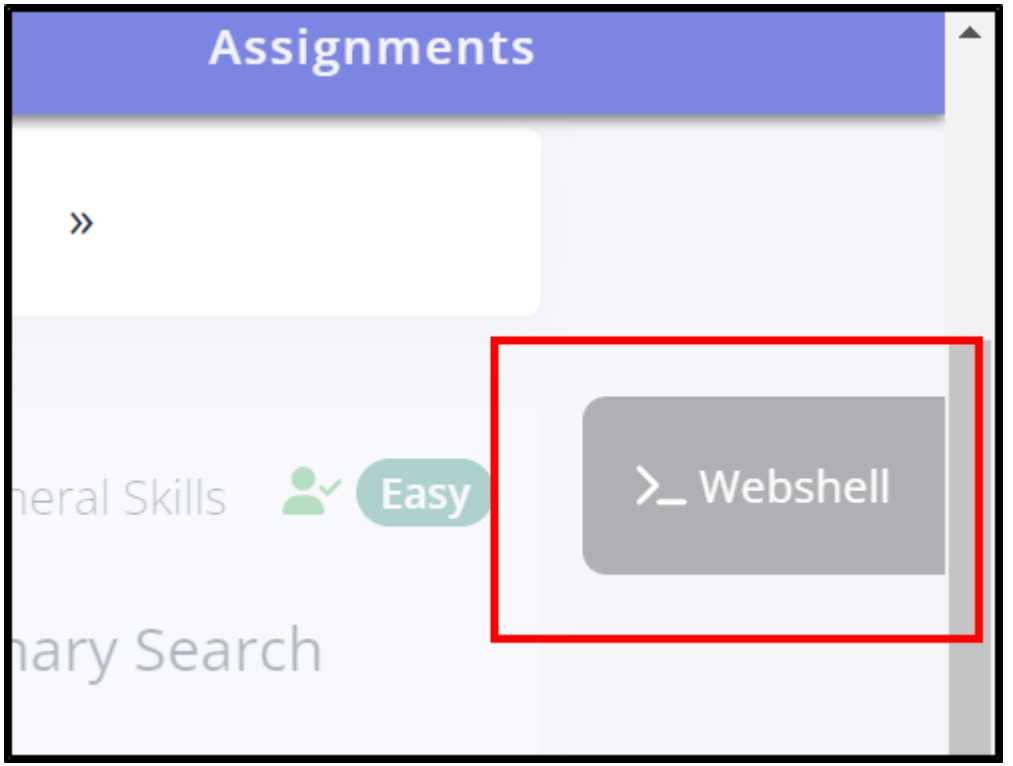
Accessing a Terminal

The first thing we need to do is open our command-line interface (CLI) terminal. After logging into PicoCTF, we can access the web-shell.

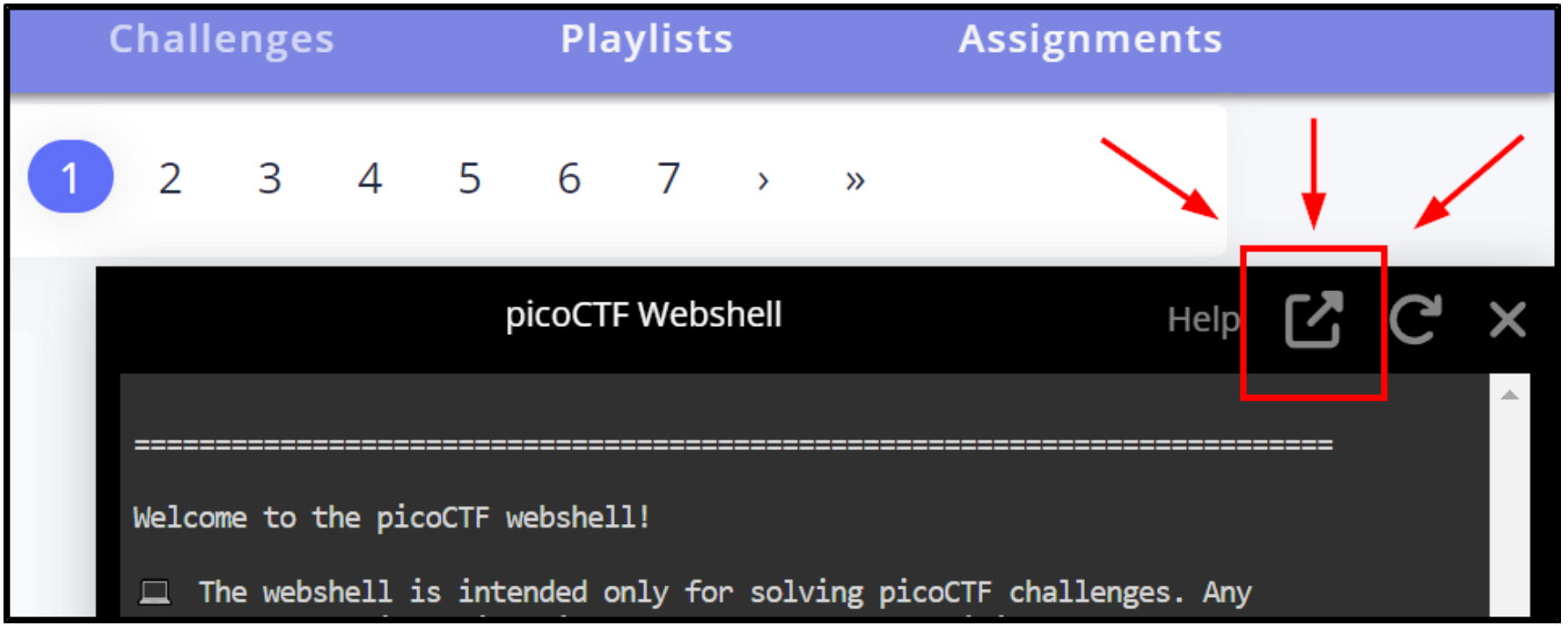


Accessing a Terminal

The button is found on the Practice webpage, and it's located at the upper-right corner of the webpage. If we can't see it, we should scroll down until it appears



Accessing a Terminal



After opening the webshell, we should click on the **Popout Webshell Terminal** button, outlined in the image above

Next Challenge: Obedient Cat

Our next challenge will help us understand downloading and reading files in the Linux terminal. Access the Obedient Cat challenge here:

https://play.picoctf.org/practice/challenge/147?category=5&page=1&search=cat

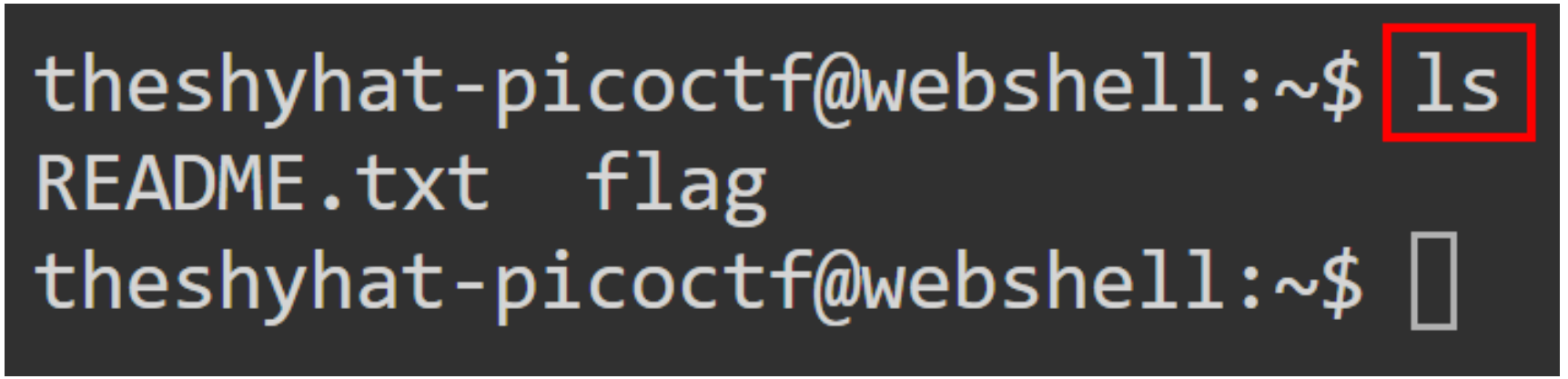
Wget – Downloading files

```
theshyhat-picoctf@webshell:~$ wget https://mercury.picoctf.net/static/217686fc11d733b80be6
2dcfcfca6c75/flag
--2024-09-30 19:42:38--  https://mercury.picoctf.net/static/217686fc11d733b80be62dcfcfca6c
75/flag
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34 [application/octet-stream]
Saving to: 'flag'

flag                100%[=========================>]      34  --.-KB/s    in 0s

2024-09-30 19:42:38 (22.6 MB/s) - 'flag' saved [34/34]
```

We use the Linux wget command to download files from remote servers. Use it like this:

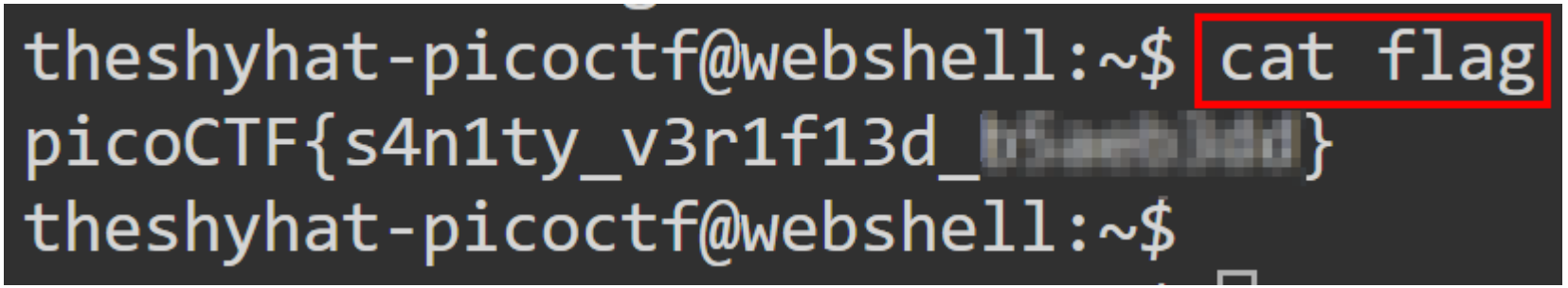
`wget <file_address>`

Ls – Listing Directory Contents



The Ls command will list out the contents of the current directory. Just type `ls`

Cat – Reading File Contents



To finish the challenge, we need to read the **flag** file using the **cat** command. Type **cat flag**, then press the enter key, and the challenge's flag string will appear

Obedient Cat

```
theshyhat-picoctf@webshell:~$ wget https://mercury.picoctf.net/static/217686fc11d733b80be6
2dcfcfca6c75/flag
--2024-09-30 19:42:38--  https://mercury.picoctf.net/static/217686fc11d733b80be62dcfcfca6c
75/flag
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34 [application/octet-stream]
Saving to: 'flag'

flag                100%[=========================>]      34  --.-KB/s    in 0s

2024-09-30 19:42:38 (22.6 MB/s) - 'flag' saved [34/34]
```

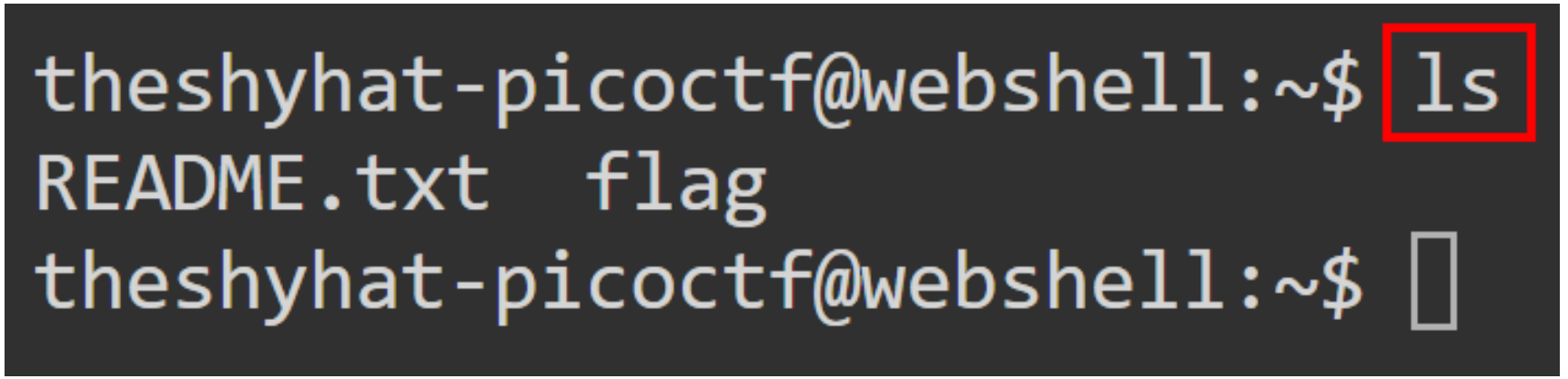
Then back at the webshell, type **wget**, then space, then paste in the address we copied from the challenge page.

Obedient Cat

```
theshyhat-picoctf@webshell:~$ wget https://mercury.picoctf.net/static/217686fc11d733b80be6
2dcfcfca6c75/flag
--2024-09-30 19:42:38--  https://mercury.picoctf.net/static/217686fc11d733b80be62dcfcfca6c
75/flag
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34 [application/octet-stream]
Saving to: 'flag'

flag                100%[=========================>]      34  --.-KB/s    in 0s

2024-09-30 19:42:38 (22.6 MB/s) - 'flag' saved [34/34]
```

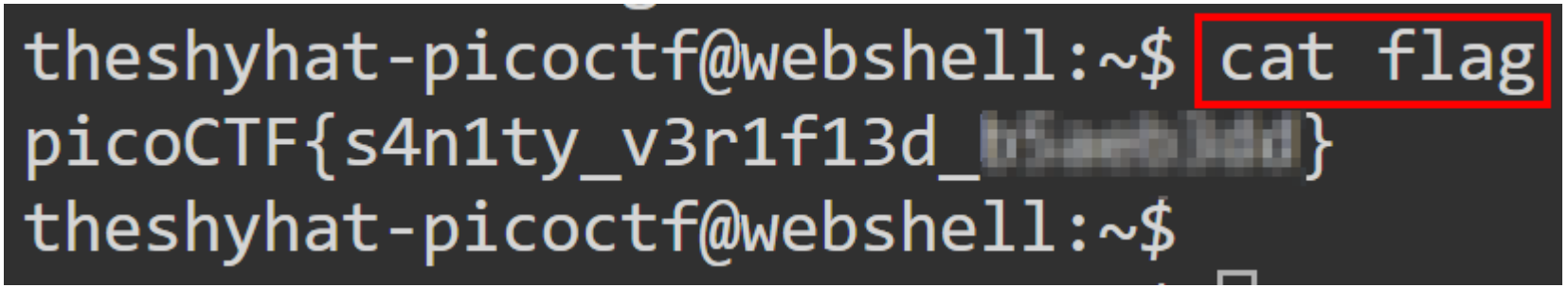
Press the Enter key, then the challenge file will be downloaded to our webshell terminal.

Obedient Cat



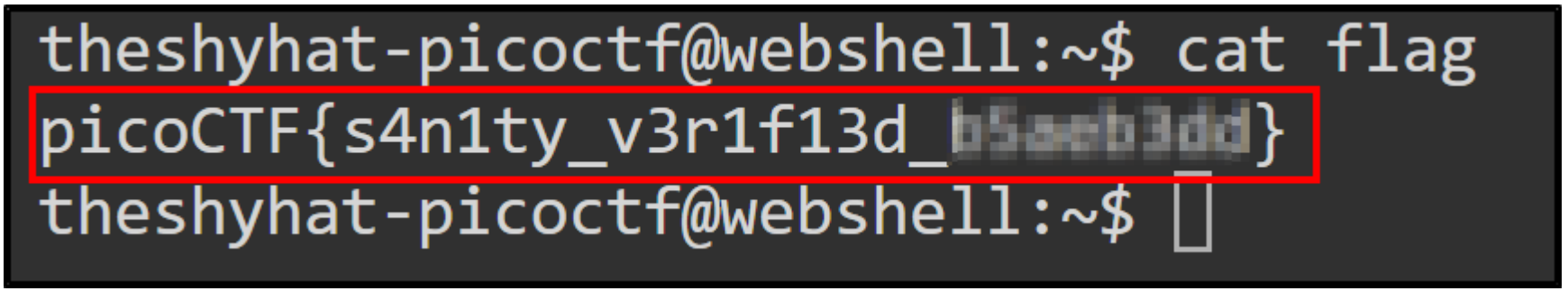
We can use the **ls** command to list all of the files in our current directory.

Obedient Cat



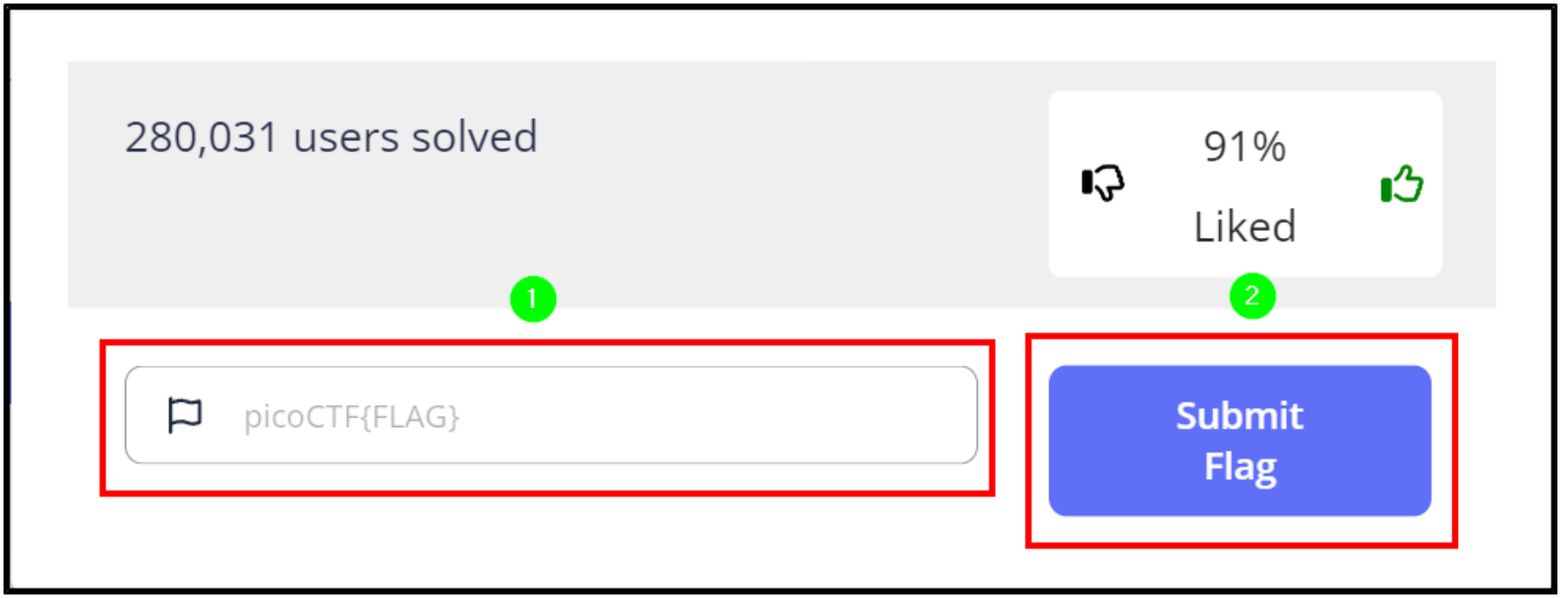
To finish the challenge, we need to read the **flag** file using the **cat** command. Type **cat flag**, then press the enter key, and the challenge's flag string will appear

Obedient Cat



To finish the challenge, we copy the flag output...

Obedient Cat



Then go back to the challenge page, and paste the flag value into the field submission field, then click on the blue **Submit flag** button

Challenge: Tab Tab Attack

The next challenge we'll cover this session is called **Tab, Tab, Attack**, which can be accessed from the following URL:

https://play.picoctf.org/practice/challenge/176?category=5&page=1&search=tab

Unzip – Extracting from archive files

```
theshyhat-picoctf@webshell:~$ unzip Addadshashanammu.zip
Archive:  Addadshashanammu.zip
   creating: Addadshashanammu/
   creating: Addadshashanammu/Almurbalarammi/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/Assurnabitashpi/
```

When we need to extract files from a .zip file, all we need to do is use the unzip command:

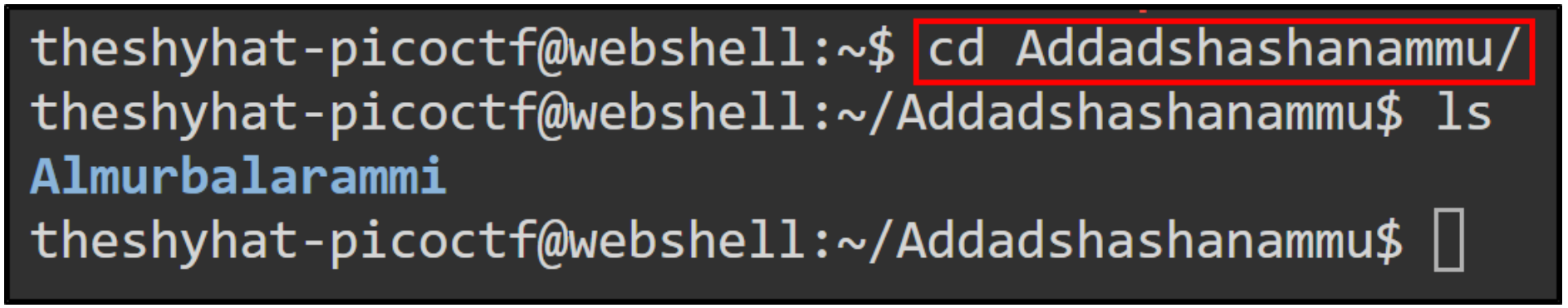
`unzip <file_name>`

Cd – Change Directory

```
theshyhat-picoctf@webshell:~$ cd Addadshashanammu/
theshyhat-picoctf@webshell:~/Addadshashanammu$ ls
Almurbalarammi
theshyhat-picoctf@webshell:~/Addadshashanammu$
```

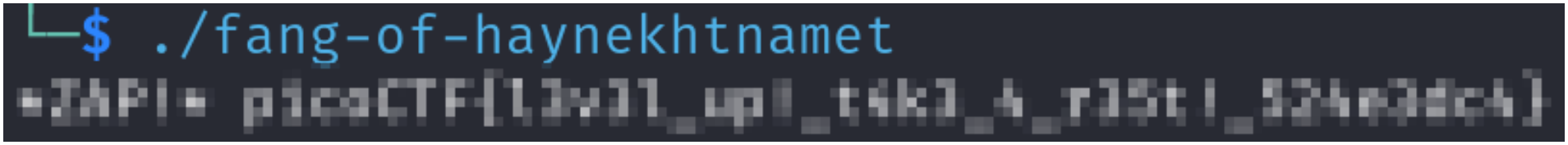
To enter the directory, type **cd** (change directory), then space, then the first two letters of the directory, **Ad**, then press the **Tab** key, and press enter.

Tab Autocomplete – Reduce Number of Required Keystrokes



We can use the Tab key on the keyboard to autocomplete the names of files or directories. Just start to type the file / directory name then hit the Tab key

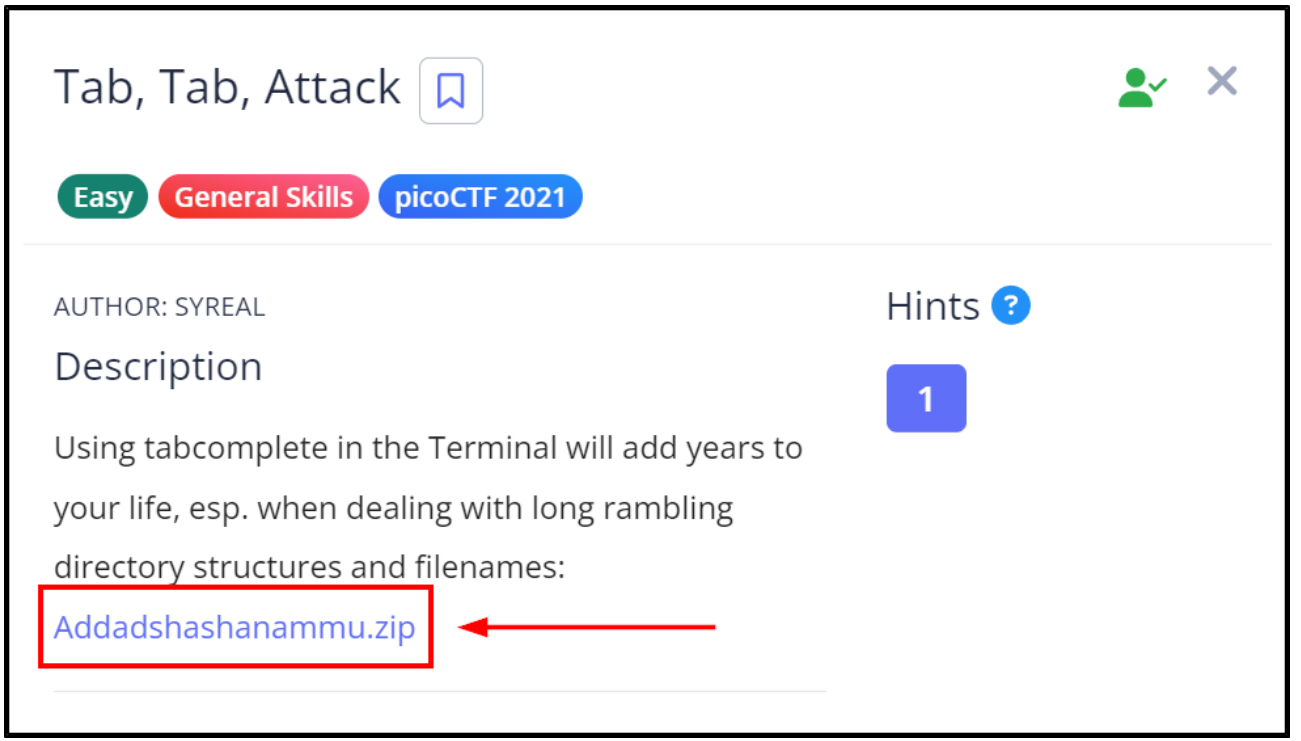
./<file_name> – Executing a file



./ in Linux refers to the current directory. To execute a file in the current directory we use the following command

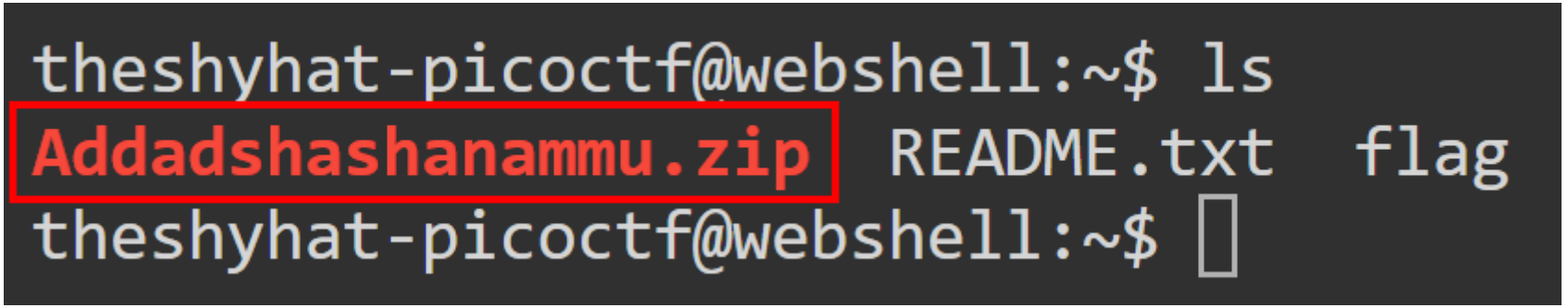
`./<file_name>`

Tab Tab Attack



Once more, we have to copy the link to the challenge file, then download it to our webshell using the **wget** command

Tab Tab Attack



After the file is downloaded, we can use **ls** to make sure it's downloaded to our directory.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ unzip Addadshashanammu.zip
Archive:  Addadshashanammu.zip
   creating: Addadshashanammu/
   creating: Addadshashanammu/Almurbalarammi/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/Assurnabitashpi/
```

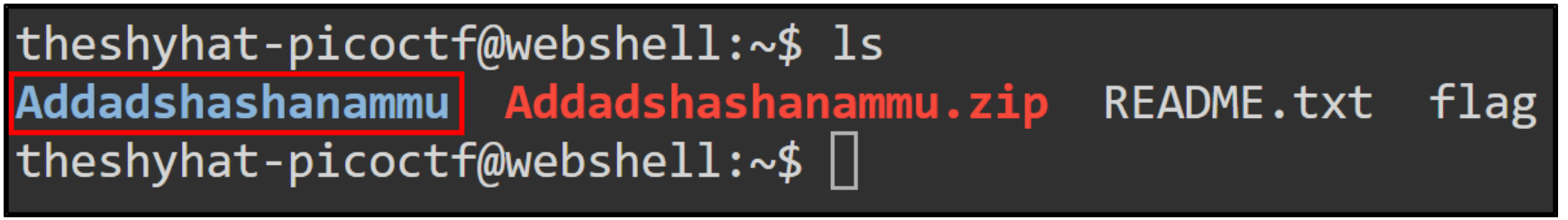
Since this file is a zip file, we need to extract the files from it using the unzip command. Type **unzip**, then space, then the letters **Ad**

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ unzip Addadshashanammu.zip
Archive:  Addadshashanammu.zip
   creating: Addadshashanammu/
   creating: Addadshashanammu/Almurbalarammi/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/Assurnabitashpi/
```

Then press the **Tab** key on the keyboard. This will auto-complete the name of the file. Tab auto-complete is a very useful method of inputting the names of files with long names.

Tab Tab Attack



If we use the **ls** command to see the contents of the directory, we see that there is a directory in here that named **Addadshashanammu**.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ ls
Addadshashanammu  Addadshashanammu.zip  README.txt  flag
theshyhat-picoctf@webshell:~$
```

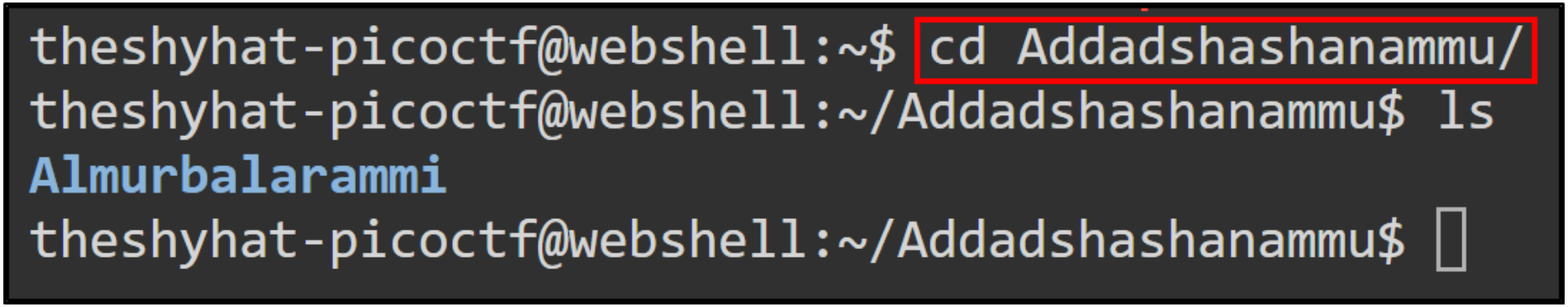
The point of this challenge is to use the tab auto-complete to enter directories with long, complicated names to obtain the flag.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ cd Addadshashanammu/
theshyhat-picoctf@webshell:~/Addadshashanammu$ ls
Almurbalarammi
theshyhat-picoctf@webshell:~/Addadshashanammu$
```

To enter the directory, type **cd** (change directory), then space, then the first two letters of the directory, **Ad**, then press the **Tab** key, and press enter.

Tab Tab Attack



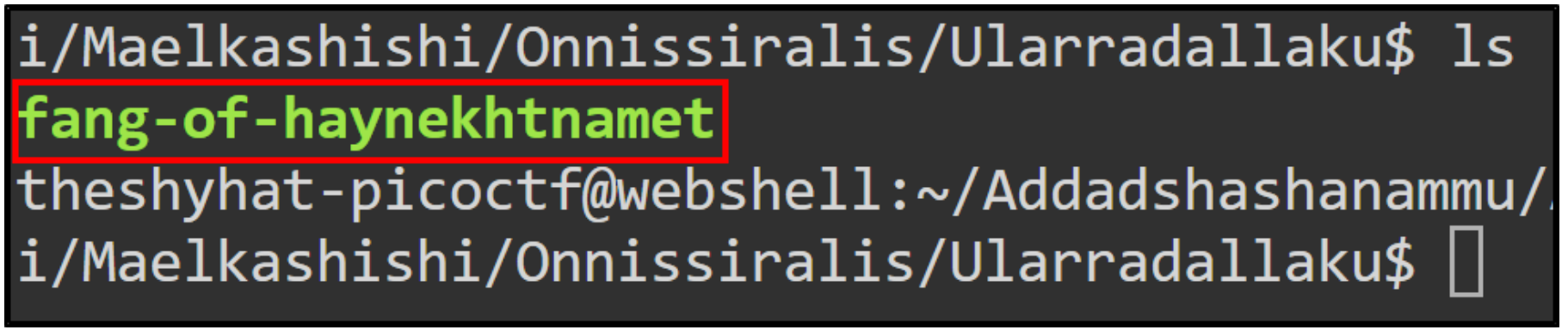
If we use the **ls** command, we see that there is another directory in here named **Almurbalarammi**.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~/Addadshashanammu$ cd Almurbalarammi/
theshyhat-picoctf@webshell:~/Addadshashanammu/Almurbalarammi$ ls
Ashalmimilkala
theshyhat-picoctf@webshell:~/Addadshashanammu/Almurbalarammi$
```

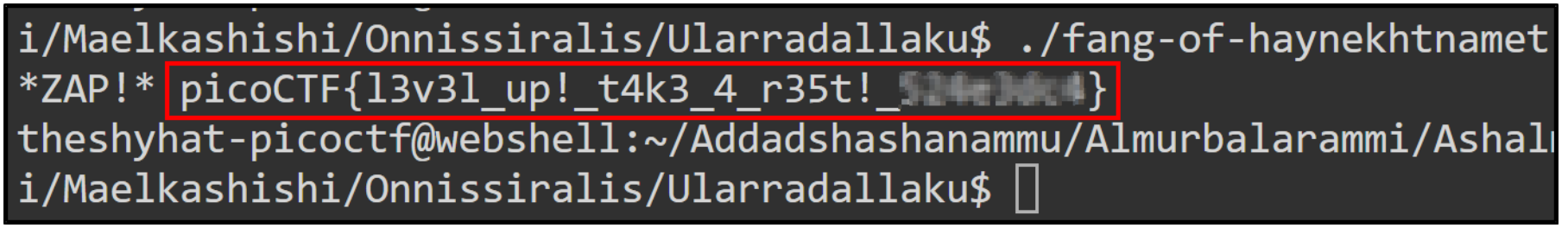
So we need to use the **cd** command and tab auto-complete to enter this new directory.

Tab Tab Attack



After doing this process 5 more times, we find this file in the directory, named **fang-of-haynekhtnamet**.

Tab Tab Attack



To run an executable file in our directory, we have to type **./** then type the first couple of letters from the file name, **fa**, then tab auto-complete.

Tab Tab Attack

```
i/Maelkashishi/Onnissiralis/Ularradallaku$ ./fang-of-haynekhtnamet
*ZAP!* picoCTF{l3v3l_up!_t4k3_4_r35t!_[illegible]}
theshyhat-picoctf@webshell:~/Addadshashanammu/Almurbalarammi/Ashali
i/Maelkashishi/Onnissiralis/Ularradallaku$
```

To complete the challenge, we repeat the process we did for the previous challenges, copying the flag, then pasting it into the flag submission field.

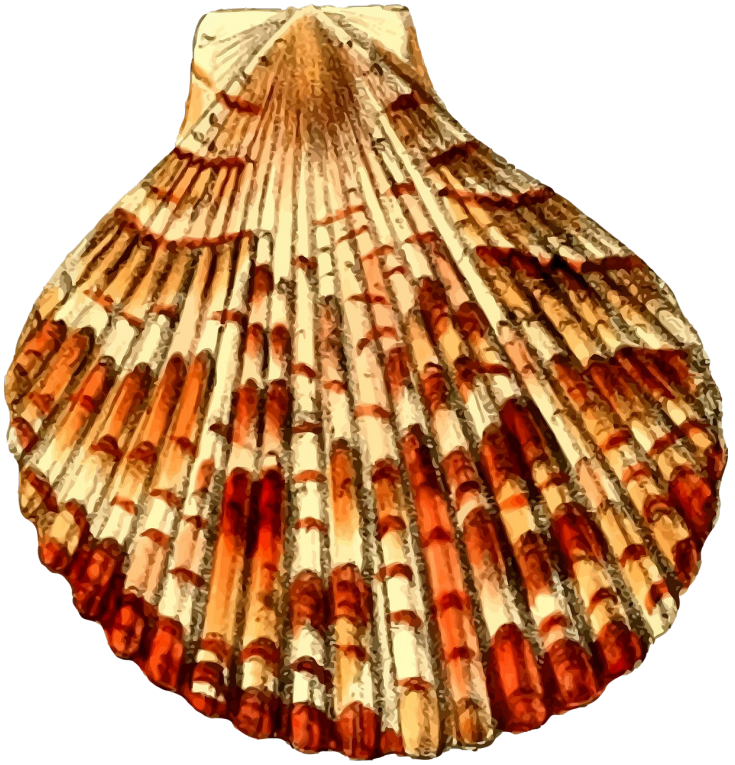
Magikarp Ground Mission Challenge

Let's wrap up our look at Linux by solving a challenge that will test our filesystem navigation skills (ls and cd commands):

https://play.picoctf.org/practice/challenge/189?category=5&page=1&search=magi

Connecting to a Server Using SSH

Let's first learn about how to connect to remote servers using SSH.



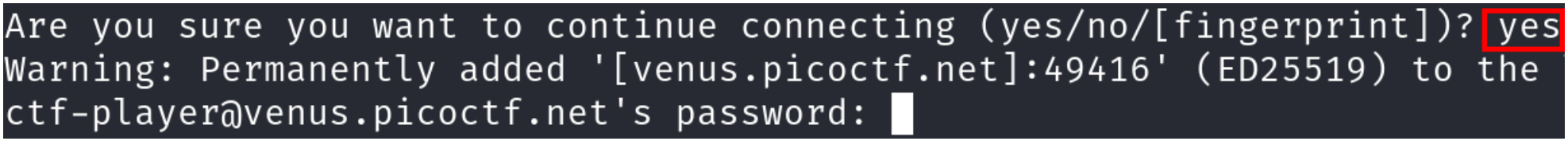
Connecting to a Server Using SSH

```
ssh ctf-player@venus.picoctf.net -p 49416
```

We can use the SSH command to login to a remote server (another computer) using a specific user account. The syntax is as follows:

`ssh <user_name>@<server_name> -p <port>`

Connecting to a Server Using SSH



When we're asked if we want to continue connected, type `yes`, and hit enter, then paste in the password

Connecting to a Server Using SSH

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[venus.picoctf.net]:49416' (ED25519) to the
ctf-player@venus.picoctf.net's password:
```

While inputting passwords in Linux, we will not see any output on the screen. This is normal. Paste or type in the password and hit enter to login

The Home Directory

Each user on a Linux system has a space where they can store their personal files. It's located at:

`/home/<username>`

For example:

`/home/theshyhat`

The Top-Level Directory

The top level directory of a Linux system is the **/** directory, where we can find all the other directories and files in the system. It can also be called the root directory, which can be confusing, since there's another directory called the root directory...

Part 2 – Python Programming

What is Programming?

According to Wikipedia, computer programming is the process of performing a particular computation (or more generally, accomplishing a specific computing result), usually by designing and building an executable computer program.

What is Programming?

Put in broader terms, programming is the process of writing instructions for a computer to execute.

What is Programming?

The instructions can result in output as simple as printing the answer to a simple math equation or a word to the screen, or as complex as the latest video editing software, games, websites, or countless other pieces of software used in our daily lives.

What is Programming?

Many different programming languages are used for writing software, and each programming language has its own advantages and disadvantages for developing different types of software.

What is Programming?

The programming language we will learn to use in these workshops is called Python.

Why Learn Python?

There are a lot of programming languages out there, so why are we learning Python?

We'll go over three reasons:

Why Learn Python?

1) Python is a very popular language with an active online community, so there's a lot of support for students.

Why Learn Python?

2) Python uses a syntax that is much closer to written English than other programming languages, so it's easier to read and understand

```
>>> print("Hello World!")
Hello World!
>>>
```

Why Learn Python?

3) Python automates some tedious operations common to other programming languages, such as memory management.

Hello, World!

Programmers learning a language usually write a “Hello, World!” program as their first exercise.

Hello, World!

To complete our **Hello, World!** exercise, we'll go over the Python print function first.

The Print Function

```
>>> print("This is how you print text in Python!")
This is how you print text in Python!
```

Virtually all programming languages have some method of printing text to the screen (often referred to as the console). In Python, the **print** function is the primary method of doing so.

The Print Function

```
>>> print("This is how you print text in Python!")
This is how you print text in Python!
```

To use the **print** function, we use the function name, **print**, then a pair of parentheses, then encase what we want printed, in either single or double quotes, between the parentheses.

The Print Function

```
>>> variable = "A variable could be a lot of different things!"
>>> print(variable)
A variable could be a lot of different things!
```

Another common use of the print function is to print the values of variables to the console. In this case, we simply input the name of the variable we want printed between the parentheses.

The Print Function

```
print "Hello, world!"
```

```
  File "main.py", line 2
    print "Hello, world!"
          ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Hello, world!")?
```

The syntax for the Print function requires parentheses to work properly. If omitted, Python will display an error message when run.

The Print Function

```
print('Printed using single quotes.')
print("Printed using double quotes.")
```

```
Printed using single quotes.
Printed using double quotes.
```

Also, if we print text, we have to use either use single quotes or double to wrap what we want to print.

Variables

Virtually all programming languages use variables, which are storage locations associated with a specific name, as well as a value. Take the following as an example:

```
my_name = 'shyhat'
```

Variables

Wherever we refer to **my_name** in the code, **shyhat** will be inserted there. So, if we have the following code:

```
my_name = 'shyhat'
print(my_name)
```

The output should be **shyhat**.

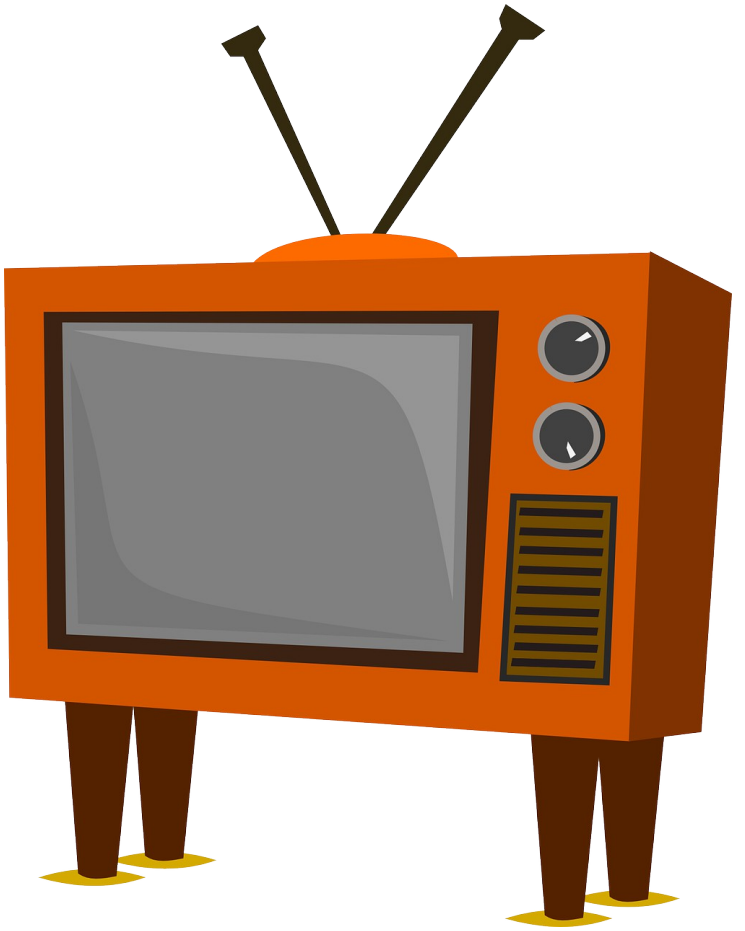
Variables

The real benefit of using variables in programming is that a variables can be referenced multiple times in the same code, saving a lot of time and accounting while writing code.

In addition, variables can change value after they've been defined.

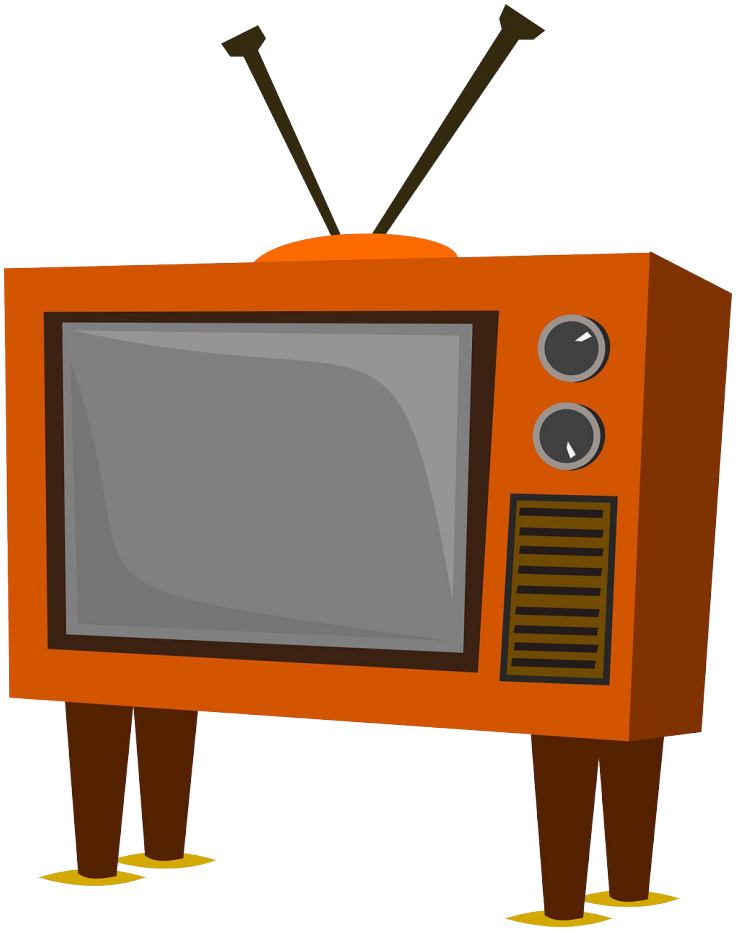
Variables

For example, say we have an online store program where we sell televisions.



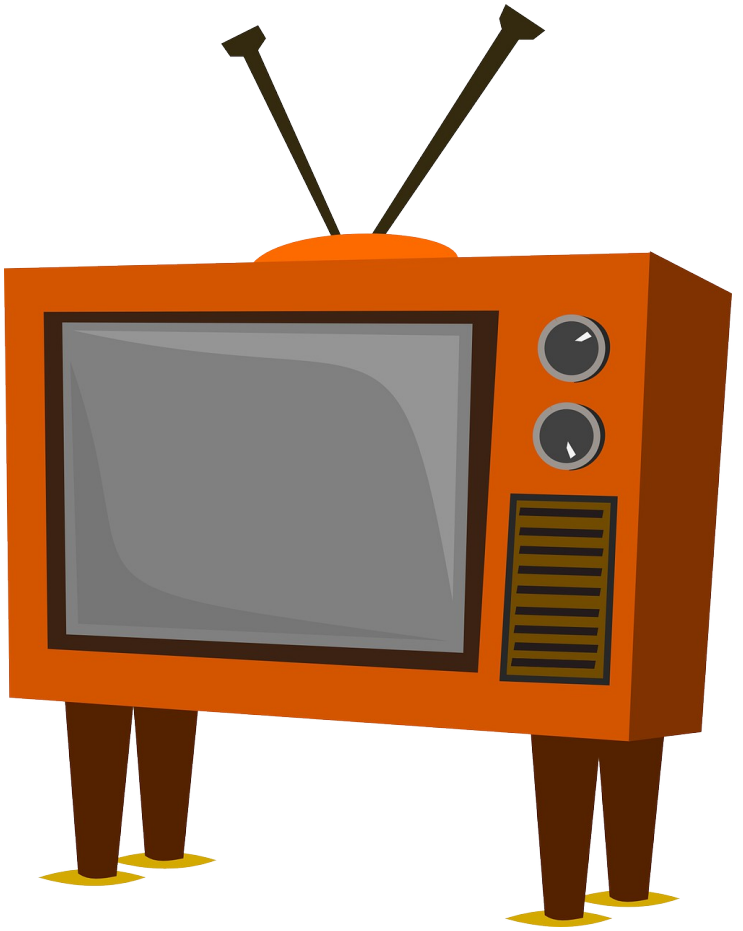
Variables

The program has two variables, **tv_price**, which is the price of a television, and **current_total**, which keeps track of the total price of the user's shopping cart.



Variables

The two variables are set to 199.99 and 0, respectively. However, if a TV is purchased, then the **total_price** variable increases by a value equal to the **tv_price** variable.



See the following:

Variables

```
tv_price = 199.99
current_total = 0

current_total = current_total + tv_price

print(current_total)
```

The output should be **199.99**

Pico – Runme.py Challenge

Let's learn how to run a Python script in the terminal by solving this PicoCTF challenge:

https://play.picoctf.org/practice/challenge/250

Conditions

Conditions are how programs make decisions as to which instructions to perform at a specific point in program execution.

Conditions

Generally, conditions trigger when a statement evaluates to **True** or **False**, so let's discuss what this means, exactly.

Conditions

The basic conditional statement in Python is the **if** statement, and **if** statements are written in code blocks like the following:

Conditions

```
password = 'mysecretpassword'

if password == 'mysecretpassword':
    print('password correct')
else:
    print('incorrect password')
```

Here, the program will print one message if the password variable is the string 'mysecretpassword' (Boolean True), and print another message if it is not (Boolean False).

Conditions

```
password = 'mysecretpassword'

if password == 'mysecretpassword':
    print('password correct')
else:
    print('incorrect password')
```

If statements start with if, then a Boolean equation, then a semicolon.

Conditions

```
password = 'mysecretpassword'

if password == 'mysecretpassword':
    print('password correct')
else:
    print('incorrect password')
```

Then an indentation on the next line, followed by instructions if the condition is met, then on the next line else:, then an indent on the next line, followed by instructions if the condition is not met.

Conditions

```
password = 'mysecretpassword'

if password == 'mysecretpassword':
    print('password correct')
else:
    print('incorrect password')

print('Was your password correct?')
```

The **else** portion of the code block is optional. After the **if** statement code block finishes, program execution continues on the next line.

Boolean Operators

Operator	Meaning
==	Equal to
<	Less than
>	Greater than
!=	Not equal to
<=	Less than or equal to
>=	Greater than or equal to

We can use any of the above Boolean operators in If Statements.

= and == are not the same

```
number = 42


if number == 42
  print("The number is 42!")
```

A common mistake using conditionals is to use the = symbol for comparisons. The == symbol is used for comparisons in conditionals.

Else Keyword

```
number = int(input("Input a number"))
if number % 2 == 0:
    print(f"{number} is an even number")
else:
    print(f"{number} is an odd number")
```

As mentioned earlier, the **else** keyword can be used with **if** conditionals to provide instructions to be executed if the **if** condition is not triggered.

Pico – PW Crack 1

Let's learn how to read and modify a Python script in the terminal by solving this PicoCTF challenge:

https://play.picoctf.org/practice/challenge/245

Functions

```
>>> print("Print() is a built-in function in Python.")
Print() is a built-in function in Python.
```

Programming functions are defined instructions that can be used multiple times in the same code.

Functions

```
>>> print("Print() is a built-in function in Python.")
Print() is a built-in function in Python.
```

If a specific set of instructions needs to be run multiple times in the same code, it is more efficient to create a function out of those instructions rather than input the same instructions again.

Functions

```
>>> print("Print() is a built-in function in Python.")
Print() is a built-in function in Python.
```

In fact, we've been making use of the most common Python function this whole time: the **print** function.

Functions

```
>>> print("Print() is a built-in function in Python.")
Print() is a built-in function in Python.
```

Any arguments that the function requires to execute properly (if any) are provided inside the parentheses when calling (running) the function.

Functions

```
def my_greeting(name):
    print('Greetings, %s!' % name)
```

To create a function, start with def, then the name of the function, then a pair of parentheses, with any arguments required inside the parentheses, then a colon.

Functions

```
def my_greeting(name):
    print('Greetings, %s!' % name)
```

Since a function is a code block, on the next line, indent (4 spaces), then input the function's instructions. If the instructions span multiple lines, each line must be indented. Any argument variables should be included in the instructions.

Functions

```
def my_greeting(name):
    print(f"Greetings, {name}!")


my_greeting("theshyhat")
```

```
Greetings, theshyhat!
```

Here, our example function is defined, then the function is called, with the corresponding output shown thereafter.

The Return Statement

```
def sales_tax_calc(amount):
    sales_tax = 1.15
    return amount * sales_tax

sales_tax_calc(55.00)
```

Return is an instruction that can be used in functions. When used, **return** stores the value specified, but doesn't print it out to the console.

The Return Statement

```
def sales_tax_calc(amount):
    sales_tax = 1.15
    return amount * sales_tax

print(sales_tax_calc(55.00))
```

```
63.25
```

So the **return** portion of a function is useful for passing values to other parts of the program, but won't output to the console unless we use the **print** function with it.

Pico – Serpentine

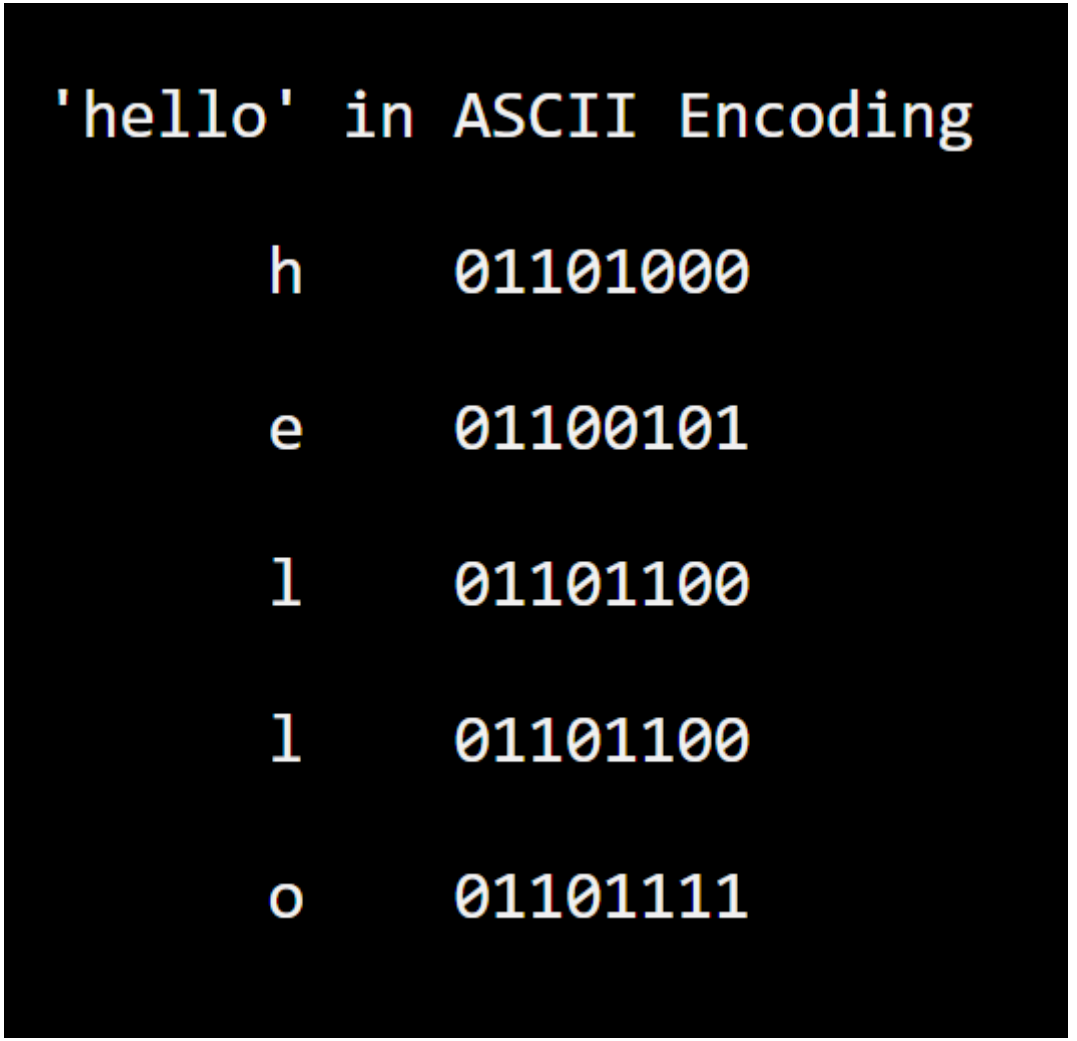
Let's learn how to read and modify a Python script in the terminal by solving this PicoCTF challenge:

https://play.picoctf.org/practice/challenge/251

Part 3 - Cryptography

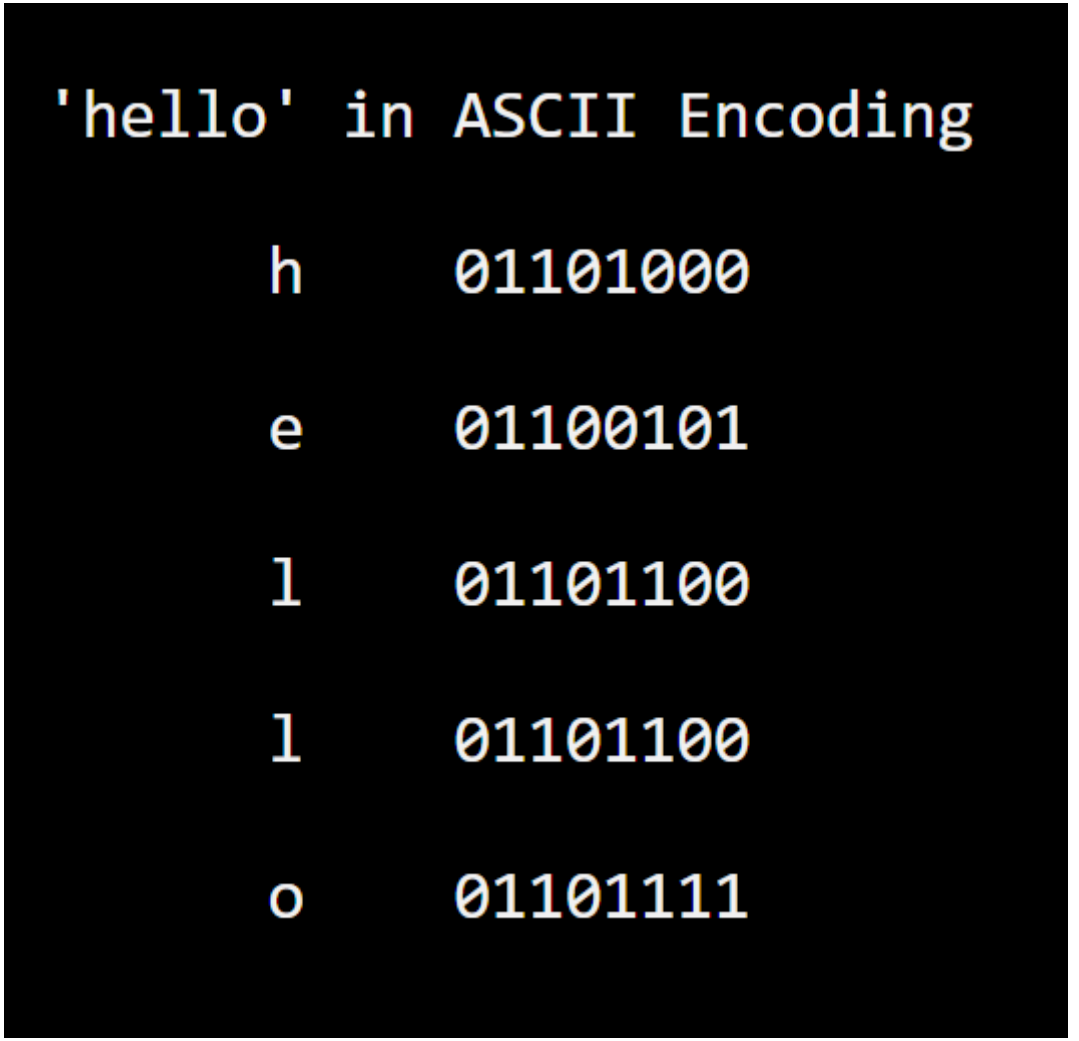
What is Encoding?

Encoding refers to the process of converting data from one form to another, often for the purpose of efficient storage, transmission, or representation.



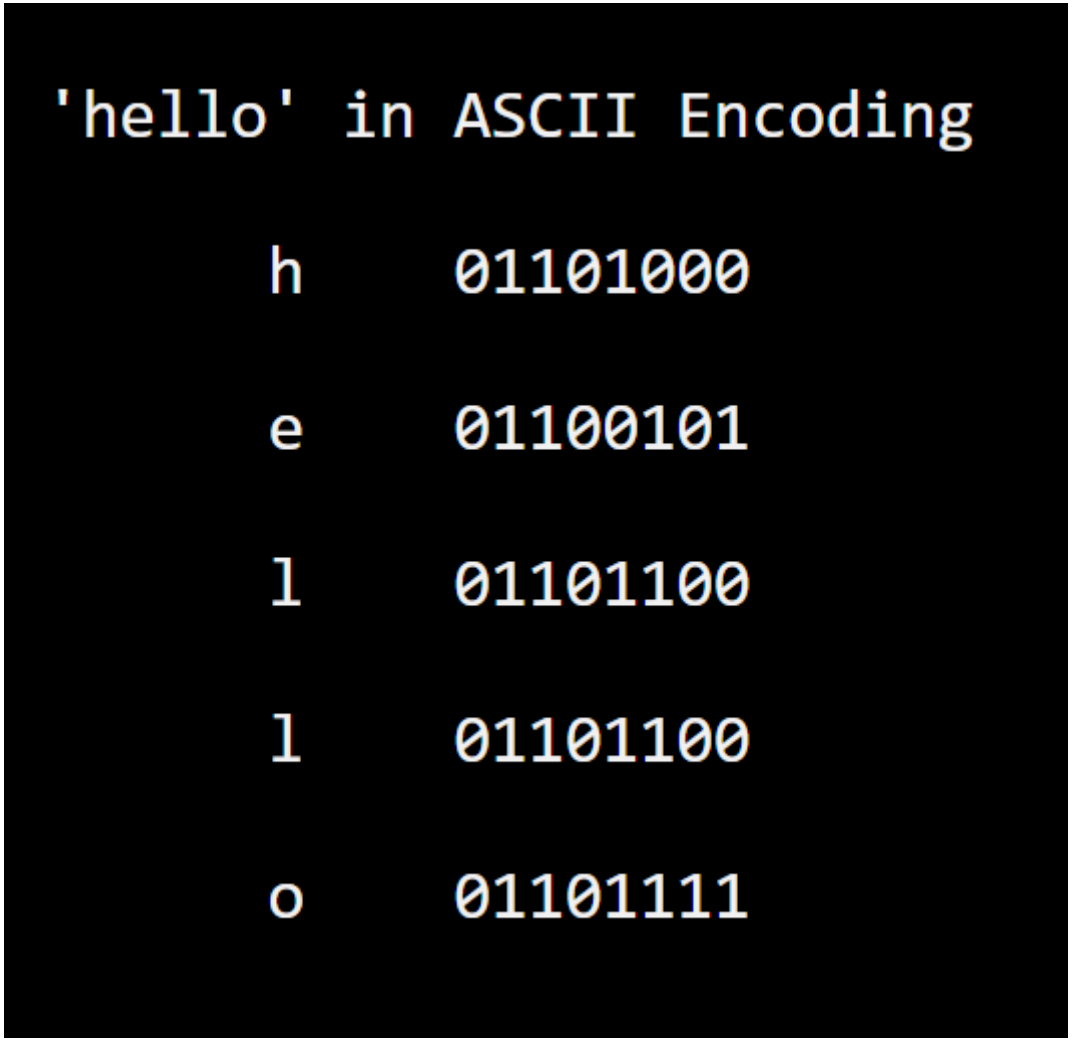
What is Encoding?

For example, before the word **hello** can appear on a computer screen, it needs to be encoded into ASCII characters from the binary numbers that computers use.



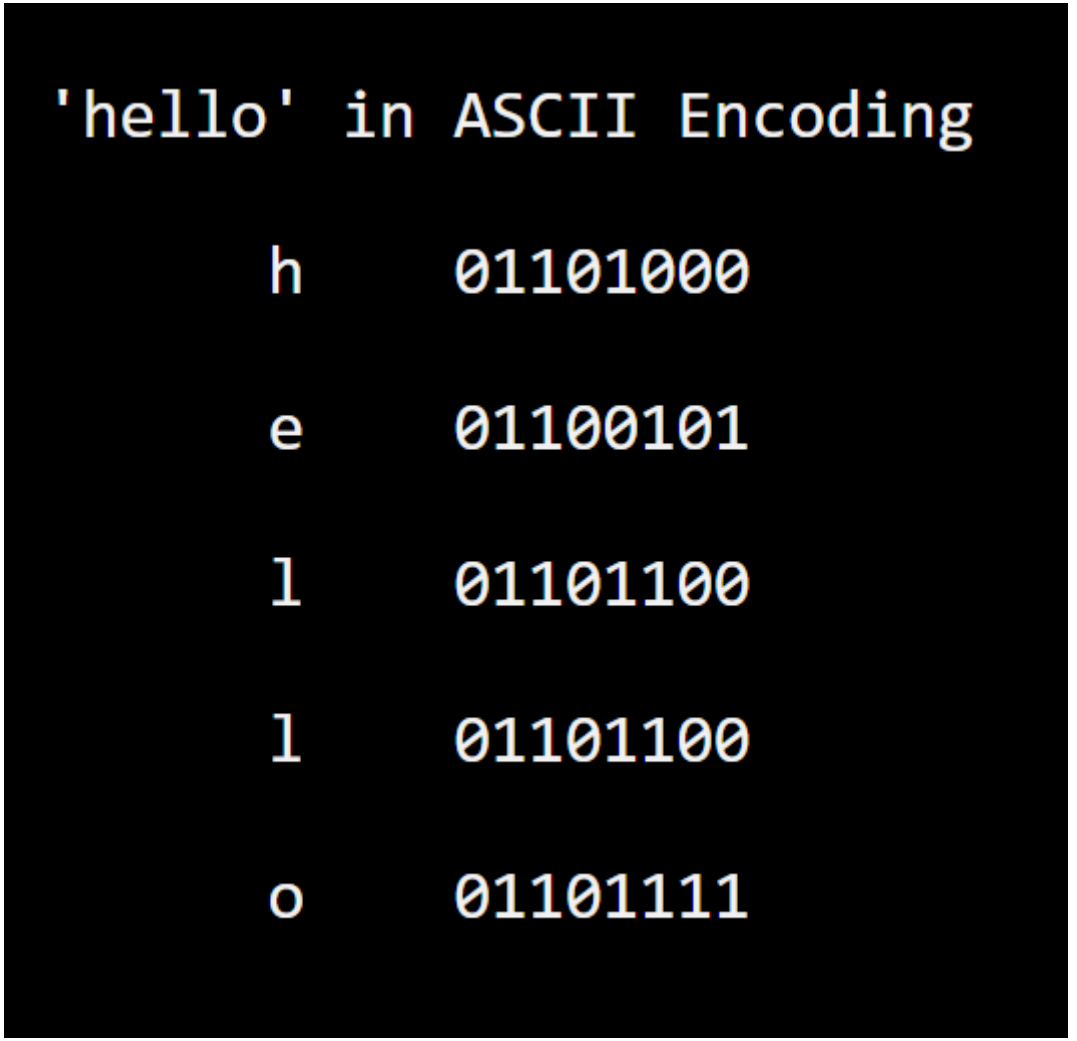
What is Encoding?

Although encoding transforms data from one form to another, it is **not** considered cryptography, because the purpose of encoding isn't to keep data secret.

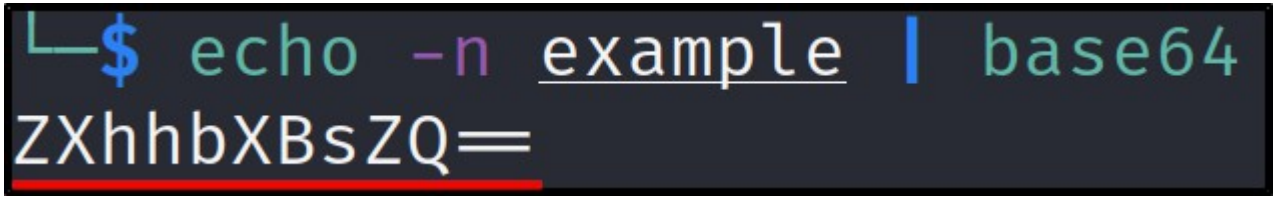


What is Encoding?

Nonetheless, encoding is a good introduction to cryptography, because it introduces us to transforming data according to different systems and rules.

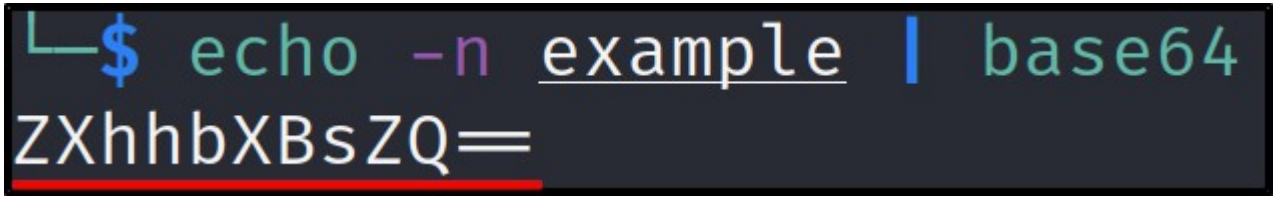


What is Base64 Encoding?



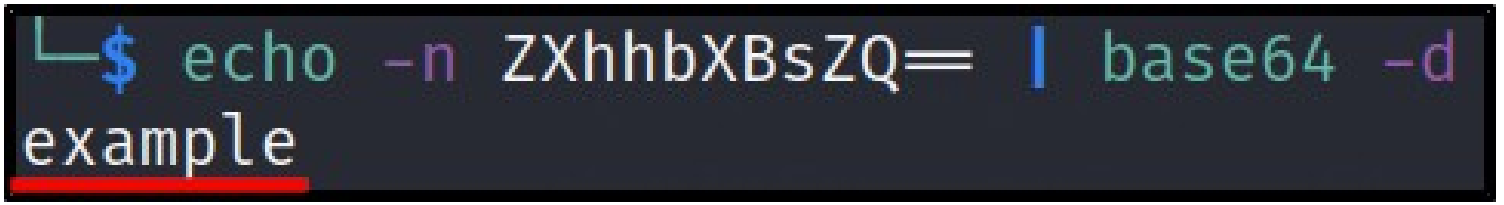
Base64 is a form of binary-to-text encoding, where data is converted to a string of letters, numbers, and symbols.

What is Base64 Encoding?



The primary use of Base64 is the encoding of binary data files (executables, pictures, etc) to facilitate transfer over computer networks (such as the internet), using different protocols and software.

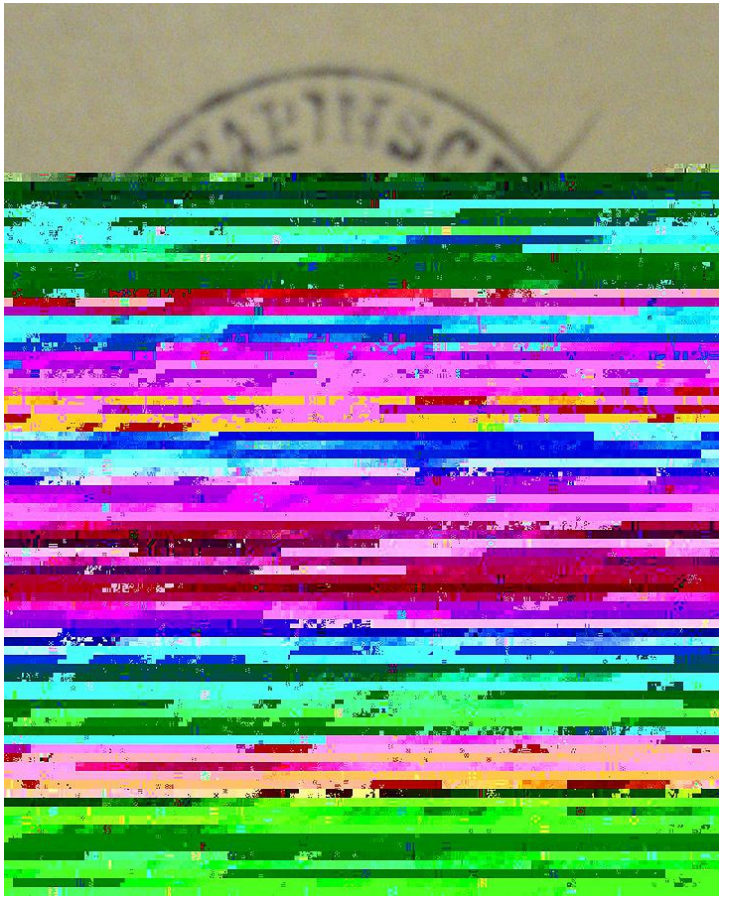
What is Base64 Encoding?



After transfer, files are decoded from Base64 back into their original formats.

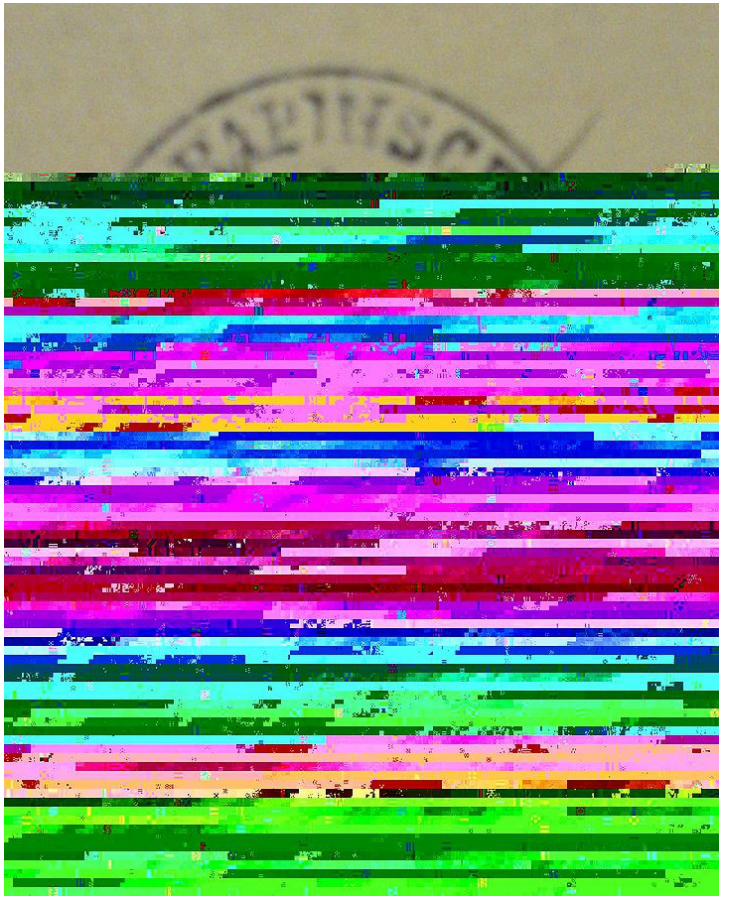
What is Base64 Encoding?

The reason why files must be encoded before file transfer is because the contents of data files may include bytes that could be interpreted as--



What is Base64 Encoding?

program instructions by the software handling the files, leading to errors and / or file corruption.



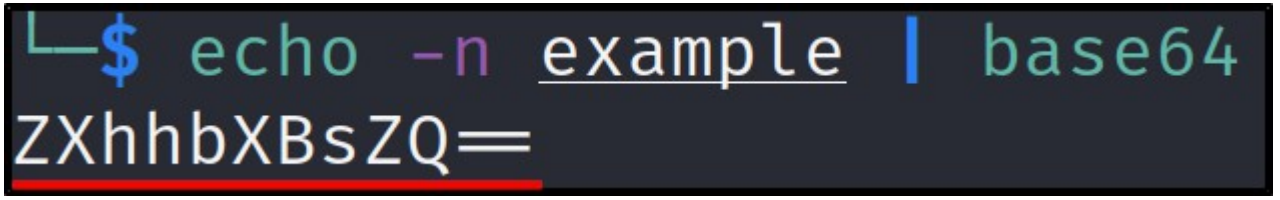
What is Base64 Encoding?

The number in the name base64 refers to the number of characters included in its character set, which includes all upper and lower case letters of the English alphabet, as well as numbers 0 to 9 and special characters + and / .

Base64 Encoding Table

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

What is Base64 Encoding?



Base64 strings are always a number of characters divisible by 4, and in the event that isn't the case, a number of equal symbols will be added to the string until the character count is divisible by 4.

Pico – Repetitions

Let's learn how to decode base64 strings in the terminal by solving this PicoCTF challenge:

https://play.picoctf.org/practice/challenge/371

Encoding is not Encryption

Although encoding transforms data from one form to another, the intention of encoding is not to hide the contents of the data, so it is not cryptography

Encoding is not Encryption

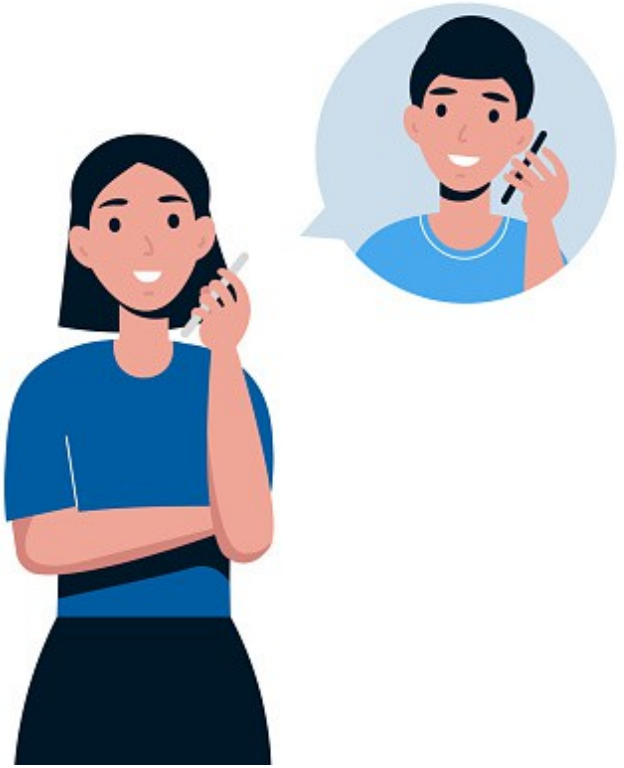
Additionally, encoding methods are meant to be well-known and easily-reversible, which further sets encoding apart from encryption

What is Cryptography?

Let's start our exploration of cryptography by going over some key terms.

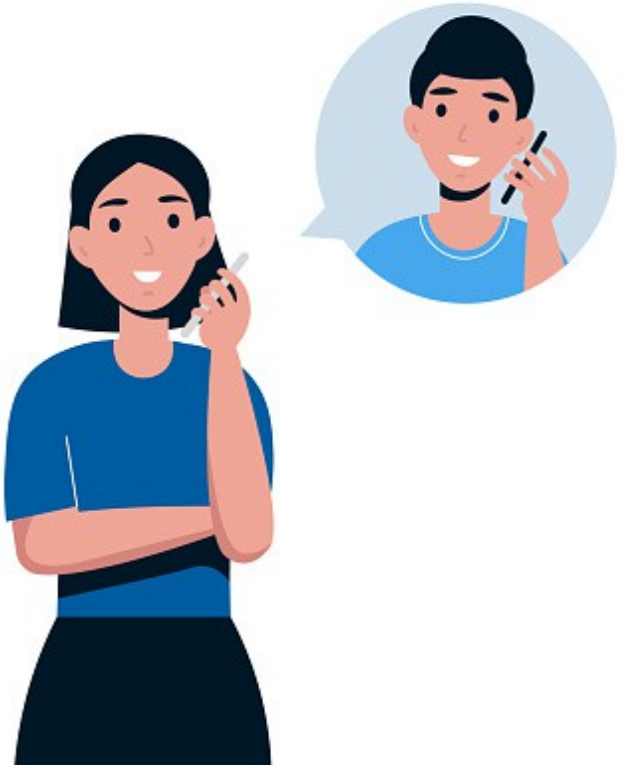
Intro to Cryptography Terms

Let's suppose that Bob wants to use Alice's streaming video app account to watch some movies.



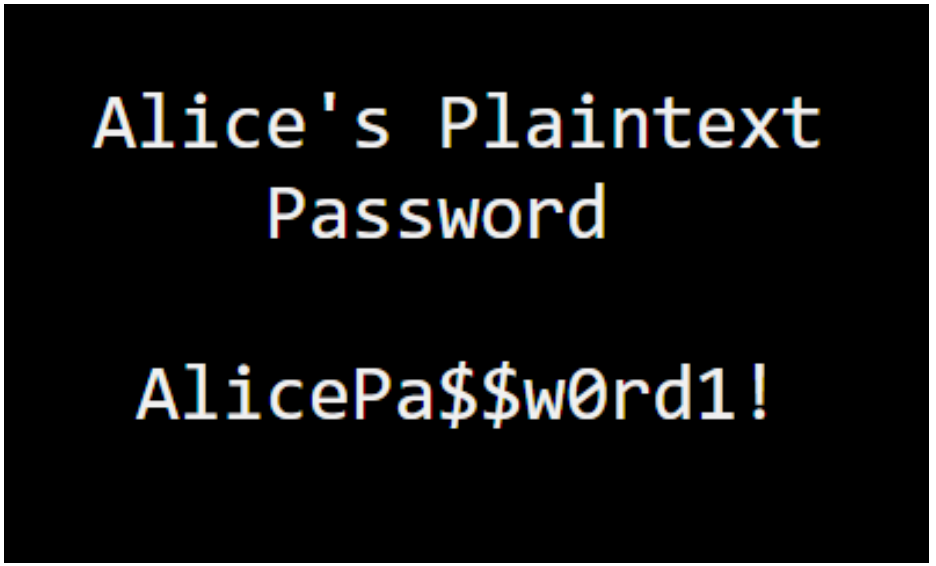
Intro to Cryptography Terms

Alice is fine with Bob using her account, but wants to send her account password to Bob secretly, to which Bob agrees.



Plaintext / Cleartext

Alice's password before it is transformed into a secret message through cryptography, is called **plaintext** or **cleartext**.



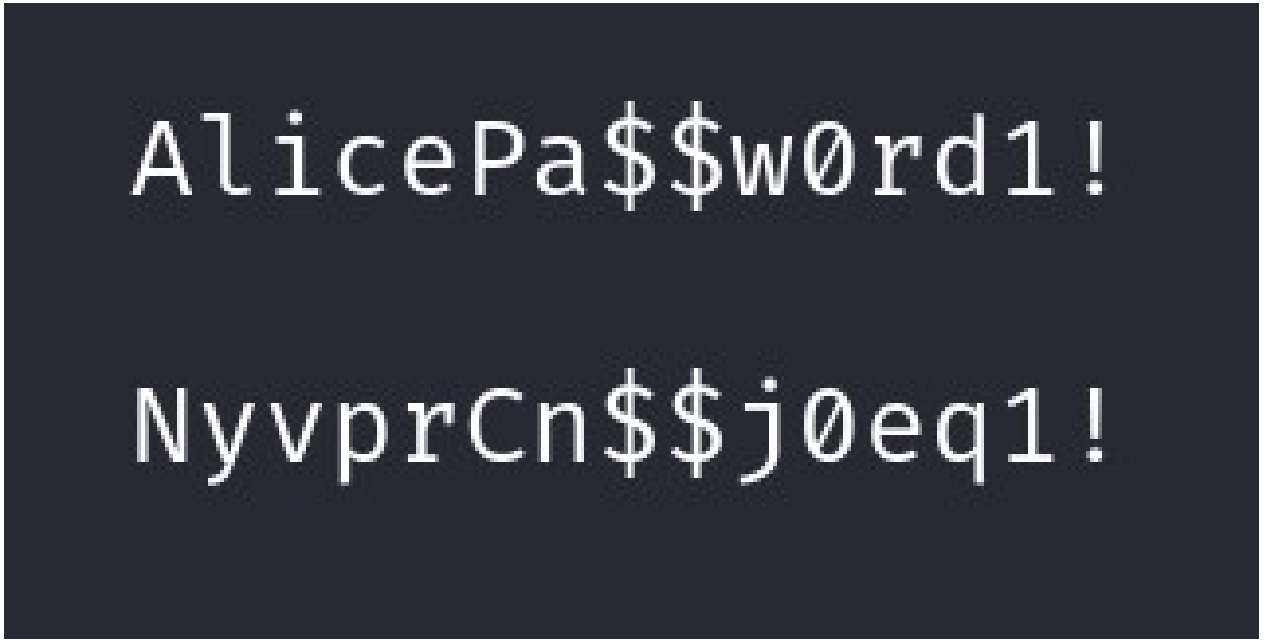
Ciphers

Alice then uses a cryptographic algorithm called a **cipher** to transform the message from its original form into a secret message.

Ciphertext

This secret form of the message is called **ciphertext**.

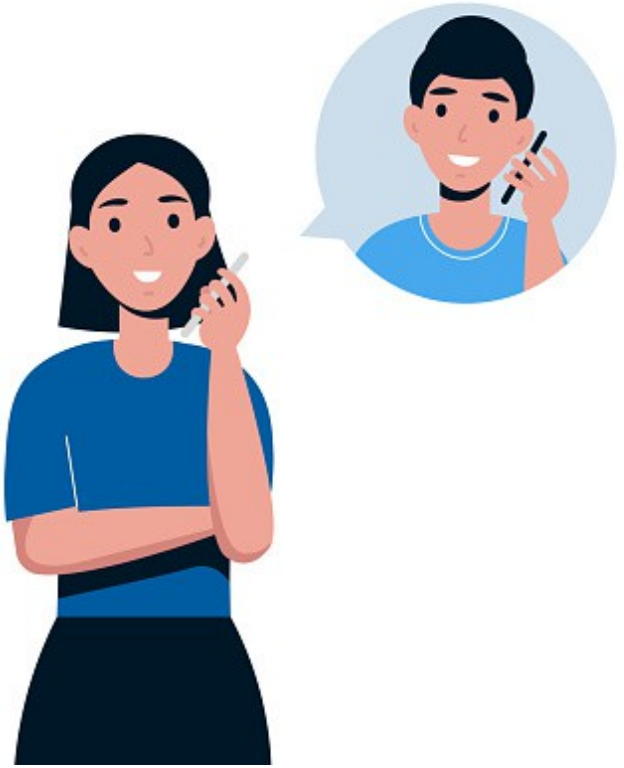
Encryption



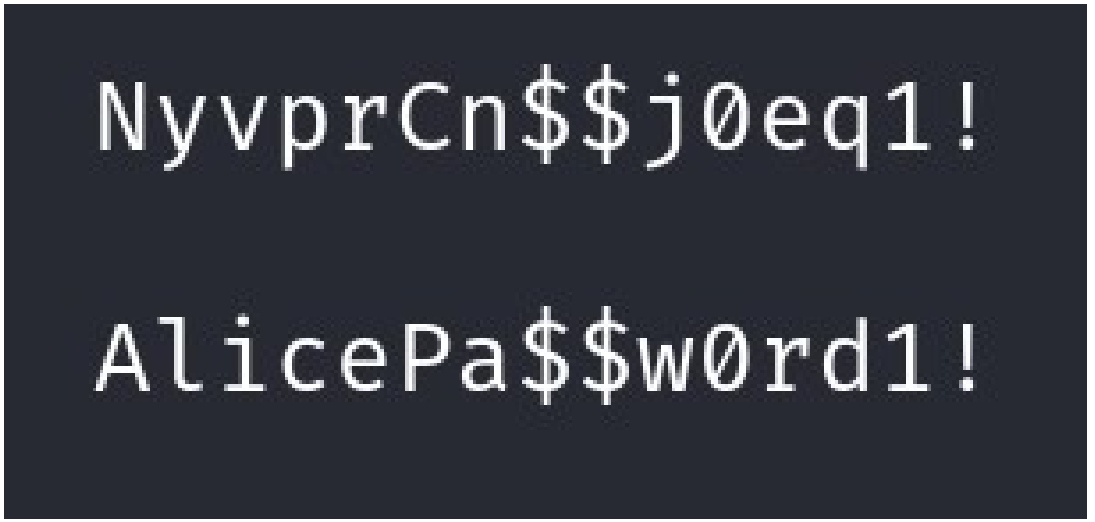
The act of transforming plaintext into ciphertext is called **encryption**.

Intro to Cryptography Terms

Alice then sends the message with the encrypted password to Bob, and tells him which cipher was used to encrypt the password



Intro to Cryptography Terms



Upon receiving the message, Bob uses the same cipher to transform the ciphertext back into plaintext. This act is called **decryption**.

Intro to Cryptography Terms

After decrypting the message from Alice, Bob can use the password to log into Alice's streaming video app account and watch movies.

Intro Cryptography Terms

Cipher – **A cryptographic algorithm used in encryption and decryption**

Plaintext – **A message or piece of text that is not encrypted**

Ciphertext – **An encrypted message or piece of text**

Encryption – **The act of transforming plaintext into ciphertext**

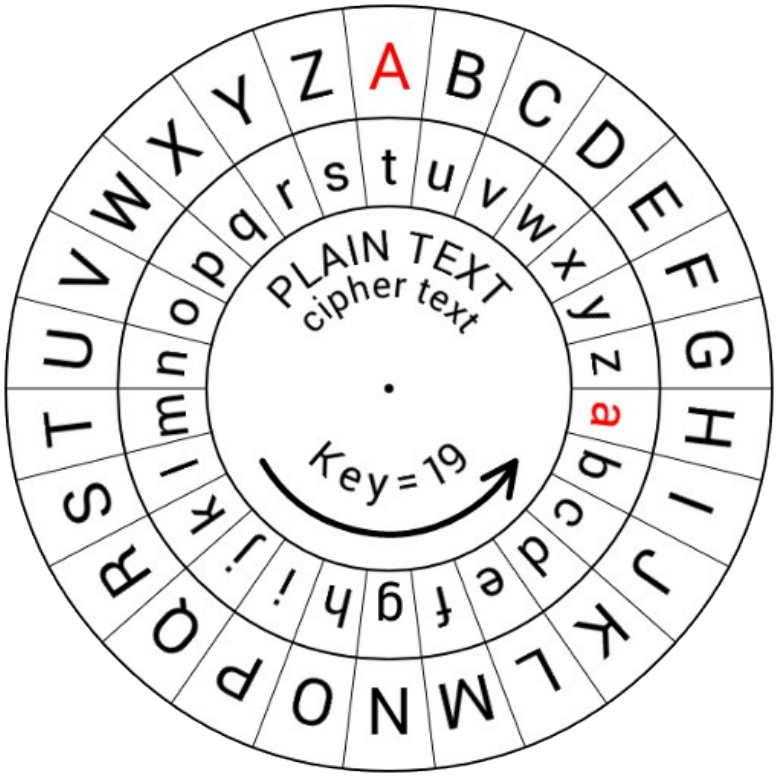
Decryption – **The act of transforming ciphertext into plaintext**

Classical Ciphers and Modern Cryptographic Ciphers

Cryptography ciphers can be divided into two categories:
classical ciphers
and **modern ciphers**

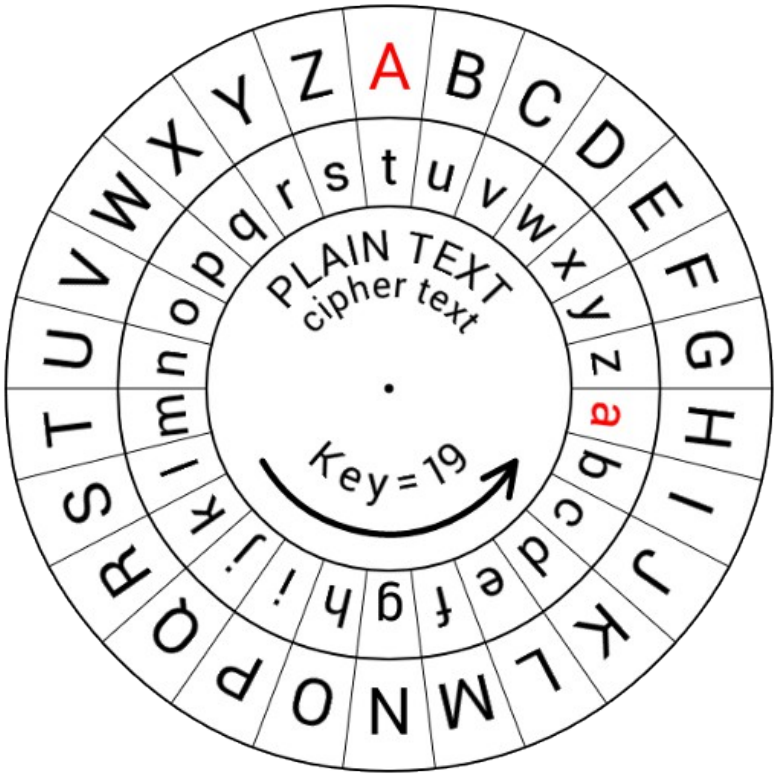
Classical Ciphers and Modern Cryptographic Ciphers

Classical ciphers refer to cryptographic ciphers used prior to the introduction of computer-aided algorithms



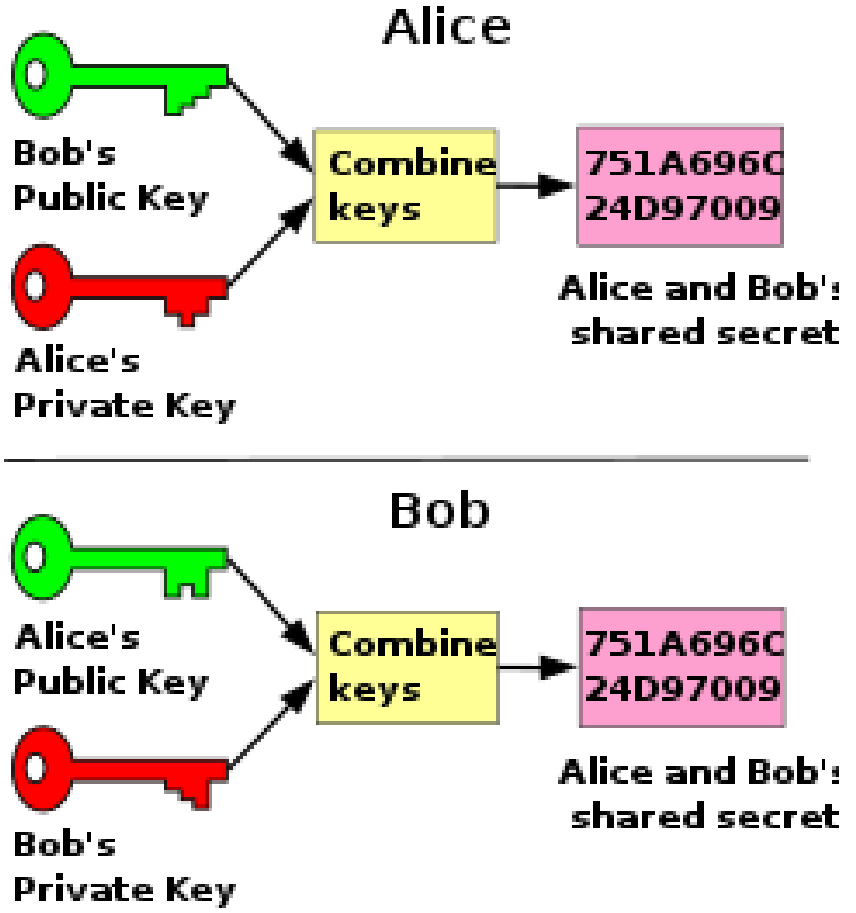
Classical Ciphers and Modern Cryptographic Ciphers

With classical ciphers, it is generally possible to perform the steps required to encrypt and decrypt messages without the aid of computers or calculators



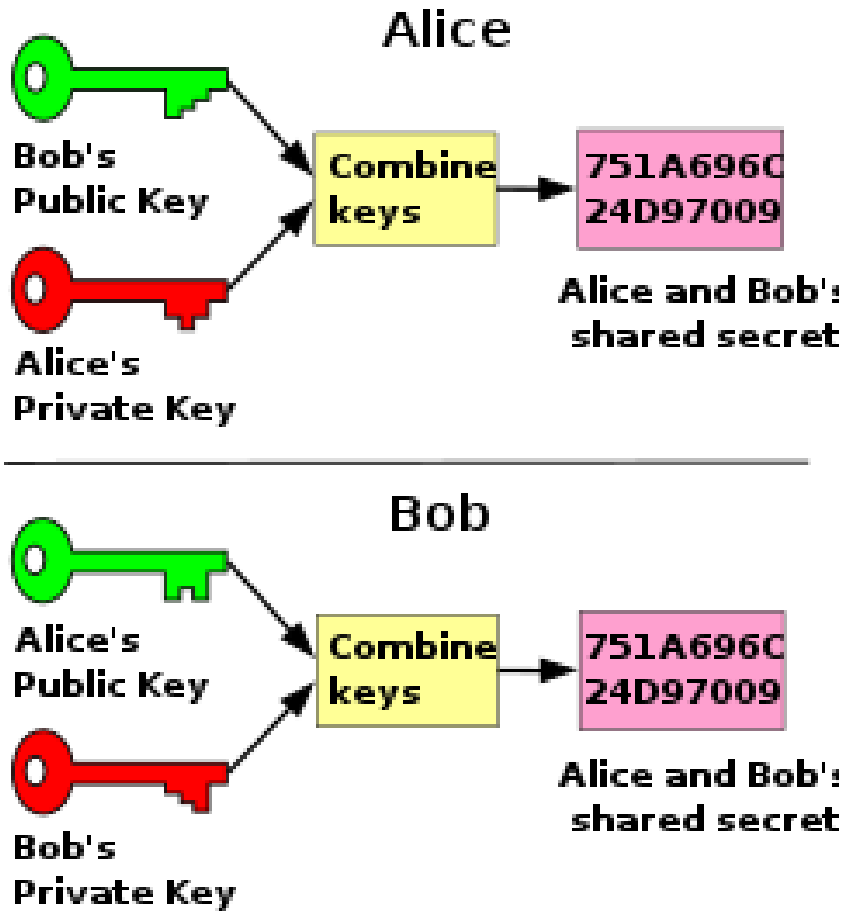
Classical Ciphers and Modern Cryptographic Ciphers

Modern ciphers are ciphers that incorporate complex mathematical operations in their encryption / decryption processes, and are impractical to use without the aid of computer processing



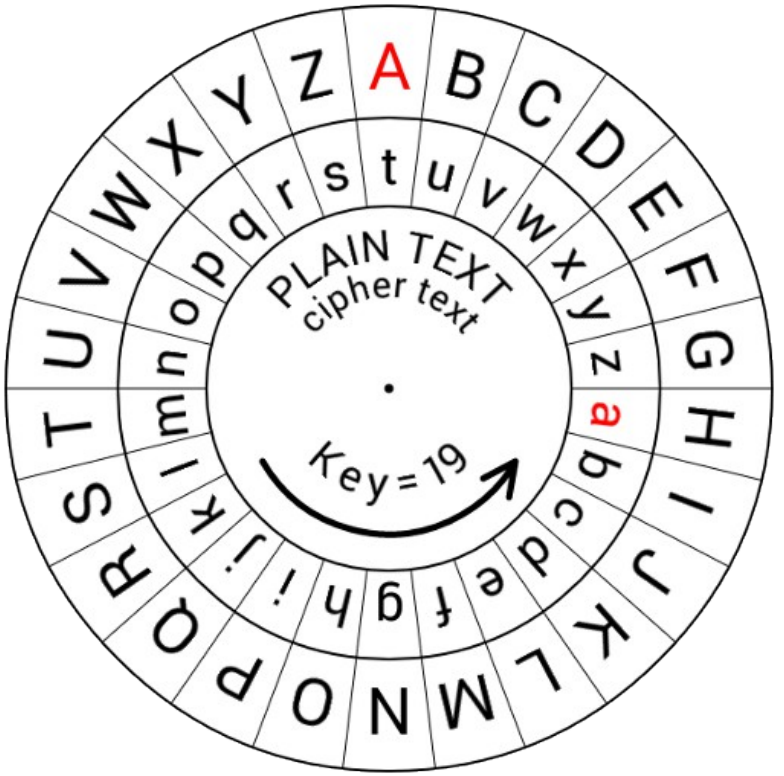
Classical Ciphers and Modern Cryptographic Ciphers

Virtually all ciphers used in computer security fall under this category



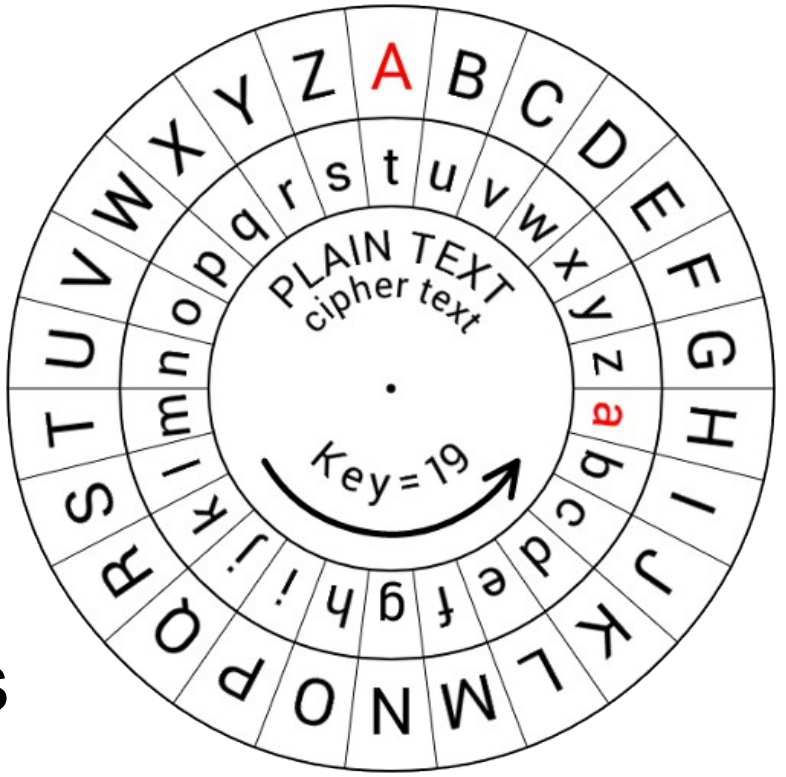
Classical Ciphers and Modern Cryptographic Ciphers

In our introduction to cryptography, we will first learn about classical ciphers for the following two reasons:



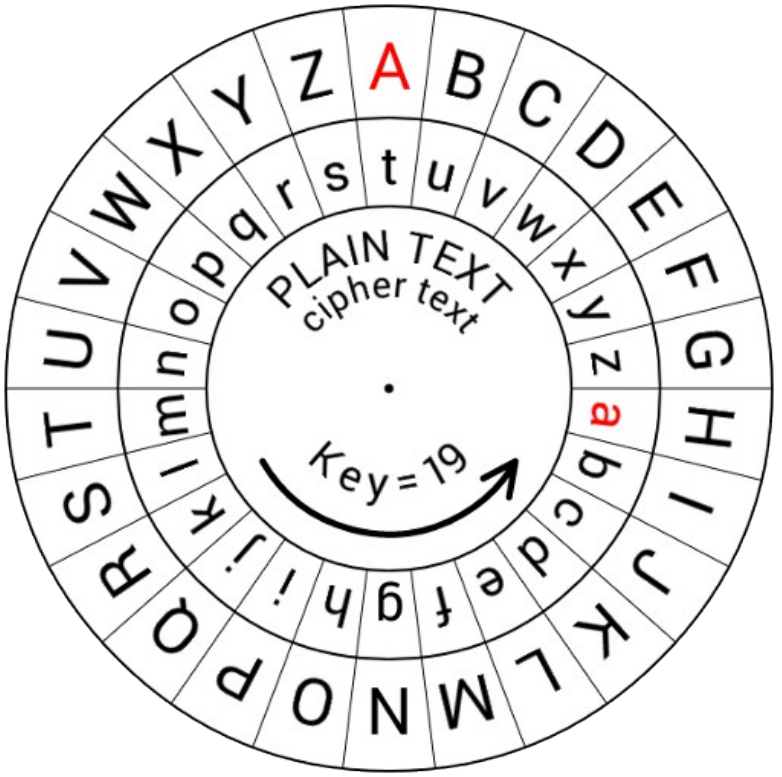
Classical Ciphers and Modern Cryptographic Ciphers

1) Beginner cryptography CTF exercises cover classical ciphers extensively, so we need to learn classical ciphers in order to engage with and solve those challenges



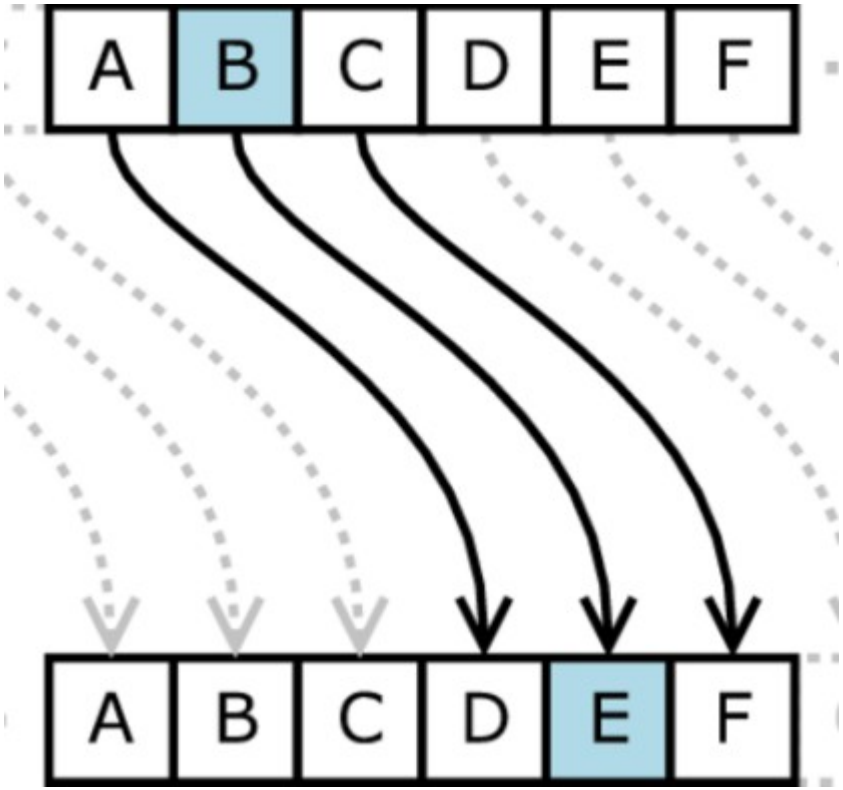
Classical Ciphers and Modern Cryptographic Ciphers

2) More importantly, as an introduction to cryptography, the methods involved in classical cryptography are much easier to understand, since they do not require complex mathematical operations



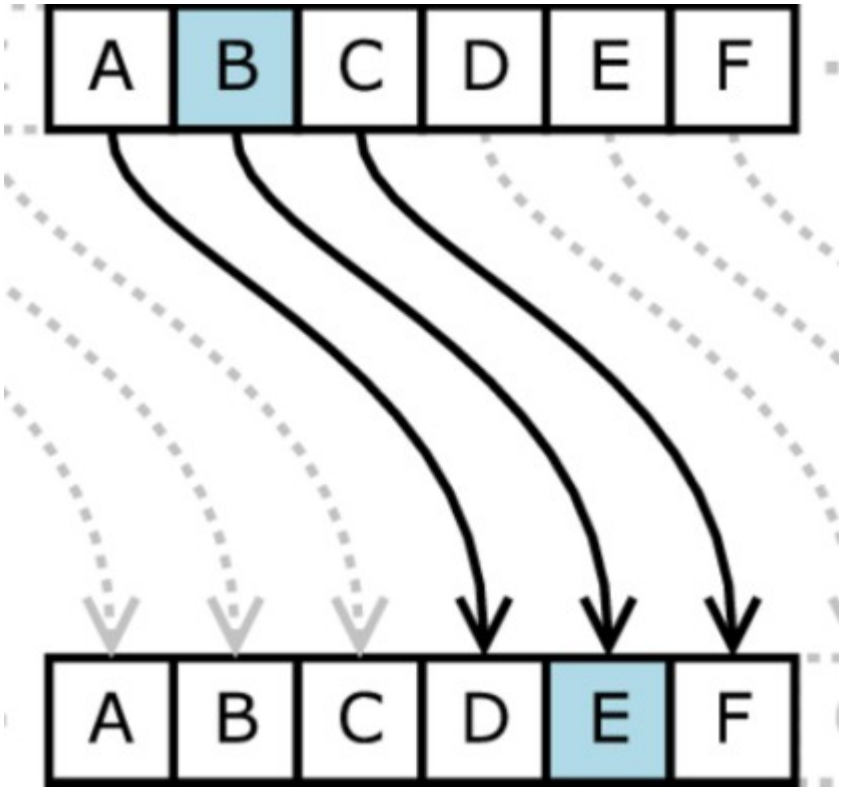
Substitution Ciphers

Substitution ciphers are a type of classical cryptographic cipher where one portion of the plaintext is substituted for a portion of ciphertext during encryption.



Substitution Ciphers

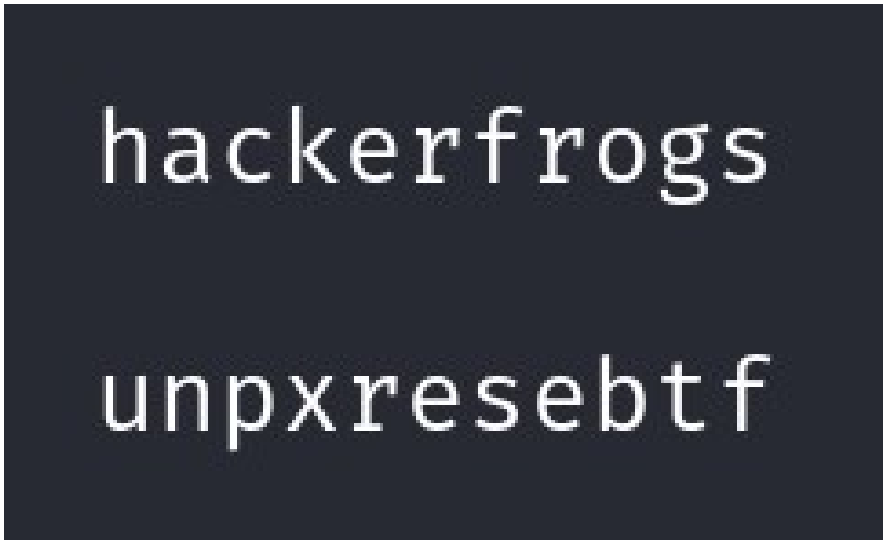
The size of the portions may be symmetrical (e.g., one character of plaintext is substituted by one character of ciphertext) or asymmetrical (e.g., one character of plaintext is substituted by two characters of ciphertext or vice versa).



ROT13 Cipher

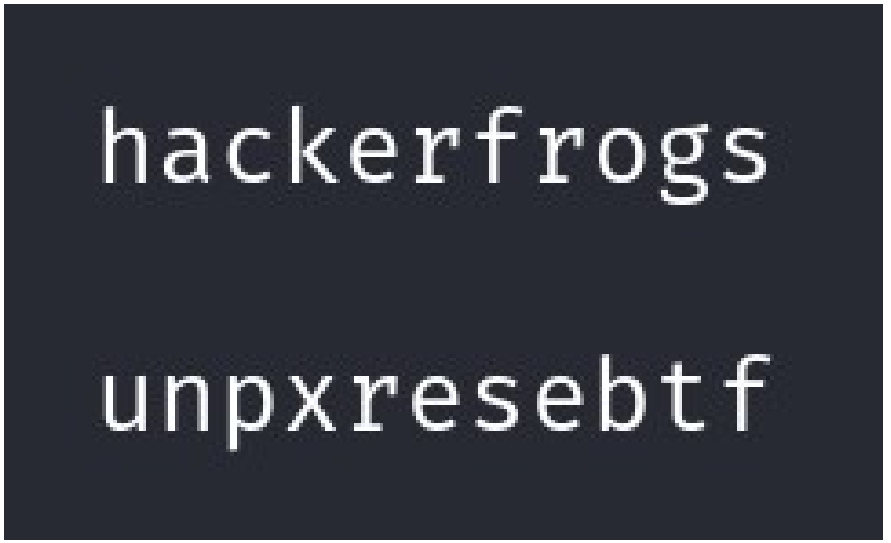
The ROT13 cipher is a simple substitution cipher where the encryption method is shift each plaintext letter 13 positions in the alphabet to form the ciphertext

ROT13 Cipher



So if we use this cipher to encrypt the plaintext `hackerfrogs`, the resulting ciphertext would be `unpxresebtf`

ROT13 Cipher



To decrypt the ciphertext we would do the same operation, shifting each ciphertext letter by 13 places in the alphabet

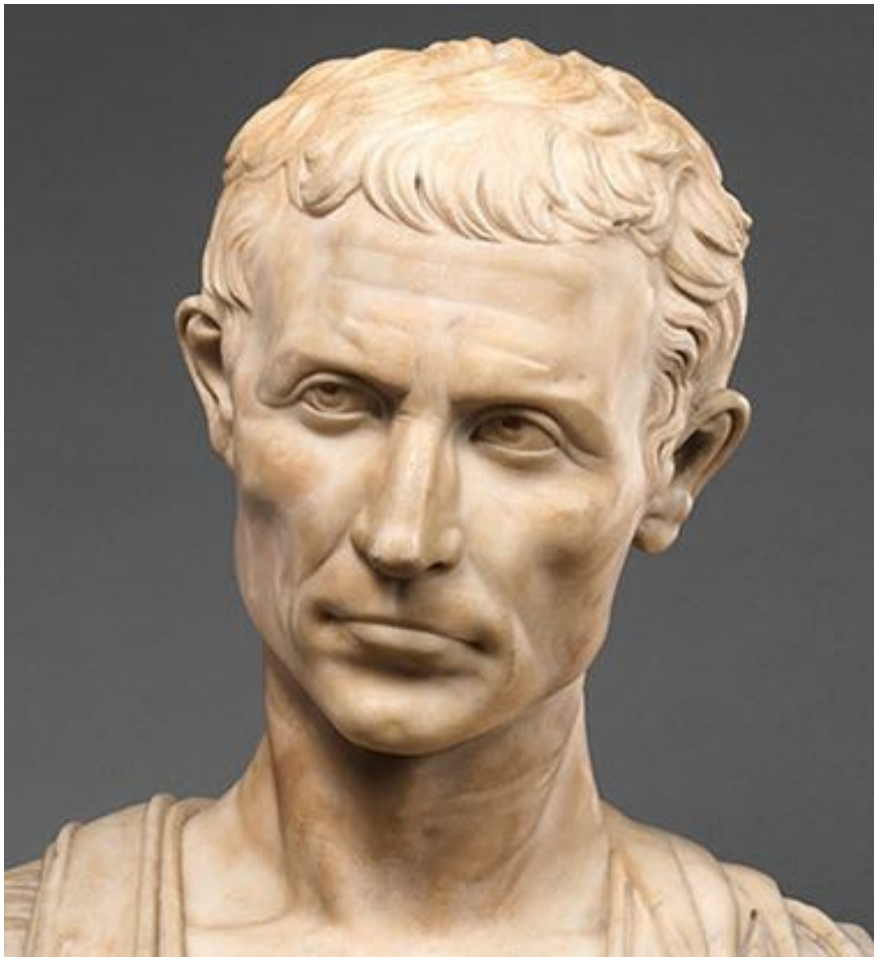
PicoCTF - 13

Let's learn more about the ROT13 cipher by working through a challenge on PicoCTF. Navigate to the following URL

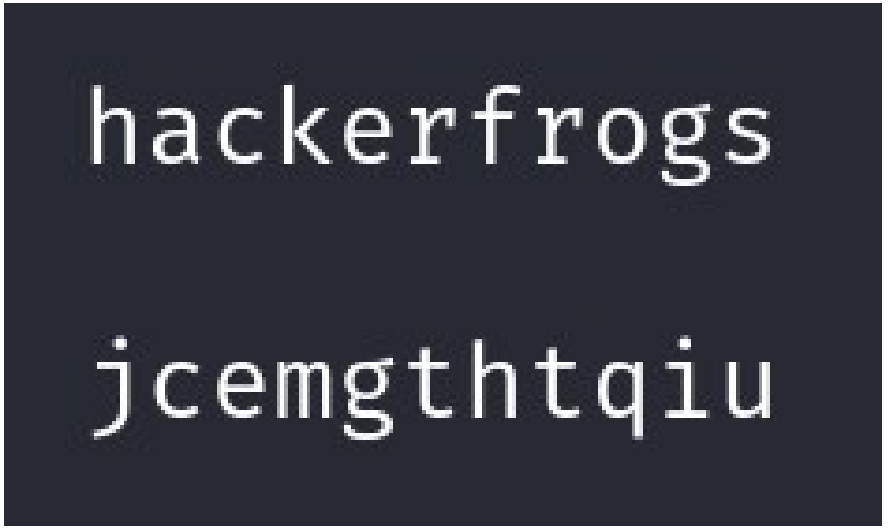
https://play.picoctf.org/practice/challenge/62?category=2&page=1

The Caesar Cipher

The Caesar Cipher is a substitution cipher where the method of encryption is to shift each letter of the plaintext by a specific number of letters in the alphabet, called the shift or key

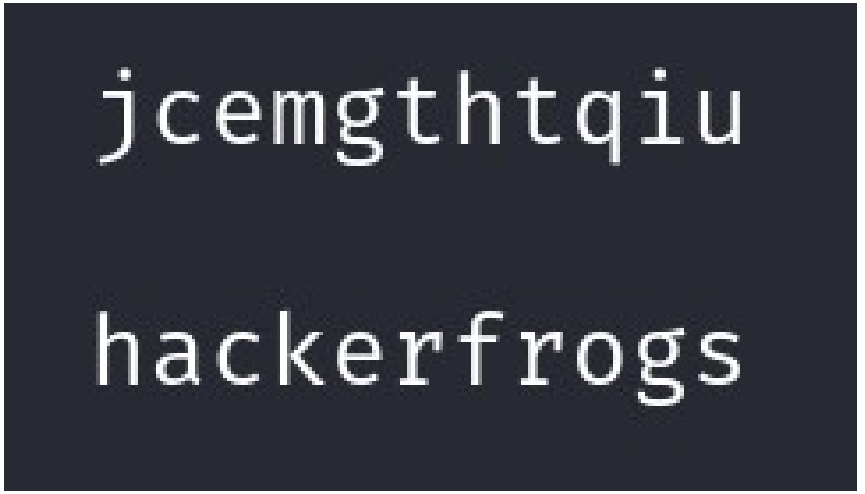


The Caesar Cipher



For example, if we use the Caesar cipher with a shift of 2 to encrypt the plaintext `hackerfrogs`, we would shift each letter two positions in the alphabet, and the resulting ciphertext would be `jcemgthtqiu`

The Caesar Cipher



To decrypt the ciphertext, we take each letter of it, and shift back two letters in the alphabet to form the plaintext

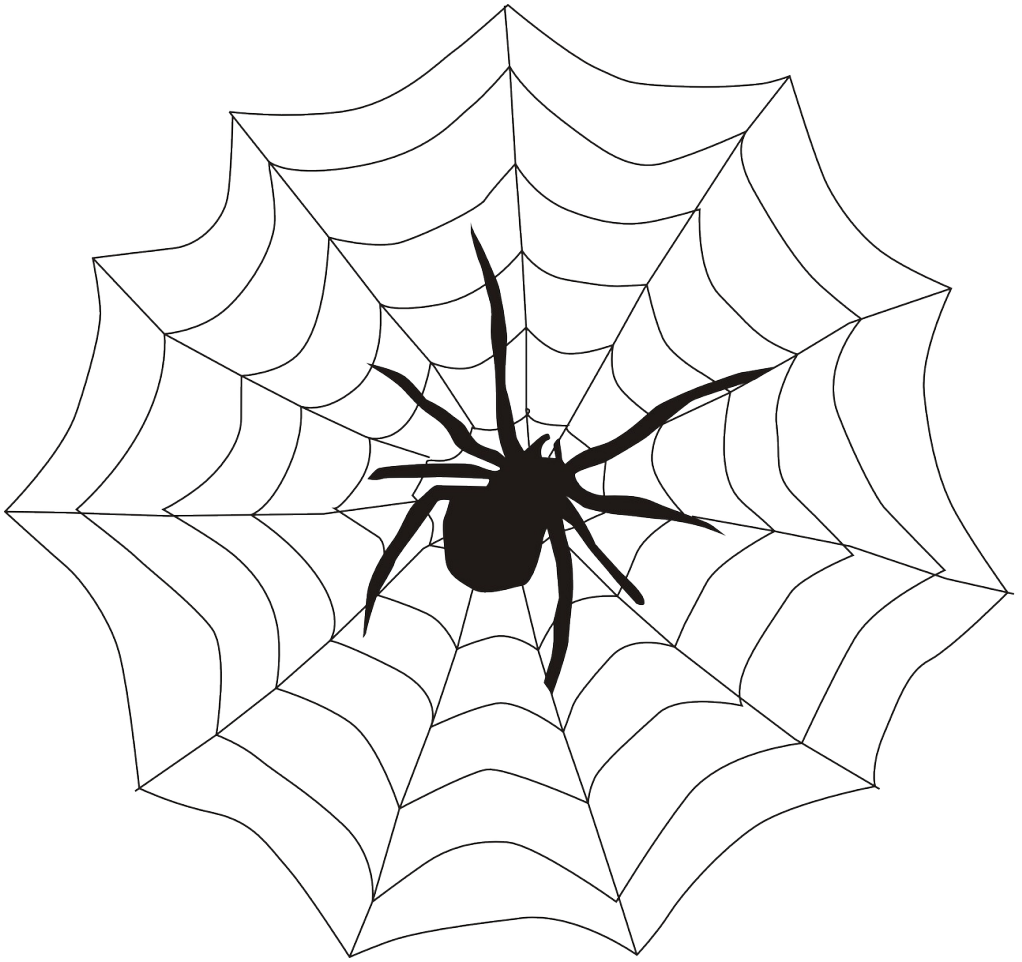
PicoCTF - Rotation

Let's learn more about Caesar cipher by working through a challenge on PicoCTF. Navigate to the following URL

https://play.picoctf.org/practice/challenge/373?category=2&page=1

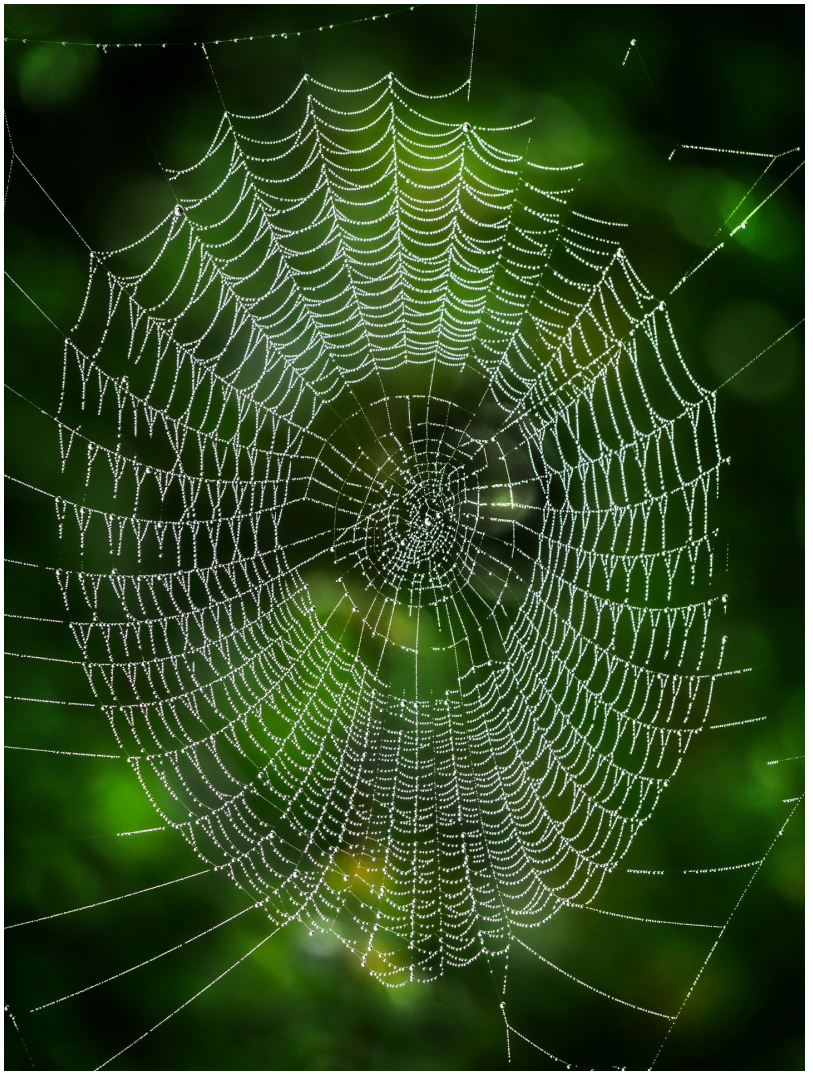
Lunch Time!

Part 4 – Web App Hacking



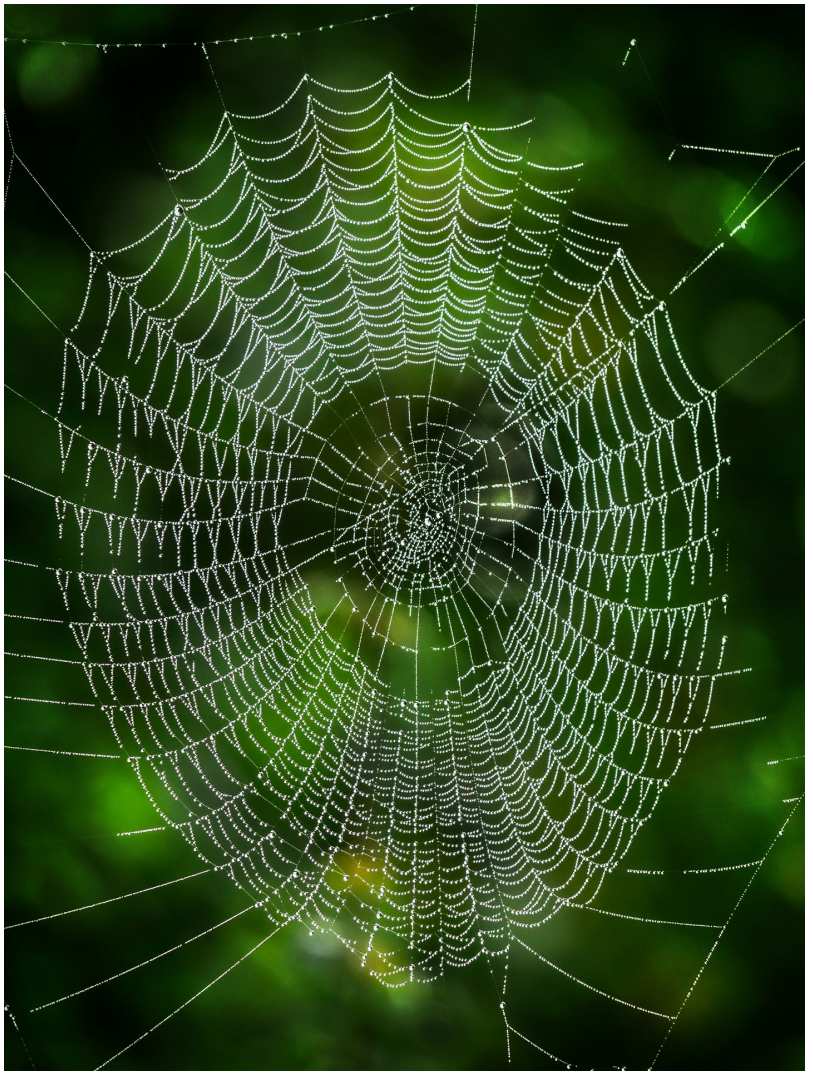
What is Web App Hacking?

Web app hacking is the intentional abuse of systems and applications accessed via the internet by a user, which benefits that user in some way.



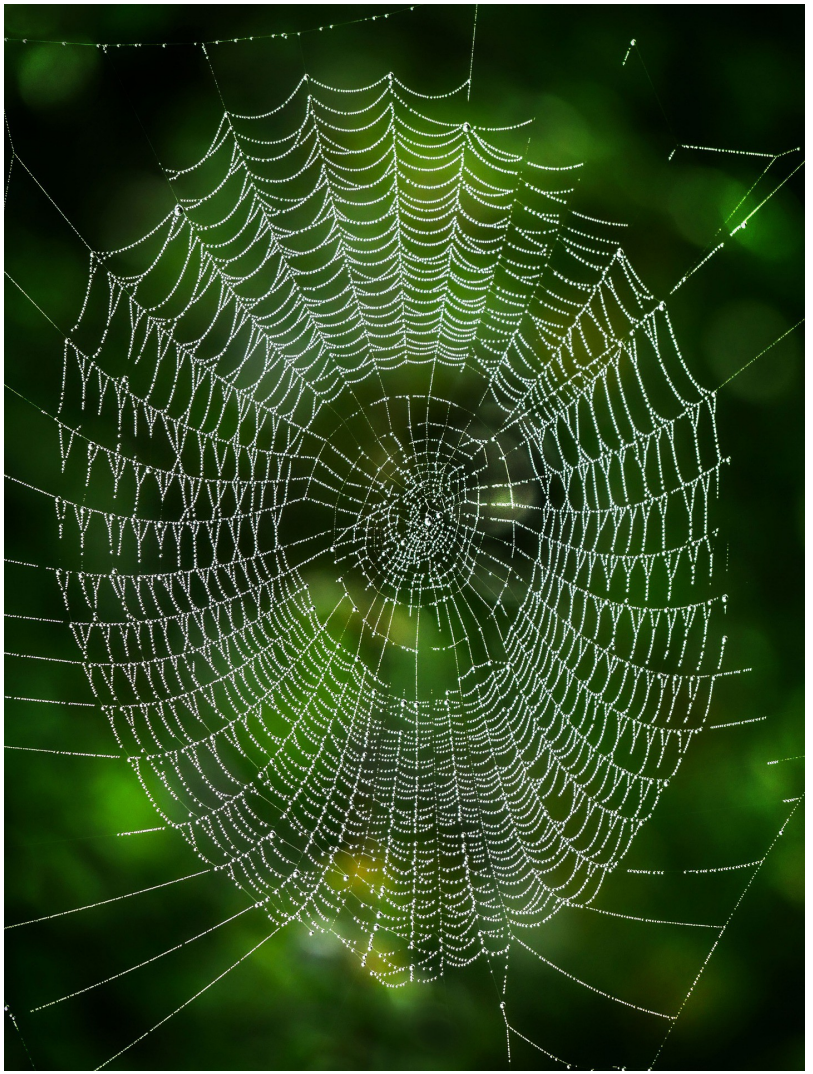
What is Web App Hacking?

This benefit can take the form of access to unauthorized data or systems, the deletion of data, denial of website services, or any number of other benefits.



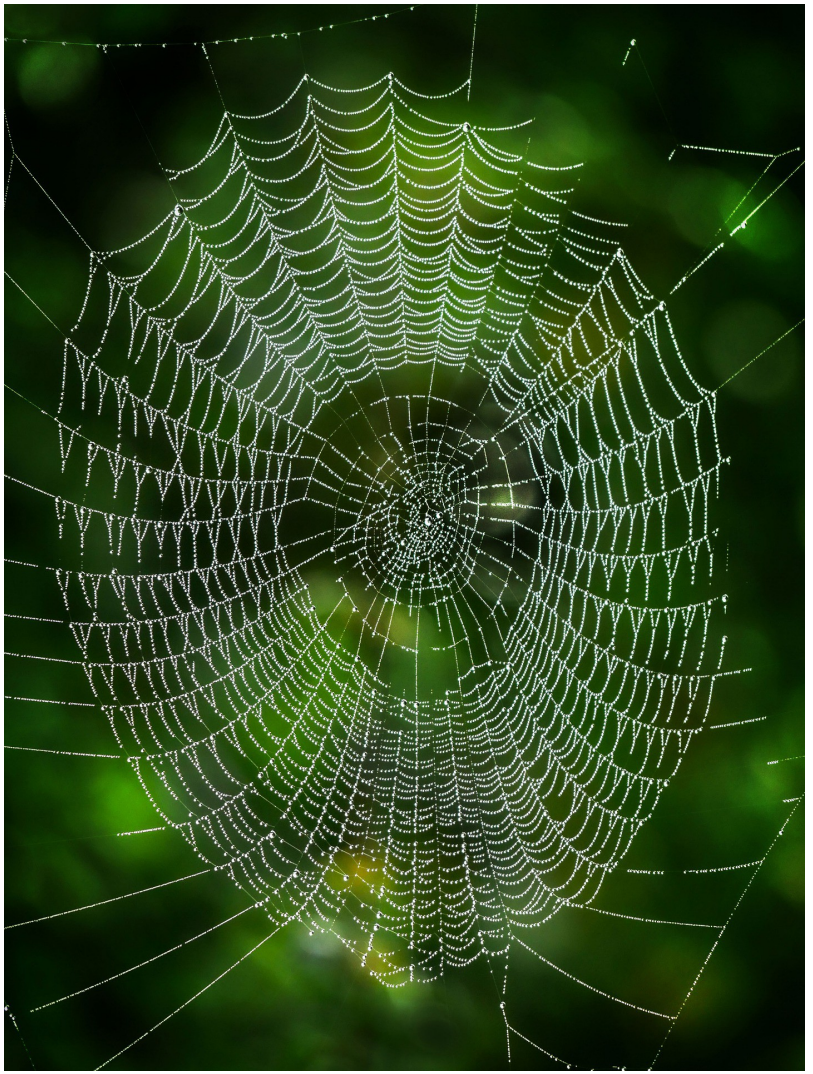
What is Web App Hacking?

In order to learn web app hacking, we must learn about the internet, how websites work, and a good amount about the technologies and software found in that space.



What is Web App Hacking?

Most students know at least the basics of using a web browser and how to use websites, so we will build on those skills to establish a collection of techniques to explore and hack web applications found on the internet.



HTTP Source Code

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css"
href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script><script
```

Web browsers render out webpages based on the HTTP code provided by the web server. Every web browser allows us to view a web page's HTTP source code.

HTTP Source Code

```
<h1>natas2</h1>
<div id="content">
There is nothing on this page
<img src="files/pixel.png">
</div>
</body></html>
```

For web app testing, this allows us to find interesting directories, developer comments, and more

PicoCTF – Inspect HTML

Let's learn more about inspecting HTML source code by working through a challenge on PicoCTF. Navigate to the following URL

https://play.picoctf.org/practice/challenge/275

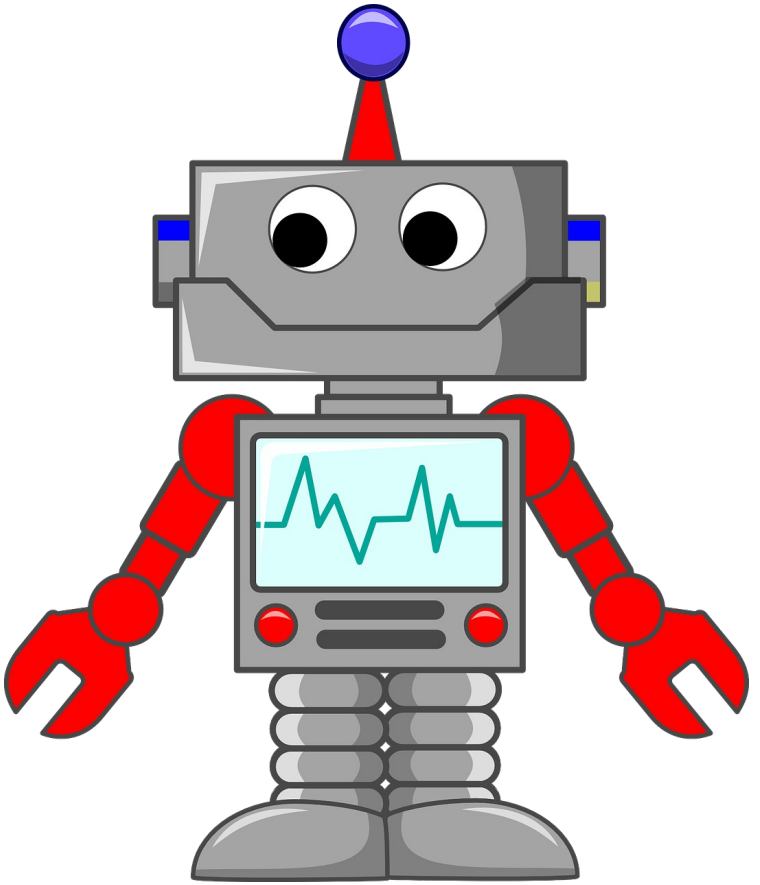
PicoCTF – Local Authority

Let's learn more about the dangers of exposed code in HTML source code by working through a challenge on PicoCTF. Navigate to the following URL

https://play.picoctf.org/practice/challenge/278

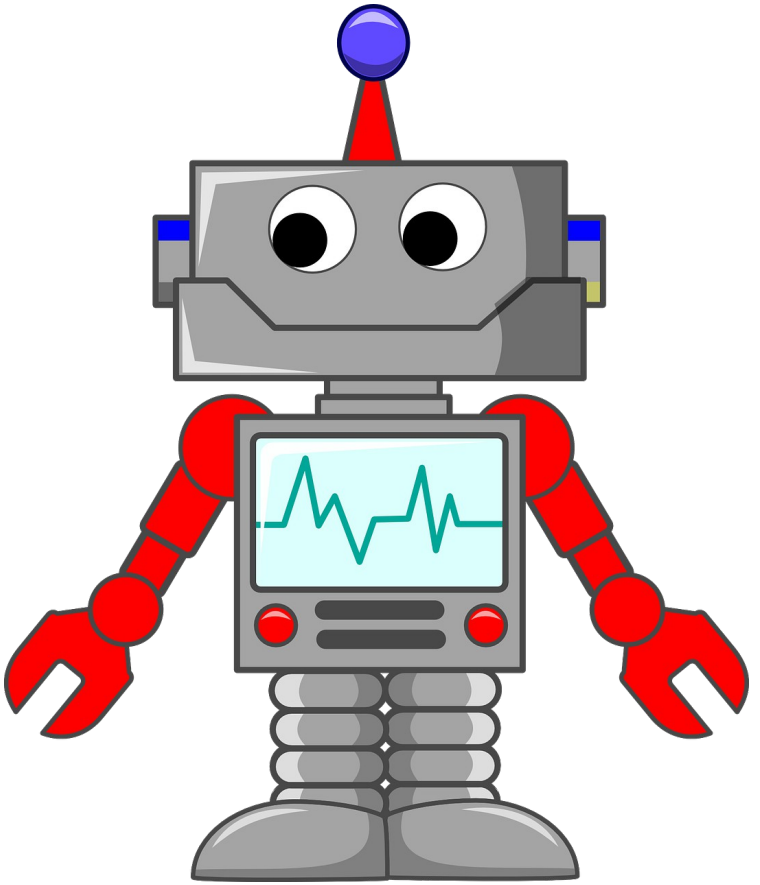
Robots.txt

Search engines (such as Google, Yahoo, DuckDuckGo, etc) use programs called robots to visit websites and map out their webpages. However, this may cause sensitive areas of websites to appear in search results.



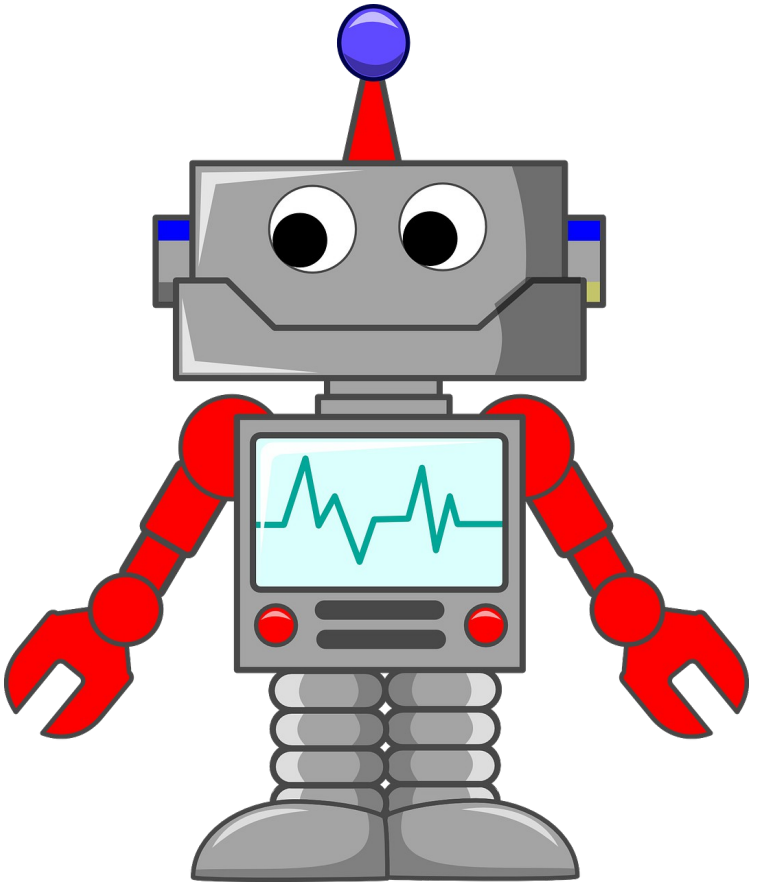
Robots.txt

In order to prevent this, website administrators can add a file called **robots.txt** to their website, which specifies which directories and / or pages of the website are off-limits to search engine robot programs.



Robots.txt

Unfortunately, if malicious users know how to find the **robots.txt** file, the contents of the file could potentially lead them to sensitive areas of the website.



PicoCTF – Where are the Robots

Let's learn about the dangers of the robots.txt file by working through a challenge on PicoCTF. Navigate to the following URL

https://play.picoctf.org/practice/challenge/4

HTTP Headers

Each time a web browser accesses a webpage, the browser makes an HTTP request to the server that hosts the page.

HTTP Headers

In each HTTP request, several headers and their values are passed along to the server to ensure that the browser and server can communicate properly.

HTTP Headers

Some examples of HTTP headers and what info they provide to the web server:

`Host` ← the website being contacted
e.g., natas4.overthewire.org

`User-Agent` ← the type of browser that is making the request
e.g., Chrome/0.2

`Accept` ← the type of data that should be sent in response
e.g., */* (any type of data)

HTTP Headers

Please keep in mind that because HTTP headers can be modified by the user before being sent, that means that the values of any HTTP headers could be spoofed (falsified), although default web browser behavior doesn't allow this.

HTTP Cookie Header

A extremely common HTTP header is the Cookie header, which is used to retain user settings or establish / maintain a user session on a website.

HTTP Cookie Header

For example, a website has a button on its user preferences page which sets the webpage background color for the website.

HTTP Cookie Header

Once the color is selected, the web server will send a Cookie to the web browser to be used anytime the website is visited, changing the webpage's background colors to whatever is specified in the Cookie.

HTTP Cookie Header

Similarly, when a user successfully logs into a website, the web server will send the web browser a Cookie that identifies which user session is being used, and the browser will use that Cookie each time that website is accessed.

HTTP Cookie Header

Any Cookie that is used for user sessions has the potential for security abuse, so it important that the Cookie values created for user sessions are not predictable at all.

PicoCTF – Power Cookie

We can practice inspecting cookies by working through a challenge on PicoCTF. Navigate to the following URL

https://play.picoctf.org/practice/challenge/288

PicoCTF – Cookie Monster Secret Recipe

We can reinforce our knowledge of cookies by working through a challenge on PicoCTF. Navigate to the following URL

https://play.picoctf.org/practice/challenge/469

Part 5 – Digital Forensics

What is Digital Forensics?

According to Wikipedia, digital forensics is a branch of forensic science encompassing the recovery, investigation, examination and analysis of material found in digital devices, often in relation to mobile devices and computer crime.

What is Digital Forensics?

The study of digital forensics (sometimes called dfir, or forensics) in cybersecurity often deals with locating and analyzing important data contained within specific types of files, entire filesystems, or within computer memory storage

What is Digital Forensics?

In our workshops, we'll be learning how to conduct forensics on different file types, including digital picture files, system memory files, and disk image files

Digital Image Files

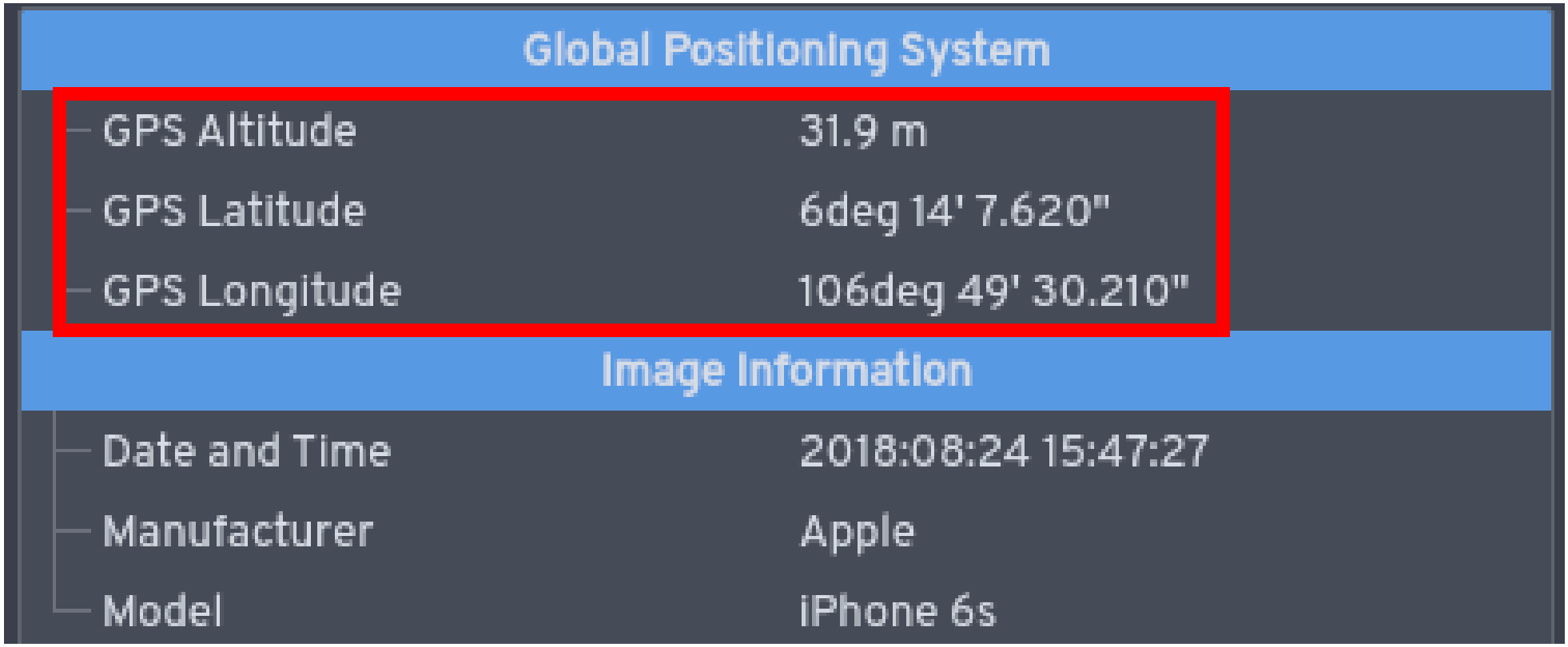
The topic of this workshop is image file forensics, i.e., digital pictures. There are a few different ways information can be hidden in these types of files.

EXIF and Image File Metadata

Metadata is data that provides information about other data.
In the context of digital image files, each file has a plethora of metadata information which is generated when the picture is taken, including:

Global Positioning System	
GPS Altitude	31.9 m
GPS Latitude	6deg 14' 7.620"
GPS Longitude	106deg 49' 30.210"
Image Information	
Date and Time	2018:08:24 15:47:27
Manufacturer	Apple
Model	iPhone 6s
Photograph Information	
Aperture	F2.2
Exposure Bias	0 EV
Exposure Mode	Auto
Exposure Program	Auto
Exposure Time	1/874 s
Flash	No, auto
FNumber	F2.2
Focal Length	4.2 mm
ISO Speed Ratings	25
Metering Mode	Multi-segment
Shutter speed	1/874 s
White Balance	Auto

EXIF and Image File Metadata



Where the image was taken (GPS Coordinates)

EXIF and Image File Metadata

When the image was taken (date and time),

EXIF and Image File Metadata

And the type of device used to create the image (manufacturer and model).

EXIF and Image File Metadata

For photos, this metadata is called Exchangeable Image File Format (EXIF). The EXIF standard is associated with digital cameras, smartphones and scanners

Global Positioning System	
GPS Altitude	31.9 m
GPS Latitude	6deg 14' 7.620"
GPS Longitude	106deg 49' 30.210"
Image Information	
Date and Time	2018:08:24 15:47:27
Manufacturer	Apple
Model	iPhone 6s
Photograph Information	
Aperture	F2.2
Exposure Bias	0 EV
Exposure Mode	Auto
Exposure Program	Auto
Exposure Time	1/874 s
Flash	No, auto
FNumber	F2.2
Focal Length	4.2 mm
ISO Speed Ratings	25
Metering Mode	Multi-segment
Shutter speed	1/874 s
White Balance	Auto

EXIF and Image File Metadata

While EXIF data is useful for archival purposes, this metadata poses a serious security risk when image files are shared, especially when uploaded to the internet

EXIF and Image File Metadata

The most obvious example of risk is the abuse of EXIF data which could allow third-parties to discover a person's residence or neighborhood through through GPS metadata

Part 1 – Picture Metadata With Pico Information

Let's learn more about image metadata, and how to access it by solving the **Information** challenge in Pico CTF:

https://play.picoctf.org/practice/challenge/186

Base64 Encoding

```
[root@localhost ~]# echo -n 'this will be encoded into base64' | base64
dGhpcyB3aWxsIGJlIGVuY29kZWQgaW50byBiYXNlNjQ=
```

Base64 is a widely-used encoding method which is commonly used to transform file and picture data for transmission over computer networks

Base64 Encoding

```
[root@localhost ~]# echo -n 'this will be encoded into base64' | base64
dGhpcyB3aWxsIGJlIGVuY29kZWQgaW50byBiYXNlNjQ=
```

It transforms text and / or data bytes into strings like the one illustrated above. Base64 also features in a lot of CTF challenges, and is used to disguise information

Base64 Encoding

```
[root@localhost ~]# echo -n 'this will be encoded into base64' | base64
dGhpcyB3aWxsIGJlIGVuY29kZWQgaW50byBiYXNlNjQ=
```

In the example above, the text 'this will be encoded into base64' is transformed into the string underlined in red.

Base64 Encoding

Base64 encoded strings can be identified as a string of characters composed of any of the characters located in the encoding table seen here.

Base64 Encoding Table

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Base64 Encoding

```
[root@localhost ~]# echo -n 'this will be encoded into base64' | base64
dGhpcyB3aWxsIGJlIGVuY29kZWQgaW50byBiYXNlNjQ=
```

The encoded strings can also contain the equals (=) sign if the encoded string character length is not evenly divisible by 4

Base64 Encoding

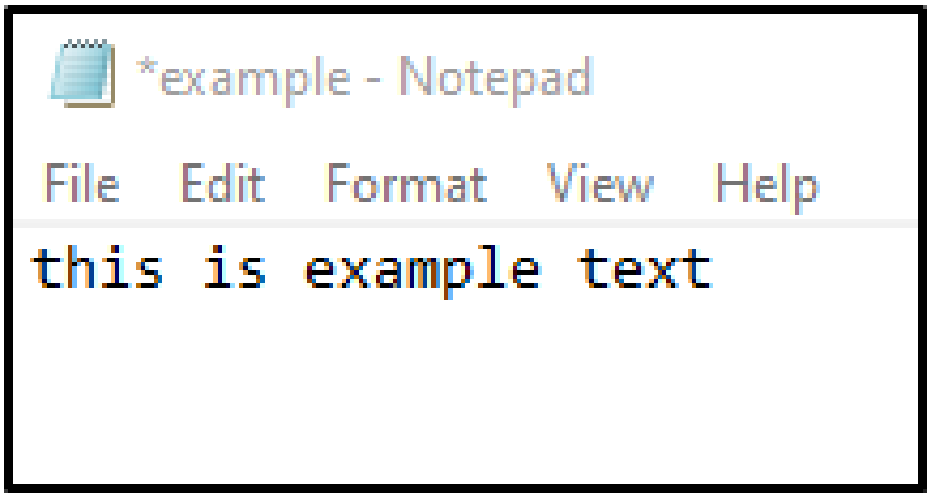
```
[root@localhost ~]# echo -n 'this will be encoded into base64' | base64
dGhpcyB3aWxsIGJlIGVuY29kZWQgaW50byBiYXNlNjQ=
```

In the underlined string, the character length is 43 without the equals character, so one is added so that the length equals 44, a length that divides evenly by 4.

The Contents of Picture Files

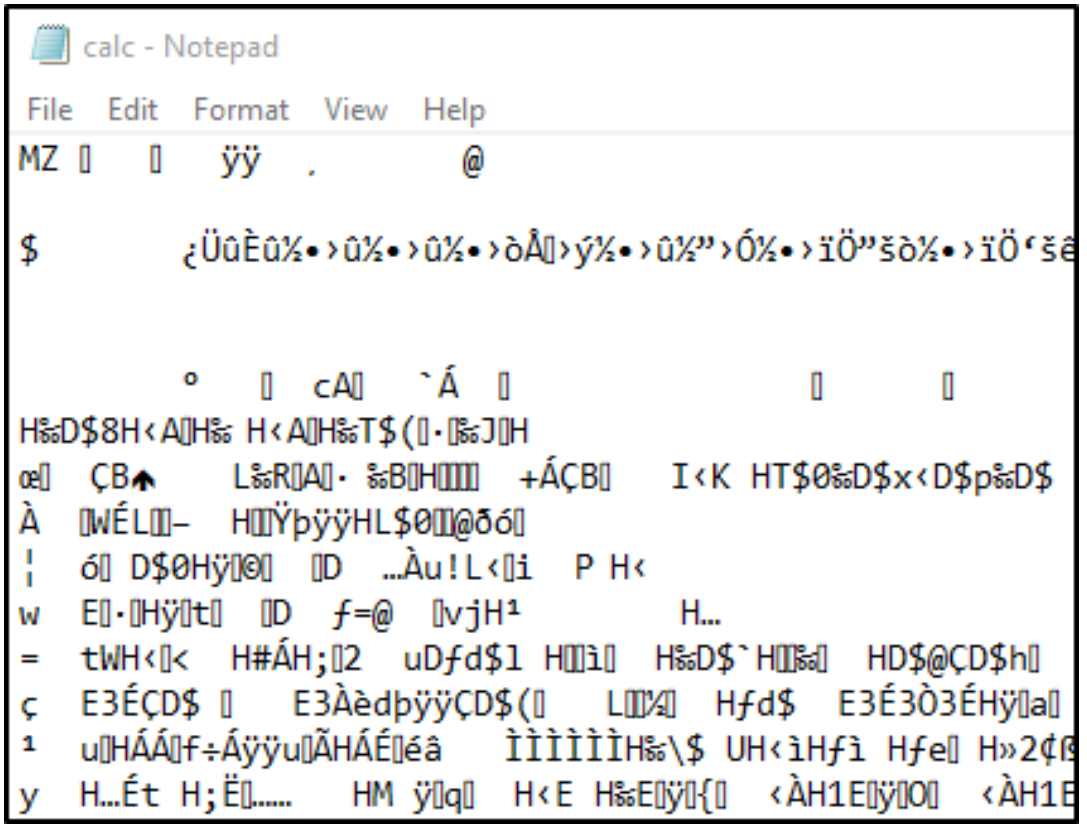
At an abstract level, files contain either human-readable **text** (letters, numbers, and symbols), and / or non-human-readable **data** (program code / instructions, etc), also called **binary data**

The Contents of Picture Files



This is an example of text file contents. It is intended to be read by users (people)

The Contents of Picture Files



This is an example of data file contents. It is not meant to be read by people, rather read and executed by machines (computers / software)

The Contents of Image Files

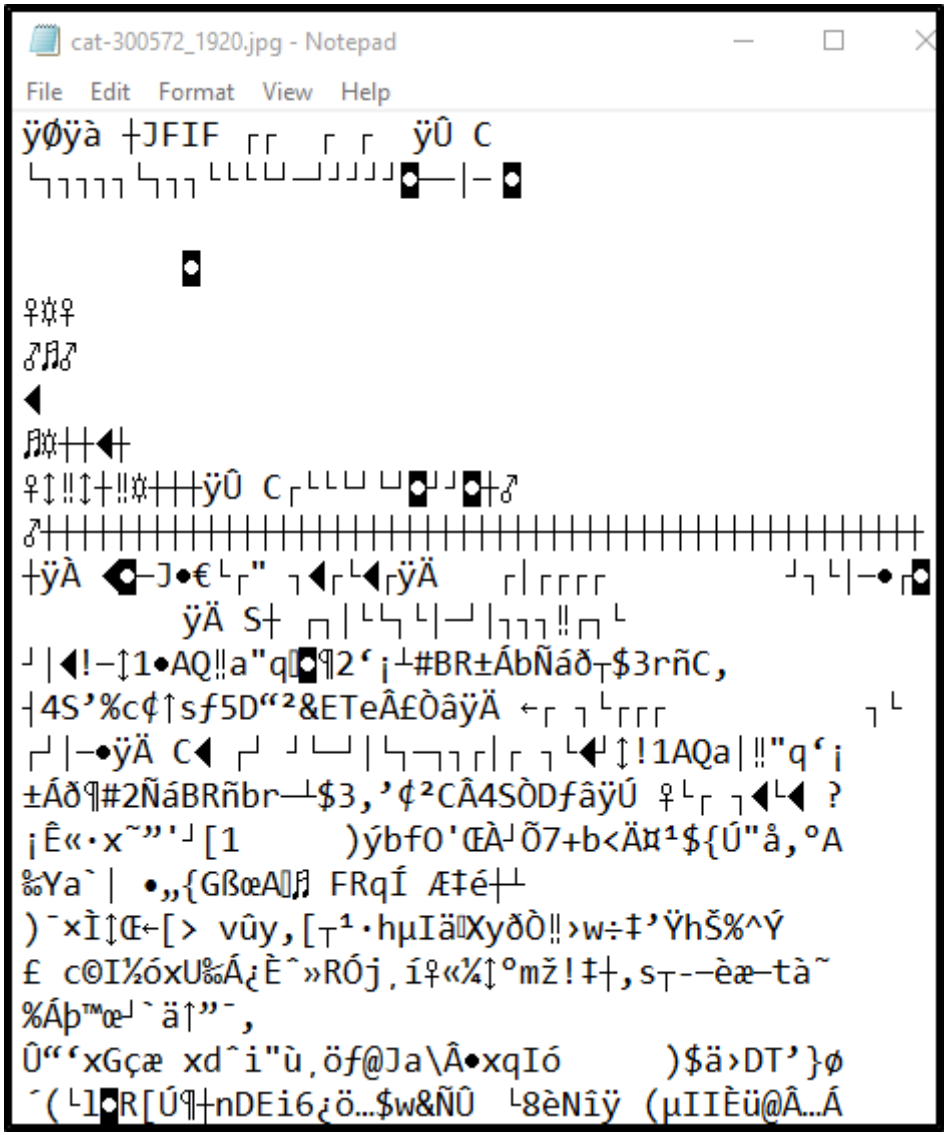
In digital picture files, the majority of their contents are non-human-readable data that is intended to be ready by an image viewing program or web browser

The Contents of Image Files

When opened with one of these programs, the file's data is read and the image appears on the screen. However, as we learned earlier, there are often text strings present inside image files

The Contents of Image Files

However, if image files are opened with a program that isn't designed to read image files, then the contents of the file will appear to be gibberish



Part 2 – Extracting Text With Pico Glory of the Garden

Let's learn more about text versus data, and how to extract text by solving the **Glory of the Garden** challenge in Pico CTF:

https://play.picoctf.org/practice/challenge/44

The Strings Command

The Strings command is used to return human-readable text (strings) from files. It's used to return text from files that contain both data bytes and text (e.g., picture files)

The Strings Command

```
theshyhat-picoctf@webshell:~/tmp$ strings garden.jpg | head
JFIF
XICC_PROFILE
HLino
mntrRGB XYZ
acspMSFT
```

If run by itself, it returns all strings, which often isn't useful, because a lot of random bytes in files can also be interpreted as ASCII characters

The Strings Command

```
theshyhat-picoctf@webshell:~/tmp$ strings -16 garden.jpg
Copyright (c) 1998 Hewlett-Packard Company
sRGB IEC61966-2.1
sRGB IEC61966-2.1
IEC http://www.iec.ch
IEC http://www.iec.ch
.IEC 61966-2.1 Default RGB colour space - sRGB
```

But if we run the command with a number argument, we can specify the minimum-length string to display in the output

What is Digital Disk Forensics?

Digital disk forensics is the examination and analysis of information stored on digital disks, such as hard drives (HDD), USB drives, or any other type of storage media.

Digital Disk Forensics

Disk forensics is often used in cybersecurity incident response to analyze and identify devices that may have been compromised in security incidents.

The Sleuthkit Software

The Sleuth Kit is a popular program used in digital disk forensics, and we can access it from the PicoCTF webshell

The Sleuthkit Software

We'll be learning some basic operations of The Sleuth Kit in this workshop to learn digital disk forensics

The MmLs Command

```
theshyhat-picoctf@webshell:/tmp/...theshyhat$ mmls disk.img
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot      Start        End          Length       Description
000:  Meta     0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------  0000000000   0000002047   0000002048   Unallocated
002:  000:000  0000002048   0000204799   0000202752   Linux (0x83)
```

The MmLs command displays the media management (Mm) of a disk image file in list (Ls) format

The MmLs Command

```
theshyhat-picoctf@webshell:/tmp/...theshyhat$ mmls disk.img
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot      Start        End          Length       Description
000:  Meta     0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------  0000000000   0000002047   0000002048   Unallocated
002:  000:000  0000002048   0000204799   0000202752   Linux (0x83)
```

We can see where the partition starts (offset in bytes), where the partition ends, and the length (size in bytes) of the partition

PicoCTF: Sleuthkit Apprentice

Let's learn more Sleuthkit commands with this PicoCTF challenge

https://play.picoctf.org/practice/challenge/300?page=1&search=sleuth

The FsStat Command

```
theshyhat-picoctf@webshell:/tmp/...theshyhat$ fsstat -o 2048 disk.flag.img
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: Ext4
Volume Name:
Volume ID: 8e023955b4e7dab7e04b7643076ccf0f
```

The FsStat command is used to display statistics (Stat) associated with a filesystem (Fs)

The FsStat Command

```
theshyhat-picoctf@webshell:/tmp/...theshyhat$ fsstat -o 2048 disk.flag.img
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: Ext4
Volume Name:
Volume ID: 8e023955b4e7dab7e04b7643076ccf0f
```

To use this command, we'll need to supply the byte offset of the disk partition (-o), and the name of the disk image

The Fls Command

```
theshyhat-picoctf@webshell:/tmp/...theshyhat$ fls -f ext4 -o 2048 -r disk.flag.img
d/d 11: lost+found
r/r 12: ldlinux.sys
```

The Fls (Filesystem Ls) command is used to list out files and directories within a specified filesystem

The Fls Command

```
theshyhat-picoctf@webshell:/tmp/...theshyhat$ fls -f ext4 -o 2048 -r disk.flag.img
d/d 11: lost+found
r/r 12: ldlinux.sys
```

To run this command, we need the format of the disk partition (-f), the offset of the disk partition (-o), run it recursively (-r), and supply the disk image name

The Icat Command

```
theshyhat-picoctf@webshell:/tmp/...theshyhat$ icat -f ext4 -o 360448 disk.flag.img 2371
picoCTF{[redacted]}
theshyhat-picoctf@webshell:/tmp/...theshyhat$
```

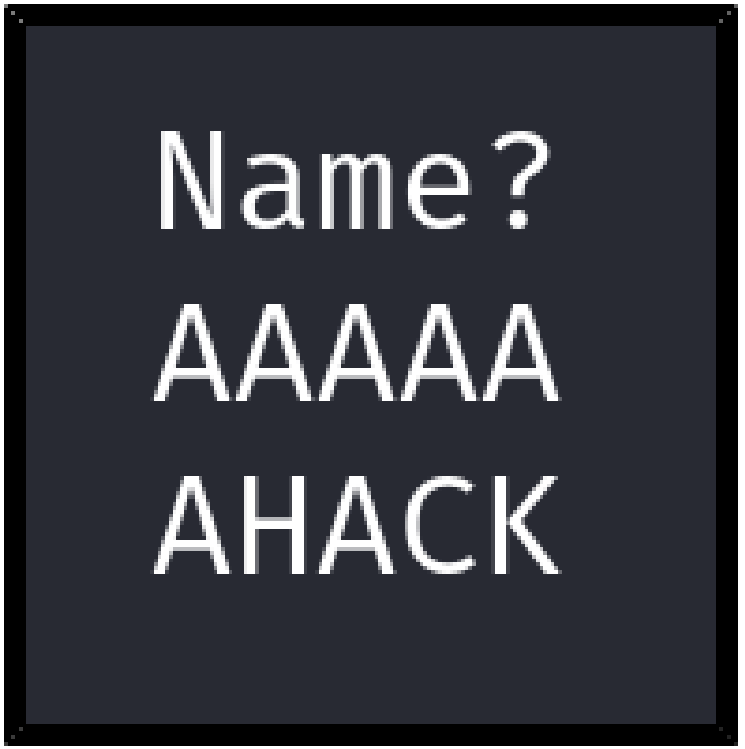
The Icat (inode cat) command is used to read specific files in a disk partition according to its inode number

The Icat Command

```
theshyhat-picoctf@webshell:/tmp/...theshyhat$ icat -f ext4 -o 360448 disk.flag.img 2371
picoCTF{[illegible]}
theshyhat-picoctf@webshell:/tmp/...theshyhat$
```

To use the command, we need to supply the format of the disk partition (-f), the offset of the disk partition (-o), and finally, the name of the disk image and the inode number (disk.flag.img 2371)

Part 6 - Binary Hacking



What is Binary Hacking?

Binary Hacking is the interaction with a compiled computer program (a.k.a binary executable) which results in an unexpected response that is beneficial to the user interacting with the program

What is Binary Hacking?

In this introductory workshop, we'll be taking a look at a couple of examples of compiled programs that have been coded in a such a way that they can be hacked by anyone who can identify the vulnerability.

Program Logic Flaws

If a program is coded in an insecure way, then it could be abused by users who can identify these flaws

Program Logic Flaws

A typical example of a program logic flaw is any shopping app which allows users to buy negative amounts of an item, which results in an "infinite money glitch"

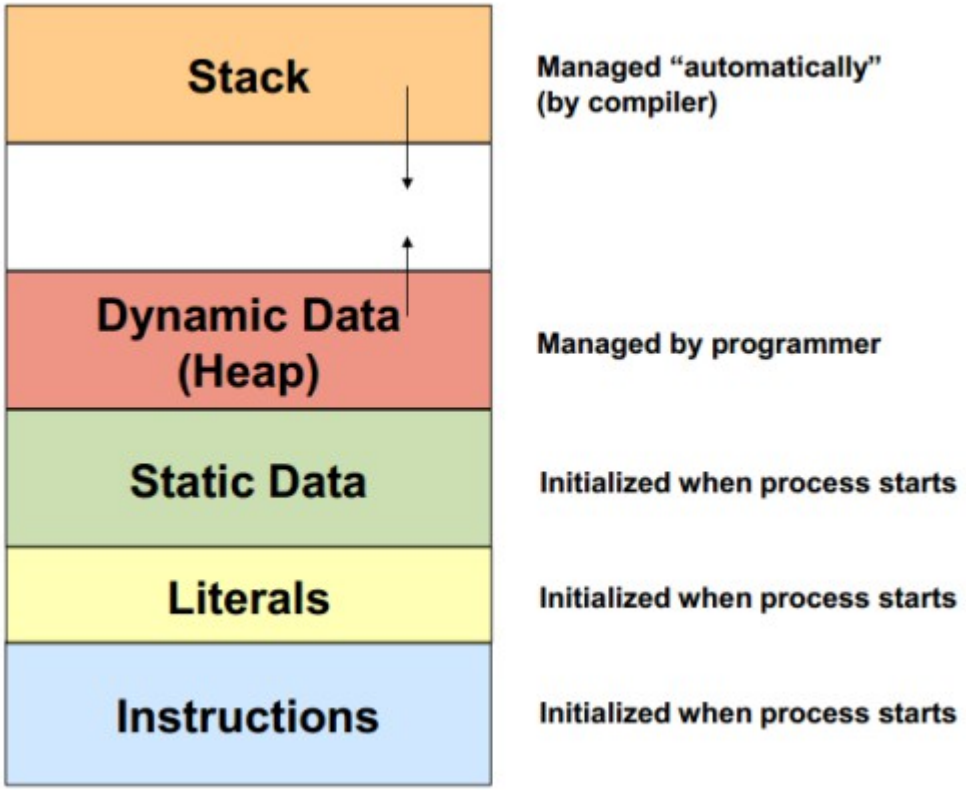
PicoCTF: RPS

Let's learn more program logic flaws with this PicoCTF challenge

https://play.picoctf.org/practice/challenge/293

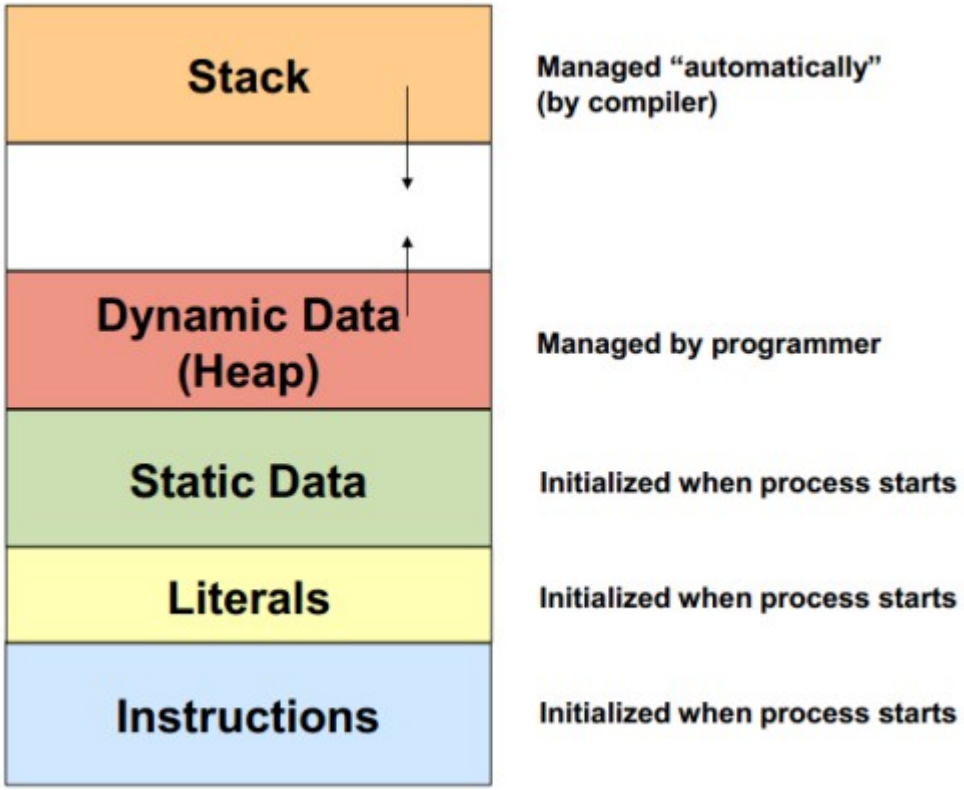
Intro to Memory Hacks

A well-known type of binary hacks involve memory vulnerabilities, which occur when a program is written in an insecure way--



Intro to Memory Exploits

which allows users to inject arbitrary data into a program's memory space, which can result in any or all of the following:



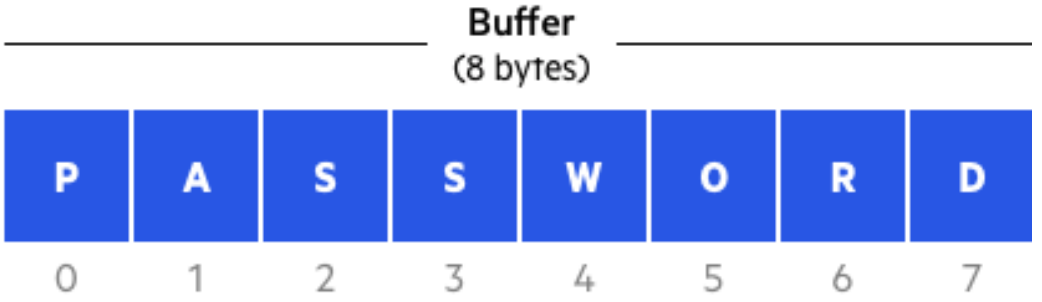
Intro to Memory Hacks

Crash the program

Allow access to other functions in the program

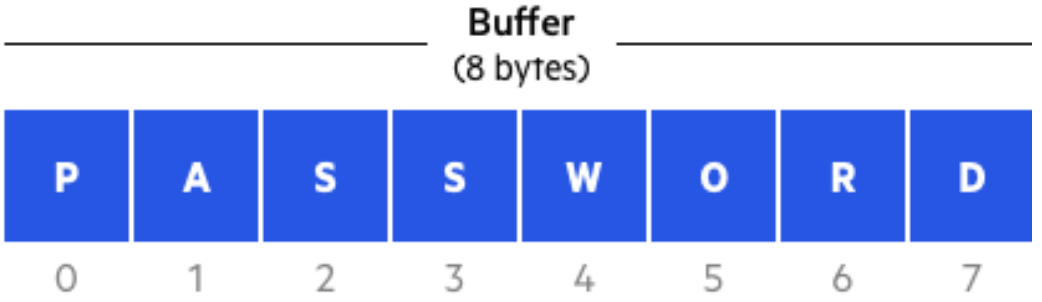
Allow arbitrary command execution on the program's host system

What are Memory Buffers?



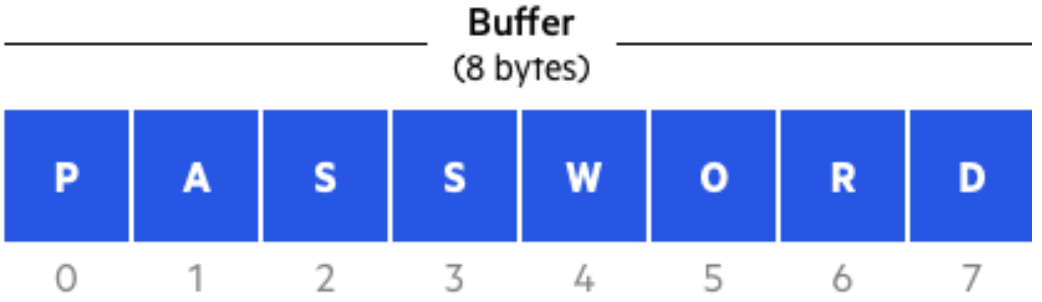
In order to understand memory hacks, we must first understand the concept of memory buffers.

What are Memory Buffers?



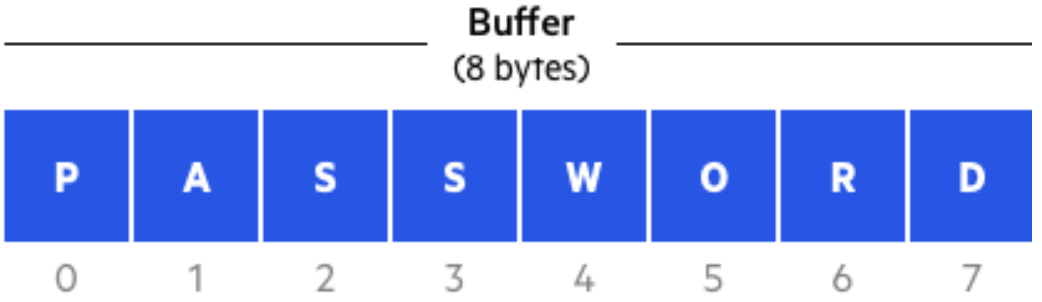
In programming, memory buffers are sections of a program's memory space that are set aside to hold data, e.g., user input.

What are Memory Buffers?



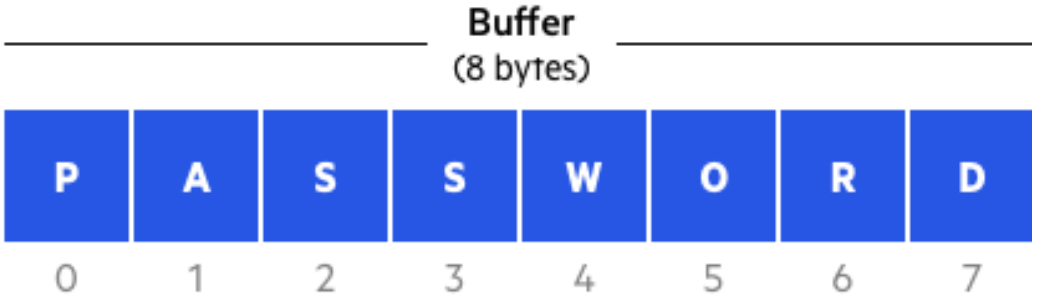
In C programming, the amount of space allocated to each memory buffer must be specified.

What are Memory Buffers?



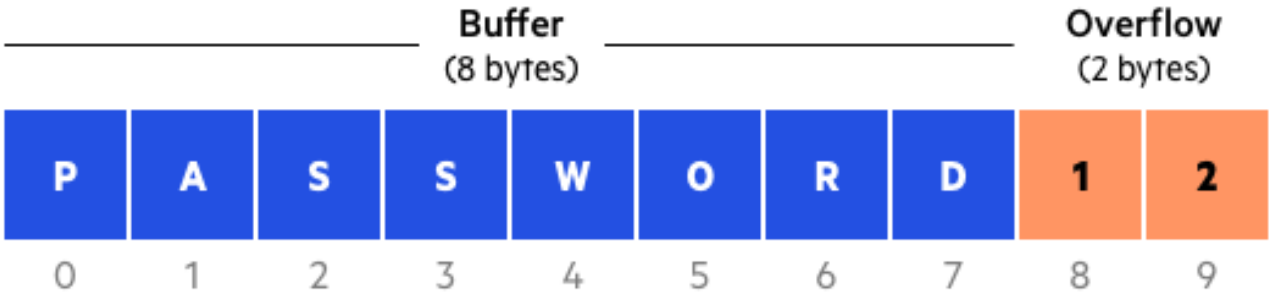
E.g., a buffer meant to store a user's password may be set to 8 bytes (each character in a password requires 1 byte), so the maximum length of that password should be 8 characters.

What are Memory Buffers?



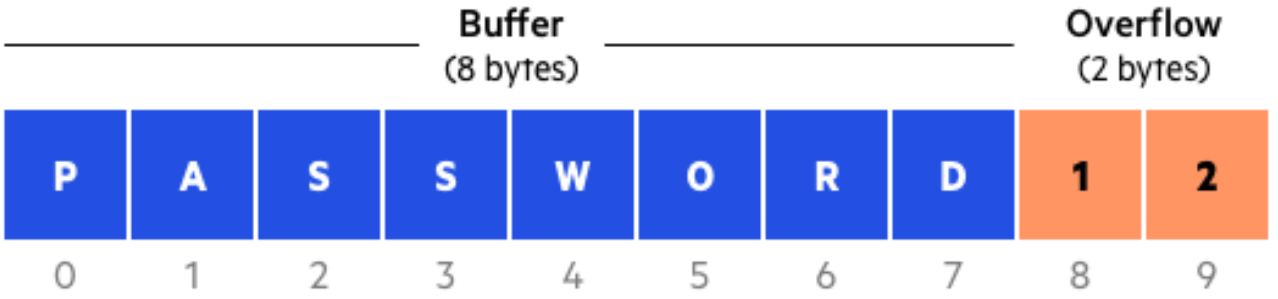
But what happens if the program receives input for that password which exceeds 8 characters?

What are Memory Buffers?



Data in excess of 8 bytes will overflow out of the specified memory buffer into the neighboring memory space, overwriting the data that was there previously.

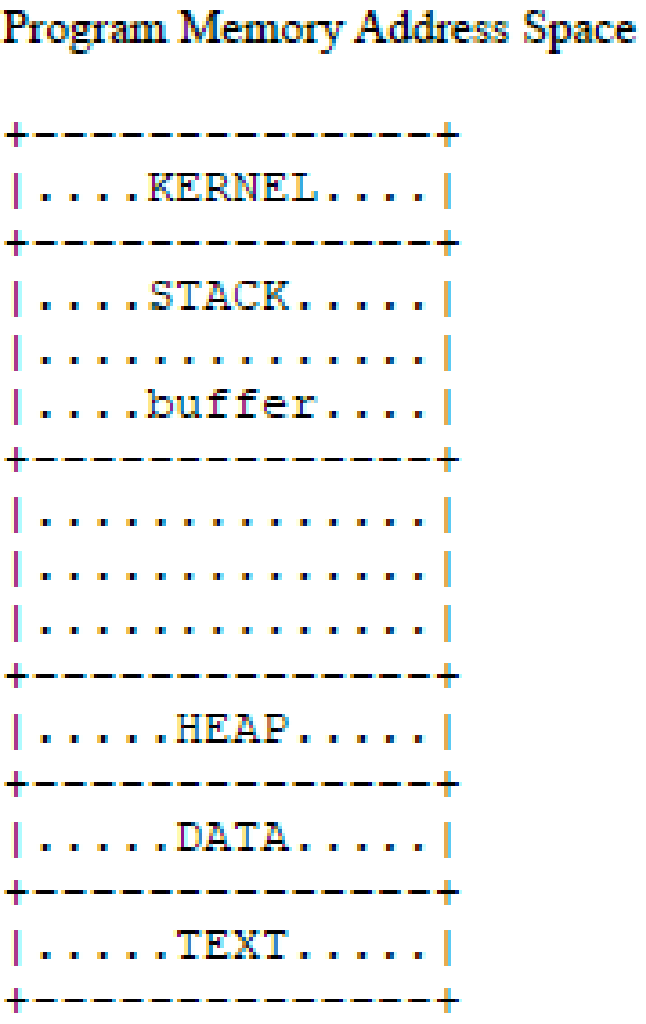
What are Memory Buffers?



In programming, the act of data escaping the memory buffer allocated to it is known as buffer overflow.

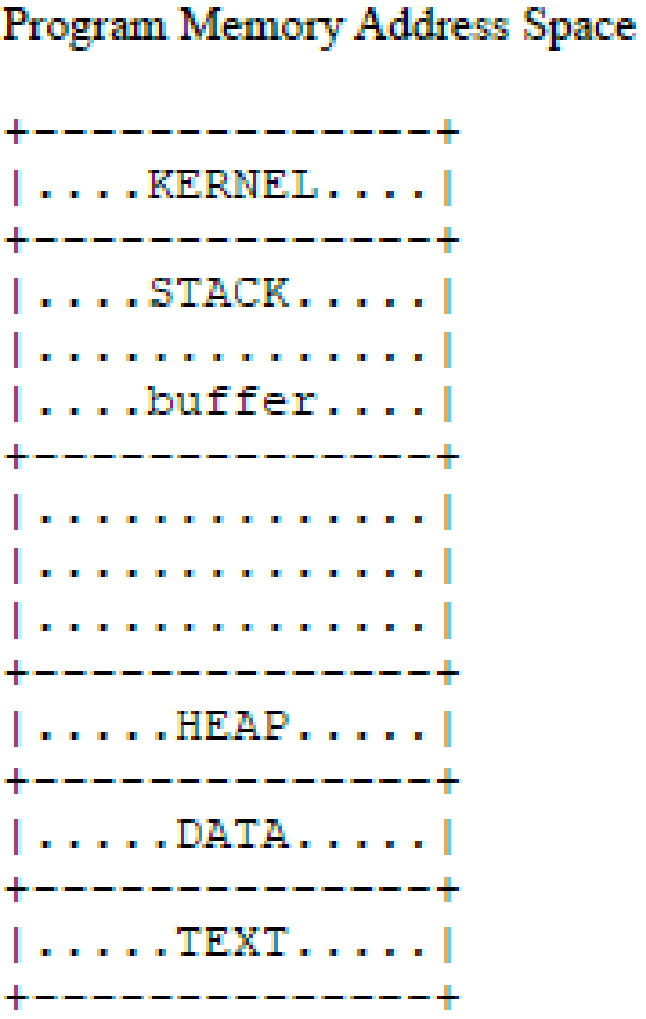
What is Memory Address Space?

In order to study memory exploitation further, we have to familiarize ourselves with program memory address space.



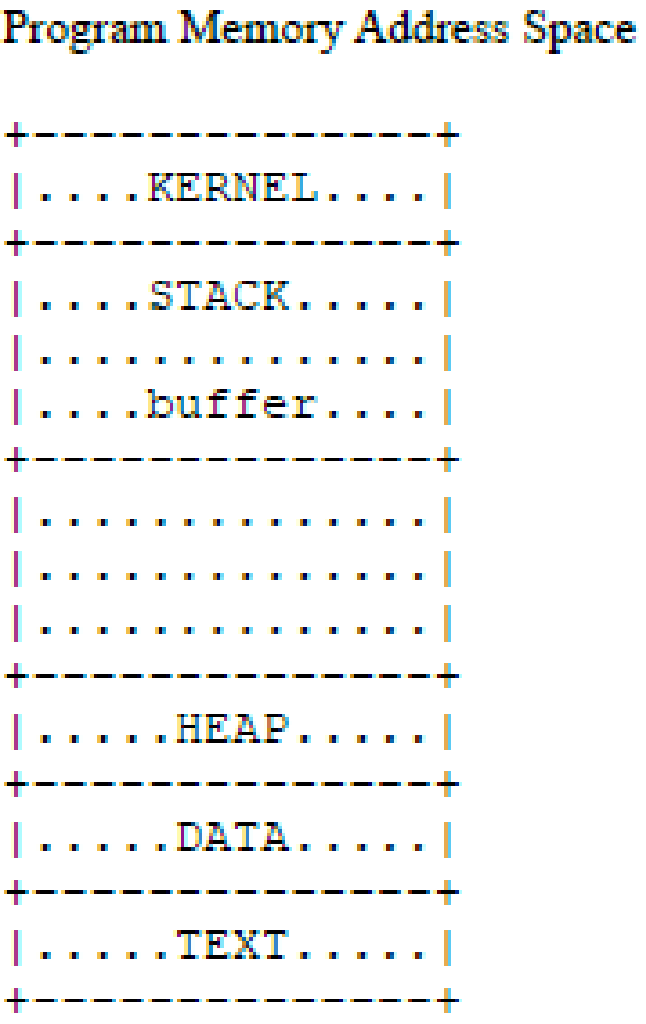
What is Memory Address Space?

When a program launches, a portion of the computer's memory is allocated to it, forming the program's memory space. The data in the memory space is divided into chunks of 4 or 8 bytes, for 32-bit and 64-bit programs, respectively.



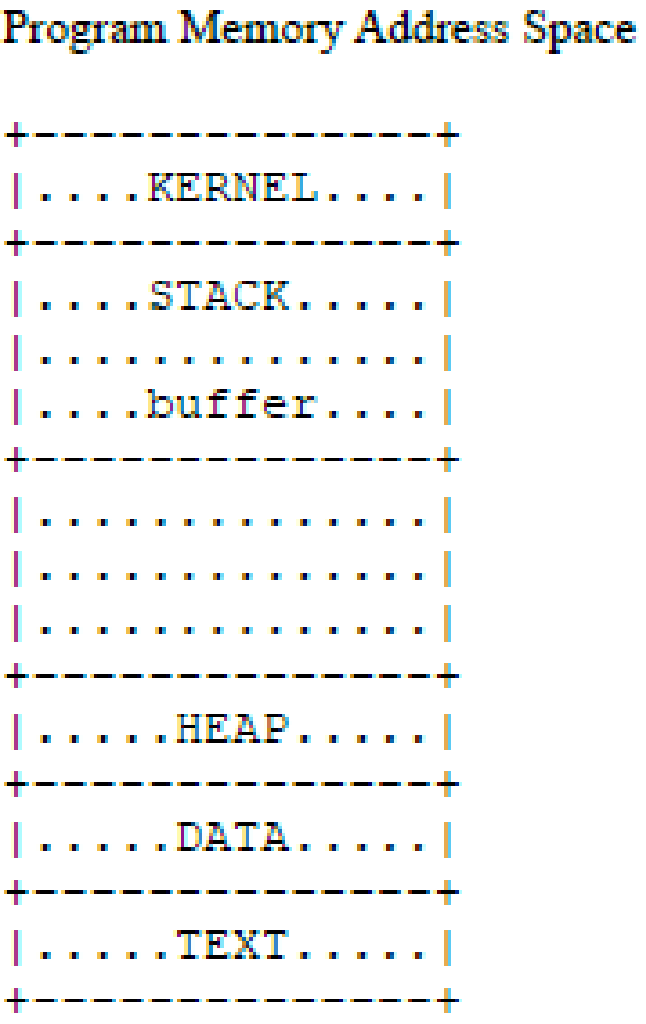
What is Memory Address Space?

In a 32-bit program, the lowest memory address is 00000000 (4 bytes, two characters per byte), or 0x00000000, and the highest address is FFFFFFFF or 0xFFFFFFFF



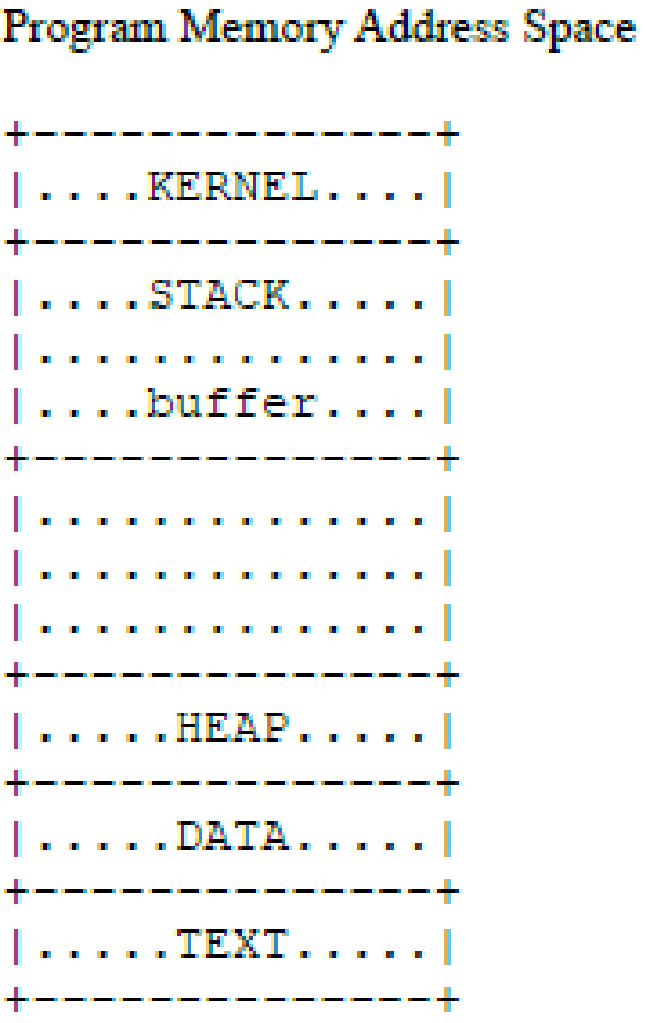
What is Memory Address Space?

The 0x pre-pended to the address indicates hexadecimal notation. There are five major divisions in the program memory space: the kernel, stack, heap, data, and text.



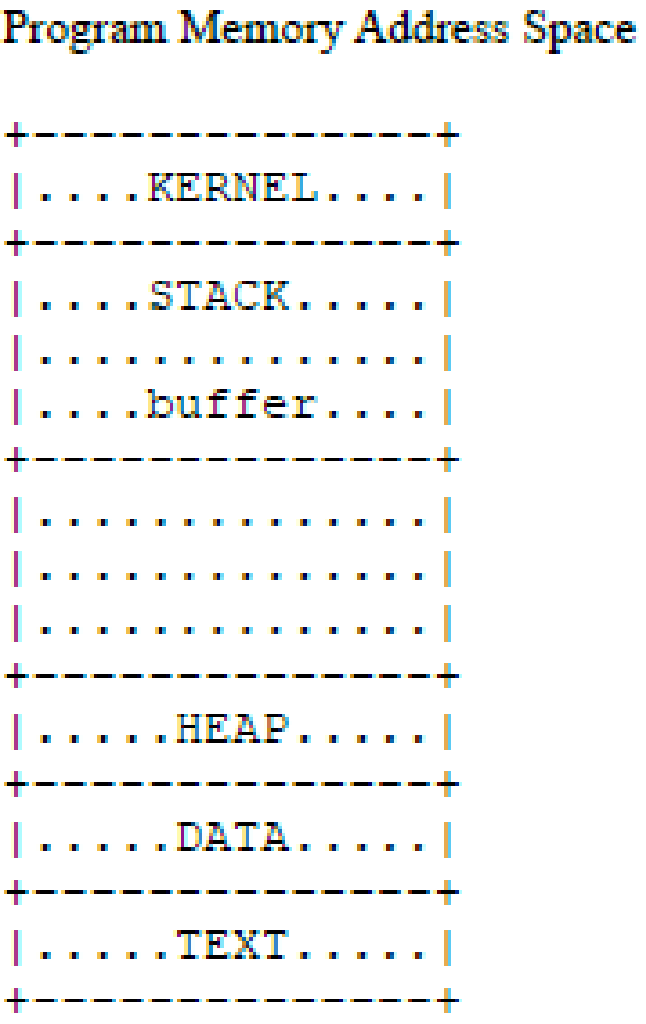
What is Kernel Memory?

The **kernel** section occupies the lowest memory addresses and contains data related to the computer's operating system processes.



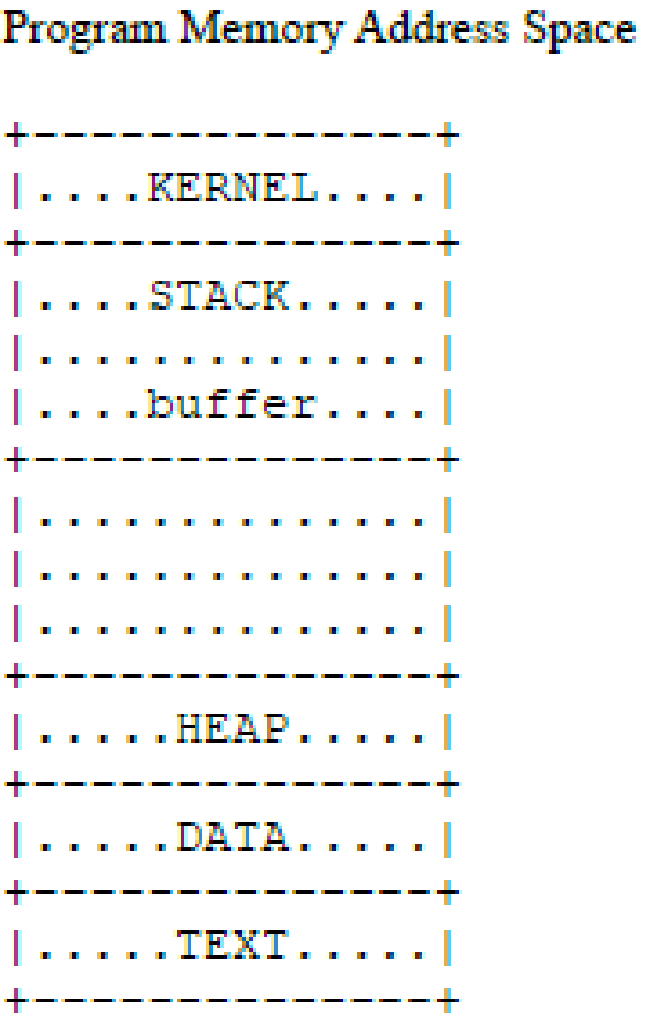
What is Stack Memory?

The **stack** section follows the kernel section, and it contains variables that store program parameters. It is a prime candidate for buffer overflow hacks

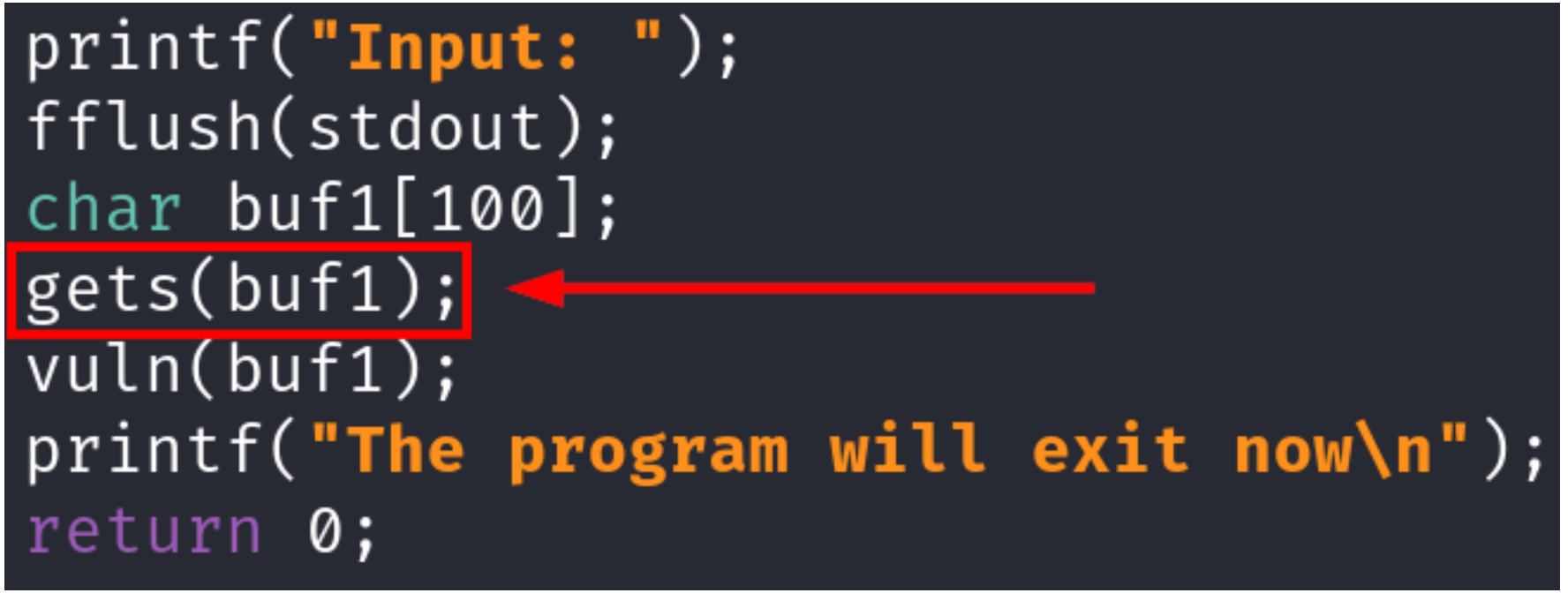


What is Stack Memory?

As data is added to the stack, it progresses to higher memory addresses, but any buffers overflows that occurs on the stack progress towards lower memory addresses.

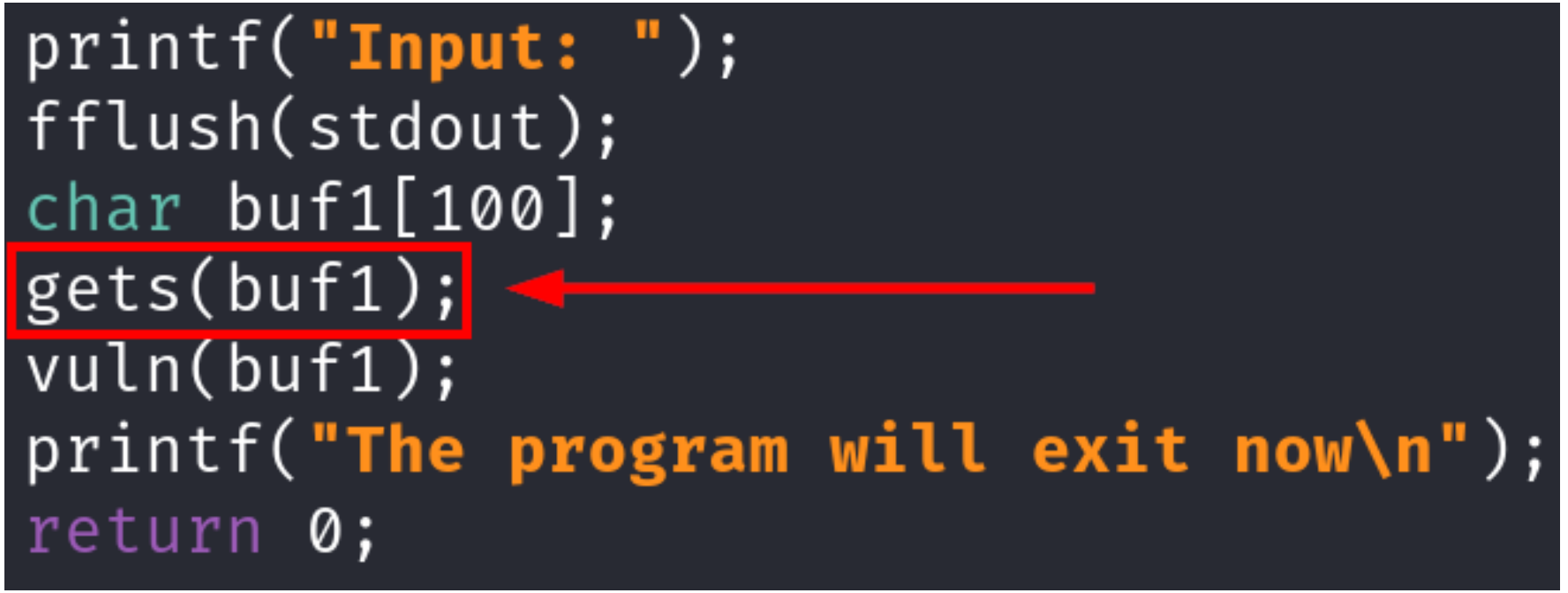


What is the Gets Function?



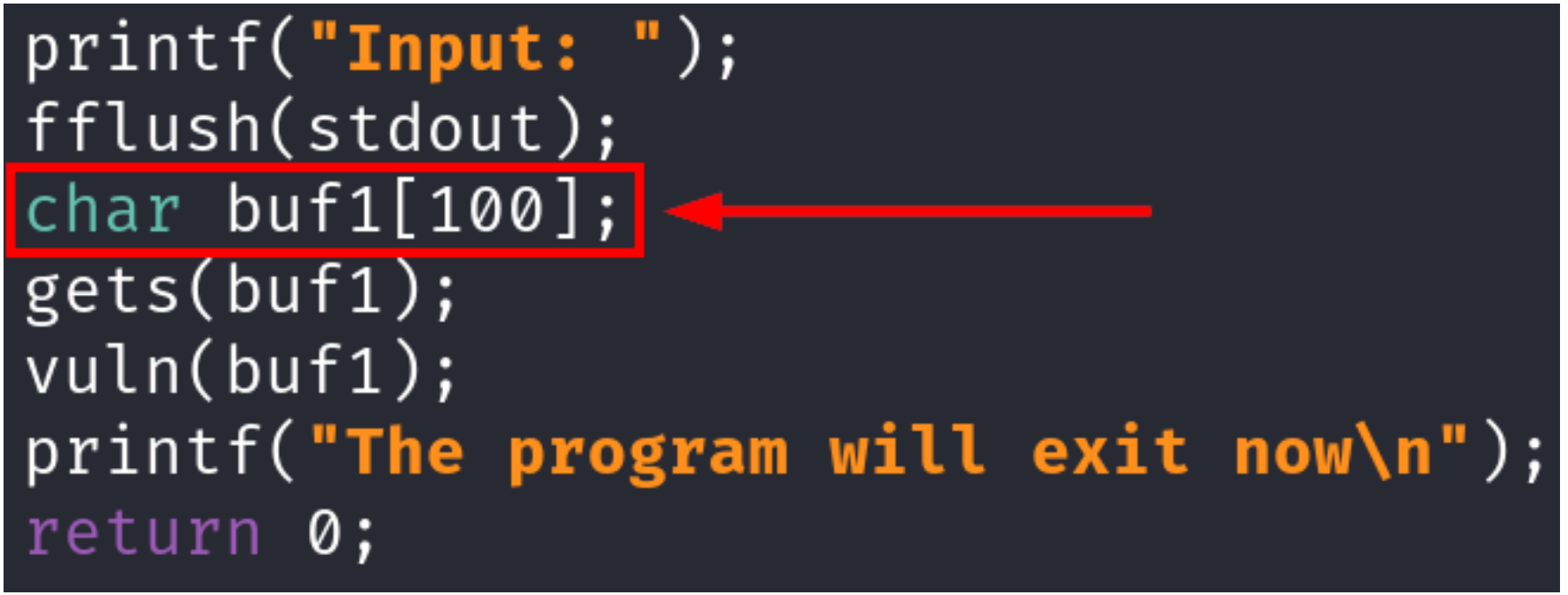
The **gets** function is used to save user input to a variable in C programming

What is the Gets Function?



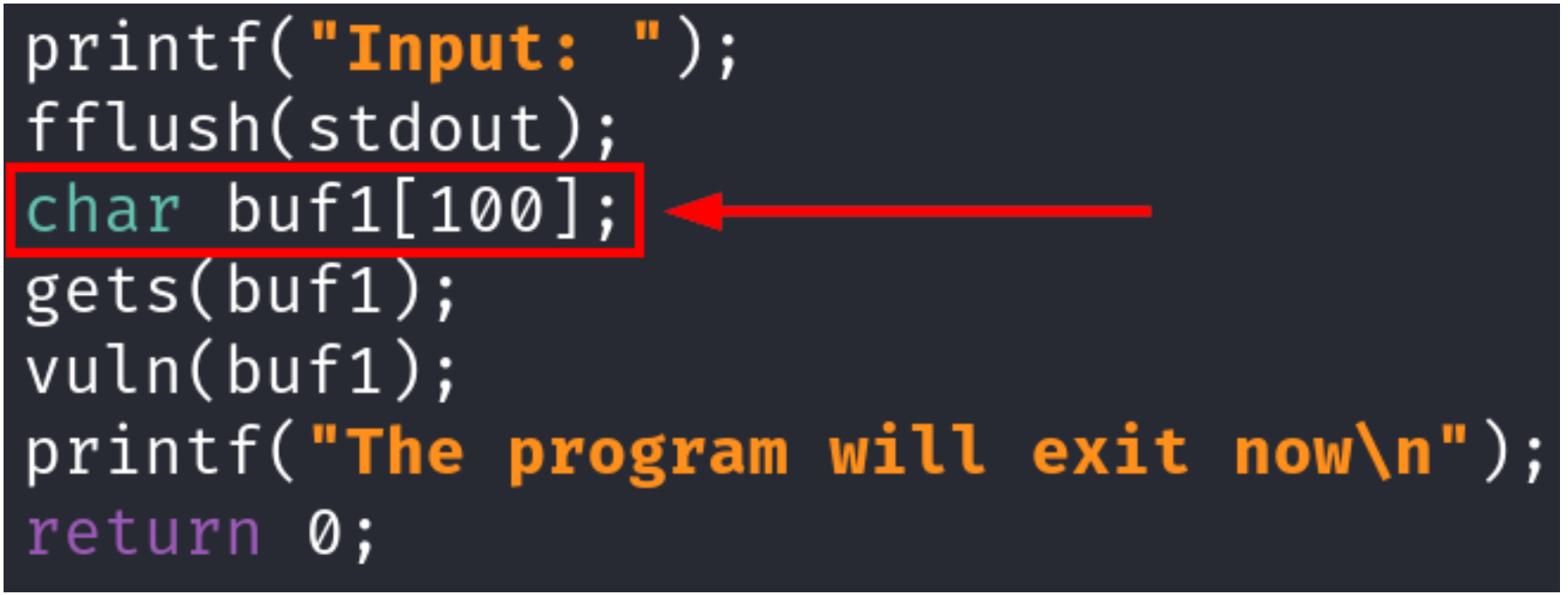
However, the **gets** function doesn't check the size of the user input, which means it can be used for buffer overflow attacks

What is the Gets Function?



If the user inputs a string that is longer than 100 characters, a buffer overflow will occur

What is the Gets Function?



In modern C programming, the Gets function was replaced by a safer function: Fgets

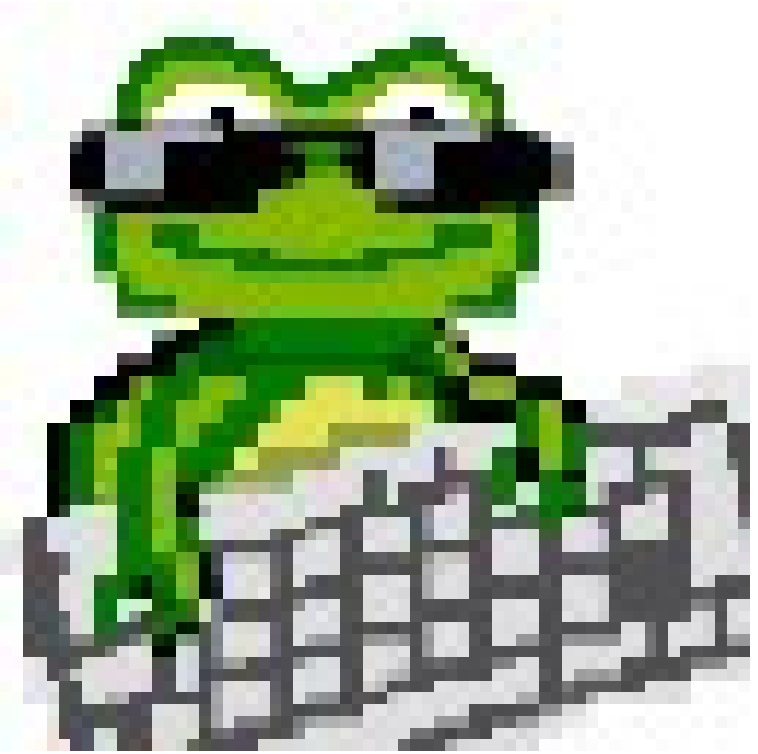
PicoCTF: Clutter Overflow

Let's learn more buffer overflow vulnerability with this PicoCTF challenge

https://play.picoctf.org/practice/challenge/216

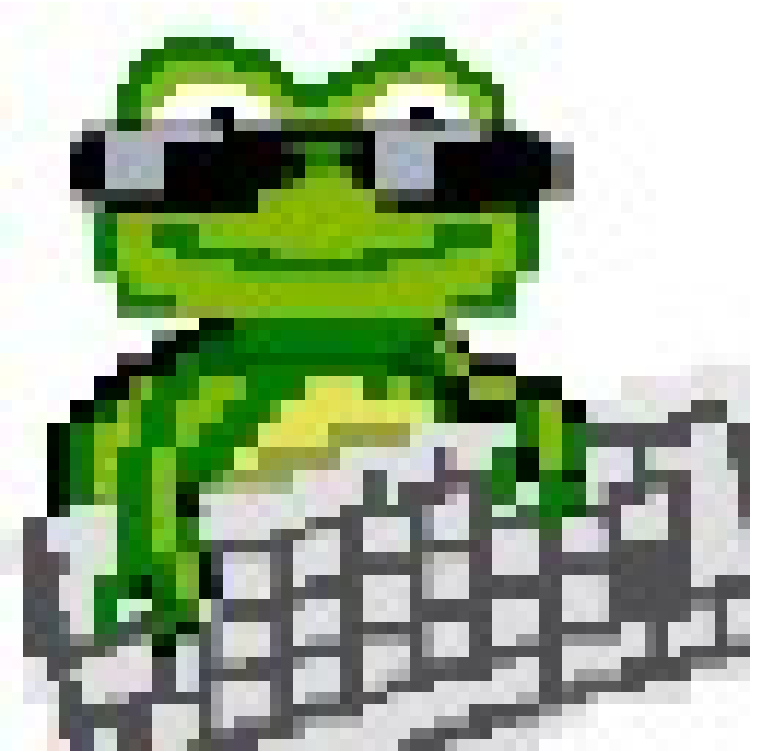
Whoami

I'm Kevin Lee, but online I go by theShyHat, and I do cybersecurity videos and live streams on platforms like YouTube and Twitch

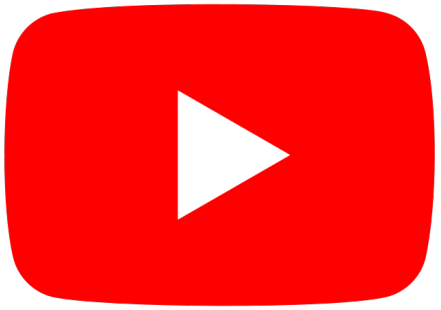


Whoami

I also teach cybersecurity for orgs, youth programs, and other groups looking for training, e.g., Youth cybersecurity summer camps



Links and Q & A



If you want to learn more cybersecurity skills with me across all sorts of topics, you can join me live on stream on Mondays, Wednesdays, and Fridays

If you have any questions or comments, feel free to ask