KHARAGPUR
OPEN SOURCE SOCIETY

CAPTURE THE FLAG!!

CRACK, LEARN, SECURE:
STEP INTO CYBERSECURITY!

Index

Who we are...

We are a group of Open Source Enthusiasts who focus on something more preliminary and relevant, "A Love for Coding".

Every year KOSS holds events for familiarizing students with UNIX tools, Linux environment, Git Development Workflow, besides conducting workshops on GUI programming, Web development, and Open Source Programs.

Why CTFs?

Apart from learning, CTFs are fun! And the fun doubles when you take part in CTFs as a team!

CTFs help you learn how systems operate, identify weaknesses, and think like an attacker. This perspective is key in protecting systems. They also open up career opportunities in digital security, helping you gain hands-on experience in the field.

What are CTFs

kossCTF{babys_first_ctf_!!!}

CTFs are (primarily online) competitions, that are essentially hacking tournaments.

- The idea is that you have to recover a string (the flag) from a (supposedly) secure system.

We will be conducting a very simple CTF alongside this presentation for you to follow along with. Please make your accounts.

Some rules:

- Don't share flags
- Don't attack the website directly
- You are allowed to use any resources available to you, but try not to GPT (or Deepseek). There's no special prize for solving them all
- You will receive plenty of hints on solving these during the presentation

10.147.8.22:8000/

Web Exploitation

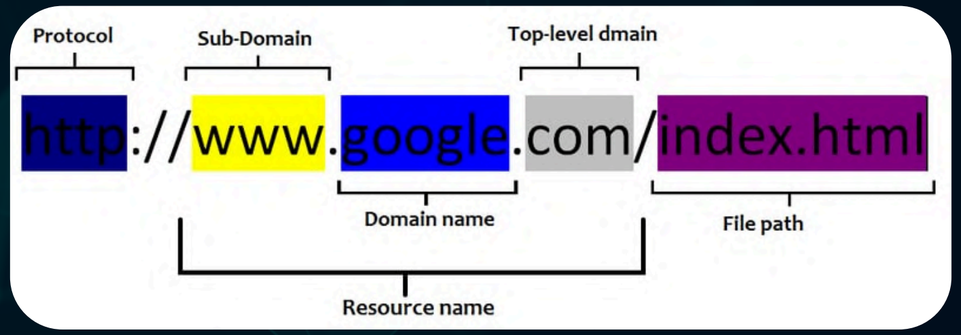
01
What happens at google.com?

02
Browser tools

03
Messing with web servers

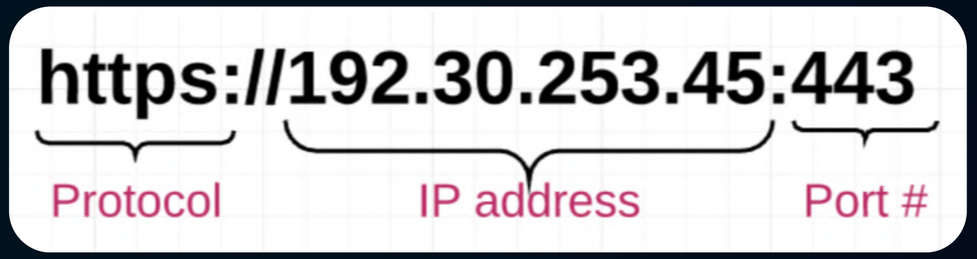
04
Famous Vulnerabilities

What happens when you enter "google.com"



Request

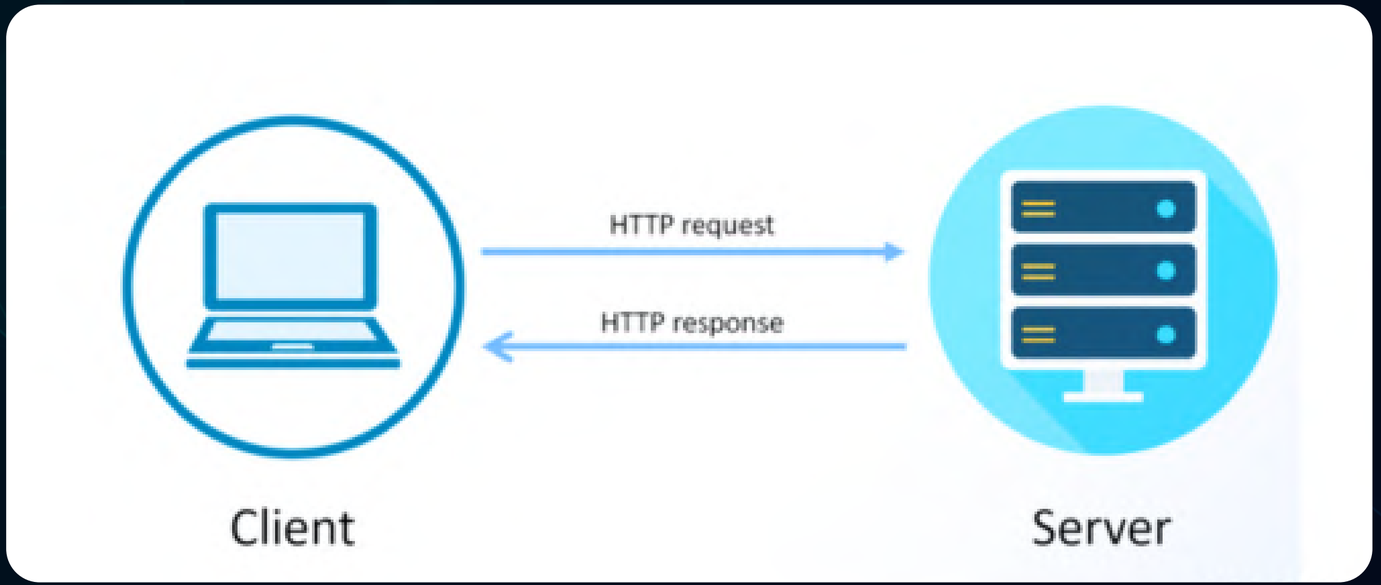
- When you type a URL in the browser, it parses the information contained in the URL and a request is sent to your internet service provider (ISP)



Response

- The server does a quick lookup and returns the IP address of the websites which redirects you to the main pages of the site

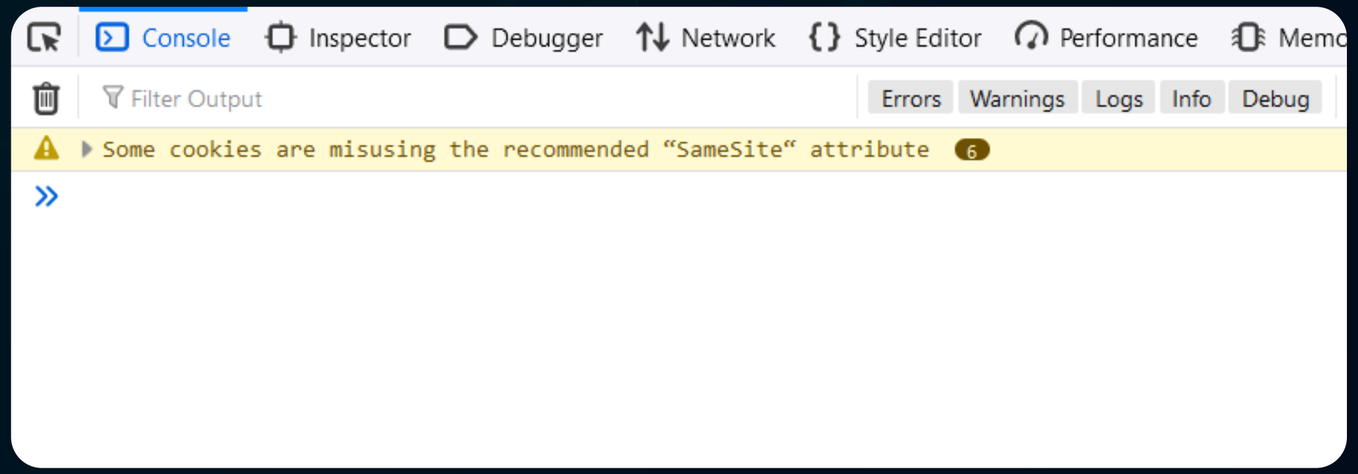
What happens when you enter "google.com"



A request is made by client to the server asking for resources, like a webpage or data. A response is the server's reply, containing the requested data or error message.

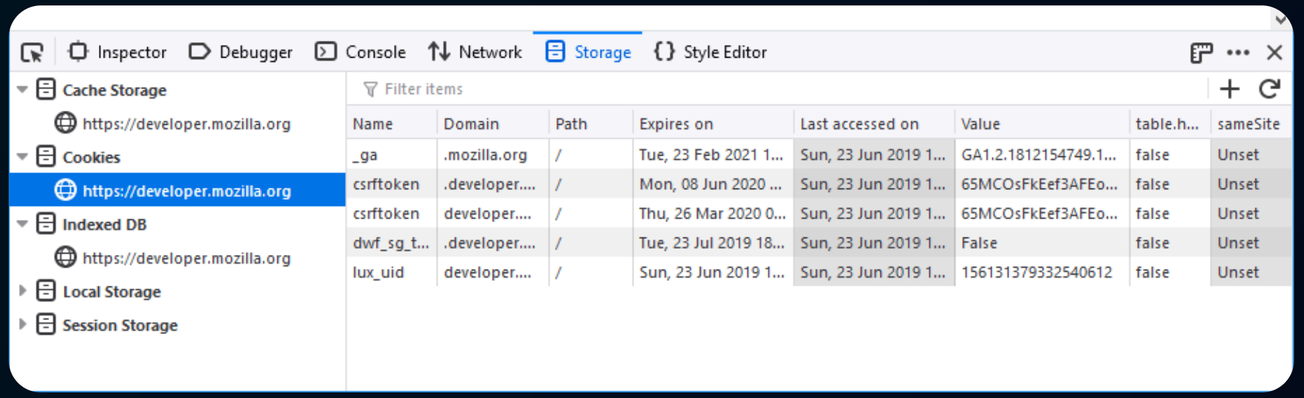
How to use your browser tools?

Web Console



- In the Inspect Console of a browser, it's possible for malicious code to be executed through injections, such as JavaScript.(an example of Cross-site Scripting)

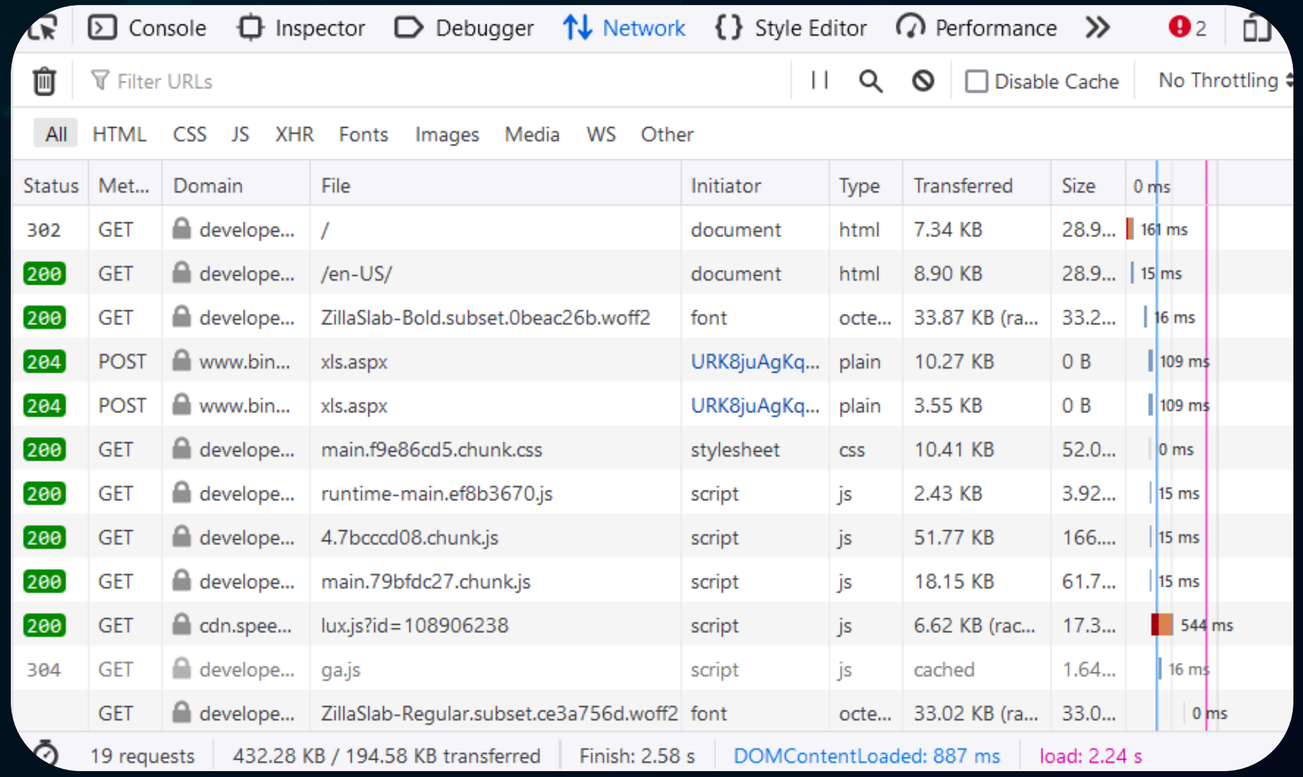
Storage



- Shows the data a website stores in the browser like cookies and session data.

How to use your browser tools?

Network Monitor



- Used to analyze a website's network traffic, to view requests, API calls, and responses.
- Interception of these requests, which is useful for identifying vulnerabilities.

How to mess with web servers?

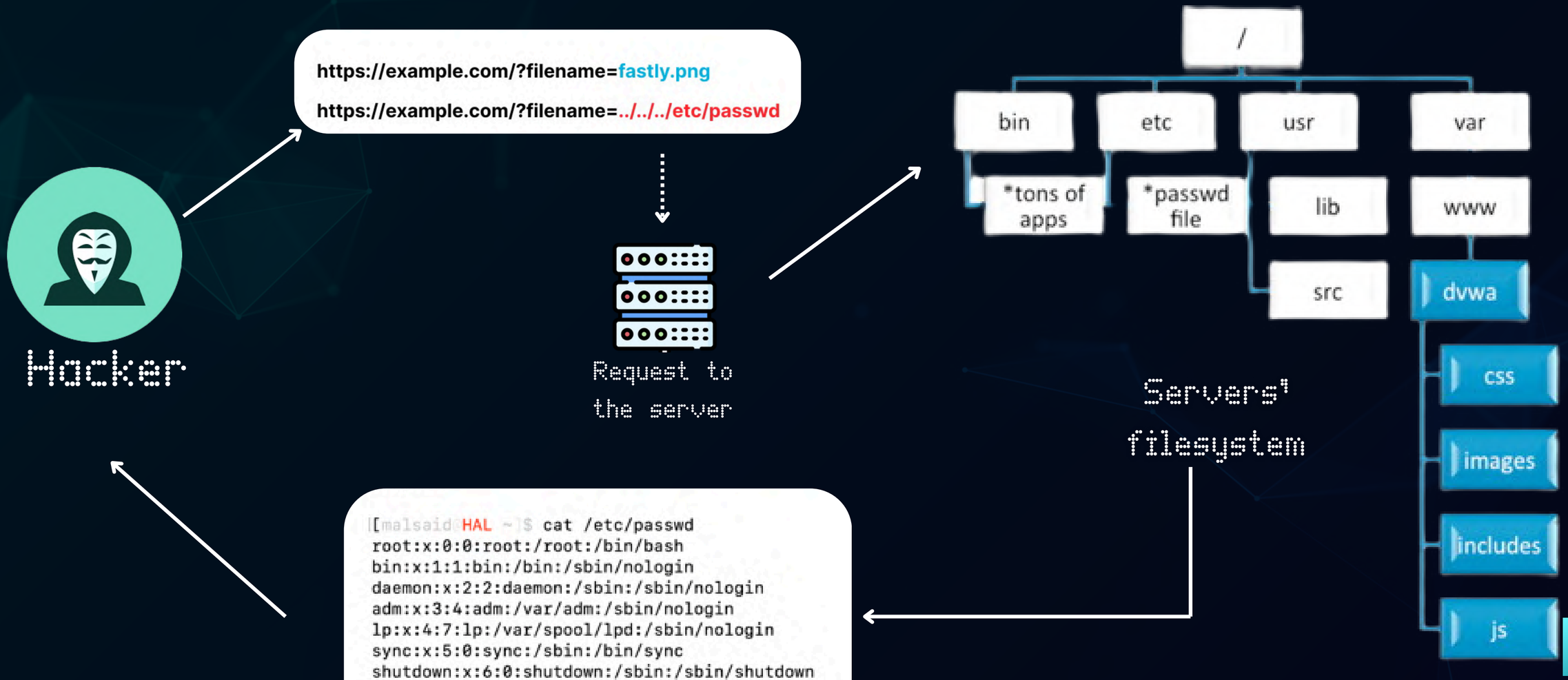
01 Cookie based authentication

02 Path traversal

Cookie-based Authentication

Cookie based authentication can be bypassed if the attacker intercepts or modifies the session cookie. Let's try it out!

Path Traversal

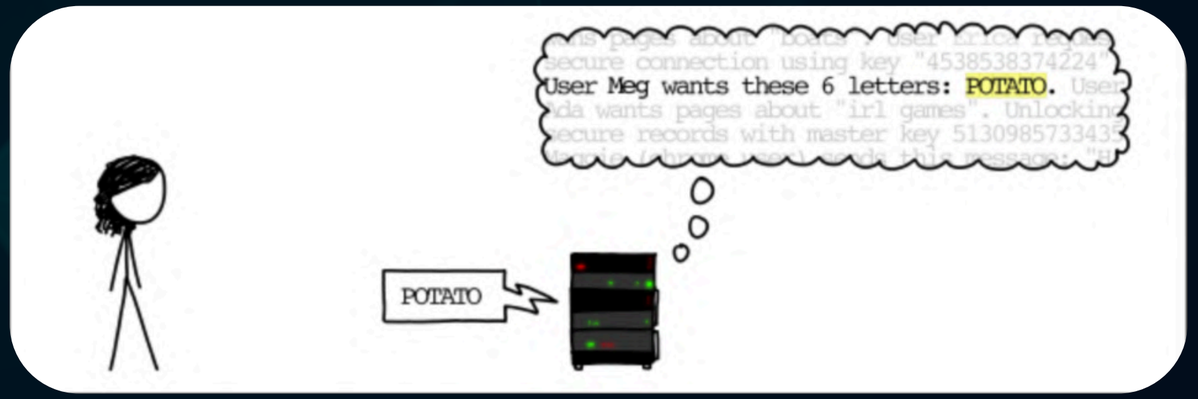


Famous Web Vulnerabilities

01
Heartbleed

02
Large scale
DDoS

Heart Bleed



- This bug allows attackers to perform man-in-the-middle attacks on websites, by taking advantage of a loophole in TLS encryption provided by OpenSSL

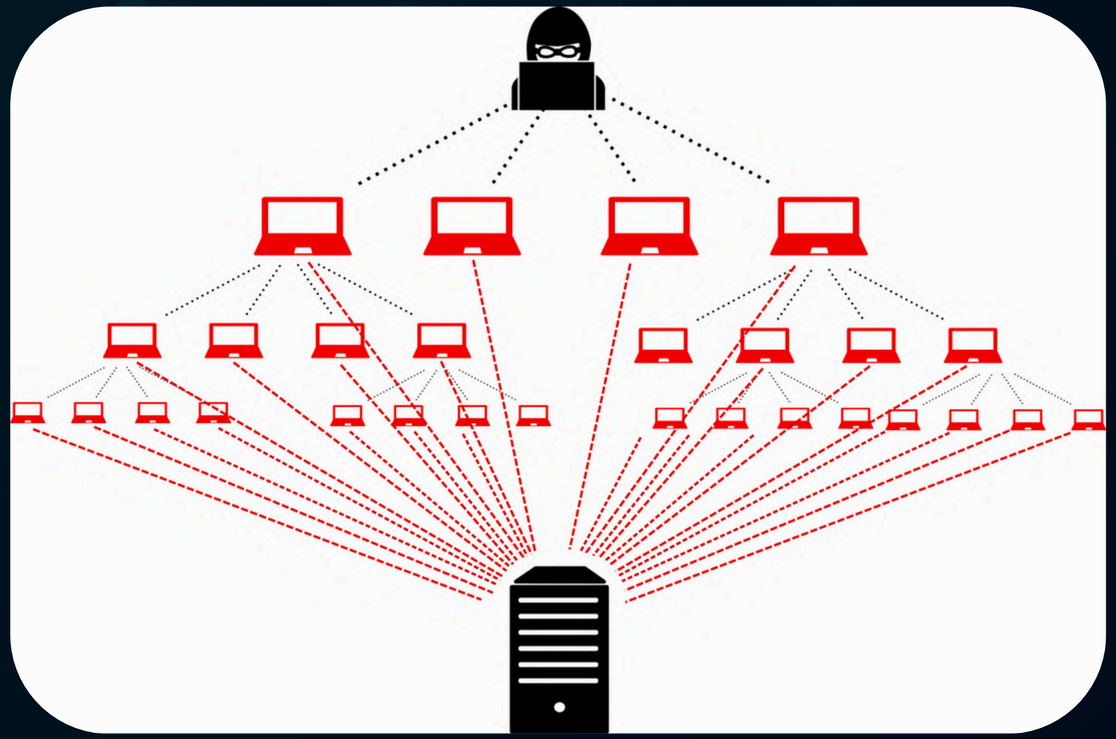
- This consists of attacker sending a heartbeat request with higher payload size.

Heart Bleed


SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
User Meg wants these 500 letters: HAT.


User Meg wants these 500 letters: HAT.
HAT. Lucas requests the "missed conne ctions" page. Eve (administrator) wan ts to set server's master key to "148 35038534". Isabel wants pages about " snakes but not too long". User Karen wants to change account password to "

Large Scale DDoS



- In a DDoS attack many computers (bots/zombies forming a botnet) controlled by a hacker flood a server with fake requests avoiding legitimate users from accessing it.
- Can occur over TCP/UDP/HTTP networks

Cryptography

01
A way to send secrets

02
Common Ciphers

03
Actually secure Crypto

04
Quantum Cryptography

05
Gravity Falls!!!

A way to send secrets

What it is: Fancy word for making sure your secrets stay secret

Why it matters:

- Keep it Private: Only the right people can see the info.
- No Tampering: Making sure no one messes with your stuff.
- Prove It's You: Confirming who you are.

Common Ciphers and Encryptions

01
Caesar Cipher

02
Vigenere Cipher

03
XOR

04
base64

05
Hashing

Back in the days of Julius Caesar

Julius Caesar was already encrypting his messages using the Caesar Cipher. It's basically shifting letters around to make messages harder to read.

Some Cool Text

Uqog Eqqn Vgzv

But here is the thing, even a child can crack this now.

Cryptography now

But we have come a long way

Some Cool Text

b7679991c582edb5b51216c02e6fdfde5993b4547df4621582112cc98a796860a

Getting back the text from these scrambled letter is next to imposible.

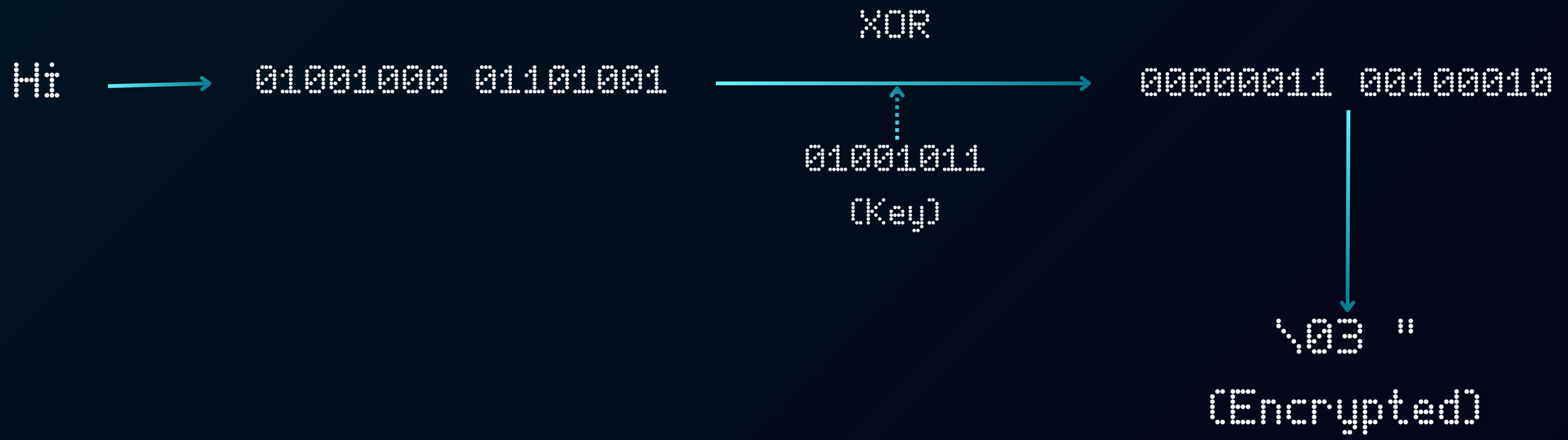
Vigenère Cipher

The Vigenère Cipher is a method of encrypting text by using a series of Caesar ciphers based on a keyword. It is a polyalphabetic substitution cipher, which means it uses multiple alphabets to encrypt the message, making it more secure than a simple Caesar cipher.

It shifts each alphabet just like in Caesar Cipher but this time, each letter is shifted by a different number, based on a key provided to it.

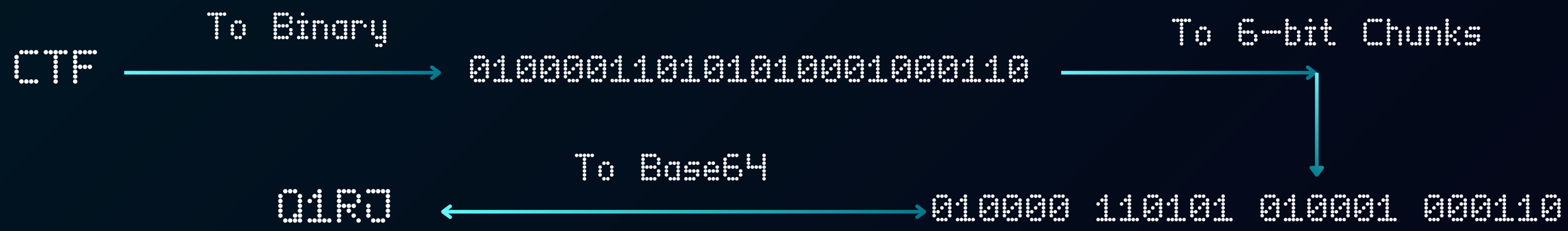
XOR

I guess you do know how information is stored in computers. It's stored in binary i.e., it consists of only zeroes and ones.
Our next encryption trick works on the binary representation of data and encrypts it using a provided key.



Base64

Base64 represents binary data in an ASCII string format. It encodes data into Base64 characters (6 bits each).



This is used for encoding API keys and Tokens. However, this is NOT an encryption. It can be easily decoded and hence not used to store sensitive data. If there are not enough bytes to make 4 chunks, = is added as padding.

Hashing

Now, this is where it gets intersting

Hashing is a process that converts data into a fixed-length string using a mathematical function. The output, called a "Hash", is unique to the input, meaning even a small change in the original data produces a completely different hash.

Remember the freaky encryption shown at the starting. That is an example of Hashing.

Hashing

Some Cool Text

b7679991c582edb5b51216c02e6fdfde5993b4547df46215821l2cc98a796860a

This is an example of SHA-256, an industry level encryption.

Here is the cool part. This cannot be reversed. Hashing is a one-way process and is used for storing passwords

Modern Cryptography

The last encryption we took a look at was awesome…
But what if I want a way to decrypt my data.

RSA : This is one of the first public-key cryptosystems and is foundational to modern cryptography. It uses two keys: a public key (for encryption) and a private key (for decryption).

RSA encryption can be theoretically reversed, but with the current compute power we have, it will take thousands of years to decrypt it without the private key.

Modern Cryptography

AES: A symmetric encryption algorithm that encrypts data using a secret key. But here's the catch, the same key is used to encrypt and decrypt data.

Like RSA, it is also reversible, but is difficult to crack with the present compute power.

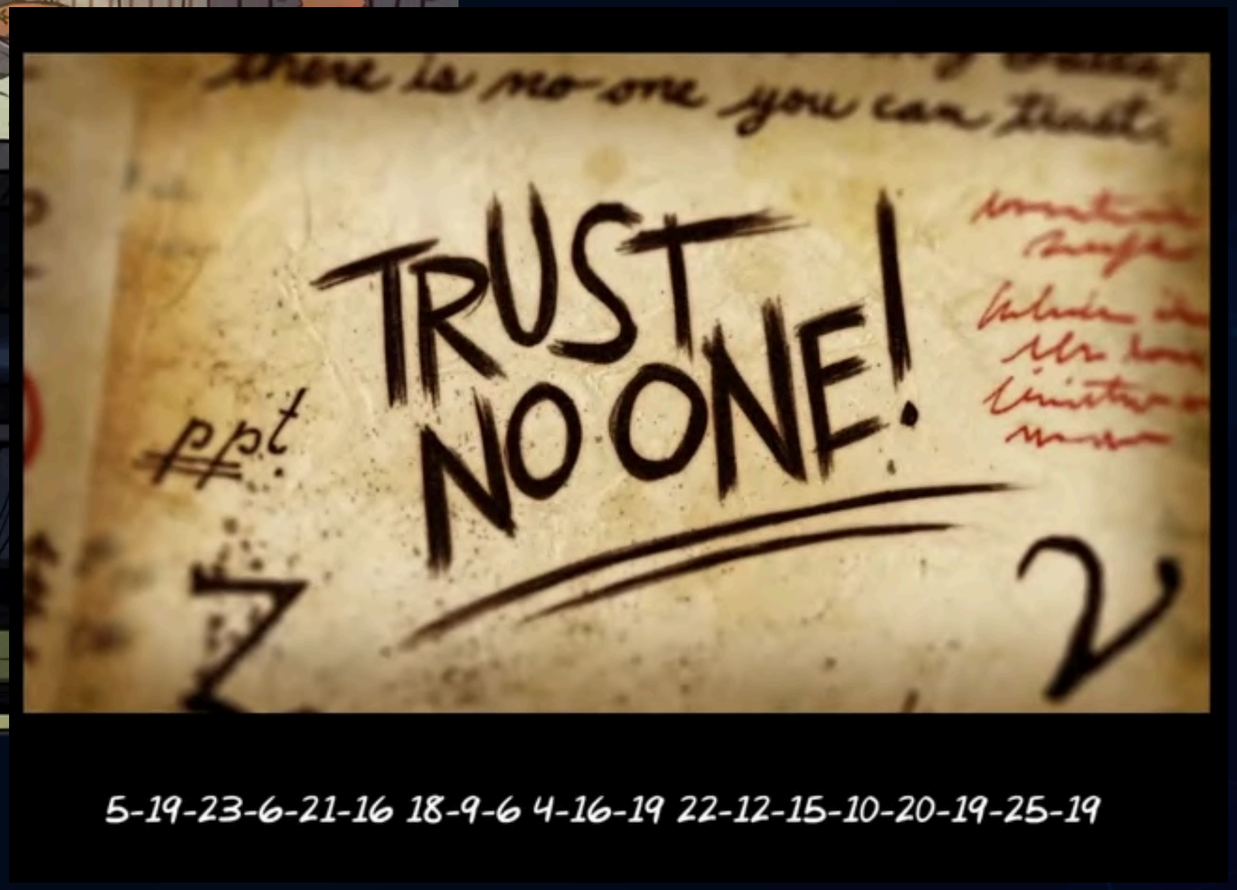
Quantum Cryptography: The Future of Security

Quantum cryptography leverages the principles of quantum mechanics to create unbreakable encryption.
Unlike traditional cryptography, which relies on mathematical complexity, quantum cryptography is based on the laws of physics, making it immune to computational attacks—even from quantum computers.

Heisenberg Uncertainity Principle

That's right... Even a quantum computer bows in front of the Laws of physics.

Gravity Falls!!! #3



Gravity Falls!!! #3

Reverse Engineering

01
What is a binary? (PDS)

02
C is not a safe language

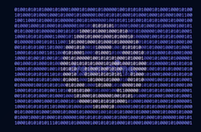
03
How can you reverse it?

04
Quick pre-requisite knowledge

05
Getting our hands dirty

Story of a binary file

When a developer writes a program, they make it machine readable by compiling it with a compiler, which converts the code to a file containing 0s and 1s. The output file is called our 'Binary File'.

source code >>>> compiler >>>> linker >>>> 

User written code in any high level language	Converts the source code to machine code	Adds the dependencies and libraries	A single executable binary file

Windows have .exe/.dll files

Unix like systems have ELFs .out/.so files

Android APKs has DEX files

Vulnerabilities in a binary file

Why is C unsafe?

C is considered unsafe because it lacks built-in memory safety features. It allows direct memory access, meaning you can read or write beyond allocated memory, leading to buffer overflows. Attackers exploit this to overwrite critical data, execute arbitrary code, or crash programs.

Unsafe memory access can lead to serious security exploits, including game hacking, DRM bypassing, router firmware modifications, and privilege escalation

Vulnerabilities in a binary file

Why is C unsafe?

```c
int main() {
    char buffer[8];
    int secret = 1234;

    printf("Enter input: ");
    scanf("%s", buffer);
    //overwrites secret if input>8chars
    printf("Buffer: %s\n", buffer);
    printf("Secret: %d\n", secret);

    return 0;
}
```

```c
int main() {
    int arr[7] = {1, 2, 3, 4, 5, 6, 7};
    printf("Valid: %d\n", arr[6]);
    printf("Invalid: %d\n", arr[8]); // Unsafe access
    return 0;
}
```

```c
int main() {
    unsigned int x = UINT_MAX;
    printf("Before overflow: %u\n", x);
    x = x + 1;  // No warning
    printf("After overflow: %u\n", x);
    return 0;
}
```

```c
int main() {
    int *ptr = malloc(sizeof(int));
    *ptr = 69;
    printf("Before free: %d\n", *ptr);
    free(ptr);

    printf("After free: %d\n", *ptr);
    return 0;
}
```

Reversing a binary file

Well...it's easy to generate binary from source code, it's not that easy to get the source code from the pool of 0s and 1s.

Heyy, tension not when Ghidra is here...

Ladies and gentlemen, gather 'round! Allow me to introduce Ghidra, the powerhouse of software reverse engineering!

For quick decompilation,

Use: dogbolt.org

meet Ghidra—the NSA's powerful, open-source reverse engineering tool. It's free, user-friendly, and perfect for unraveling complex code.

Quick pre-requisite knowledge

Static Analysis : Inspects binaries without execution.

Dynamic Analysis : Inspects binaries with execution, to see how it's interacting with system. Unsafe if it contains malicious code. [Run in sandboxed environment]

Static Analysis tools: Ghidra, IDA pro

Dynamic Analysis tools: GDB, Ltrace

- file <pathToFile>: Get file info
- ls -l <pathToFile>: Get file permissions
- chmod +x <pathToFile>: Give executable permission

Getting our hands dirty

Forensics

01
Sniffing for information

02
Common file formats

03
Image forensics

04
Network traffic

05
Get Netflix movies for free!!!

06
What real digital forensics looks like

What is Digital Forensics?

Everything you do online leaves a footprint

- Digital Forensics involves tracking that footprint

You can hide all sorts of information in all sorts of files. In this section we're going to look at this aspect of forensics called Steganography

- images, pdfs, audio files, and so much more

Common File formats

```
/ctfs/mm
$ file ben10.txt
ben10.txt: JPEG image data, JFIF standard 1.01, aspect ratio, density 1×1, segment length 16, Exif
Standard: [TIFF image data, big-endian, direntries=4, xresolution=62, yresolution=70], progressive,
precision 8, 338×601, components 3
```

in steganography you're often working with images, audio files or archives.

- But beware, you can't always trust the file extension!
- This is when you use the `file` command!

How does `file` work???

Image Steganography

Metadata Analysis

Every file in your filesystem has some metadata associated with it. And certain file formats allow you to modify this metadata with user defined values

```
❯ exiftool meow.jpg
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
...
User Comment                    : a29zc0NURntsb29rX2Zvcl90aGVfZGF0YV91bmRlcl90aGVfZGF0YX0=
```

Strings

```
❯ strings hackme
libc.so.6
exit
__isoc99_sscanf
...
Enter the password:
kossCTF{remeber_to_strip_your_binaries_kids}
...
```

Image Steganography

File carving

Sometimes you can add a file INSIDE another file.

```
file1 start
...
...
...
file1 end (supposedly)
——
file2 start
...
...
...
file2 end (file actually ends now)
——
```

```
❯ binwalk -ave torrent.pcap
for/torrent/extractions/torrent.pcap

DECIMAL                    HEXADECIMAL                DESCRIPTION
8861                       0×229D                     ZIP archive,
                                                      file count: 2,
                                                      total size:
                                                      192209460 bytes

[+] Extraction of zip data at offset 0×229D completed successfully
```

Computer Networks

Network Forensics

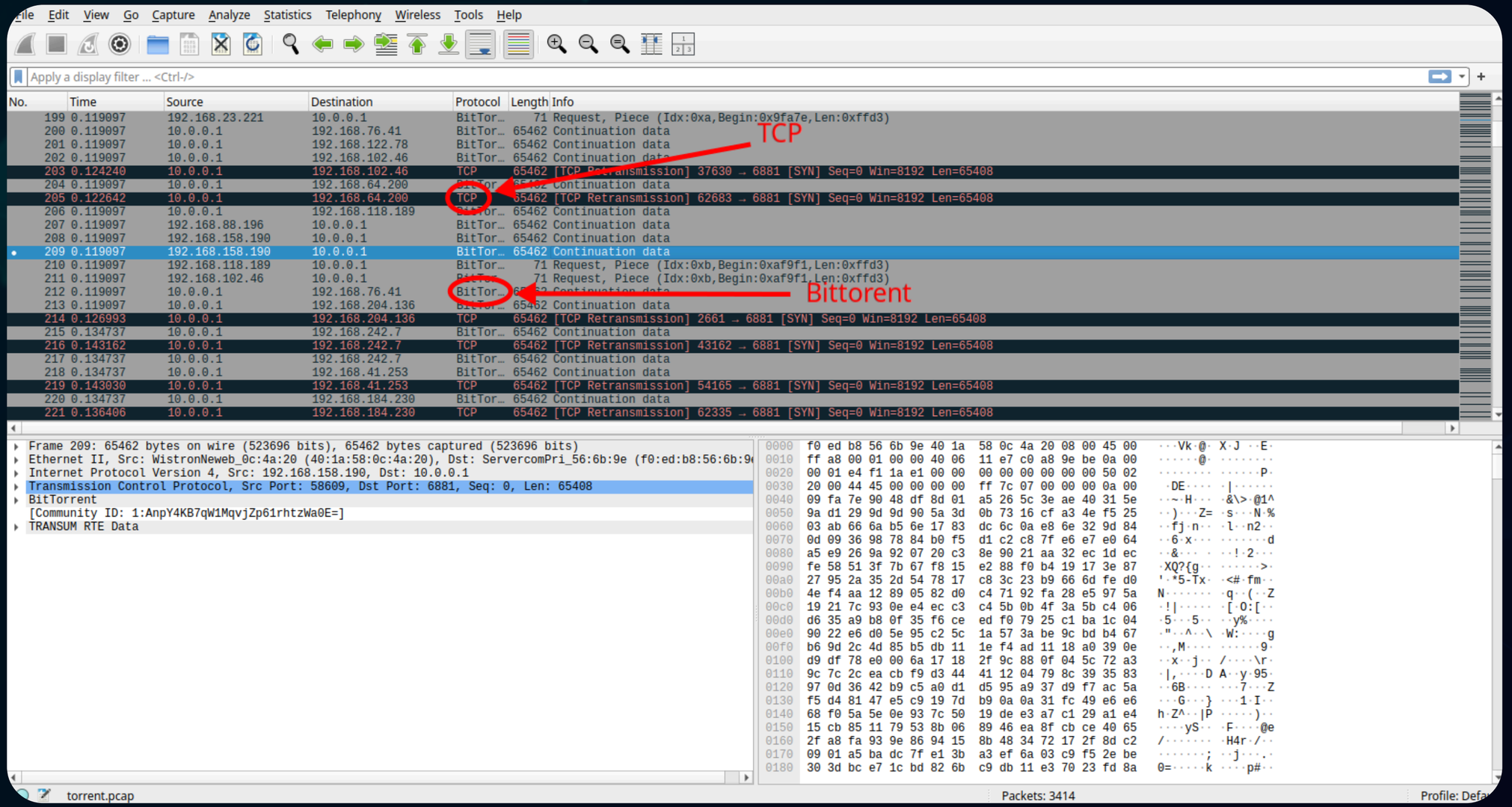
As you saw in the web module, all communication happens through "requests". These requests are sent through "packets" and given the scale of the internet, you can imagine there must be MANY packets being sent every second.

- Network forensics deals with understanding these packets

Packets and Protocols

This is a very (very) simplified explanation, but just remember that these packets follow certain protocols that define exactly the contents of the packet and any other required details. You saw this a little in heartbleed. Knowledge about various protocols and their common vulnerabilities becomes essential after a point.

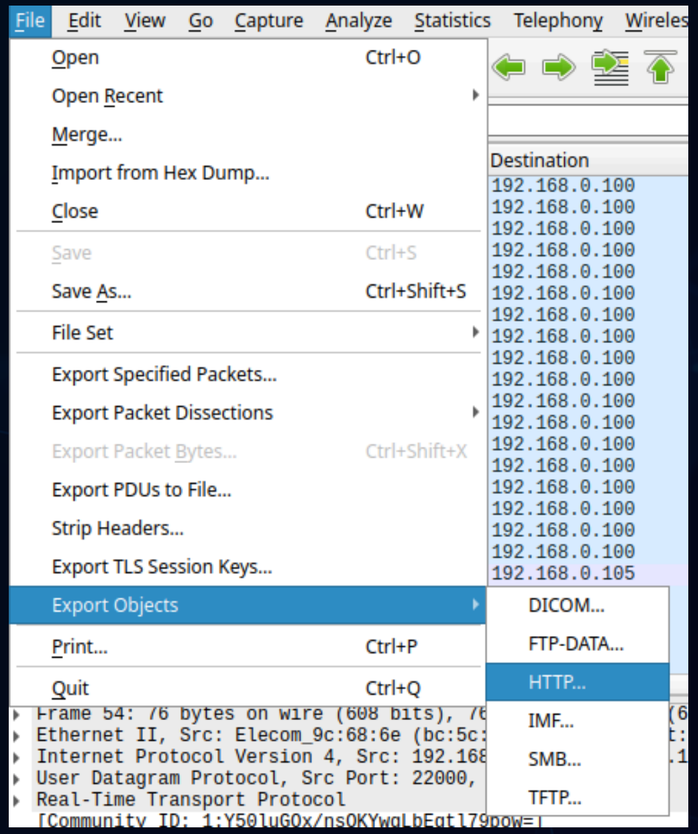
WireShark Computer Networks



Free Netflix Movies?

What happens when you watch some content?

- The video is being sent to your machine via packets
- If you capture these packets (with wireshark) could you extract this video?
  - Maybe not for Netflix movies (there's something called DRM protected content)
  - But you can extend this idea to get other cool stuff
- Live streams must be sent via unencrypted packets (RTP protocol), you can extract them from pcaps.
- Similarily files shared via 'weaker' protocols like HTTP can be extracted.
- Other examples I've seen:
  - Reconstructing a minecraft world from the server packets
  - Recreating a file from the torrent packets



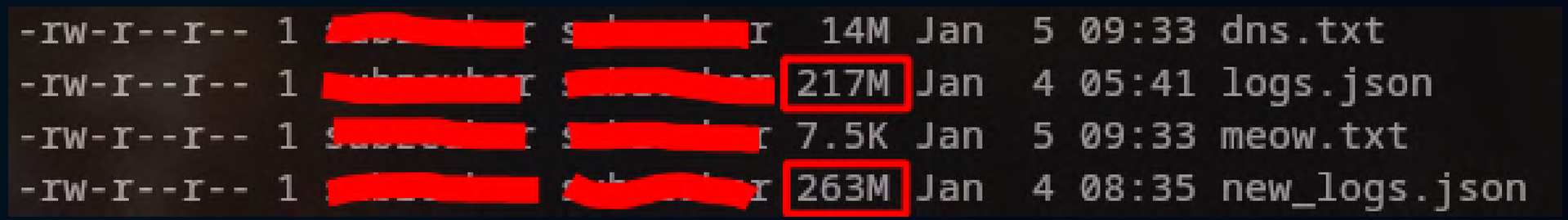
What does Real DFIR Look like?

Like we mentioned in the beginning of this section, everything leaves a footprint.

What are some examples of recovering footprints?

- recovering deleted files?
- breaking password protection?
- or slogging through log files large enough to crash your text editor?

All of the above! And a lot more. DFIR is more defense oriented. Going through very large .pcaps or logs to find suspicious activity or analysing file systems and recovering data etc is what these guys do.



opening these ate half of my ram

Open Source Intelligence (OSINT)

OSINT is about gathering and analyzing information that's out there for everyone to see. Imagine you're a detective. But instead of dusting for fingerprints, you're combing through social media posts, public records, and online forums. That's OSINT in action!

Let's dive into some basic OSINT challenges!

How to get better at CTFs?

use linux

learn a scripting language

read writeups

write writeups

ctftime.org

References and Resources

GEN:

- Careers in cybersecurity
- Beginner’s guide

Web:

- What happens at "google.com"
- Browser Tools
- Heartbleed XKCD
- Heartbleed

CRYPTO:

- Gravity falls (available on KGP’s DC++)

FORENSICS:

- Image Steganography

OSINT:

- Beginner’s guide to OSINT

Official Solutions

https://github.com/kossiitkgp/KossCTF-2025

Feel free to add your writeups and solutions if you think you did something unique or cool

Feedback Form

We value your feedback and would appreciate you filling out this form

Feedback

THANK YOU

REACH OUT TO US: