

CTF Crash Course: Part 2

Master OSINT, steganography, and digital forensics – essential techniques for competitive hacking and investigation challenges.



What is OSINT?

Open Source Intelligence (OSINT) is the art of gathering publicly available information from digital and physical sources. Unlike covert intelligence, OSINT relies on legitimate, accessible data – websites, databases, social media, and historical records. In CTF challenges, OSINT skills help you uncover hidden clues, verify identities, locate servers, and trace digital footprints before technical exploitation begins.



OSINT Tools & Techniques



Reverse Image Search

Google Images, TinEye, Yandex — identify where photos originate, find duplicates, and locate metadata clues embedded in image sources.



WHOIS & DNS

Query domain registration data to uncover website owners, registrar details, IP addresses, and hosting providers.



Wayback Machine

Archive.org captures historical snapshots of websites. Find deleted content, old configurations, and leaked information archived over time.



Sherlock & Maltego

Automate usernames across platforms, map digital relationships, and visualize OSINT connections in comprehensive investigation graphs.

OSINT in Action: Real-World Example

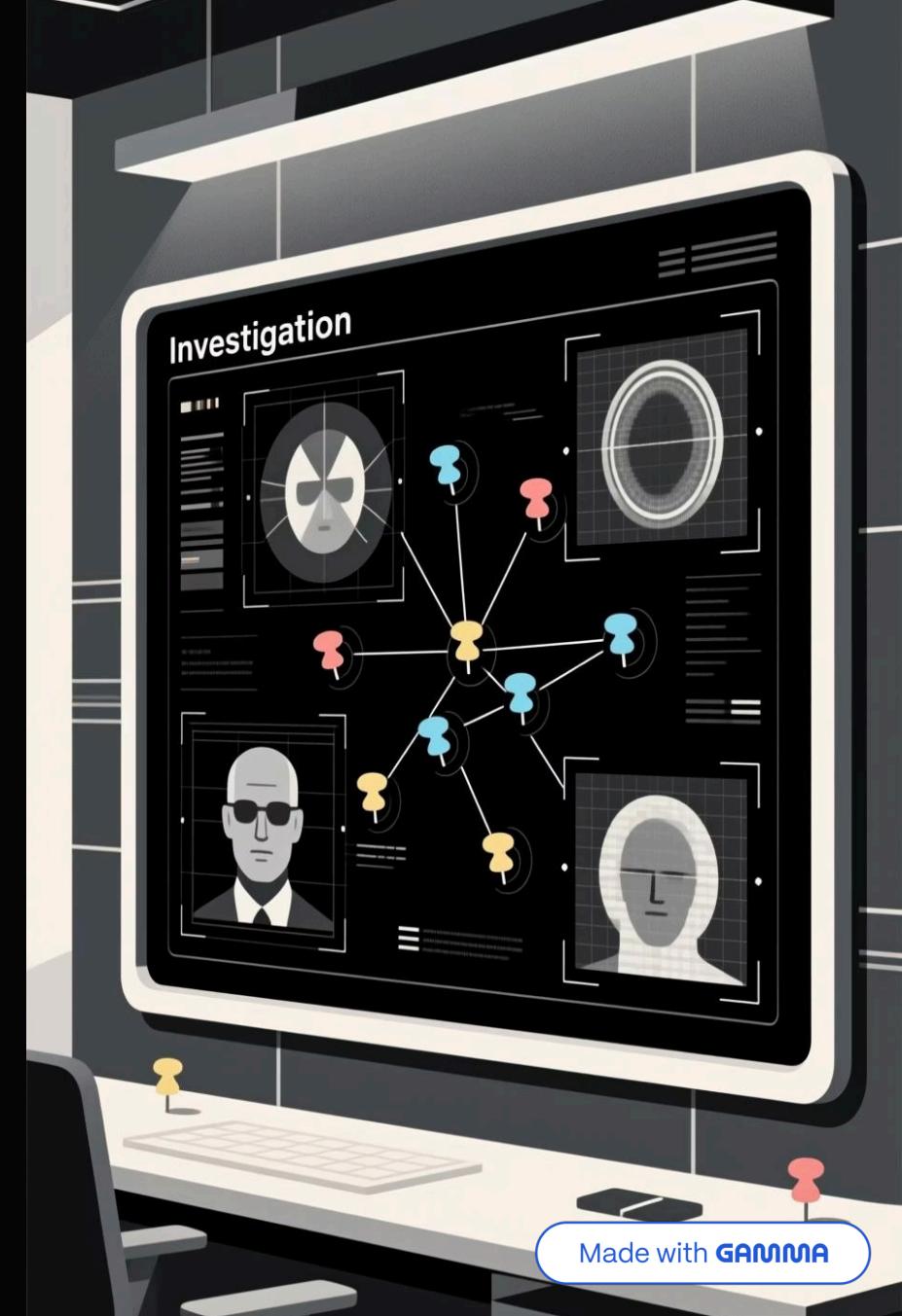
The Challenge

A CTF prompt provides a suspicious screenshot with minimal context.

Your goal: identify the location, the person, and the server where data was hosted.

The Process

- Extract metadata using exiftool
 - GPS coordinates, timestamps, device info
- Reverse image search to find original sources and linked accounts
- Query WHOIS for domain details and hosting providers
- Cross-reference usernames on social media via Sherlock
- Archive.org reveals past website versions with credential hints

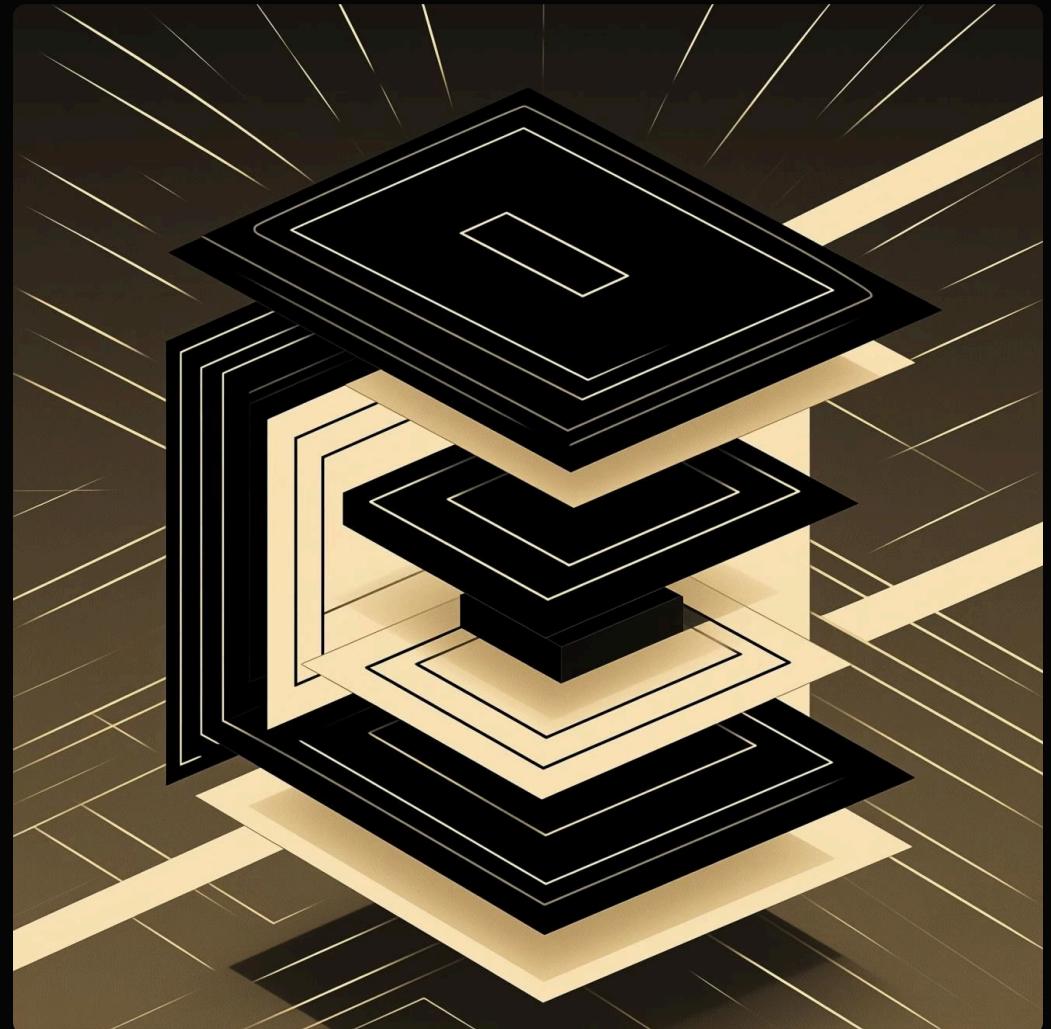


Steganography Unveiled

Steganography vs. Cryptography

Cryptography scrambles data so only authorized users can decrypt it – everyone sees encrypted text.

Steganography hides data inside other files (images, audio, video) – no one suspects data exists. CTF challenges often combine both: hide encrypted data inside an image, then solve the puzzle.



Steganography: Hiding & Extracting Data

1

Image Steganography

Embed data in pixel least-significant bits (LSBs). Tools like Steghide and zsteg extract hidden files from PNG, JPG, and BMP without altering visible appearance.

2

Audio Steganography

Hide data in audio frequencies using LSB encoding or phase shifting. Extract using spectral analysis tools or dedicated audio stego software.

3

Text Steganography

Conceal messages in whitespace, Unicode characters, or word patterns. Flag characters embedded in seemingly innocent sentences reveal secrets under close inspection.

4

Metadata Embedding

Exploit EXIF, metadata headers, and file structure gaps. Exiftool reveals hidden authors, GPS coordinates, timestamps, and custom fields flagged with clues.

```
Jøægistenabøl :~ 5888
Jdhe:
  ↳ conðe:
    (slicxestramnæt and) )ððð.,
    cðouze
      trtonuelne-swðnñit);
    (ððtakat zøg)
  ↳
]
steghide :::
  (iwianus-couretine- zøð),
  ↳
    (arcannptlxxt. zøg))
  cñouna
  ( (tcansint rtut (9;8))
  ( enutor 5sg)
  (ðenloar-temeital zøð)
  ↳
)
steghid :::::
  :
  (corxeajcitrñfønt zøð)
  laarorc-
  (onocrtace-itagz zøð)
  ahøa
  (oni aiøt zøg)
  ↳
)
Termøða PialinæsfOne. f::
  (counte
    (cotaselbn=tatiol zøð)
  ↳
    (ouk tic 9øg),
  ↳
)
```

Essential Steganography Tools

Steghide

Embed and extract data from JPEG, BMP, WAV files with password protection.
Command: `steghide extract -sf image.jpg`

Zsteg

Scan PNG and BMP files for all LSB steganographic layers automatically. Quickly detect hidden text or binary payloads without guessing parameters.

Exiftool

Read and write metadata in images, audio, and documents. Extract GPS, timestamps, author info, and custom hidden fields embedded by challenge creators.

Digital Forensics Essentials

Digital forensics in CTFs involves extracting data from corrupted files, analyzing binary structures, capturing network traffic, and recovering deleted information. Challenges test your ability to uncover hidden layers: files within files, encrypted payloads, obfuscated code, and recoverable fragments from seemingly broken data.

01

File Analysis

Use **binwalk** to scan files for embedded filesystems, archives, and signatures. Extract nested data layers automatically.

02

String Extraction

Run **strings** to pull readable text from binary files. Flags, URLs, and hints often hide in plaintext within executables.

03

Network Forensics

Capture and analyze traffic with **Wireshark**. Inspect packets, follow TCP streams, decode credentials, and uncover command-and-control communications.

04

Memory Forensics

Extract running processes, decrypted passwords, and cached data from memory dumps using Volatility and similar frameworks.



binwalk analysis
corrupted file extraction
nested fifasoneys.

Forensics in Action: Extracting Hidden Data

The Challenge

You receive a corrupted image file. Running `file` shows conflicting headers. The image won't open normally.

The Solution

- **binwalk -e file.bin** – extracts nested data automatically
- Reveals hidden ZIP archive inside PNG metadata
- **strings file.bin** – pulls readable text and URLs
- Unzip archive to discover encrypted flag or next clue
- Combine with steganography tools if layered encoding detected

Challenge Recap: The Complete Picture

Real CTF puzzles combine OSINT, steganography, and forensics into one integrated challenge. Here's how they work together:

OSINT
Gather context, locate resources, identify targets and infrastructure.

Flag Found

Combine outputs to solve the puzzle and claim your victory.



Steganography
Extract hidden credentials, API keys, and encrypted payloads from files.

Forensics
Analyze corrupted data, recover fragments, and decrypt final flag.

Pro Tip: Always document your methodology. Screenshot tool outputs, note command syntax, and keep organized notes — you'll reference them in future challenges.