Knowledge Notes: Cybersecurity, Web Tech & CTF

Rohith Karning
rohithkarning@gmail.com
Linkedin
GITHUB

August 25, 2025

Contents

1 Ciphers: Encryption & Decryption

1.1 Concept

- Encryption = converting **plaintext** (readable message) into **ciphertext** (unreadable form).
- Decryption = converting ciphertext back into plaintext using a secret key.
- Secure communication requires strong algorithms resistant to brute force or frequency analysis.

1.2 Caesar Cipher

- One of the earliest substitution ciphers.
- Each letter is shifted forward or backward by a fixed number (e.g., shift 3: A $\rightarrow$ D).
- Very weak by modern standards:
  - Only 25 possible shifts.
  - Easily breakable via brute force or letter frequency analysis.

1.3 Modern View

- Used historically, but now replaced by stronger algorithms like AES, RSA, and ECC.

2 Steganography

2.1 Introduction

- The art of **concealing information** inside another medium (image, video, text, audio).
- Unlike encryption, the goal isn't to make data unreadable, but to make it invisible.

2.2 Image Steganography

- Example: Hide secret text inside pixels of an image (Least Significant Bit (LSB) encoding).
- Looks like a normal image to humans, but tools can extract the hidden payload.

2.3 Difference from Encryption

- Encryption: hides **content**.
- Steganography: hides **existence**.
- They can be combined for layered security.

3 Web Technology Concepts

3.1 Metadata

- "Data about data".
- Provides descriptive information about the properties of the actual content.
- Examples:
  - Images → EXIF data (camera, GPS coordinates, timestamps).
  - Web files → title tags, comments, headers.
- Useful in forensics, OSINT, and CTFs to retrieve hidden information.

3.2 Web Archives

- Store historical snapshots of websites.
- Example: Wayback Machine at archive.org.
- Use case: Access websites that are taken down, check older versions for clues, or restore lost content.

3.3 GitHub OSINT

- GitHub accounts often reveal:
  - Project files (`README.md`) containing notes, contact info, or internal hints.
  - Commit history which may leak credentials or API keys.
  - Hidden branches with sensitive code.

4 WASM & WAT Files

- **.wasm** – WebAssembly binary file, runs at near-native speed in browsers.
- **.wat** – Text-based human-readable version of a `.wasm` file.
- Often used in CTF challenges:
  - Reverse engineer logic from `.wasm`.
  - Translate back into C-like pseudo code.

5 Machine Learning Basics

- Steps to train an ML model:
  1. Data collection.
  2. Data preprocessing (cleaning, normalization).
  3. Choose model type (Regression, Classification, Neural Networks).

4. Train model on dataset.
5. Evaluate performance (accuracy, precision, recall).
6. Deploy model for real-world use.

- Applied in security: anomaly detection, malware classification, phishing detection.

6 CTF Notes & Investigation Tips

- Always check code for **hidden comments**.
- Use `Ctrl + Shift + I` in browser to inspect elements.
- Look inside:
  - HTML **title tag** (sometimes contains hints).
  - Source code comments.
  - Network tab: inspect response headers or hidden API data.
  - Metadata in files (images, PDFs, Word documents).
  - GitHub repository commits, branches.
  - `.wasm/.wat` files for hidden logic.
- "Think like a puzzle solver" – every unintended file/field could be a clue.

7 Miscellaneous Notes

- DBWhatsApp – exploration of WhatsApp database files often used in forensics.
- README.md – crucial starting point for GitHub repos, contains summaries of purpose.
- Network responses – check JSON/XML for leftover debug fields.