Project Report: Information Stealer Simulation

1. Objective

The primary objective of this project was to simulate the behavior of an **Information Stealer (Infostealer)** — a type of malware designed to extract sensitive data from a compromised host. The project aimed to build a safe, educational proof-of-concept that would help understand the data theft techniques used by real-world malware families while ensuring ethical boundaries within a lab environment.

2. Project Summary

This project involved the development of a Python-based **data exfiltration simulation tool** that mimics how malicious actors collect credentials, system metadata, clipboard content, and public IP addresses. The simulation was designed to demonstrate what data is commonly targeted during system compromise and how it can be programmatically collected.

The tool was developed with a strong emphasis on **transparency, ethical use, and local-only operation** — no actual data was exfiltrated externally. All collected data was stored in structured local files inside a `stolen_data/` directory.

3. Key Functionalities

✅ Chrome Credential Theft

- Accesses Chrome's `Login Data` SQLite database.
- Decrypts saved usernames and passwords using AES-GCM (Advanced Encryption Standard - Galois/Counter Mode) and Windows DPAPI (Data Protection API).
- Utilizes master keys from Chrome's `Local State` file to complete decryption.

✅ Clipboard Data Access

- Uses `pyperclip` to fetch any current text copied to the system clipboard.
- Demonstrates the risk of temporary sensitive data leakage (e.g., passwords or tokens).

✅ System Information Reconnaissance

- Gathers host metadata including:
  - OS name and version
  - Hostname and user profile
  - MAC address and local IP address
  - CPU architecture and platform

✅ Public IP Collection

- Connects to `https://api.ipify.org` to determine the public IP address of the host machine.

✅ Secure Data Storage

- Stores all retrieved information in organized `.json` and `.txt` files under a `stolen_data/` folder.
- Ensures readability and traceability for later review in training scenarios.

4. Technology Stack

Library / Module	Purpose
`sqlite3`	To interact with Chrome's Login Data database
`os`, `json`	To handle file creation and structured data storage
`win32crypt`, `Cryptodome`	For decryption using DPAPI and AES-GCM
`pyperclip`	Clipboard access on Windows
`requests`	HTTP-based public IP retrieval
`base64`, `hashlib`	Encoding utilities for binary-safe operations

⚠️ Note: Tool works on **Windows only** due to Chrome's path structure and dependency on DPAPI.

5. Ethical Considerations

This project was strictly conducted within an **isolated virtual lab** and was never used on production or unauthorized systems. All data extraction was performed on the developer's own system for testing purposes. No exfiltration or external transmission mechanisms were implemented.

The project was developed to help security professionals and students:

- Understand how sensitive data is targeted
- Learn how malware operates at a system level
- Develop defensive awareness about what needs to be protected

6. Learning Outcomes

- Gained hands-on experience in **decrypting real browser-stored credentials**
- Understood how adversaries exploit **client-side data** such as clipboard and profile info
- Practiced implementing **cryptographic operations (AES-GCM, DPAPI)**
- Learned techniques used in **real infostealers** and how to simulate them safely
- Reinforced secure coding practices and ethical red teaming awareness

7. Conclusion

The Information Stealer Simulation project effectively showcased the **data reconnaissance capabilities** that attackers often employ during the initial or post-compromise phase. By safely simulating these behaviors, I gained a deeper appreciation of how sensitive data is stored, accessed, and potentially exposed.

This project strengthened my understanding of endpoint vulnerabilities and will contribute toward better **detection engineering**, **blue-team defense tactics**, and **cyber threat emulation**.