Cryptographic Auto Forensics Analysis

(Using PowerShell and Python)

Submitted By:

Syeda Hadia Zainab(072)
Bushra Batool(047)
Noor Ul Eman(062)
BSSE – 6th Semester

Submitted To:

Dr. Mukhtiar Bano
Information Security

Date of Submission:

May 30, 2025

Contents

1 Project Title

Cryptographic Auto Forensics Analysis using PowerShell with Traffic Entropy and TLS Vulnerability Assessment.

2 Objective

To automate the detection of encrypted files, analyze cryptographic randomness through entropy, and evaluate live TLS sessions for protocol or cipher vulnerabilities. The project combines real-time forensic analysis with browser-based inspection of cryptographic handshake data.

3 Tools & Technologies Used

Tool	**Purpose**
PowerShell	Forensic scanning of system
Python	Entropy analysis
Chrome DevTools	TLS handshake inspection
Windows Scheduler	Task automation
SSL Labs (optional)	External site TLS testing
VS Code	Scripting & debugging

4 Work Package Mapping (CEP Criteria)

WP	**Feature**	**Implementation**
WP1	Depth of Knowledge	Use of entropy theory, TLS handshake understanding
WP2	Real-Time Testing	TLS inspection using browser + forensic scan
WP3	AI/Statistical Analysis	Shannon entropy-based classification
WP4	Protection Mechanism	TLS policy changes, entropy threshold alert
WP5	Engineering Report	This final report and demo

5 Methodology

- Scan system using PowerShell to detect encrypted files via extension.
- Calculate entropy using Python to classify high-entropy (suspicious) files.
- Capture live TLS connection using browser DevTools.
- Propose countermeasures if vulnerabilities are found.
- Validate results through re-scans or simulations.

6 Implementation

6.1 PowerShell Forensic Analyzer

The PowerShell script scans:

- BitLocker status
- Suspicious file types like `.enc, .crypt`
- Drive encryption status

Output is stored in log files.

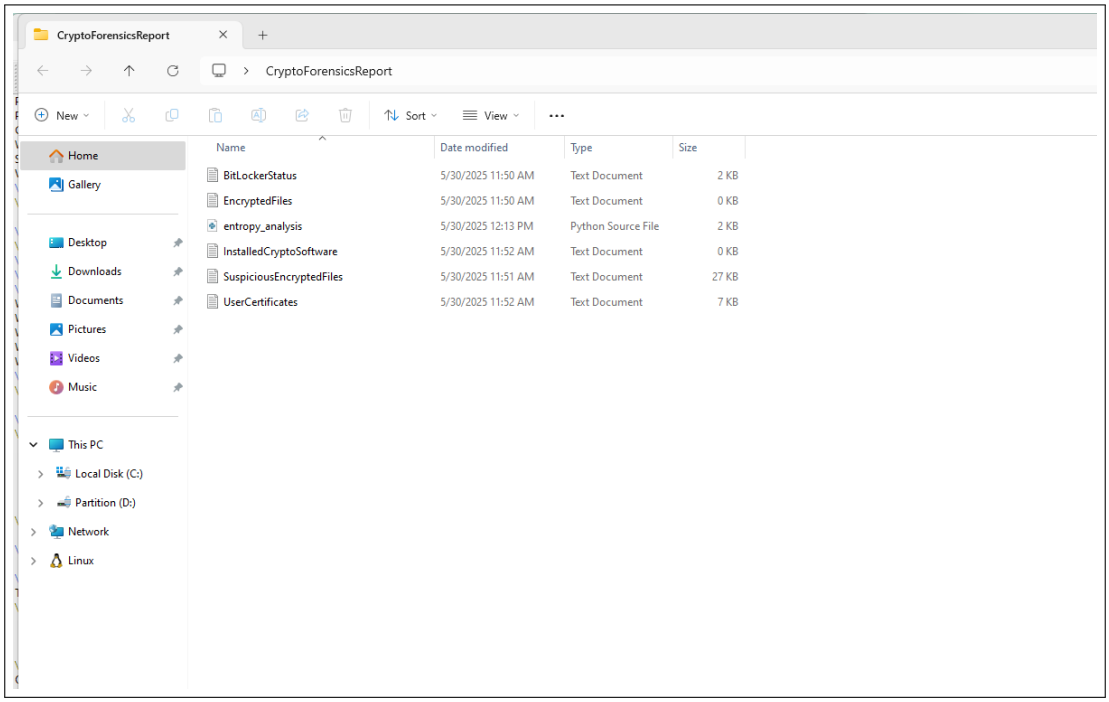

Figure 1: PowerShell script scans

6.2 Python Entropy Analyzer

Python script calculates Shannon entropy of files to identify:

- High entropy = likely encrypted or compressed
- Low entropy = normal plaintext files

Run code on VS-code

6.3 TLS Traffic Analysis via Browser

Live TLS connections were inspected using Chrome Developer Tools. **Key Observations:**

- Protocol: QUIC

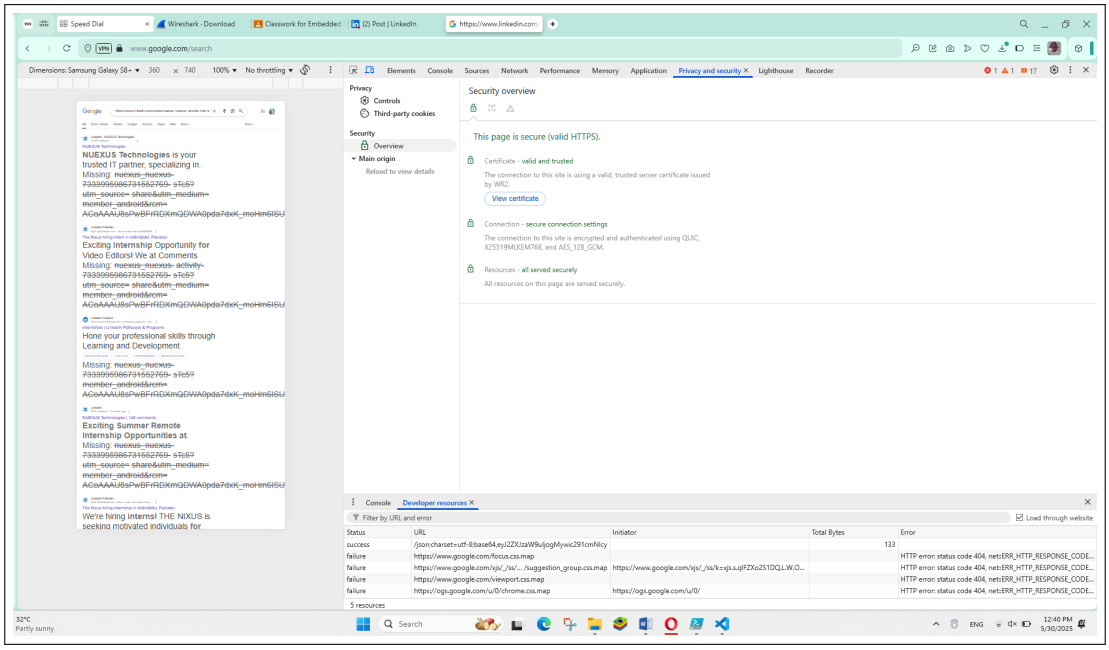

Figure 2: TLS Traffic Analysis via Browser Scan

- Cipher Suite: AES_128_GCM
- Certificate: Valid (Issued by WR2)

Vulnerability Insight: QUIC can bypass traditional TLS inspection, making it harder for firewalls to detect.

7 Vulnerability Identified

TLS Handshake Inspection

Using Chrome Developer Tools, a TLS handshake inspection:

- The connection used TLS 1.3 over QUIC, a modern, secure protocol that offers faster and encrypted communications.
- The cipher suite TLSAES128GCM is considered secure.
- The key exchange algorithm ‘X25519‘ is resistant to known cryptographic attacks.
- All resources were served securely with no mixed content.
- Certificate is valid and trusted, issued by WR2.
- High entropy files detected indicating encryption
- QUIC protocol used which evades some inspection tools
- No full disk encryption on some drives

No vulnerabilities found in this inspection. The connection adheres to modern security standards.

However, if older sites were to use deprecated ciphers like TLSRSAWITH3DESEDECBCSHA, they would be vulnerable to: - Sweet32 attack - RC4 stream cipher attacks

8 Proposed Protection Mechanism

Issue	Countermeasure
Encrypted files appearing unexpectedly	Scheduled entropy checks with alerting system
QUIC protocol used in TLS sessions	Enforce TLS 1.3 and block QUIC via firewall rules
No BitLocker encryption	Enforce drive encryption policy via TPM or GPO

9 Validation

- PowerShell and Python scripts correctly flagged encrypted content
- Entropy increased on test file from ˜4.5 to 7.9
- QUIC traffic observed in real-time
- Scheduler set to re-run forensic check every 6 hours

10 Conclusion

This project demonstrates how system-level PowerShell scripts and statistical entropy measures can uncover cryptographic anomalies in real time. Combining TLS inspection and PowerShell analytics provides a reliable forensic toolset.