Department of Computer Science Technology
Network and Cyber Security Technology Program
CET233- Digital Forensics
Spring 2025- Project Ideas
Instructor: Dr. Ayman Taha
Eng. Maryam Adel

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Digital Forensics Project Ideas

1. AI-Assisted Network Traffic Forensics for Intrusion Detection

➢ Description:

This project involves real-time monitoring and analysis of network traffic using AI-powered techniques to detect anomalies, cyber threats, and malicious activities such as DDoS attacks, malware communications, and unauthorized access.

➢ Objective:
- Build a real-time intrusion detection system (IDS)
- Use machine learning (ML) to detect malicious activities
- Analyze packet captures and log files

➢ Tools:
- Zeek (Bro) – Network forensics
- Wireshark – Packet capture
- Suricata – IDS/IPS
- TensorFlow / Scikit-learn – AI-based anomaly detection
- ELK Stack (Elasticsearch, Logstash, Kibana) – Log visualization

➢ Project Steps:
1. Set up a test network with simulated traffic
2. Capture network traffic using Wireshark
3. Use Zeek to extract metadata from network logs
4. Train an AI model to classify malicious vs. normal traffic
5. Deploy Suricata for real-time intrusion alerts
6. Integrate logs with ELK for forensic analysis
7. Validate model performance with attack scenarios

2. Memory Forensics and Rootkit Investigation

➢ Description:

This project focuses on analyzing RAM dumps to detect hidden processes, malware, and rootkits that evade traditional security measures.

➢ Objective:
- Extract and analyze RAM images
- Identify malicious processes and injected DLLs
- Detect rootkits and hidden malware

➢ Tools:

- Volatility 3 – Memory forensics
- Rekall – Advanced RAM analysis
- LiME (Linux Memory Extractor) – RAM acquisition
- Sysinternals Suite (Procmon, Autoruns) – Process tracking

➢ Project Steps:

1. Acquire a memory dump from a compromised system
2. Analyze running processes and detect anomalies
3. Investigate hidden DLLs and injected code
4. Use Volatility plugins to detect rootkits
5. Generate forensic reports on identified threats

3. Automated Digital Evidence Acquisition with Chain of Custody

➢ Description:

Create an automated forensic evidence acquisition system that ensures the chain of custody and maintains evidence integrity.

➢ Objective:

- Automate disk and memory imaging
- Implement chain of custody tracking
- Verify evidence integrity with cryptographic hashes

➢ Tools:

- Autopsy (Sleuth Kit) – Digital forensics
- Guymager – Disk imaging
- dc3dd & ddrescue – Disk acquisition
- Log2Timeline (Plaso) – Timeline reconstruction

➢ Project Steps:

1. Collect evidence from a compromised system
2. Create forensic disk images with integrity verification
3. Document chain of custody with timestamps
4. Analyze logs for digital artifacts
5. Generate forensic reports

4. Web Server Log Forensics and Attack Detection

➢ Description:

Analyze web server logs to detect cyberattacks like SQL injection, brute-force logins, XSS, and DDoS attempts.

➢ Objective:

- Parse web logs for attack patterns
- Identify malicious IPs and user behavior
- Generate forensic reports

- ➢ Tools:
  - GoAccess – Log analysis
  - ELK Stack – Log ingestion and visualization
  - OSSEC – Host-based intrusion detection
  - ModSecurity – Web application firewall logs
- ➢ Project Steps:
  1. Collect web logs from Apache/Nginx servers
  2. Parse logs to extract suspicious activities
  3. Correlate attack patterns with known threats
  4. Visualize logs in ELK Stack
  5. Generate forensic reports on attack findings

5. IoT Forensics: Investigating Smart Device Attacks

- ➢ Description:

Investigate IoT security incidents by analyzing firmware, logs, and network traffic.

- ➢ Objective:
  - Extract and analyze IoT device firmware
  - Detect unauthorized access and exploit attempts
  - Reverse-engineer IoT malware
- ➢ Tools:
  - Binwalk & Ghidra – Firmware analysis
  - Wireshark / Tcpdump – Network monitoring
  - RouterSploit – IoT penetration testing
  - Autopsy – Log and disk image analysis
- ➢ Project Steps:
  1. Obtain IoT firmware from a target device
  2. Extract and analyze firmware code
  3. Monitor network traffic for anomalies
  4. Identify security vulnerabilities
  5. Document forensic findings

6. Open-Source Endpoint Detection & Response (EDR) Analysis

- ➢ Description:

Deploy an open-source EDR solution to monitor endpoint security events and investigate malware behavior.

- ➢ Objective:
  - Deploy an EDR system on multiple endpoints
  - Detect and analyze suspicious process execution
  - Correlate findings with MITRE ATTACK
- ➢ Tools:
  - Wazuh – Open-source SIEM/EDR

- Sysmon + ELK Stack – Windows event correlation
- Osquery + Auditd – Linux forensic monitoring

➢ Project Steps:

1. Deploy EDR agents on endpoints
2. Monitor process and registry changes
3. Analyze alerts and correlate logs
4. Detect and classify threats
5. Generate forensic incident reports

7. Mobile Forensics: Android & iOS Investigation

➢ Description:

Analyze mobile devices for forensic artifacts, malware, and security breaches.

➢ Objective:

- Recover deleted files and metadata
- Reverse-engineer suspicious Android/iOS apps
- Analyze network traffic and logs

➢ Tools:

- Autopsy + Sleuth Kit – Mobile disk image analysis
- MobSF – Mobile Security Framework
- JADX / APKTool – Android reverse engineering
- Wireshark – Mobile traffic analysis

➢ Project Steps:

1. Extract mobile device data
2. Analyze apps for security flaws
3. Reverse-engineer suspicious APKs
4. Monitor network activity for leaks
5. Generate forensic reports