Luke Nguyen

Final Project Proposal

Repository Link: https://github.com/lukee-n/cecs478-final-project

Problem Statement

Wired Equivalent Privacy (WEP) was one of the earliest standards for wireless network security, but it is now known to be fundamentally insecure due to weaknesses in its RC4-based encryption scheme. The use of a 24-bit Initialization Vector (IV), combined with predictable key scheduling patterns, makes WEP susceptible to passive traffic capture and offline key recovery. Although WEP is deprecated and no longer widely used, these vulnerabilities remain a foundational lesson in wireless security, cryptographic weaknesses, and digital forensics.

This project aims to build a reproducible, containerized WEP forensics pipeline that demonstrates how an attacker can extract IVs from captured traffic, perform statistical key recovery, and measure how quickly the WEP key can be revealed using modern tooling. The system will take encrypted 802.11 WEP PCAPs as input, extract relevant attack data, run automated cracking attempts, and generate metrics such as packet thresholds and recovery times.

This work matters because understanding WEP's downfall provides insight into modern wireless security design, responsible cryptographic deprecation, and forensic workflows used by analysts investigating compromised or legacy systems.

Threat Model

Assets

- Encrypted 802.11 WEP wireless frames contained in public or synthetic PCAP files.
- The confidentiality of the (synthetic) WEP key used in test captures.

Adversary

A passive attacker whose only capability is to capture wireless traffic.
The adversary does not perform active attacks (e.g., deauthentication, packet injection).
They rely solely on:

- IV collection
- Statistical weaknesses in RC4
- FMS/Korek-style key recovery

Attack Surfaces

- Repeated and predictable IV values
- Weak RC4 Key Scheduling Algorithm (KSA)
- Short 24-bit IV space causing rapid reuse
- Legacy WEP configurations (40-bit / 104-bit keys)

Defensive / Analytic Focus

Rather than exploiting real networks, this project analyzes:

- How many packets are required to reveal a WEP key
- How attacker success correlates with IV distribution quality
- Whether the dataset shows recognizable weak IV patterns
- How quickly modern systems can perform full key recovery

This aligns with forensic workflows, not offensive exploitation.

Success Metrics

The project's measurable outcomes include:

1. **Key Recovery Success Rate**
   Whether the pipeline correctly recovers the known WEP key from test datasets.

2. **Packet Threshold Measurement**
   Minimum number of WEP-encrypted packets needed for a successful crack.

3. **Time-to-Crack**
   Time taken from PCAP ingestion to key recovery on a standard environment.

4. **IV Distribution Analysis**
   Summary of weak vs. strong IV occurrences in the dataset.

5. **Reproducibility**
   A deterministic, one-command build (make bootstrap) that reproduces results.

Dataset / PCAP Plan

Because this project is explicitly offline and uses public or synthetic data, ethical and legal risks are minimized. Sources include:

- Aircrack-ng WEP test PCAPs, intentionally created for research and instruction
- Synthetic captures generated using tools such as airbase-ng and captured with tcpdump in an isolated lab VM
- No real user traffic and no collection from active wireless networks

The project will document:

- Dataset source
- Capture method (if self-generated)
- IV anonymization details
- Exact PCAPs used for reproducibility

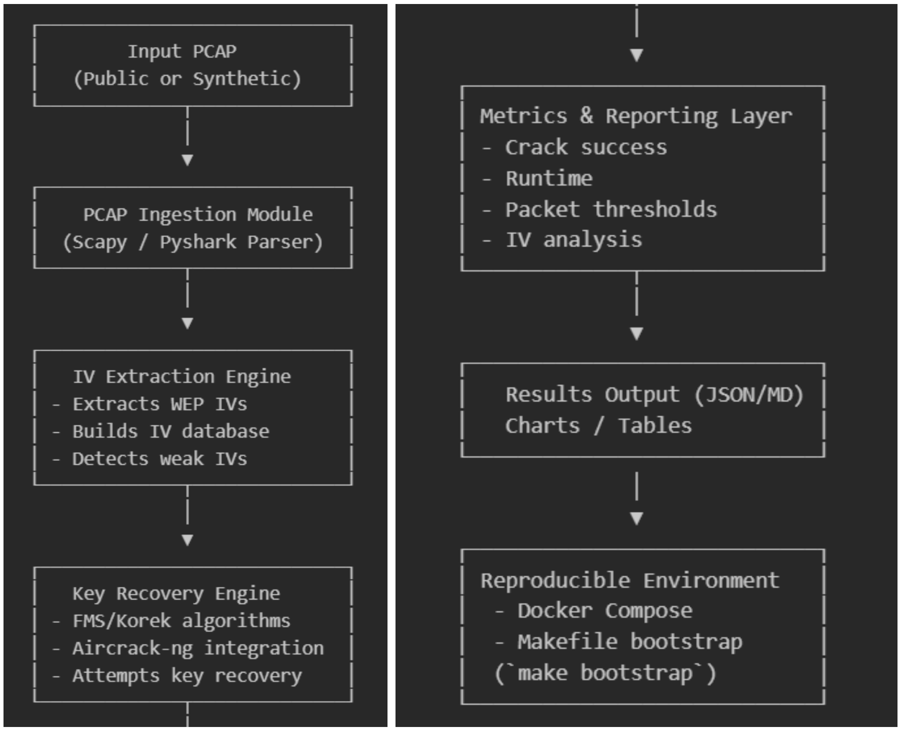
Risks and Ethics

Ethical handling of wireless data is central to this project. Mitigations include:

- **No live network attacks:** No deauth, replay, or packet-injection techniques will be performed against real devices.
- **Public / synthetic PCAPs only:** Ensures no personal data, MAC addresses, or sensitive metadata is captured.
- **Anonymization:** All PCAPs will be inspected and cleaned of unnecessary identifying information.
- **Academic intent:** The project demonstrates cryptographic weaknesses for educational purposes only.

By operating entirely offline with non-sensitive datasets, the project aligns with ethical research expectations.

Architecture Diagram



```
┌────────────────────────────┐
│        Input PCAP          │
│   (Public or Synthetic)    │
└────────────────────────────┘
              │
              ▼
┌────────────────────────────┐
│   PCAP Ingestion Module    │
│ (Scapy / Pyshark Parser)   │
└────────────────────────────┘
              │
              ▼
┌────────────────────────────┐
│   IV Extraction Engine     │
│ - Extracts WEP IVs         │
│ - Builds IV database       │
│ - Detects weak IVs         │
└────────────────────────────┘
              │
              ▼
┌────────────────────────────┐
│   Key Recovery Engine      │
│ - FMS/Korek algorithms     │
│ - Aircrack-ng integration  │
│ - Attempts key recovery    │
└────────────────────────────┘
              │
              ▼
┌────────────────────────────┐
│ Metrics & Reporting Layer  │
│ - Crack success            │
│ - Runtime                  │
│ - Packet thresholds        │
│ - IV analysis              │
└────────────────────────────┘
              │
              ▼
┌────────────────────────────┐
│  Results Output (JSON/MD)  │
│  Charts / Tables           │
└────────────────────────────┘
              │
              ▼
┌────────────────────────────┐
│ Reproducible Environment   │
│  - Docker Compose          │
│  - Makefile bootstrap      │
│ (`make bootstrap`)         │
└────────────────────────────┘
```