Kev's PCAPER Handbook

Table of Contents

---

1. Introduction

PCAPER is a Python-based tool that simplifies analyzing PCAP files. With an interactive menu, support for EBCDIC decoding, suspicious pattern detection, credential extraction, and now a JSON-based malicious IP list, PCAPER provides a comprehensive environment for network forensic analysis.

---

2. What is a PCAP File?

A **PCAP (Packet Capture)** file is a binary format capturing raw network traffic. Commonly produced by tools like tcpdump or Wireshark, PCAP files let you:

- Investigate security incidents
- Troubleshoot network issues
- Understand protocols and sessions

---

3. What is EBCDIC and How Does it Differ From ASCII?

EBCDIC is an older, IBM mainframe-centric encoding, while **ASCII** is the widely adopted standard on modern systems.

- **EBCDIC**: Used primarily on IBM mainframes; less common today.

- **ASCII (and UTF-8)**: A standard encoding for modern computing.

PCAPER allows choosing EBCDIC decoding (cp037 code page) for legacy traffic analysis.

---

4. Overview of PCAPER

PCAPER reads PCAP files, decodes payloads, identifies suspicious activity, extracts credentials, merges PCAPs, reconstructs TCP streams, and analyzes against malicious IP lists from a JSON file. It's designed to be user-friendly, presenting a menu-driven interface and paginated outputs.

---

5. Key Features & Functionalities

5.1. Menu System and Analysis Options

Interact with PCAPER via a simple menu, selecting from various analysis tasks without memorizing complex commands.

5.2. EBCDIC and ASCII Decoding Support

Choose EBCDIC if analyzing mainframe traffic; otherwise, default to ASCII (UTF-8).

5.3. Suspicious Payload Detection (XSS/SQLi)

PCAPER scans payloads for known indicators of XSS and SQL injection, warning you when suspicious patterns are detected.

5.4. Credential Extraction

PCAPER searches payloads for keywords (user=, password=, logon, etc.) to identify potential credentials. This is heuristic and may yield false positives, but it can highlight sensitive information in the capture.

5.5. Malicious IP Identification via JSON-based List

Load a list of known malicious (C2, hacker) IP addresses from a JSON file. PCAPER annotates these IPs in summaries, sessions, and other outputs, helping you spot adversarial infrastructure quickly.

5.6. Merging Multiple PCAPs

Combine multiple PCAP files into one for unified analysis.

5.7. TCP Stream Reconstruction

Rebuild TCP streams to show conversations (commands and responses) in a chronological, readable format.

5.8. Searching Payloads

Search all payloads for user-defined strings, useful for pinpointing relevant data.

5.9. Starting a New Analysis

Easily start fresh with another PCAP without restarting the program.

---

6. Installation & Requirements

- **Python 3.6+**
- **Dependencies:**
- pip install scapy tabulate ebcdic
- **PCAP Files:** Ensure you have .pcap files ready to analyze.

---

7. Setting Up the Malicious IP JSON File

Create a JSON file named malicious_ips.json in the same directory as pcaper.py:

```
{
  "1.2.3.4": "Known C2 server",
  "5.6.7.8": "Known Hacker IP"
}
```

You can add/remove entries as needed. Each key is an IP (string), and each value is a description.

When you run PCAPER, it loads this file. If the file doesn't exist or fails to parse, PCAPER continues without malicious IP annotations.

---

8. Running PCAPER

1. **Create or confirm malicious_ips.json exists.**
2. **Run:**
3. python pcaper.py
4. **Input the PCAP file path and choose EBCDIC or ASCII decoding.**

PCAPER then displays the main menu.

---

9. Understanding the Menu

The menu uses alternating colours for better readability: green for odd lines, red for even lines, white for prompts.

Menu Options:

1. **Print summary of the capture:**
   Shows total sessions, total packets, top ports, top IPs (with malicious tags).
2. **Show sessions and transactions:**
   Displays each session, packets, commands/responses, and malicious IP notes.
3. **Search for a specific string in payloads:**
   Find sessions containing a user-defined keyword.
4. **Print detailed report (XSS/SQLi warnings):**
   Shows suspicious activity summary (XSS/SQLi) and overall stats.
5. **Analyse against known malicious actors:**
   Uses the loaded JSON file to identify sessions involving known malicious IPs.
6. **Passwords we think were captured:**
   Attempts to identify credentials from captured traffic.
7. **Merge multiple PCAP files into one:**
   Combine several PCAPs into a single file.
8. **Reconstruct TCP streams:**
   Shows conversations in a linear, human-readable format.
9. **Start a new analysis:**
   Begin analysing another PCAP file without restarting PCAPER.
10. **Exit:**
    Quit the application.

Pagination Controls: For large outputs, PCAPER presents data in pages. After each page:

- Press **N** to go to the next page.
- Press **M** to return to the main menu.
- Press Enter at the end to return to the main menu.

---

10. Technical Details

10.1. Decoding Logic

- **EBCDIC (cp037)** vs. **ASCII (UTF-8)**:
    - If EBCDIC chosen, payloads decode using cp037 code page.
    - Otherwise, default UTF-8 decoding is applied.

10.2. Command/Response Heuristics

The first packet of a session defines which side is “command” and which is “response.” This heuristic may not always be perfect but usually provides a good starting point.

10.3. Handling Large Outputs (Pagination)

Outputs are paginated to avoid overwhelming the user with too much data at once.

10.4. JSON-based Malicious IP Loading

At runtime, PCAPER reads malicious_ips.json. If loaded successfully, the MALICIOUS_IPS dictionary is populated. IPs in sessions are checked against this dictionary, and if a match is found, the IP is annotated with the provided description.

---

11. Best Practices & Security Considerations

- **Up-to-date Threat Intelligence:**
  Keep malicious_ips.json current by adding or removing entries as threat intel changes.
- **Sensitive Data Handling:**
  Credentials found may be sensitive. Handle with care and comply with legal/privacy requirements.
- **False Positives:**
  Pattern-based detection (XSS/SQLi, credentials) may produce false positives. Use judgment.

---

12. Troubleshooting

- **File Not Found:**
  Ensure the PCAP and malicious_ips.json paths are correct.
- **No Results:**
  If nothing is found, the PCAP may not contain relevant traffic, or try adjusting filters.

- **Slow Performance:**
  For very large PCAPs, consider filtering with tcpdump before analysis.

---

13. Future Extensions

- **Live Threat Intel Feeds:**
  Automatically update malicious_ips.json from a threat intelligence API.
- **Protocol-specific Parsing:**
  Add protocol-aware parsers for HTTP, FTP, SMTP, etc., for more accurate credential detection.
- **Improved Heuristics:**
  Enhance command/response detection and suspicious pattern matching.

---

14. Conclusion

PCAPER simplifies PCAP analysis by providing a menu-driven interface and integrating various features—from legacy EBCDIC decoding to suspicious pattern detection and malicious IP lookups via a JSON file. By following this handbook, you can set up and use PCAPER effectively, improving your network forensic capabilities.

Remember! Use PCAPER responsibly, ensure you maintain your malicious IP list (malicious_ips.json) and keep your threat intelligence data fresh for the best possible insights.