


<p>POLITECHNIKA WROCŁAWSKA</p>  <p>Wydział Informatyki i Telekomunikacji</p>	<p>Wydział: Informatyki i Telekomunikacji Kierunek: Cyberbezpieczeństwo Stopień: II Rok studiów: 2 Termin: <i>Piątek</i></p>
<p align="center">Informatyka śledcza – Projekt / Seminarium Opracowanie systemu ekstrakcji danych ze zrzutów transmisji sieciowych</p>	
<p>Data: 23.12.2024</p>	<p>Skład grupy: 1. 2.</p>

1. Wprowadzenie.....	1
2. Zakres funkcjonalny.....	1
3. Wymagania i zależności.....	2
4. Architektura rozwiązania.....	2
5. Uruchomienie i użycie.....	2
6. Wnioski i możliwość rozwoju.....	3

1. Wprowadzenie

Celem projektu jest stworzenie aplikacji, która pozwoli na automatyczną analizę plików .pcap (zawierających zrzuty ruchu sieciowego). Dzięki temu można w prosty sposób filtrować i wyodrębniać pakiety interesujących protokołów (DNS, TCP, HTTP) oraz dodatkowo kategoryzować i zabezpieczać wyekstrahowane informacje.

2. Zakres funkcjonalny

1. Analiza protokołów

- Użytkownik może zdecydować, czy chce wyświetlić/wyeksportować tylko ruch DNS, ruch TCP, ruch HTTP czy wszystkie pakiety.
- Pakiety są analizowane przy pomocy biblioteki **Scapy**, która umożliwia parsowanie nagłówków i sprawdzanie typów protokołów oraz portów.

2. Kierunek ruchu

- Skrypt rozpoznaje, czy pakiet jest ruchem przychodzącym (incoming), czy wychodzącym (outgoing), używając prostej reguły sprawdzania prywatnego adresu IP (np. 192.168.x.x, 10.x.x.x).
- W razie potrzeby można tę regułę rozbudować i dostosować do konkretnej topologii sieci.

3. Zapis do bazy SQLite

- Dane o pakietach są zapisywane w tabeli packets bazy SQLite.
- Dla każdego pakietu przechowywane są m.in. czas (timestamp), adres źródłowy i docelowy, porty, protokół, lista tagów (DNS_traffic, HTTP_traffic, incoming, outgoing) oraz długość pakietu.

4. Archiwizacja z hasłem (AES)

- Po zakończeniu analizy użytkownik może spakować wygenerowaną bazę .db do archiwum .zip zaszyfrowanego hasłem.
- Stosowane jest szyfrowanie **AES** dzięki bibliotece pyzipper.
- Chroni to bazę przed nieautoryzowanym odczytem.

5. Interfejs wiersza poleceń (CLI)

- Aplikacja prowadzi użytkownika krok po kroku: wyświetla listę plików .pcap w katalogu, umożliwia wybranie jednego lub wielu z nich, pyta o filtry ruchu, nazwę bazy itp.
- Możliwa jest wielokrotna analiza różnych plików w jednym uruchomieniu.

3. Wymagania i zależności

- **Język programowania:** Python 3.7+
- **Biblioteki:**
 - **Scapy** – do wczytywania i parsowania pakietów z plików .pcap.
 - **pyzipper** – do tworzenia archiwów ZIP z szyfrowaniem AES.
- **Dodatkowo:**
 - SQLite (biblioteka sqlite3 jest częścią standardowej dystrybucji Pythona).
 - W niektórych środowiskach może być potrzebny kompilator C/C++ (Visual C++ Build Tools na Windows), gdyby pojawiły się problemy z instalacją zależności.

4. Architektura rozwiązania

1. **Główny skrypt** (np. `pcap_cli_app.py`) realizuje logikę interfejsu użytkownika (CLI).
2. **Biblioteka Scapy** wczytuje plik `.pcap` i dostarcza pakiety do dalszej analizy.
3. **Filtry** decydują, czy dany pakiet ma zostać zapisany do bazy (DNS, TCP, HTTP).
4. **Baza SQLite** (domyślnie `nazwa.db`) zapisuje szczegóły pakietów w tabeli `packets`.
5. **Pyzipper** służy do spakowania i zaszyfrowania bazy w formie `.zip` na żądanie.

5. Uruchomienie i użycie

1. Zainstaluj wymagane biblioteki (`requirements.txt` lub `pip install scapy pyzipper`).
2. Uruchom skrypt:

Python

```
python pcap_cli_app.py
```

3. Wybierz pliki `.pcap` do przetworzenia.
4. Wskaż, jaki rodzaj ruchu chcesz wyodrębnić.
5. Podaj (lub zaakceptuj) nazwę bazy `.db`.
6. Po zakończeniu analizy możesz zaszyfrować wynikowe `.db` do archiwum `.zip` z hasłem.

6. Wnioski i możliwość rozwoju

Rozwiązanie ułatwia szybką analizę ruchu sieciowego w plikach `.pcap`, zwłaszcza w kontekście ćwiczeń z cyberbezpieczeństwa czy inspekcji na potrzeby diagnostyczne.

W przyszłości można dodać:

- Integrację z narzędziami takimi jak Zeek/Suricata,
- Interfejs graficzny (GUI),
- Bardziej zaawansowane reguły rozpoznawania protokołów (np. TLS, SSH),
- System alertów i raportów (np. generowanie wykresów).