



## **Podstawy Teleinformatyki**

### **Sprawozdanie I**

✉ EvilPostman ✉

Graficzny modyfikator pakietów On-The-Fly

prowadzący:  
mgr. inż. Przemysław Walkowiak

Zespół
<b>Nazwisko i Imię</b>
Wieczorek Jarosław
Orczyk Krzysztof
Pawlaczyk Dominika

## Spis treści:

1. Charakterystyka ogólna projektu.	3
2. Uzasadnienie wyboru tematu.	3
3. Wymagania (z podziałem na aktorów).	3
4. Architektura systemu, narzędzia.	4
5. Opis najważniejszych protokołów.	4
6. Harmonogram.	4
7. Podział prac.	4

## **1. Charakterystyka ogólna projektu:**

Aplikacja ma za zadanie umożliwić użytkownikowi przechwycenie przesyłanych pakietów w sieci, a następnie przy pomocy graficznego interfejsu wykonać modyfikację ich parametrów w celu ponownej retransmisji. Odpowiednia konfiguracja systemu linux, a dokładne reguły iptables np. NFQUEUE, pozwoli aplikacji na przechwytywanie pakietów przesyłanych w sieci i modyfikowanie ich w locie ("on-the-fly"). Aplikacja ta może zostać wykorzystana jako serwer proxy lub uproszczona wersja programu Burp.

## **2. Uzasadnienie wyboru tematu:**

- Chęć zagłębienia się w bibliotekę Scapy i praktycznego jej wykorzystania.
- Potrzeba własnych narzędzi do wykonywania testów penetracyjnych.  
Z dużym naciskiem na możliwość szybkiej rozbudowy i dostosowania do własnych potrzeb.
- Tematyka związana z bezpieczeństwem komputerowym jest interesująca.
- Prostota rozwiązania - wykorzystanie takich metod jak command (zwraca konstruktory z parametrami dla przechwyconego pakietu).
- Duża liczba materiałów dotyczących programowania sieciowego.

## **3. Wymagania (z podziałem na aktorów):**

Aktor	Opis
Stacja A	Komputer roboczy z , którego pakiet jest przechwytywany.
Stacja B	Klient odbierający pakiet ze Stacji A.
Stacja C	Urządzenie, na którym uruchomiony jest program do modyfikacji pakietów. W celu przetestowania poprawności modyfikacji pakietów w locie zostanie wykonany atak ARP spoofing na Stacje A i Stacje B.

### **Funkcjonalne:**

- Przechwytywanie pakietów.
- Modyfikacja wybranych wartości pakietów.
- Retransmisja zmodyfikowanych pakietów.
- Interfejs graficzny użytkownika, umożliwiający zmiany wartości przechwyconych pakietów.
- Możliwość modyfikowania pakietów on-the-fly.

## **Niefunkcjonalne:**

- Działanie pod kontrolą systemu Linux. (np. Ubuntu 16.04).
- Wersja desktopowa aplikacji.
- Napisana w języku Python - wersja 3.X.
- Wykorzystana biblioteka Scapy - biblioteka służąca do sniffowania, a także generowania pakietów sieciowych różnych protokołów.
- Dodatkowa konfiguracja reguł iptables NFQUEUE - dla karty sieciowej w systemie Linux.
- Interfejs graficzny skonstruowany w Tkinter, PyQt5 lub Ncurses.  
Biblioteka Sphinx dla python 3.x - do tworzenia dokumentacji dla języka Python.

## **4. Architektura systemu, narzędzia (w tym również narzędzia wspomagające prace zespołowe biblioteki, kodeki).**

- Github - repozytorium, na którym umieszczane będą wszystkie materiały związane z realizacją projektu oraz kod.
- Spyder3 - lekkie funkcjonalne środowisko do pythona, zawiera wiele udogodnień takich jak inspekcja zmiennych w trakcie działania programu.
- Najbardziej popularne wersje linuxa: Ubuntu, Debian, Arch.

## **5. Opis najważniejszych protokołów.**

- Ethernet, UDP, TCP, IP, ARP, DHCP itd. Liczba obsługiwanych protokołów będzie zależna od szybkości realizowania prac.

## **6. Harmonogram:**

- Utworzenie repozytorium na Githubie.
- Zaprojektowanie systemu przechwytywania pakietów.
- Zaprojektowanie intuicyjnego interfejsu pozwalającego na modyfikacje wartości.
- Automatyczna zmiana pakietu “filtry” np. if ARP in pkt -> ustaw src IP na 18.18.18.18.

## **7. Podział prac:**

- Ze względu na zakres projektu, praca będzie podzielona po równo między wszystkie osoby. tzn. że każda z osób w zespole będzie pracowała wspólnie nad każdym elementem aplikacji.

## Przykład odebranego pakietu ARP:

Hardware Type	Protocol Type
Hardware length	Protocol length <b>Operation</b> Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)	
Sender protocol address (For example, 4 bytes for IP)	
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)	
Target protocol address (For example, 4 bytes for IP)	