Wydział Matematyki
i Nauk Informacyjnych
POLITECHNIKA WARSZAWSKA

Model uczenia maszynowego do wykrywania anomalii sieciowych

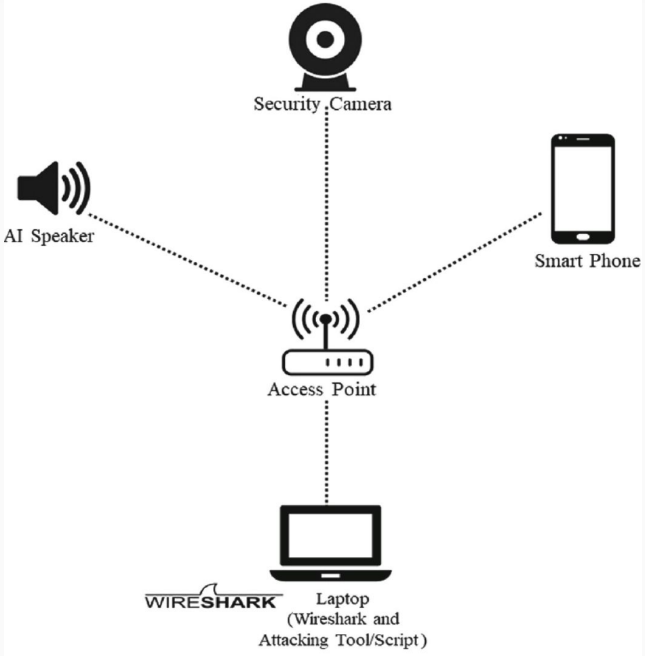
Inżynieria Cyberbezpieczeństwa

01.06.2022

Autorzy:
Adrian Stańdo, Mateusz Stączek, Kinga Ułasik

Plan

- Analiza problemu
- Metody rozwiązania
  - Wybrany zbiór danych
  - Wykorzystane narzędzia
  - Model uczenia maszynowego
- Prezentacja końcowej aplikacji
- Podsumowanie



Analiza problemu - gdzie występuje?

Każda sieć z dostępem z zewnątrz jest narażona na:

- Niepożądany dostęp do sieci
- Ataki
  - DDOS
  - MITM,
  - scan, malware i inne
- Inne nietypowe wzory w ruchu sieciowym

Analiza problemu - sposoby ochrony

- Odłączenie od sieci
- Firewall
- Oprogramowanie antywirusowe
- NIDS - network intrusion detection system

- **Śledzenie całego ruchu sieciowego**

Metody rozwiązania - wykorzystane narzędzia

Język programowania:

- Python

CICflowmeter

- Narzędzie do generowania i analizy ruchu sieciowego

Platforma GitLab - źródło, z którego pobraliśmy część kodów:

- https://gitlab.com/hieulw/cicflowmeter - CICflowmeter

Metody rozwiązania - wykorzystane narzędzia

Biblioteki:

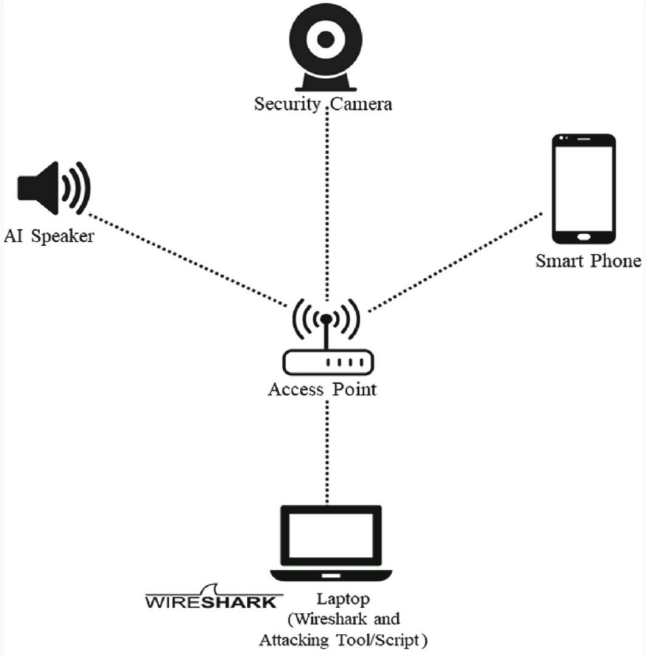
- GUI
  - pygubu - wizualne projektowanie aplikacji użytkownika
  - tkinter, PIL- do poprawek i edycji GUI
- Model
  - sklearn - posiada zaimplementowanych wiele modeli uczenia maszynowego
  - pickle - zapis/odczyt wytrenowanego modelu zapakowanego w aplikacji
- Inne
  - pandas, numpy

Metody rozwiązania - inspiracje

Artykuł: A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks

Autorzy: Imtiaz Ullah, Qusay H. Mahmoud

Data publikacji: maj 2020



Metody rozwiązania - zbiór danych

Przetwarzanie danych w artykule:

Wireshark -> pcap -> CICflowmeter -> csv flows -> Python script -> **clean csv**

Wynik:

- Plik csv, 80 kolumn
- Bez adresów IP
- **Bez wyjaśnień nazw kolumn**

Flow_Duration in	Tot_Fwd_Pkts i…	Tot_Bwd_Pkts i…	TotLen_Fwd_Pk…	TotLen_Bwd_Pk…	Fwd_Pk
75	1	1	982	1430	
5310	1	2	0	0	
141	0	3	0	2806	
151	0	2	0	2776	
153	2	1	886	420	

Metody rozwiązania - trenowanie modelu ML

Model do klasyfikacji: Random Forest

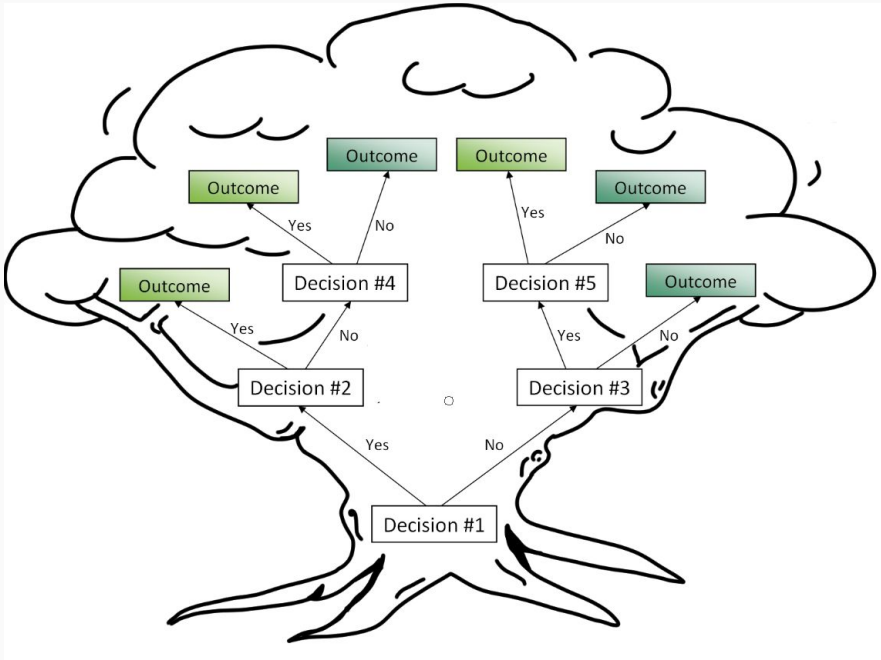
(wiele drzew trenowanych na części danych, a potem uśrednianie)

Kryterium podziału: gini

Max głębokość: bez limitu

Liczba drzew: 100

$$Gini = 1 - \sum_j p_j^2$$

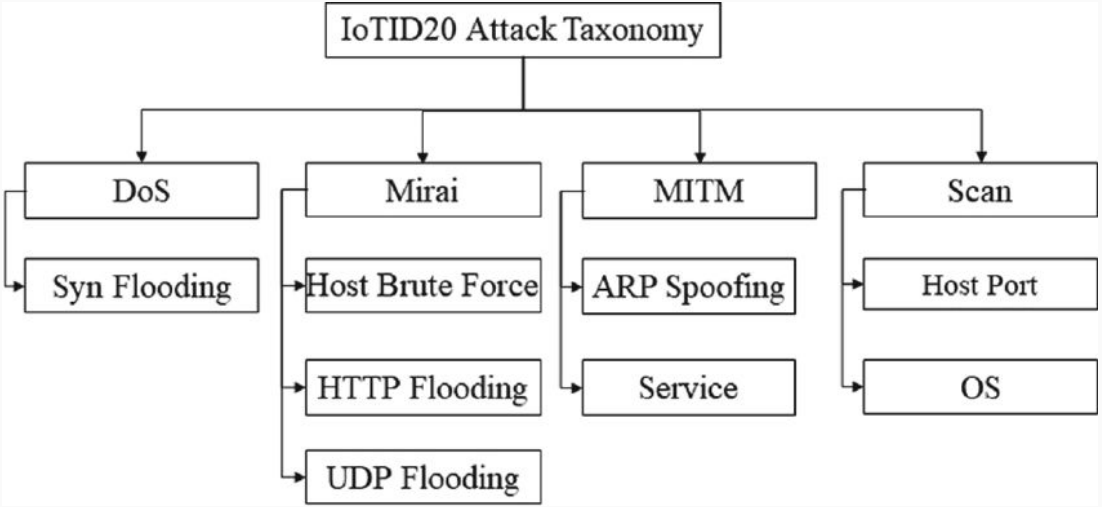


Metody rozwiązania - trenowanie modelu ML

Model: Random Forest

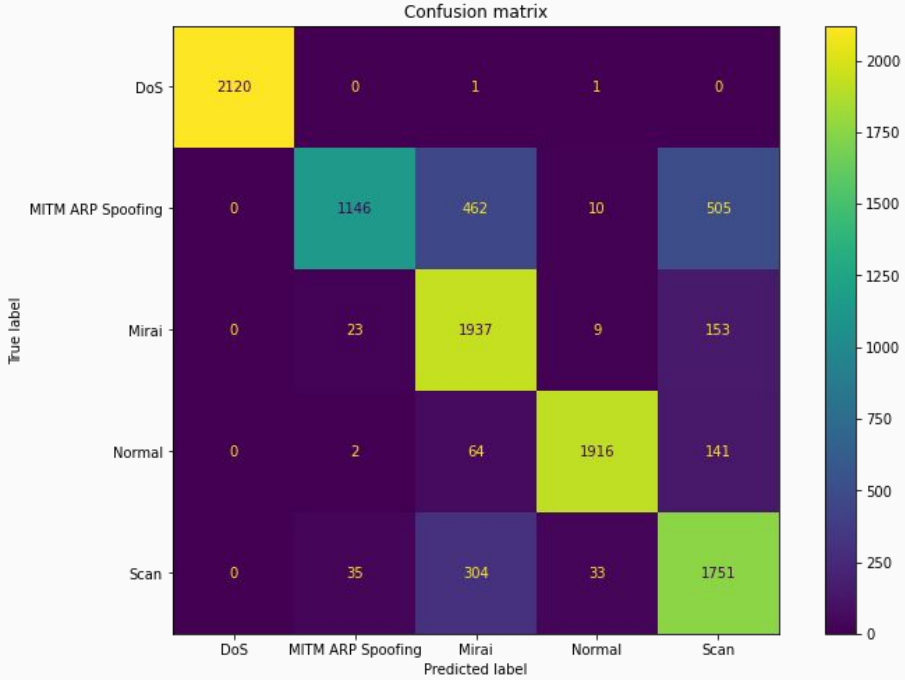
Liczba klas: 4

- DoS (Denial of Service)
- Mirai (botnet)
- MITM (Man in the Middle)
- Scan (port scanning)



Liczba podklas: 8 (nie używaliśmy)

Wyniki modelu



Wartości metryk:

Metryka	**Zbiór treningowy**	**Zbiór testowy**
Accuracy	0.8406	0.8358
F1 score weighted	0.8385	0.8332
ROC AUC weighted	0.9783	0.9782

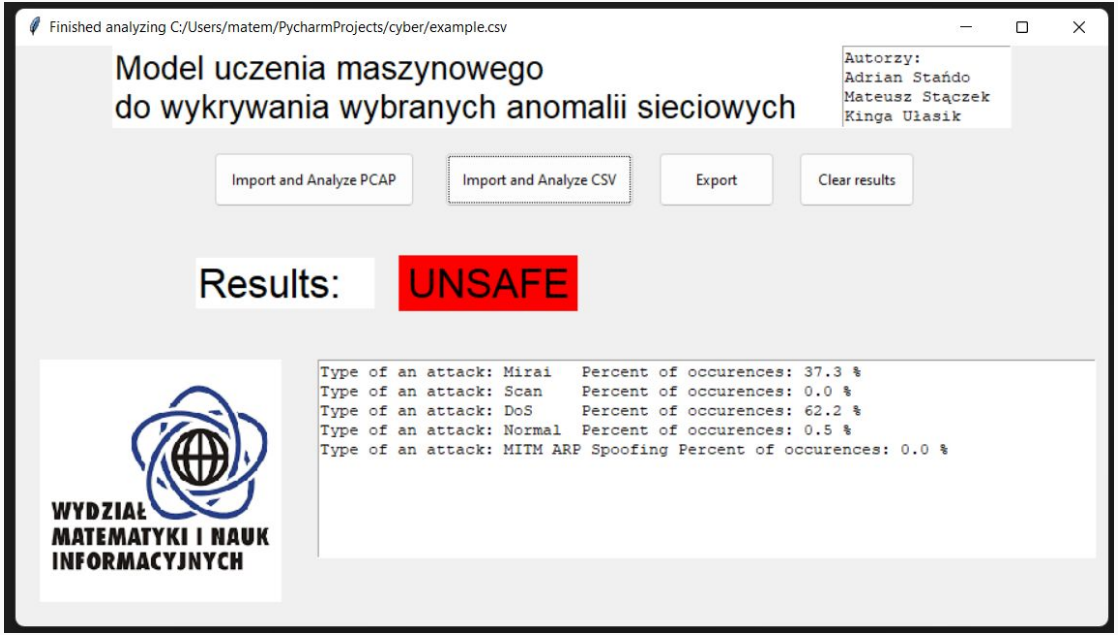
Aplikacja końcowa - działanie

Plik wejściowy:

- Plik *.pcap lub *.csv

Wyjście programu:

- Napis SAFE / UNSAFE
- Predykowany procent ataków
- Podział na typu ataków



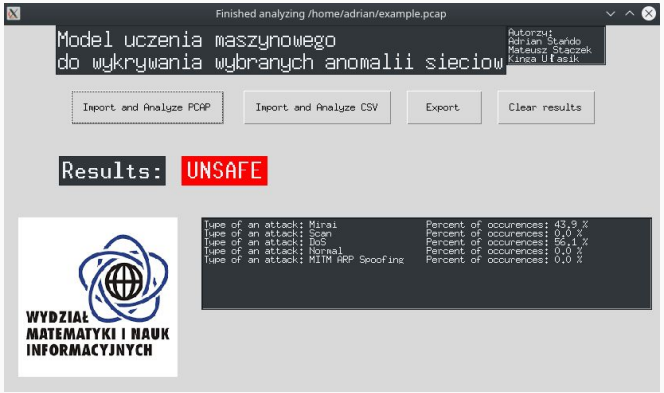
Prezentowana aplikacja działa poprawnie na systemie Linux.

Przykładowe użycie aplikacji

1. Zebranie pakietów np. przy pomocy programu Wireshark lub komendą: *sudo tcpdump -i any -w example.pcap*
2. Uruchomienie aplikacji komendą: *python3 gui_app.py*
3. Wczytanie wynikowego pliku do aplikacji.
4. Obserwacja wyników.

```
adrian@adrian-komputer:~$ sudo tcpdump -i any -w example.pcap
[sudo] hasło użytkownika adrian:
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
^C5613 packets captured
5786 packets received by filter
0 packets dropped by kernel
adrian@adrian-komputer:~$
```

```
Type of an attack: Mirai                Percent of occurences: 43.9 %
Type of an attack: Scan                 Percent of occurences: 0.0 %
Type of an attack: DoS                  Percent of occurences: 56.1 %
Type of an attack: Normal               Percent of occurences: 0.0 %
Type of an attack: MITM ARP Spoofing    Percent of occurences: 0.0 %
```



Na zakończenie…

Co może nasza aplikacja powiedzieć dowolnemu programowi w języku C?

- Nie masz klasy.

Podsumowanie

1. Stworzona przez nas aplikacja analizuje ruch sieciowy modelem uczenia maszynowego.
2. Użytkownik jasno widzi, że jest bezpieczny lub nie.
3. Wykrywane są różne rodzaje anomalii ruchu sieciowego.
4. Niestety, w trakcie testów okazało się, że nasz model działa poprawnie tylko w przypadku danych ze sztucznej sieci, na której był trenowany.
5. Na danych rzeczywistych, pochodzących z normalnego przeglądania Internetu, predykowane są wyłącznie ataki (brak normalnego ruchu).
6. Jest to znany problem w dziedzinie Data Science, który ma nazwę Data Drift.

Wydział Matematyki
i Nauk Informacyjnych

POLITECHNIKA WARSZAWSKA

Dziękujemy za uwagę!