

Politechnika Warszawska
Wydział Fizyki
Kryptografia i bezpieczeństwo informacji dla fizyków

Sprawozdanie z zadania nr. 8 na temat:

Pasywna identyfikacja systemów operacyjnych na podstawie własności pakietów sieciowych

Wykonali:

Maciej Czarnecki

Denys Morokov

Fizyka techniczna II stopień, 2 rok

1. Wstęp

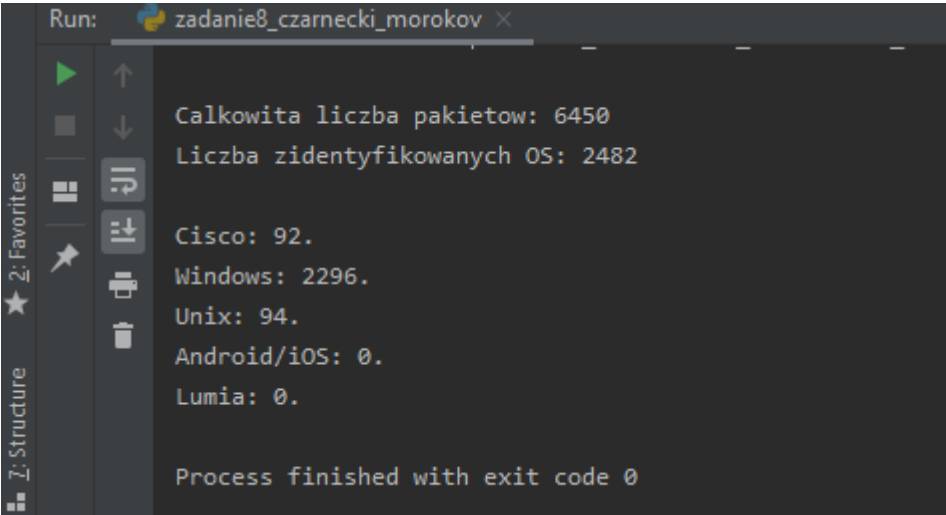
1.1. Cel: celem zadania jest zidentyfikowanie systemu operacyjnego w obserwowanym ruchu sieciowym na podstawie dowolnych parametrów pakietu lub zawartości i znaleźć i uruchomić program wyszukujący i identyfikujący OS hostów w sieci lokalnej.

1.2. Wykorzystana technologia: skrypt został zaimplementowany w Python wykorzystujący bibliotekę Scapy. Analizator ruchu sieciowego Wireshark. Pasywny skaner sieciowy p0f.

2. Opracowanie wyników

W celu stworzenia pliku, który będzie zawierał ruch sieciowy użyto Wireshark, który w ciągu 5-7 minut zapisywał ruch sieciowy, następnie uzyskane dane zapisano do pliku *ruch_sieciowy.pcapng*. W celu zidentyfikowania systemu operacyjnego w obserwowanym ruchu zastosowano dwie metody:

1. Napisano skrypt w Pythonie, który na podstawie parametru TTL (Time To Life) określano system operacyjny urządzenia, z którego dany pakiet został wysłany. Wynik dla wygenerowanego pliku wyżej przedstawia rysunek 1.

A screenshot of a terminal window titled 'Run: zadanie8_czarnecki_morokov'. The terminal displays the output of a script. It shows the total number of packets (6450) and the number of identified OSes (2482). It then lists the counts for various operating systems: Cisco (92), Windows (2296), Unix (94), Android/iOS (0), and Lumia (0). The process ends with 'Process finished with exit code 0'.

```
Run: zadanie8_czarnecki_morokov x
Calkowita liczba pakietow: 6450
Liczba zidentyfikowanych OS: 2482

Cisco: 92.
Windows: 2296.
Unix: 94.
Android/iOS: 0.
Lumia: 0.

Process finished with exit code 0
```

Rys. 1. Wynik identyfikacji OS

2. Użyto pasywny skaner sieciowy p0f.

```
master@master-VirtualBox:~/Pulpit$ p0f -r ruch_sieciowy.pcapng | grep os | sort | uniq -c
1 .-[ 10.2.11.20/51424 -> 52.114.77.33/443 (host change) ]-
1 .-[ 10.2.11.20/51425 -> 104.18.16.5/443 (host change) ]-
1 .-[ 10.2.11.20/51438 -> 52.114.77.33/443 (host change) ]-
1 .-[ 10.2.11.20/51439 -> 104.244.42.129/443 (host change) ]-
1 .-[ 10.2.11.20/51451 -> 5.135.104.110/443 (host change) ]-
1 [+] Closed 1 file descriptor.
25 | os      = ???
6 | os      = Linux 3.x
3 | os      = Windows 7 or 8
26 | os      = Windows NT kernel
8 | os      = Windows NT kernel 5.x
1 | raw_sig  = 1:?Cache-Control,Connection=[Keep-Alive],Accept=[/*/*],?If-Modified-Since,User-Agent,Host:Accept-Encoding,Accept-Language,
Accept-Charset,Keep-Alive:Microsoft-CryptoAPI/10.0
1 | raw_sig  = 1:Connection=[Keep-Alive],Accept=[/*/*],?If-Modified-Since,?If-None-Match,User-Agent,Host:Accept-Encoding,Accept-Language,
Accept-Charset,Keep-Alive:Microsoft-CryptoAPI/10.0
2 | raw_sig  = 1:Host,Connection=[keep-alive],Upgrade-Insecure-Requests=[1],DNT=[1],User-Agent,Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9],?Referer,Accept-Encoding=[gzip, deflate],Accept-Language=[ru,pl;q=0.9]:Accept-Charset,Keep-Alive:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36
5 | reason   = os_diff
master@master-VirtualBox:~/Pulpit$
```

Rys. 2. Wynik identyfikacji OS przy użyciu p0f

3. Podsumowanie

Dla uzyskanego ruchu sieciowego zidentyfikowano systemy operacyjne na dwa sposoby: poprzez napisanie własnego skryptu w Python i przez użycie programu p0f.