Sebastian Złotek

Metody i środki analizy ruchu sieciowego z uwzględnieniem aplikacji i trendów statystycznych

Rzeszów 2022

POLITECHNIKA RZESZOWSKA
im. IGNACEGO ŁUKASIEWICZA

WYDZIAŁ MATEMATYKI I FIZYKI STOSOWANEJ
POLITECHNIKI RZESZOWSKIEJ

Spis treści

WYDZIAŁ MATEMATYKI I FIZYKI STOSOWANEJ
POLITECHNIKI RZESZOWSKIEJ

Wprowadzenie

Analiza ruchu sieciowego to istotna kwestia szczególnie w kontekście informatyki śledczej czy monitoringu bezpieczeństwa sieci. Wiele firm stale inwestuje w infrastrukturę IT, aby dynamicznie się rozwijać i zwiększać swoją konkurencyjność. Prowadzi to do znaczącego wzrostu natężenia ruchu sieciowego. W systemie pojawia się coraz więcej urządzeń, a do tego rośnie ilość przechowywanych danych, które należy odpowiednio zabezpieczać. Utrzymanie jego wydajności i bezpieczeństwa na najwyższym poziomie nie polega jednak wyłącznie na inwestowaniu w nowe urządzenia. Problemy z działaniem często spowodowane są np.:

- Niepoprawną konfiguracją,
- „zatorami" w transferze danych,
- instalacją oprogramowania pochodzącego z niepewnych źródeł, zewnętrznymi atakami hakerów.

Im bardziej rozbudowana infrastruktura IT, tym trudniej jest ją kontrolować. Często okazuje się, że klasyczne oprogramowanie monitorujące
jest niewystarczające. Dużo więcej możliwości zapewniają narzędzia do inteligentnej analizy systemów informatycznych.

Jednak najsłabszym ogniwem każdego systemu informatycznego jest jego użytkownik. Pracownik firmy może np. zainstalować na komputerze oprogramowanie pochodzące z niepewnego źródła. Ruch sieciowy mogą spowalniać przestarzałe, dawno już nieaktualizowane programy.

Badany zbiór danych

Zbiór danych zawiera informacje dotyczące przepływu danych w sieci komputerowej. Dane posiadają około 60 kolumn i 500 tysięcy wierszy. Każda kolumna została opisana w tabeli poniżej (str.3).

Do przeprowadzenia analizy użyte zostało środowisko RStudio, oraz wersja języka programowania R 4.2.0.

	X.U.FEFF.ts	src_ip	src_port	dst_ip	dst_port	proto	service	duration
1	1554198358	3.122.49.24	1883	192.168.1.152	52976	tcp	-	80549.530260
2	1554198358	192.168.1.79	47260	192.168.1.255	15600	udp	-	0.000000
3	1554198359	192.168.1.152	1880	192.168.1.152	51782	tcp	-	0.000000
4	1554198359	192.168.1.152	34296	192.168.1.152	10502	tcp	-	0.000000
5	1554198362	192.168.1.152	46608	192.168.1.190	53	udp	dns	0.000549
6	1554198364	192.168.1.79	33269	192.168.1.255	15600	udp	-	0.000000
7	1554198364	192.168.1.152	34296	192.168.1.152	10502	tcp	-	0.000000
8	1554198364	192.168.1.152	1880	192.168.1.152	51782	tcp	-	0.000000
9	1554198369	192.168.1.152	1880	192.168.1.152	51782	tcp	-	0.000000
10	1554198369	192.168.1.152	34296	192.168.1.152	10502	tcp	-	0.000000

Widok danych w Rstudio (10 wierszy)

Tabela typów danych i kolumn

ID	Nazwa kolumny	Typ danych	Opis
1	ts	Time	Znacznik czasu połączenia między identyfikatorami przepływu
2	src_ip	String	Adres IP nadawcy (źródła)
3	src_port	Number	Numer portu nadawcy (źródła)
4	dst_ip	String	Adres IP odbiorcy
5	dst_port	Number	Numer portu odbiorcy
6	proto	String	Nazwa protokołu warstwy transportowej
7	service	String	Protokoły wykrywane dynamicznie, takie jak DNS, HTTP i SSL
8	duration	Number	Czas połączeń pakietowych, który jest szacowany przez odjęcie „czasu ostatniego widzianego pakietu” i „czasu pierwszego widzianego pakietu”
9	src_bytes	Number	Bajty wychodzące od nadawcy (źródła)

10	dst_bytes	Number	Bajty wychodzące od odbiorcy
11	conn_state	String	Status połączenia, S0 - połączenie bez informacji zwrotnej, S1 – połączenie powiodło się, REJ – próba połączenia została odrzucona
12	missed_bytes	Number	Liczba utraconych bajtów

13	src_pkts	Number	Liczba początkowa pakietów wychodzących od nadawcy (źródła)
14	src_ip_bytes	Number	Liczba oryginalnych bajtów IP, czyli całkowita długość pola nagłówka IP systemów źródłowych
15	dst_pkts	Number	Liczba początkowa pakietów wychodzących od odbiorcy
16	dst_ip_bytes	Number	Liczba bajtów docelowego IP, czyli całkowita długość pola nagłówka IP systemów docelowych.

17	dns_query	string	Nazwa domeny, do której kierowane są zapytania DNS
18	dns_qclass	Number	Wartości, które określają klasy zapytań DNS
19	dns_qtype	Number	Wartość określająca typy zapytań DNS
20	dns_rcode	Number	Wartości kodów odpowiedzi DNS
21	dns_AA	Bool	Informacja, czy odpowiedź DNS jest otrzymana w sposób bezpośredni (autorytatywny) bez pośrednika
22	dns_RD	Bool	Informacja czy żądane zapytanie rekursywne DNS zostało zwrócone
23	dns_RA	Bool	Informacja czy żądane zapytanie rekursywne DNS jest dostępne

24	dns_rejected	Bool	Informacja, czy zapytanie DNS zostało odrzucone

25	ssl_version	String	Wersja SSL dostępna na serwerze
26	ssl_cipher	String	Pakiet szyfrów SSL wybrany przez serwer
27	ssl_resumed	Bool	SSL wskazuje sesję, która może być używana do inicjowania nowych połączeń, gdzie T oznacza, że połączenie SSL jest inicjowane
28	ssl_established	Bool	Oznacza nawiązanie połączenia między dwiema stronami, gdzie T oznacza nawiązanie połączenia
29	ssl_subject	String	Certyfikat X.509 oferowany przez serwer
30	ssl_issuer	String	Informacja o właścicielu/inicjatorze sll i certyfikatu cyfrowego

31	http_trans_depth	Number	Pipelining HTTP, informacja o możliwości wysyłania kilku żądań jednocześnie
32	http_method	String	Informacje zwrotne HTTP takie jak GET, POST i HEAD
33	http_uri	String	URI użyte w żądaniu HTTP
35	http_version	String	Wykorzystywane wersje protokołu HTTP, takie jak V1.1
36	http_request_body_len	Number	Rzeczywiste rozmiary nieskompresowanej zawartości danych przesyłanych od klienta HTTP
37	http_response_body_len	Number	Rzeczywiste rozmiary nieskompresowanej zawartości danych przesyłanych z serwera HTTP
38	http_status_code	Number	Kody stanu zwracane przez serwer HTTP
39	http_user_agent	Number	Wartości nagłówka User-Agent w protokole HTTP
40	http_orig_mime_types	String	Uporządkowane wektory typu mime z systemu źródłowego w protokole HTTP

41	http_resp_mime_types	String	Uporządkowane wektory typu mime z systemu docelowego w protokole HTTP

42	weird_name	String	Nazwy zaistniałych anomalii/naruszeń związanych z protokołami
43	weird_addl	String	Dodatkowe informacje związane z anomaliami lub naruszeniami protokołu
44	weird_notice	bool	Wskazuje, czy naruszenie/nieprawidłowość zostały zwrócone do użytkownika

45	label	Number	Oznacza rekordy ataki oraz adresy, bez wykrytych ataków, gdzie 0 oznacza rekordy normalne(niezaatakowane), a 1 oznacza ataki
46	type	String	Oznacza kategorie ataków, takie jak adresy niezaatakowane(normal), DoS, DDoS i ataki typu backdoor.

Cel

Celem badania jest dokonanie analizy ruchu sieciowego. Sprawdzamy za metodą funkcji statystycznych takich jak średnia, odchylenie standardowe czy korelacja, ruch w sieci (przepustowość, przesyłane bajty).

Analizie podlegać będą dane przepływu sieciowego.

Sposoby monitorowania ruchu sieciowego

Informacje dostarczane przez oprogramowania służące do monitorowania sieci umożliwiają:

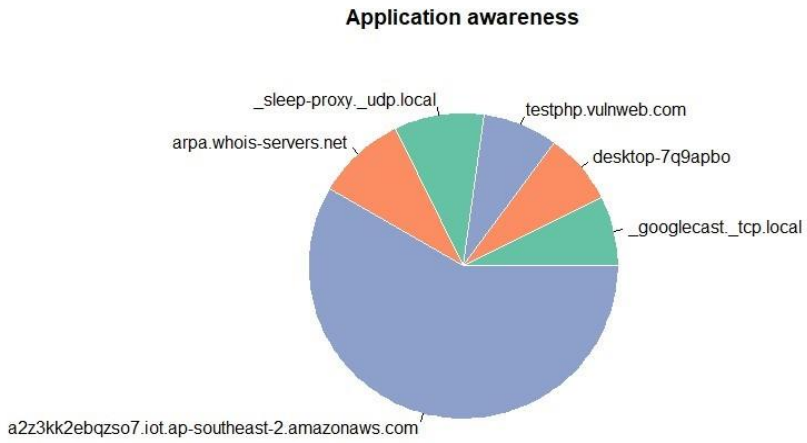
- Monitorowanie i prognozowanie trendów sieciowych,
- Identyfikacje głównych mówców, (określanie jacy użytkownicy i jakie aplikacje wykorzystują całą przepustowość), Monitorowanie przepustowości, Zarządzanie urządzeniami.

Przykładowymi programami są:

- NetFlow Traffic Analyzer,
- Wireshark,
- OpenNMS.

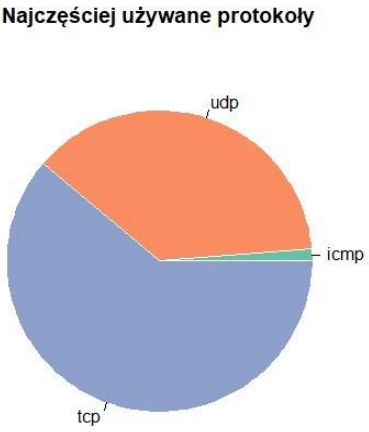
Analiza danych

Poniższe wykresy przedstawiają najczęściej używane strony internetowe. Z pierwszego wykresu wynika, że najczęściej odwiedzaną witryną jest amazonaws.com jest to ponad 50%, pozostałe strony mają mniej więcej taką liczbę połączeń.



Najczęściej używane strony internetowe

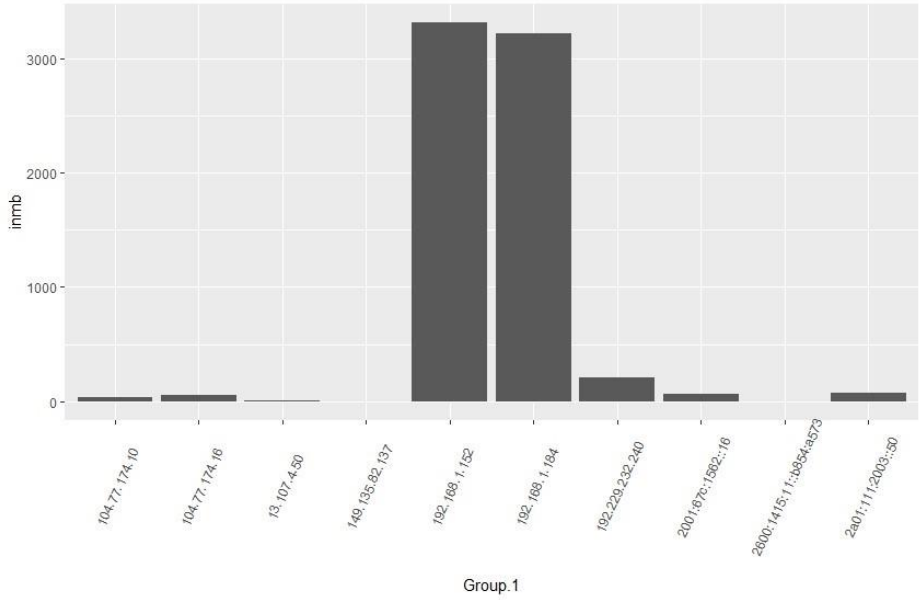
Na wykresie drugim możemy zauważyć, że najczęstszym protokołem używanym przez adresatów są protokoły transportowe TCP i UDP.



Najczęściej używane protokoły

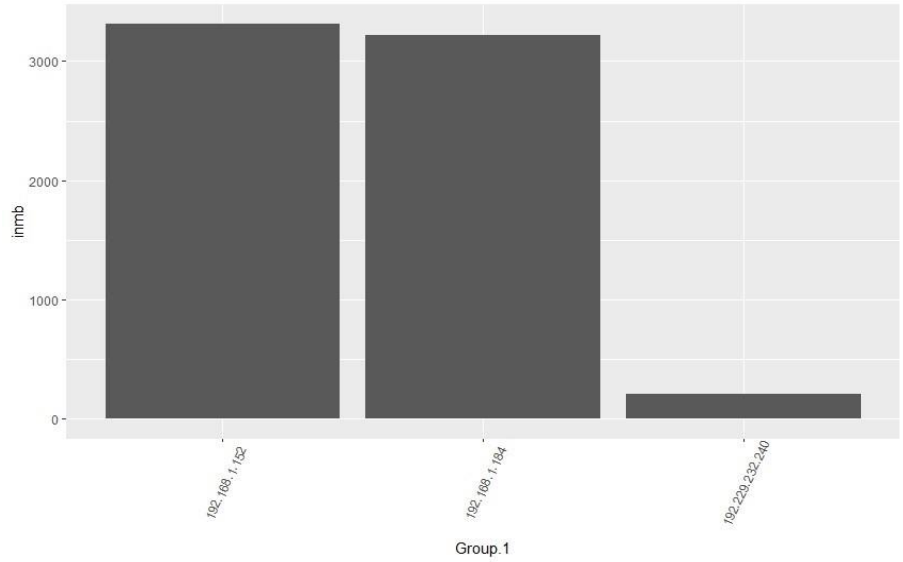
Wykresy zużycia pasma dla podanych adresów IP przedstawiają ilość przesyłanych danych w megabajtach. IP o adresach 192.168.1.152 i 192.168.1.164 zużywają ponad 3 GB, gdzie reszta adresów IP zużywa pasma w znacząco mniejszym stopniu.

Na podanym wykresie widzimy dysproporcje pomiędzy adresami IP zużywającymi najwięcej pasma (megabajtów), a adresami IP, które zużywają o wiele mniej transferu. Jak widać istnieje przepaść w transferze pomiędzy dziesięcioma adresami. Z czego można wywnioskować, iż nadawcy dwóch pierwszych znacznie częściej używają sieci do bardziej wymagających zadań.



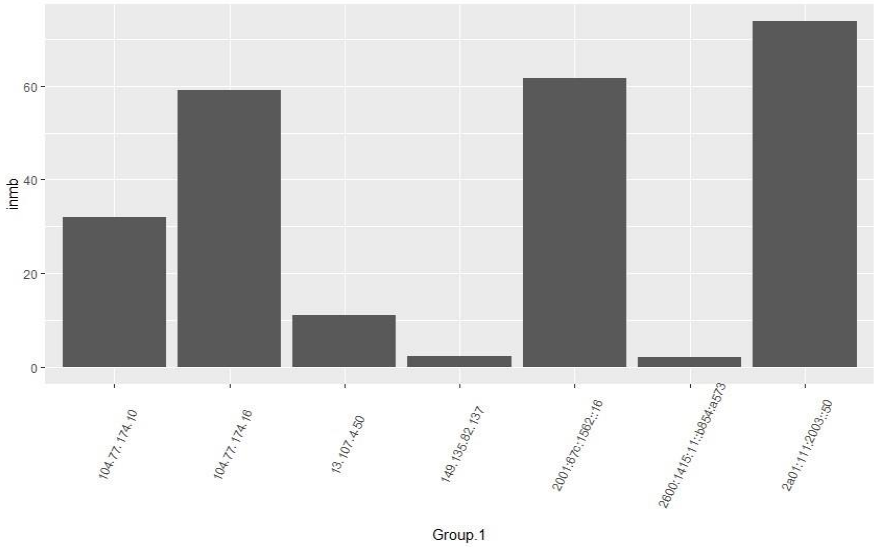
Suma zużytych megabajtów

Na następnym zrzucie ekranu widzimy wcześniejszą dysproporcję w zużyciu pasma. Porównując dwa adresy wykorzystujące najwięcej transferu z trzecim (Jest to ponad 15 razy więcej MB).



3 adresy IP które wykorzystują najwięcej transferu.

Pozostałe adresy nie różnią się już znacząco pod względem zużycia danych (Jest to maksymalnie kilkadziesiąt megabajtów).

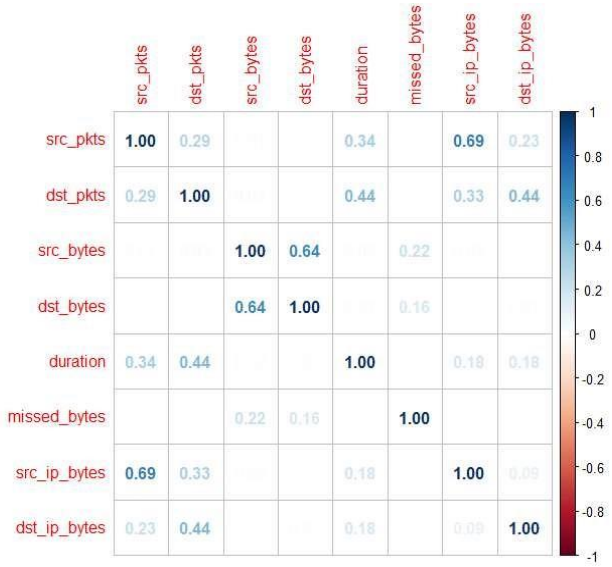


Porównanie zużytych danych w megabajtach

Korelacja – związek pomiędzy dwiema zmiennymi losowymi. zależność dwóch zmiennych oznacza, że znając wartość jednej z nich, dałoby się przynajmniej w niektórych sytuacjach dokładniej przewidzieć wartość drugiej zmiennej, niż bez tej informacji.

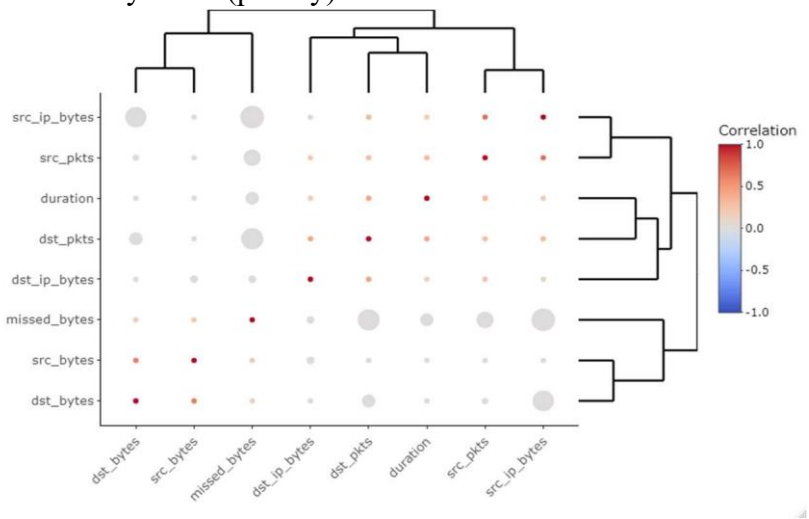
Na obu wykresach możemy zauważyć, że zmienne nie mają, aż tak dużego powiązania ze sobą, w związku z tym korelacje możemy uznać jako słabą, najmocniejsze

powiązania mają ze sobą dane nadawcy, lecz nadawca według korelacji ma naprawdę słabą korelację (około 0.2).



Heatmapa korelacji

Za pomocą drugiego wykresu możemy zobaczyć bardziej dokładne wartości korelacji. Jest to możliwe dzięki uruchomieniu kodu w środowisku Rstudio i najechanie kursorem na poszczególne wartości na wykresie (punkty).



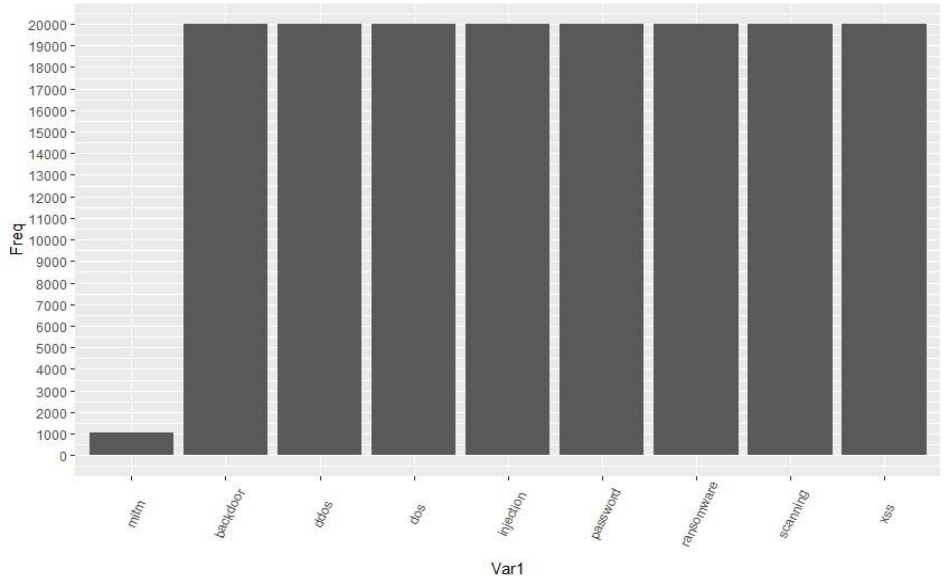
Heatmapa korelacji

Ilość ataków, prób włamań, z wykresu można wywnioskować, że było ich dokładnie 170 000, ponadto jeden adres IP mógł zostać zaatakowany kilka razy.

Wystepujące ataki w analizie:

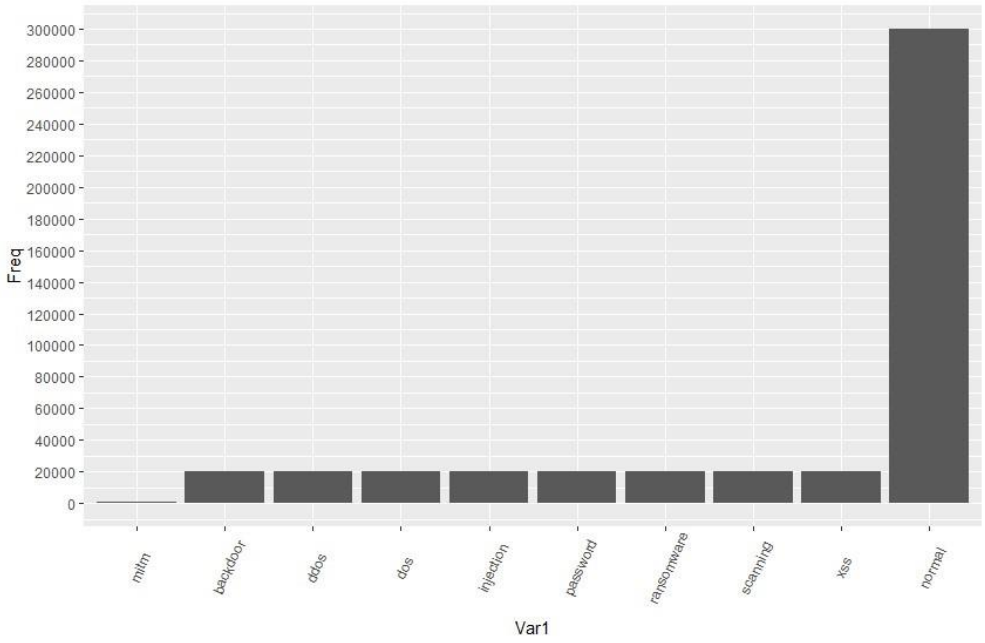
- MITM (Man in the middle) – to prosty atak sieciowy, którego zamysłem jest podsłuchiwanie wymiany danych pomiędzy stronami komunikacji lub ewentualnej jej modyfikacji.

- Backdoor – metoda pominięcia normalnych procesów uwierzytelniania lub szyfrowania systemu, produktu lub urządzenia (np. domowego routera).
- DDOS i DOS – te ataki to ciągłe wysyłanie określonych typów pakietów na adres IP atakowanego serwisu.
- Injection – jest to wstrzykiwanie złośliwych ciągów znaków, któee mogą przybierać wiele form. Mogą one wykorzystywać wbudowane funkcje oprogramowania rejestrującego dane w logach.
- Password – do tego ataku dochodzi gdy atakujący przy użyciu hasła próbuje uzyskać dostęp do kilku kont w jednej domenie.
- Ransonware - oprogramowanie, które blokuje dostęp do systemu komputerowego lub uniemożliwia odczyt zapisanych w nim danych.
- Scanning – jest to zazwyczaj przygotowanie do bardziej niebezpiecznego ataku sieciowego. Haker skanuje na komputerze docelowym porty UDP/TCP używające usługi sieciowe i określa stopień podatności komputera na bardziej niebezpieczne ataki sieciowe.
- XSS – polega on na wstrzyknięciu do przeglądarki ofiary fragmentu javascript bądź innego języka skryptowego, który może być uruchomiony w przeglądarce.



Sprawdzenie najczęstszych typów ataków na dany adres IP

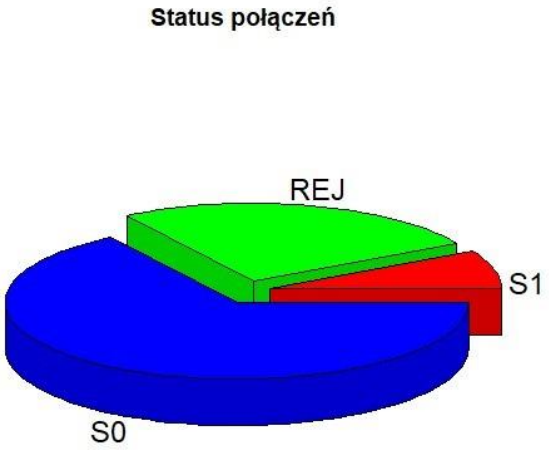
Porównanie niezaatakowanych adresów IP z próbami włamań. Większość adresów nie została zaatakowana co możemy wywnioskować z kolumny „normal”, jest to dokładnie 300 000 adresów IP, w porównaniu zaatakowanych jest około 170 000.



Sprawdzenie najczęstszych typów ataków na dany adres IP

Z wykresu możemy powiedzieć, że ponad połowa połączeń nie uzyskała informacji zwrotnej (S0, kolor niebieski), przy czym widać również, że najmniejszą wartością jest S1, która przedstawia udane połączenia. Około 25% prób połączeń zostało odrzuconych (REJ, kolor zielony). Statystyka jest dość zaskakująca, gdyż na co dzień przeważają połączenia zakończone sukcesem.

- S0 – połączenie bez informacji zwrotnej
- S1 – połączenie powiodło się
- REJ – próba połączenia została odrzucona



Wykres statusu połączeń

Wyniki

Poniższa tabela przedstawia dokładniejsze wyniki liczbowe wcześniej przedstawionego wykresu opisującego zużyte pasmo w megabajtach dla poszczególnych adresów IP.

	Group.1	inmb
1512	192.168.1.152	3311.80
1515	192.168.1.184	3214.51
1558	192.229.232.240	206.20
4091	2a01:111:2003::50	73.81
2427	2001:67c:1562::16	61.60
96	104.77.174.16	59.11
95	104.77.174.10	32.03
358	13.107.4.50	11.02
765	149.135.82.137	2.41
3977	2600:1415:11::b854:a573	2.17
3976	2600:1415:11::b854:a561	1.76
1526	192.168.1.33	1.08
4057	2606:2800:147:120f:30c:1ba0:fc6:3001	0.91
1532	192.168.1.79	0.75
4849	8.43.85.13	0.66
1517	192.168.1.190	0.61
1528	192.168.1.37	0.61
5055	91.189.92.20	0.53
2426	2001:67c:1560:8001::14	0.44
5056	91.189.92.38	0.38
4360	52.229.207.60	0.38
4055	2606:2800:147:120f:30c:1ba0:fc6:265a	0.37
1303	18.184.104.180	0.35
1521	192.168.1.195	0.34
391	13.35.146.38	0.34

Utracone bajty dla każdego IP (Nie uwzględnia wszystkich danych)

Średnia utraconych bajtów dla IP posortowane malejąco. W tabeli widać, że tworzące ramkę danych uwzględnia również adresy IPv6, a największa średnia wynosi 16147878.5 bajtów.

	Group.1	x
2427	2001:67c:1562::16	16147878.5000000
4091	2a01:111:2003::50	9674243.1250000
96	104.77.174.16	7748218.7500000
95	104.77.174.10	4797845.8571429
765	149.135.82.137	2532195.0000000
1558	192.229.232.240	1533457.8439716
3977	2600:1415:11::b854:a573	759925.3333333
3976	2600:1415:11::b854:a561	614400.0000000
1515	192.168.1.184	309377.0581000
4849	8.43.85.13	173464.2500000
4107	2a01:111:f330:1790::a01	124870.0000000
4057	2606:2800:147:120f:30c:1ba0:fc6:3001	86921.9090909
391	13.35.146.38	71772.0000000
2426	2001:67c:1560:8001::14	65843.5714286
4656	65.52.108.90	63188.0000000
1512	192.168.1.152	57058.2116099
2263	20.42.24.50	56381.0000000
2816	205.185.216.42	52648.0000000
839	151.101.2.49	48300.3333333
3850	23.206.242.18	46720.0000000
5044	91.189.88.149	45736.3333333
4106	2a01:111:f307:1794::a21	42686.0000000
5056	91.189.92.38	36367.0909091
4056	2606:2800:147:120f:30c:1ba0:fc6:3000	21860.2500000
4587	64.4.16.214	20705.5000000
3984	2600:1415:11:49b::25bb	20020.0000000

Średnia utraconych bajtów (Nie uwzględnia wszystkich danych)

W odchyleniu standardowym ukazanym w poniższej tabeli widzimy, że adresy IP ze średniej powtarzają się również w odchyleniu standardowym. Największe odchylenie w utraconych danych to około 30205694.2 bajtów.

	Group.1	x
2427	2001:67c:1562::16	30205694.1996173
1515	192.168.1.184	20622644.8399071
4091	2a01:111:2003::50	12676863.5754689
96	104.77.174.16	11172135.1559969
95	104.77.174.10	7565732.4494651
1558	192.229.232.240	6411590.3633040
1512	192.168.1.152	4433650.7244826
3976	2600:1415:11::b854:a561	746899.7426697
4849	8.43.85.13	346928.5000000
358	13.107.4.50	240524.2634884
2426	2001:67c:1560:8001::14	174205.7154323
391	13.35.146.38	160487.0708811
3977	2600:1415:11::b854:a573	153953.9133810
5056	91.189.92.38	111842.7961752
839	151.101.2.49	83658.6313558
5044	91.189.88.149	76358.7843430
4656	65.52.108.90	74565.8242897
5055	91.189.92.20	74067.9764129
410	13.35.149.60	42929.7893309
4056	2606:2800:147:120f:30c:1ba0:fc6:3000	40010.1940375
389	13.35.146.29	32215.5927550
5054	91.189.92.19	29475.8967163
4057	2606:2800:147:120f:30c:1ba0:fc6:3001	26533.2998342
5046	91.189.88.161	24826.5000000
3849	23.206.242.11	22514.6996871
2425	2001:67c:1360:8001::17	17136.0000000

Odchylenie utraconych danych (Nie uwzględnia wszystkich danych)

Korelacja Pearsona, którą testowaliśmy sprawdza nam wyniki korelacji do porównania z algorytmami korelacji do użytych wcześniej (str. 11) wykresów korelacji w heatmapie. W informacji zwrotnej widzimy wybrane kolumny, wartość t (która brana jest do obliczenia poziomu istotności p), właśnie ten poziom istotności (próg, wedle którego oceniamy z jakim prawdopodobieństwem różnice, które zaobserwowaliśmy są dziełem przypadku.) oraz liczbę swobody stopni df (z teoretycznego punktu widzenia stopnie swobody odnoszą się do liczby niezależnych obserwacji / wyników / porównań występujących w badanej przez nas grupie obserwacji). Dalej w informacji zwrotnej mamy hipotezę, czyli dopuszczenie pewnych prawidłowości w tym wypadku że korelacja nie będzie równa 0, a dalej za pomocą próbek mamy wartość korelacji, działa to wszystko tak samo dla wszystkich 3 przykładów.

```
> korpkts

        Pearson's product-moment correlation

data:  src_pkts$src_pkts and src_pkts$dst_pkts
t = 135.55, df = 197406, p-value < 0.00000000000000022
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2877661 0.2958374
sample estimates:
     cor
0.291807

> korbytes

        Pearson's product-moment correlation

data:  src_pkts$src_bytes and src_pkts$dst_bytes
t = 365.36, df = 197406, p-value < 0.00000000000000022
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6325117 0.6377751
sample estimates:
      cor
0.6351508

> kordest

        Pearson's product-moment correlation

data:  src_pkts$duration and src_pkts$dst_bytes
t = 4.4617, df = 197406, p-value = 0.000008135
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.005630462 0.014452165
sample estimates:
       cor
0.01004151
```

Sprawdzanie korelacji

Poniższe trzy listy przedstawiają dokładne wartości zużytych megabajtów, które zostały ujęte na ww. wykresach (str.9 i 10).

	Group.1	inmb
1512	192.168.1.152	3311.80
1515	192.168.1.184	3214.51
1558	192.229.232.240	206.20
4091	2a01:111:2003::50	73.81
2427	2001:67c:1562::16	61.60
96	104.77.174.16	59.11
95	104.77.174.10	32.03
358	13.107.4.50	11.02
765	149.135.82.137	2.41
3977	2600:1415:11::b854:a573	2.17

Zużyte megabajty (Top 10)

	Group.1	inmb
1512	192.168.1.152	3311.80
1515	192.168.1.184	3214.51
1558	192.229.232.240	206.20

Zużyte megabajty (Top 3)

	Group.1	inmb
4091	2a01:111:2003::50	73.81
2427	2001:67c:1562::16	61.60
96	104.77.174.16	59.11
95	104.77.174.10	32.03
358	13.107.4.50	11.02
765	149.135.82.137	2.41
3977	2600:1415:11::b854:a573	2.17

Zużyte megabajty (Top 7)

Zestawienie pakietów i długość trwania połączenia dla danych połączeń. Możemy zauważyć, że przy pierwszych połączeniach wprowadzanych do ramki liczba utraconych bajtów wynosiła zero, lecz we wcześniejszych analizach (str.9) widać, że nie we wszystkich połączeniach udało się bez utraty bajtów nawiązać połączenie. Dzięki długości trwania połączenia (jest to czas połączeń pakietowych, który jest szacowany przez odjęcie „czasu

ostatniego widzianego pakietu" i „czasu pierwszego widzianego pakietu") możemy zobaczyć, że długość połączeń jest dla większości krótka. Spis pakietów i bajtów ma za zadanie pokazać, że nie każde połączenie zużywa dużą ilość pasma, a w większości są to wcześniej wspomniane krótkie połączenia.

	src_pkts	dst_pkts	src_bytes	dst_bytes	duration	missed_bytes	src_ip_bytes	dst_ip_bytes
1	252181	2	1762852	41933215	80549.530260	0	14911156	236
2	1	2	0	0	0.028326	0	52	104
3	11	7	616	392	10.037018	0	924	588
4	4	4	224	224	3.009074	0	336	336
5	3	121942	33593178	0	22290.894560	0	1405	6360012
6	2	41109	66365725	0	45971.656380	0	4311	1648806
7	1	2	0	0	0.374279	0	40	80
8	1	2	0	0	0.371422	0	40	80
9	1	2	0	0	2.066015	0	40	80
10	1	2	0	0	1.214163	0	40	80
11	144	2	0	0	22.169931	0	8652	104
12	1	3	0	0	3.180604	0	52	144
13	36	2	0	0	5.964989	0	2196	104
14	266	1	0	0	32.292298	0	15656	52
15	219	3	0	0	25.425793	0	13328	156
16	1	2	0	0	1.896998	0	52	104
17	1	4	0	0	6.043497	0	52	208
18	1	4	0	0	6.043121	0	52	208
19	8	3	651	1	6.921624	0	2010	120
20	5	1	0	0	7.963314	0	260	52
21	1	4	0	0	4.322967	0	52	208
22	1	4	0	0	4.262362	0	52	208
23	1	4	0	0	4.321351	0	52	208
24	1	4	0	0	4.323088	0	52	208
25	4	1	0	0	2.191474	0	208	40
26	4	1	0	0	2.191369	0	208	40
27	27	2	0	0	4.845009	0	1512	104

Spis pakietów i czas trwania połączeń (Nie uwzględnia wszystkich danych)

Status połączenia pokazany na wykresie (str. 14), jest opisany również w tabeli poniżej, po wartości kolumny Freq możemy dokładnie określić ilość połączeń dla danego statusu, przy czym statusy i ich objaśnienie to:

- S0 – połączenie bez informacji zwrotnej
- S1 – połączenie powiodło się
- REJ – próba połączenia została odrzucona

Jak i na wykresie widać tu, że połączenie S0, bez informacji zwrotnej jest wartością największą z ilością 113495 wystąpień, odrzucenie połączenia REJ na drugim miejscu z 45036 wystąpieniami. Zaskakujące jest S1, czyli udane połączenie, które posiada przypisane do siebie „jedynie" 13843 rekordy.

	Var1	Freq
1	S1	13843
2	REJ	45036
3	S0	113495

Podsumowanie połączeń

Podsumowanie ramki danych wypisuje wszystkie dostępne kolumny z ramki danych, a następnie podsumowuje statystyki opisowe m.in. długość danej kolumny, średnią, medianę, wartości minimalne oraz maksymalne, kwantyle pierwszego i trzeciego rzędu, odchylenia standardowe, typy danych oraz klas. Jest to o tyle przydatne, że dzięki właśnie funkcji summary możemy w łatwy sposób analizować szybko dużą liczbę kolumn, mając przy tym ogromną ilość informacji do analizy, które później możemy przenieść w bardziej rozbudowane złożone funkcję i sprawdzić ich poprawność.

```
> summary(netflowds) #podsumowanie całej ramki danych
  X.U.FEFF.ts          src_ip             src_port       dst_ip             dst_port       proto              service            duration          src_bytes
 Min.   :1554198358   Length:461043      Min.   :    1   Length:461043      Min.   :    0   Length:461043      Length:461043      Min.   :    0.00   Min.   :         0
 1st Qu.:1554294210   Class :character   1st Qu.:34296   Class :character   1st Qu.:   53   Class :character   Class :character   1st Qu.:    0.00   1st Qu.:         0
 Median :1556240048   Mode  :character   Median :43530   Mode  :character   Median :  136   Mode  :character   Mode  :character   Median :    0.00   Median :         0
 Mean   :1555623627                      Mean   :39326                      Mean   : 7759                                         Mean   :    8.47   Mean   :    120993
 3rd Qu.:1556326776                      3rd Qu.:52725                      3rd Qu.: 9197                                         3rd Qu.:    0.02   3rd Qu.:        55
 Max.   :1556549129                      Max.   :65534                      Max.   :65530                                         Max.   :93516.93   Max.   :3890855126
   dst_bytes          conn_state          missed_bytes          src_pkts          src_ip_bytes         dst_pkts          dst_ip_bytes        dns_query
 Min.   :       0   Length:461043      Min.   :     0   Min.   :   0.00   Min.   :     0   Min.   :    0.00   Min.   :     0   Length:461043
 1st Qu.:       0   Class :character   1st Qu.:     0   1st Qu.:   1.00   1st Qu.:    40   1st Qu.:    0.00   1st Qu.:     0   Class :character
 Median :       0   Mode  :character   Median :     0   Median :   1.00   Median :    63   Median :    0.00   Median :     0   Mode  :character
 Mean   :  119693                      Mean   : 15895   Mean   :   7.18   Mean   :  1820   Mean   :    8.33   Mean   :  1953
 3rd Qu.:     250                      3rd Qu.:     0   3rd Qu.:   2.00   3rd Qu.:   186   3rd Qu.:    1.00   3rd Qu.:   353
 Max.   :3913853482                    Max.   :1854527046   Max.   :252181.00   Max.   :47367248   Max.   :313943.00   Max.   :86395523
  dns_qclass        dns_qtype         dns_rcode           dns_AA             dns_RD             dns_RA             dns_rejected       ssl_version        ssl_cipher
 Min.   :   0.0   Min.   :  0.000   Min.   :0.00000   Length:461043      Length:461043      Length:461043      Length:461043      Length:461043      Length:461043
 1st Qu.:   0.0   1st Qu.:  0.000   1st Qu.:0.00000   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Median :   0.0   Median :  0.000   Median :0.00000   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   : 139.9   Mean   :  5.631   Mean   :0.08372
 3rd Qu.:   0.0   3rd Qu.:  0.000   3rd Qu.:0.00000
 Max.   :32769.0   Max.   :255.000   Max.   :5.00000
 ssl_resumed        ssl_established    ssl_subject         ssl_issuer         http_trans_depth   http_method        http_uri           http_version
 Length:461043      Length:461043      Length:461043      Length:461043      Length:461043      Length:461043      Length:461043      Length:461043
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


 http_request_body_len http_response_body_len http_status_code   http_user_agent    http_orig_mime_types http_resp_mime_types  weird_name         weird_addl
 Min.   :   0.0000     Min.   :         0     Min.   :  0.0000   Length:461043      Length:461043        Length:461043        Length:461043      Length:461043
 1st Qu.:   0.0000     1st Qu.:         0     1st Qu.:  0.0000   Class :character   Class :character     Class :character     Class :character   Class :character
 Median :   0.0000     Median :         0     Median :  0.0000   Mode  :character   Mode  :character     Mode  :character     Mode  :character   Mode  :character
 Mean   :   0.0286     Mean   :        65     Mean   :  0.1145
 3rd Qu.:   0.0000     3rd Qu.:         0     3rd Qu.:  0.0000
 Max.   :2338.0000     Max.   :  13424384     Max.   :404.0000
 weird_notice          label                type
 Length:461043      Min.   :0.0000   Length:461043
 Class :character   1st Qu.:0.0000   Class :character
 Mode  :character   Median :0.0000   Mode  :character
                    Mean   :0.3493
                    3rd Qu.:1.0000
                    Max.   :1.0000
```

Podsumowanie ramki danych

Kod

```r
# Administracja systemów rozproszonych
# Koczynasz Jonasz, Nowak Filip, Złotek Sebastian P06

#Instalacja bibliotek
install.packages("ggplot2")
install.packages("RColorBrewer")
install.packages("dplyr")
install.packages("tidyverse")
install.packages("scales")
install.packages("installr")
install.packages("ggvis")
install.packages("ggpubr")
install.packages("corrplot")
install.packages("heatmaply")
install.packages("plotrix")

#Potrzebne biblioteki
library("plotrix")
library("heatmaply")
library("corrplot")
library("ggpubr")
library('ggvis')
library("dplyr")
library("tidyverse")
library("ggplot2")
library("RColorBrewer")

options(scipen=999) #usuwamy postać wykładniczą

setwd("D:/ASR_PROJEKT") #ustawiamy folder główny

netflowds <- read.csv("Train_Test_Network.csv", encoding = "UTF-8") #wczytujemy plik csv

netflowds

summary(netflowds) #podsumowanie całej ramki danych

(netflowds3 <-netflowds[netflowds$`dns_query`!="(empty)" & netflowds$`dns_query`!="-",]) #wykluczanie niepotrzebnych linii

sorttowanie <- (sort(table(netflowds3$'dns_query'),DECREASING=F)) #wybieramy najczęściej używane strony internetowe

paletabarw <- brewer.pal(6, "Set2") #kolory oraz opcje wykresu najczęściej używanych stron
(takiwykres <- pie(tail(sorttowanie),
                   border="white" ,
                   main = "Application awareness",
                   col = paletabarw))
dev.off()

sorttowanie2 <- (sort(table(netflowds$'proto'),DECREASING=F)) #wybieramy najczęściej używane protokoły
```

```r
paletabarw <- brewer.pal(3, "Set2") #kolory oraz opcje wykresu najczęściej używanych protokołów
(takiwykres <- pie(tail(sorttowanie2),
                   border="white" ,
                   main = "Najczęściej używane protokoły",
                   col = paletabarw))

dev.off() #zapisanie wykresu

mean(netflowds$'duration') #średnie, sumy oraz odchylenia standardowe utraconych bajtów i czasu trwania połączenia

mean(netflowds$'missed_bytes')

sd(netflowds$'duration')

sd(netflowds$'missed_bytes')

sum(netflowds$'duration')

sum(netflowds$'missed_bytes')

#średnia zużytych bajtów
bytesperipmean <- (aggregate(netflowds$'missed_bytes', by=list(netflowds$'dst_ip'), FUN = mean))

bytesperipmeansorted <- bytesperipmean[with(bytesperipmean, order(x, Group.1, decreasing = T)), ]

#odchylenie standardowe zużytych bajtów
bytesperipsd <- (aggregate(netflowds$'missed_bytes', by=list(netflowds$'dst_ip'), FUN = sd))

bytesperipsdsorted <- bytesperipsd[with(bytesperipsd, order(x, Group.1, decreasing = T)), ]

#suma zużytych bajtów
bytesperip <- (aggregate(netflowds$'missed_bytes', by=list(netflowds$'dst_ip'), FUN = sum))

bytesperipsorted <- bytesperip[with(bytesperip, order(x, Group.1, decreasing = T)), ]

bytesperipsorted <- bytesperipsorted %>% #zaokraglenie i zamiana bajtów na megabajty
  mutate(inmb = round((bytesperipsorted$x / 1048576), 2))

bytesperipsorted$x <- NULL

TES1 <- head(bytesperipsorted, n = 10)

TES2 <- head(bytesperipsorted, n = 3)

TES3 <- tail(TES1, n = 7)

str(bytesperipsorted)

ggplot(TES1, aes(Group.1, inmb)) + geom_bar(stat = 'identity') + theme(axis.text.x = element_text(angle=65, vjust=0.6)) #top 10 zużycia pasma po IP
```

```r
ggplot(TES2, aes(Group.1, inmb)) + geom_bar(stat = 'identity') + theme(axis.text.x = element_text(angle=65, vjust=0.6)) #top 3 zużycia pasma po IP

dev.off()

ggplot(TES3, aes(Group.1, inmb)) + geom_bar(stat = 'identity') + theme(axis.text.x = element_text(angle=65, vjust=0.6)) #porównanie zużytych megabajtów d

dev.off()

view(head(bytesperipsorted, n = 50))

src_pkts <- netflowds %>% #informacje o pakietach, bajtach i długości połączenia
  select(src_pkts, dst_pkts, src_bytes, dst_bytes, duration,  missed_bytes, src_ip_bytes, dst_ip_bytes ) %>%
  filter(src_pkts >0 & dst_pkts >0 )

cor(src_pkts$src_pkts,src_pkts$dst_pkts, method="spearman") #korelacja pakietów

length(src_pkts$src_pkts)
length(src_pkts$dst_pkts)

#wykresy korelacji
corrplot(corr = cor(src_pkts), method='number') #heatmapa korelacji

str(netflowds)

r <- cor(src_pkts) #dokładniejszy opis wykresu i wartości korelacji
cor.test.p <- function(x){
  FUN <- function(x, y) cor.test(x, y)[["p.value"]]
  z <- outer(
    colnames(x),
    colnames(x),
    Vectorize(function(i,j) FUN(x[,i], x[,j]))
  )
  dimnames(z) <- list(colnames(x), colnames(x))
  z
}
p <- cor.test.p(src_pkts)

heatmaply_cor(
  r,
  node_type = "scatter",
  point_size_mat = p,
  point_size_name = "p value",
  label_names = c("x", "y", "Correlation")
)

#-----------------------------------------------

koripbytes <- cor.test(src_pkts$src_ip_bytes, src_pkts$dst_ip_bytes, method="pearson") #sprawdzanie czy wartości korelacji są poprawne
koripbytes
```

```r
koripbytes <- cor.test(src_pkts$src_ip_bytes, src_pkts$dst_ip_bytes, method="pearson") #sprawdzanie czy wartości korelacji są poprawne
koripbytes

korpkts <- cor.test(src_pkts$src_pkts, src_pkts$dst_pkts, method="pearson")

korpkts

korbytes <- cor.test(src_pkts$src_bytes, src_pkts$dst_bytes, method="pearson")

korbytes

kordest <- cor.test(src_pkts$duration, src_pkts$dst_bytes, method="pearson")

kordest

#sprawdzenie najczęstszych typów ataków na dany adres IP
type1 <- netflowds[netflowds$'type'!="normal",]

typeanalysis <- (sort(table(type1$'type'),DECREASING=F))

ggplot(data.frame(typeanalysis), aes(Var1, Freq)) + geom_bar(stat = 'identity') + theme(axis.text.x = element_text(angle=65, vjust=0.6)) + scale_y_continuous(breaks=seq(0,20000,by=1000))

dev.off()

typeanalysis2 <- (sort(table(netflowds$'type'),DECREASING=F))

ggplot(data.frame(typeanalysis2), aes(Var1, Freq)) + geom_bar(stat = 'identity') + theme(axis.text.x = element_text(angle=65, vjust=0.6)) + scale_y_continuous(breaks=seq(0,300000,by=20000))

dev.off()

#sprawdzenie czy dane połączenie powiodło się, gdzie:
#S0 połączenie bez informacji zwrotnej
#S1 połączenie powiodło się
#REJ próba połączenia została odrzucona
status_polaczenia <- (sort(table(netflowds$'conn_state'),DECREASING=F))

view(status_polaczenia)

stat_data <- data.frame(status_polaczenia)

namedata <- stat_data %>%
  select(Var1) %>%
  filter((Var1 == "S0") | (Var1 == "S1") | Var1 == "REJ") #wybieranie danych kolumn i wierszy

freqdata <- stat_data %>%
  filter((Var1 == "S0") | (Var1 == "S1") | Var1 == "REJ") %>%
  select(Freq)

podsumowanie_polaczen <- stat_data %>% #spis polaczen
  select(Var1,Freq)%>%
  filter(Var1 == "S0" | Var1=="S1" | Var1=="REJ")

pie3D(as.numeric(unlist(freqdata)),labels = as.character(unlist(namedata)),explode = 0.1, main = "Status połączeń ")

dev.off()
```

Wnioski

Dzięki analizie sieci z wykorzystaniem języka programowania R, przeanalizowaliśmy przepustowość sieci, typy ataków hakerskich, zużyte megabajty, a także najczęściej używane protokoły .

Duże przedsiębiorstwa z rozwinięta infrastrukturą powinny używać oprogramowania do monitorowania ruchu sieciowego w czasie rzeczywistym, aby móc zarządzać ruchem sieciowym i wykorzystaniem łącza.

Źródła

- https://stovaris.pl/analiza-ruchu-sieciowego-dlaczego-jest-tak-potrzebna/
- https://www.manageengine.com/pl/netflow/network-traffic-monitor.html
- https://pl.joecomp.com/free-network-internet-traffic-monitor-tools-for-windows-10-8-7
- https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i?path=%2FTrain_Test_datasets%2FTrain_Test_Network_dataset
- https://www.naukowiec.org/wiedza/statystyka/stopnie-swobody_718.html