

1. Traceroute

Używając komendy `tracert -d` (gdzie `-d` oznacza, że ma nie rozpoznawać adresów jako nazw hostów), można zobaczyć jaką drogę pokonuje pakiet i ile routerów spotyka.

```
Sposób użycia: tracert [-d] [-h maks_przes] [-j lista_hostów] [-w limit_czasu]
                 [-R] [-S adres_źródłowy] [-4] [-6] nazwa_celu

Opcje:
-d             Nie rozpoznawaj adresów jako nazw hostów.
-h maks_przes  Maksymalna liczba przeskoku w poszukiwaniu celu.
-j lista_hostów Swoobodna trasa źródłowa według listy lista_hostów
                (tylko IPv4).
-w limit_czasu Limit czasu oczekiwania na odpowiedź w milisekundach.
-R             Śledź ścieżkę błędzenia (tylko IPv6).
-S adres_źródłowy Adres źródłowy do użycia (tylko IPv6).
-4            Wymuś używanie IPv4.
-6            Wymuś używanie IPv6.
```

Poniżej inne dodatkowe opcje tej komendy:

W tym przypadku pakiet dociera do witryny `google.pl` w 9 przeskoku, jednak liczba ta może być różna przy każdym wywołaniu komendy oraz może mijąć inne routery.

Jeśli wśród wyników pojawia się gwiazdka (*), oznacza to, że nie było odpowiedzi na wysłany pakiet, a powodem mógł być przykładowo problem z siecią.

```
Śledzenie trasy do www.google.pl [2a00:1450:401b:808::2003]
z maksymalną liczbą 30 przeskoku:

  1      3 ms      3 ms      3 ms      2a02:a317:a040:8280:ae22:5ff:fe19:d9a8
  2      267 ms   32 ms   13 ms      2a02:a304:0:177::1
  3       17 ms    9 ms   10 ms      2a02:a300:180:40:0:1201:0:1
  4       15 ms   25 ms   20 ms      2001:730:2c00:5474:8034
  5       15 ms   16 ms   18 ms      2001:730:2c00:5474:8035
  6       16 ms   18 ms   19 ms      2001:4860:1:1:0:1aae:0:29
  7       14 ms   18 ms   14 ms      2001:4860:0:1184::1
  8       27 ms   16 ms   15 ms      2001:4860:0:1:30dd
  9       22 ms   21 ms   21 ms      2a00:1450:401b:808::2003

Śledzenie zakończone.
```

W Wiresharku zawężając poszukiwania przy pomocy `icmp` lub w tym przypadku `icmpv6`, można zobaczyć jak wygląda badanie trasy pakietów.

The image shows a Wireshark packet capture of an ICMPv6 traceroute. The packet list pane displays 204 packets, with the first 9 being echo requests and the rest being replies. The packet details pane shows the selected packet (No. 204) with its source and destination addresses. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
137	15.698997	2a02:a300:180:40:0:1201:0:1	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
138	15.699494	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=365, hop limit=3 (no response found!)
139	15.709492	2a02:a300:180:40:0:1201:0:1	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
145	16.700167	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=366, hop limit=4 (no response found!)
146	16.710302	2001:730:2c00:5474:8034	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	206	Time Exceeded (hop limit exceeded in transit)
147	16.716544	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=367, hop limit=4 (no response found!)
148	16.721687	2001:730:2c00:5474:8034	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	206	Time Exceeded (hop limit exceeded in transit)
149	16.742242	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=368, hop limit=4 (no response found!)
150	16.763161	2001:730:2c00:5474:8034	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	206	Time Exceeded (hop limit exceeded in transit)
153	17.744178	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=369, hop limit=5 (no response found!)
154	17.769007	2001:730:2c00:5474:8035	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
155	17.769716	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=370, hop limit=5 (no response found!)
156	17.777471	2001:730:2c00:5474:8035	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
157	17.778818	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=371, hop limit=5 (no response found!)
158	17.796598	2001:730:2c00:5474:8035	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
169	18.779246	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=372, hop limit=6 (no response found!)
170	18.795782	2001:4860:1:1:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
171	18.796394	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=373, hop limit=6 (no response found!)
172	18.814451	2001:4860:1:1:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
173	18.814994	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=374, hop limit=6 (no response found!)
174	18.834995	2001:4860:1:1:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
175	19.420438	fe80::ae22:5ff:fe19:d9a8	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	86	Neighbor Solicitation for 2a02:a317:a040:8280:95ca:b72e:de80:2615 from ac:22:05:19:d9:a8
176	19.420579	2a02:a317:a040:8280:95ca:b72e:de80:2615	fe80::ae22:5ff:fe19:d9a8	ICMPv6	86	Neighbor Advertisement 2a02:a317:a040:8280:95ca:b72e:de80:2615 (sol, ovr) is at a0:f3:c1:2e:95:6c
179	19.815301	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=375, hop limit=7 (no response found!)
180	19.829777	2001:4860:0:11:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
181	19.829906	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=376, hop limit=7 (no response found!)
182	19.847079	2001:4860:0:11:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
183	19.848449	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=377, hop limit=7 (no response found!)
184	19.852878	2001:4860:0:11:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
189	20.849402	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=378, hop limit=8 (no response found!)
190	20.876871	2001:4860:0:11:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
191	20.877467	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=379, hop limit=8 (no response found!)
192	20.893481	2001:4860:0:11:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
193	20.893947	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=380, hop limit=8 (no response found!)
195	20.909794	2001:4860:0:11:0:1aae:0:29	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	174	Time Exceeded (hop limit exceeded in transit)
199	21.895430	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=381, hop limit=9 (reply in 200)
200	21.918093	2a00:1450:401b:808::2003	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	126	Echo (ping) reply id=0x0001, seq=381, hop limit=116 (request in 199)
201	21.918707	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=382, hop limit=9 (reply in 202)
202	21.939718	2a00:1450:401b:808::2003	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	126	Echo (ping) reply id=0x0001, seq=382, hop limit=116 (request in 201)
203	21.940323	2a02:a317:a040:8280:95ca:b72e:de80:2615	2a00:1450:401b:808::2003	ICMPv6	126	Echo (ping) request id=0x0001, seq=383, hop limit=9 (reply in 204)
204	21.961962	2a00:1450:401b:808::2003	2a02:a317:a040:8280:95ca:b72e:de80:2615	ICMPv6	126	Echo (ping) reply id=0x0001, seq=383, hop limit=116 (request in 203)

Next Header: ICMPv6 (58)
Hop Limit: 1
Source Address: 2a02:a317:a040:8280:95ca:b72e:de80:2615
Destination Address: 2a00:1450:401b:808::2003

Internet Control Message Protocol v6

0000 ac 22 05 19 d9 a8 a0 f3 c1 2e 95 6c 86 dd 00 00:..1..
0000 00 00 00 48 3a 01 2a 02 a3 17 a0 40 82 80 95 ca ...H:.*...@...
0020 b7 2e de 80 26 15 2a 00 14 50 40 1b 08 08 00 00 ...&.*...P@...
0030 00 00 00 20 03 00 00 96 36 00 01 01 65 00 00:..6....

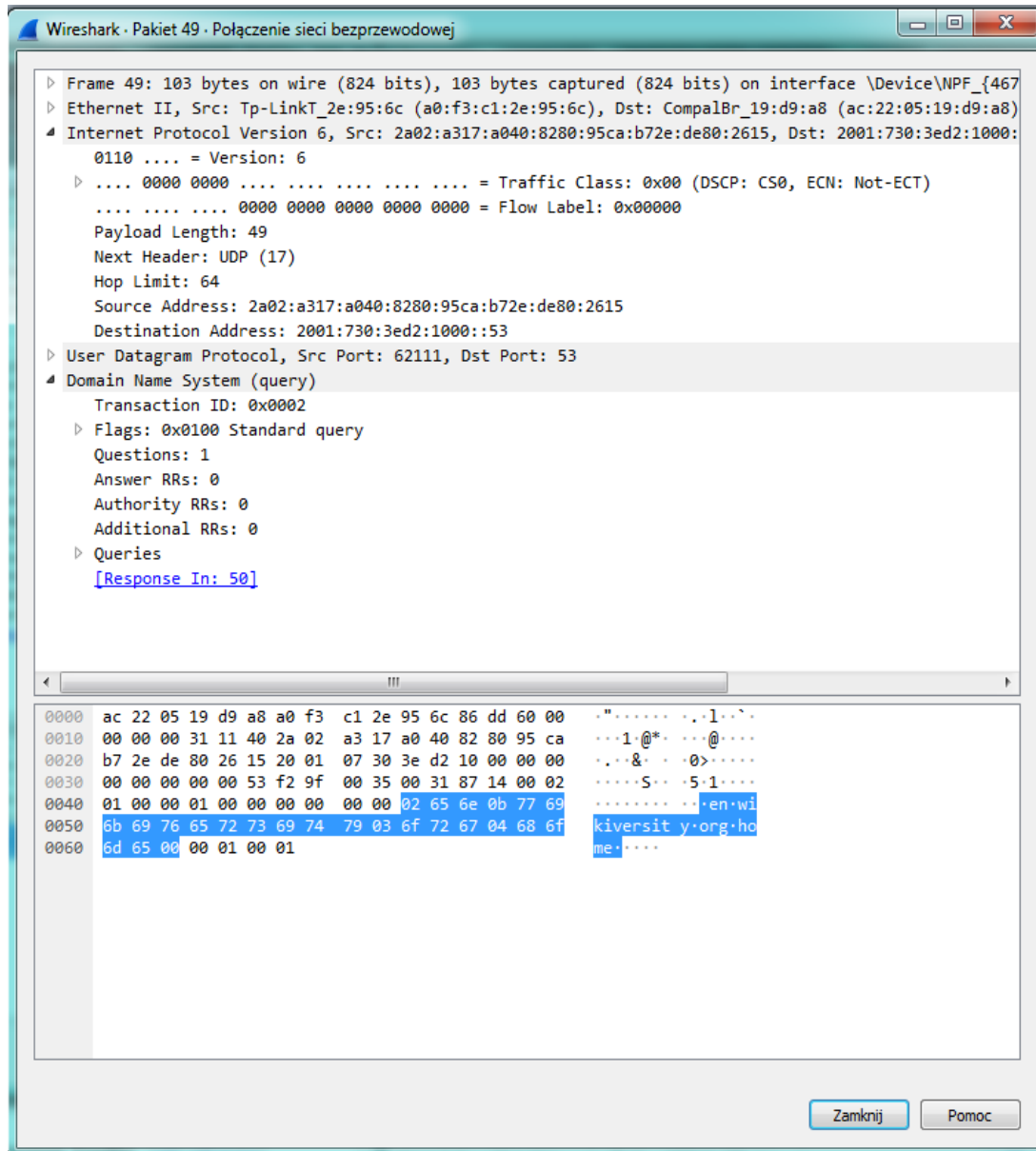
Połączenie sieci bezprzewodowej: 'live capture in progress'

Pakietów: 910 · Wyświetlonych: 68 (7.5%)

Profil: Default

[illegible]

Sprawdzając pakiety, można zauważyć, że Source Address odpowiada naszemu adresowi IP, natomiast Destination Address odpowiada adresowi IP serwera DNS. Ponadto można sprawdzić, że adres DNS nie jest szyfrowany, tj. jest zawarty w bajtach 74-98, które przyporządkowane są nazwie.



Analiza ruchu HTTP:

W przeglądarce wpisujemy wybrany adres strony internetowej korzystającej z HTTP.

ip.addr == 69.94.77.202						
No.	Time	Source	Destination	Protocol	Length	Info
2035	15.706193	192.168.0.17	69.94.77.202	TCP	66	65296 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2036	15.752351	69.94.77.202	192.168.0.17	TCP	62	443 → 65296 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1420 WS=128
2037	15.752461	192.168.0.17	69.94.77.202	TCP	54	65296 → 443 [ACK] Seq=1 Ack=1 Win=17040 Len=0
2038	15.752857	192.168.0.17	69.94.77.202	TLSv1.2	236	Client Hello
2040	15.807130	69.94.77.202	192.168.0.17	TCP	56	443 → 65296 [ACK] Seq=1 Ack=183 Win=30336 Len=0
2041	15.809612	69.94.77.202	192.168.0.17	TLSv1.2	1474	Server Hello
2042	15.810717	69.94.77.202	192.168.0.17	TCP	1474	443 → 65296 [ACK] Seq=1421 Ack=183 Win=30336 Len=1420 [TCP segment of a reassembled PDU]
2043	15.810717	69.94.77.202	192.168.0.17	TLSv1.2	481	Certificate, Server Key Exchange, Server Hello Done
2044	15.810771	192.168.0.17	69.94.77.202	TCP	54	65296 → 443 [ACK] Seq=183 Ack=3268 Win=17040 Len=0
2045	15.816351	192.168.0.17	69.94.77.202	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2046	15.868318	69.94.77.202	192.168.0.17	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message

Trzy pierwsze pakiety zarejestrowane przez Wiresharka to TCP three-way handshake. SYN wysłany do celu (69.94.77.202), serwer odsyła segmenty SYN oraz ACK, a następnie odpowiedź potwierdzenia odebrania segmentu wysłana przez nadawcę.

Następnie poprzez protokół TLS klient wysyła wiadomość (Client Hello) wraz z wersją protokołu i innymi informacjami, na którą odpowiada serwer odsyłając Server Hello wraz z informacjami o połączeniu. Następuje wymiana klucza publicznego serwera i przejście do kolejnej fazy łączenia (Server Hello Done).

Klient wysyła wstępny klucz sesji, zaszyfrowany za pomocą otrzymanego klucza publicznego, po czym klient i serwer generują klucz sesji. Od momentu nadania Change Cipher Spec możliwe jest przesyłanie zaszyfrowanych danych.

3. Cache

Przy pierwszym wejściu na wybraną stronę, można zaobserwować TCP handshake, hosty przesyłają kolejno segmenty SYN, SYN ACK oraz ACK, po czym przesyłane są Client Hello etc. oraz dane.

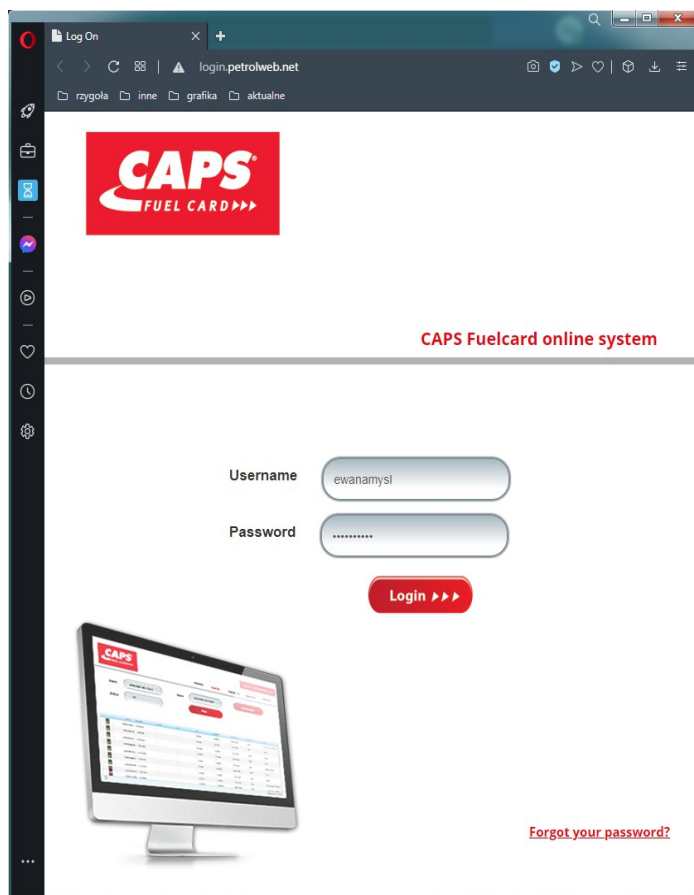
ip.addr == 128.30.52.100						
No.	Time	Source	Destination	Protocol	Length	Info
10	1.682336	192.168.0.17	128.30.52.100	TCP	66	52567 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
13	1.824773	128.30.52.100	192.168.0.17	TCP	66	443 → 52567 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM=1 WS=512
14	1.824904	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=1 Ack=1 Win=17040 Len=0
15	1.840891	192.168.0.17	128.30.52.100	TLSv1.3	571	Client Hello
19	1.991024	128.30.52.100	192.168.0.17	TLSv1.3	1474	Server Hello, Change Cipher Spec, Application Data
20	1.993711	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=1421 Ack=518 Win=67072 Len=1420 [TCP segment of a reassembled PDU]
21	1.993788	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=518 Ack=2841 Win=17040 Len=0
22	1.994834	128.30.52.100	192.168.0.17	TCP	1310	443 → 52567 [PSH, ACK] Seq=2841 Ack=518 Win=67072 Len=1256 [TCP segment of a reassembled PDU]
23	1.994834	128.30.52.100	192.168.0.17	TLSv1.3	1474	Application Data [TCP segment of a reassembled PDU]
24	1.994834	128.30.52.100	192.168.0.17	TLSv1.3	160	Application Data, Application Data
25	1.994905	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=518 Ack=5623 Win=17040 Len=0
26	2.012042	192.168.0.17	128.30.52.100	TLSv1.3	118	Change Cipher Spec, Application Data
27	2.013568	192.168.0.17	128.30.52.100	TLSv1.3	560	Application Data
29	2.150548	128.30.52.100	192.168.0.17	TLSv1.3	133	Application Data
30	2.151634	128.30.52.100	192.168.0.17	TLSv1.3	133	Application Data
31	2.151810	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=1088 Ack=5781 Win=16880 Len=0
32	2.156772	128.30.52.100	192.168.0.17	TLSv1.3	100	Application Data
33	2.157276	192.168.0.17	128.30.52.100	TLSv1.3	85	Application Data
36	2.184167	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=5827 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
37	2.184473	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=7247 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
38	2.184473	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=8667 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
39	2.184473	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=10087 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
40	2.184578	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=1119 Ack=11507 Win=17040 Len=0
41	2.184772	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=11507 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
42	2.184772	128.30.52.100	192.168.0.17	TLSv1.3	1418	Application Data
43	2.184772	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=14291 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
44	2.185014	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=1119 Ack=15711 Win=17040 Len=0
46	2.290079	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=15711 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
47	2.290685	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=17131 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
48	2.290685	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=18551 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
49	2.290685	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=19971 Ack=1088 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
50	2.291547	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=1119 Ack=21391 Win=17040 Len=0
51	2.322388	128.30.52.100	192.168.0.17	TLSv1.3	1474	Application Data [TCP segment of a reassembled PDU]
52	2.322909	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=22811 Ack=1119 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
53	2.322909	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=24231 Ack=1119 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
54	2.322909	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=25651 Ack=1119 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
55	2.323018	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=1119 Ack=27071 Win=17040 Len=0
56	2.323192	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=27071 Ack=1119 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
57	2.323192	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=28491 Ack=1119 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
58	2.323192	128.30.52.100	192.168.0.17	TCP	1474	443 → 52567 [ACK] Seq=29911 Ack=1119 Win=68096 Len=1420 [TCP segment of a reassembled PDU]
59	2.323421	192.168.0.17	128.30.52.100	TCP	54	52567 → 443 [ACK] Seq=1119 Ack=31331 Win=17040 Len=0

Z kolei po odświeżeniu strony w przeglądarce nie następuje wymiana handshake'ów.

ip.addr == 128.30.52.100						
No.	Time	Source	Destination	Protocol	Length	Info
14	1.599984	192.168.0.17	128.30.52.100	TLSv1.2	168	Application Data
17	1.827280	128.30.52.100	192.168.0.17	TLSv1.2	93	Application Data
19	1.871220	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=40 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
20	1.871330	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=1460 Win=65320 Len=0
21	1.872083	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=1460 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
22	1.872083	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=2880 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
23	1.872083	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=4300 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
24	1.872139	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=5720 Win=65320 Len=0
25	1.872527	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=5720 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
26	1.872527	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=7140 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
27	1.872527	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=8560 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
28	1.872573	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=9980 Win=65320 Len=0
29	1.873628	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=9980 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
30	1.873628	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=11400 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
31	1.873700	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=12820 Win=65320 Len=0
34	2.098834	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=12820 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
35	2.099486	128.30.52.100	192.168.0.17	TLSv1.2	1474	Application Data [TCP segment of a reassembled PDU]
36	2.099486	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=15660 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
37	2.099486	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=17080 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
38	2.099570	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=18500 Win=65320 Len=0
39	2.099850	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=18500 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
40	2.099850	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=19920 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
41	2.099850	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=21340 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
42	2.099919	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=22760 Win=65320 Len=0
43	2.100186	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=22760 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
44	2.100186	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=24180 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
45	2.100186	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=25600 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
46	2.100239	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=27020 Win=65320 Len=0
47	2.100532	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=27020 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
48	2.100532	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=28440 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
49	2.100532	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=29860 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
50	2.100777	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=31280 Win=65320 Len=0
51	2.101441	128.30.52.100	192.168.0.17	TLSv1.2	1474	Application Data [TCP segment of a reassembled PDU]
52	2.101441	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=32700 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
53	2.101441	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=34120 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
54	2.101500	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=35540 Win=65320 Len=0
55	2.101818	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=35540 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
56	2.101818	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=36960 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
57	2.101818	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=38380 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]
58	2.101897	192.168.0.17	128.30.52.100	TCP	54	53121 → 443 [ACK] Seq=115 Ack=39800 Win=65320 Len=0
59	2.102916	128.30.52.100	192.168.0.17	TCP	1474	443 → 53121 [ACK] Seq=39800 Ack=115 Win=135 Len=1420 [TCP segment of a reassembled PDU]

4. Logowanie poprzez protokół HTTP

Korzystając z <http://login.petrolweb.net>, można sprawdzić jak wygląda logowanie w protokole HTTP.



Po wprowadzeniu danych, w Wiresharku można zobaczyć, że podany login i hasło nie są zaszyfrowane i tym samym bardzo łatwo można je zdobyć.

The image shows a Wireshark network traffic capture. The main pane displays a list of packets, with packet 142 selected. The packet details pane on the right shows the structure of the selected packet:

- Frame 142: 738 bytes on wire (5904 bits), 738 bytes captured (5904 bits) on interface Device\NPF{46776988-79F}
- Ethernet II, Src: Tp-LinkT_2e:95:6c (a0:f3:c1:2e:95:6c), Dst: CompalBr_19:d9:a8 (ac:22:05:19:d9:a8)
- Internet Protocol Version 4, Src: 192.168.0.17, Dst: 193.110.250.6
- Transmission Control Protocol, Src Port: 54321, Dst Port: 80, Seq: 1, Ack: 1, Len: 684
- Hypertext Transfer Protocol
 - HTML Form URL Encoded: application/x-www-form-urlencoded
 - Form item: "UserName" = "ewanamys1"
 - Key: UserName
 - Value: ewanamys1
 - Form item: "Password" = "qerty1234"
 - Key: Password
 - Value: qerty1234

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII. The ASCII portion clearly shows the encoded form data: `Name=ewanamys1&password=qerty1234`.

5. Zapytania DNS podczas wywoływania strony

Wchodząc przykładowo na onet.pl, można w Wiresharku zauważyć nie tylko zapytania dla serwera DNS dotyczące samej witryny, ale też podstron serwisu, jak i narzędzi reklamowych (tutaj chociażby Google Analytics).

The image shows a Wireshark network traffic capture of DNS traffic. The main pane displays a list of packets, with packet 27 selected. The packet details pane on the right shows the structure of the selected packet:

- Frame 27: 299612 bytes on wire (2396896 bits), 299612 bytes captured (2396896 bits) on interface Device\NPF{46776988-79F}
- Ethernet II, Src: Tp-LinkT_2e:95:6c (a0:f3:c1:2e:95:6c), Dst: CompalBr_19:d9:a8 (ac:22:05:19:d9:a8)
- Internet Protocol Version 4, Src: 192.168.0.17, Dst: 193.110.250.6
- Transmission Control Protocol, Src Port: 54321, Dst Port: 80, Seq: 1, Ack: 1, Len: 684
- Hypertext Transfer Protocol

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII. The ASCII portion clearly shows the DNS query: `Standard query 0xf3fa A www.onet.pl`.

6. Analiza ruchu FTP

No.	Time	Source	Destination	Protocol	Length	Info
68	9.592472	195.144.107.198	192.168.0.17	FTP	81	Response: 220 Microsoft FTP Service
69	9.592785	192.168.0.17	195.144.107.198	FTP	70	Request: USER anonymous
70	9.637948	195.144.107.198	192.168.0.17	FTP	92	Response: 331 Password required for anonymous.
71	9.638354	192.168.0.17	195.144.107.198	FTP	79	Request: PASS chrome@example.com
72	9.684242	195.144.107.198	192.168.0.17	FTP	79	Response: 530 User cannot log in.
73	9.684530	192.168.0.17	195.144.107.198	FTP	60	Request: QUIT
74	9.734714	195.144.107.198	192.168.0.17	FTP	68	Response: 221 Goodbye.
82	9.825235	195.144.107.198	192.168.0.17	FTP	81	Response: 220 Microsoft FTP Service
83	9.825532	192.168.0.17	195.144.107.198	FTP	65	Request: USER ewan
84	9.872585	195.144.107.198	192.168.0.17	FTP	87	Response: 331 Password required for ewan.
85	9.872927	192.168.0.17	195.144.107.198	FTP	71	Request: PASS qwerty1234
86	9.921914	195.144.107.198	192.168.0.17	FTP	79	Response: 530 User cannot log in.
87	9.922265	192.168.0.17	195.144.107.198	FTP	60	Request: QUIT
88	9.967404	195.144.107.198	192.168.0.17	FTP	68	Response: 221 Goodbye.

Podobnie jak w przypadku logowania na stronie korzystające z protokołu HTTP, login i hasło nie zostały zaszyfrowane.

7. Logowanie poprzez protokół HTTPS

The image displays a Wireshark network traffic capture of an HTTPS connection. The main packet list on the left shows a series of packets, with packet 34 selected. The packet details pane on the right shows the structure of the selected packet, which is a TLSv1.2 Record. The record consists of a TLS header and an encrypted application data payload. The packet bytes pane at the bottom shows the raw hex and ASCII data of the selected packet.

Wireshark - Pakiet 34 - Połączenie sieci bezprzewodowej

Destination Port: 59032
[Stream index: 9]
[TCP Segment Len: 46]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2938818712
[Next Sequence Number: 47 (relative sequence number)]
Acknowledgment Number: 458 (relative ack number)
Acknowledgment number (raw): 2556934042
0101 = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window: 82
[Calculated window size: 82]
[Window size scaling factor: -1 (unknown)]
Checksum: 0x0309 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[SEQ/ACK analysis]
[Timestamps]
TCP payload (46 bytes)

Transport Layer Security
TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 41
Encrypted Application Data: f059cef4d55523a9e943d46fb2d83cd2059e0da368417406726ed7430987b01bf7e8e25...
[Application Data Protocol: http-over-tls]

0000 a0 f3 c1 2e 95 6c ac 22 05 19 d9 a0 08 00 45 001..E..
0010 00 56 b2 43 40 00 3a 06 6a 44 d5 b4 8d ac c0 a8 ..V.C@.:jD..
0020 00 11 01 bb e6 98 af 2a d0 98 98 67 b7 9a 50 18*g.P..
0030 00 52 03 89 00 00 17 03 03 00 29 f0 59 ce f4 d5 R.....)Y..
0040 55 23 a9 e9 43 d4 6f b2 d8 3c d2 05 9e 0d 0a 36 U@.C.o.<....6
0050 84 17 40 67 26 ed 74 30 98 7b 01 bf 7e 8e 25 d8 .@g8.t0-{:~%5
0060 6e 25 2b 7b n%+{

W protokole HTTPS nie jesteśmy w stanie podejrzeć przesyłanych informacji, dane są szyfrowane za pomocą protokołu TLS. Można wyświetlić ilość przesyłanych danych na porcie 443 etc., jednak bez klucza prywatnego nie jesteśmy w stanie stwierdzić, co zawierają. Klucz publiczny jest udostępniany szyfrującemu dane, jednak klucz prywatny jest dostępny wyłącznie dla wysyłającego informacje i tylko on może je odczytać.