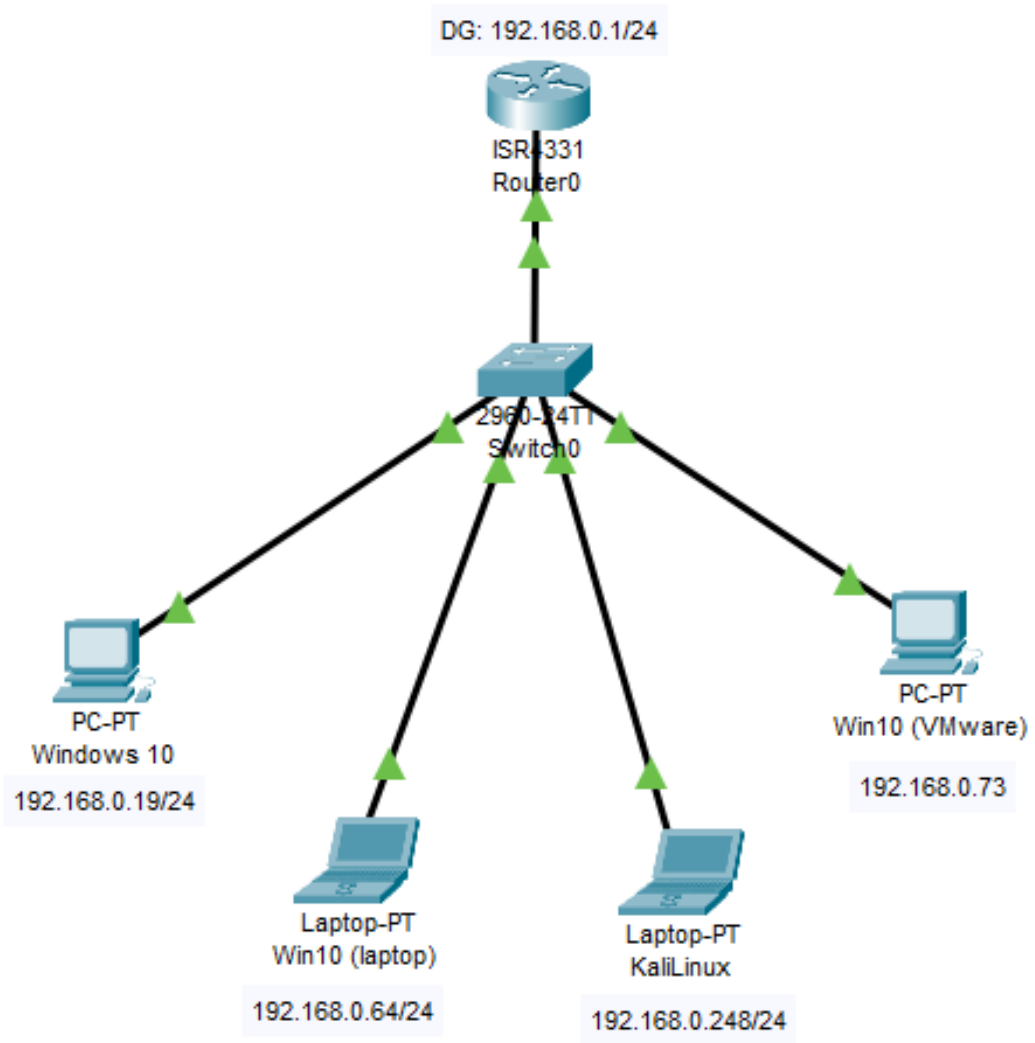
Laboratorium 5. – Szymon Szkarłat

Zadanie 1. – Przygotowanie środowiska testowego
W moim środowisku testowym znajdują się 3 urządzenia z systemem Windows. Komputer stacjonarny (win10), laptop (win11), maszyna wirtualna (win10) oraz maszyna wirtualna z KaliLinux. Rysunek techniczny, który miej więcej odzwierciedla aktualne połączenie sieci.
Do sporządzenia rysunków technicznych podczas wykonywania tego laboratorium użyłem popularnego programu od firmy CISCO, a mianowicie CISCO Packet Tracert.



Zadanie 2. – Pozyskiwanie informacji z sieci przy użyciu skanera Nmap

1. Sprawdzenie adresu IP oraz ustawień sieci (ifconfig && route in) mojej sieci.

```
┌──(kali㉿kali)-[~]
└─$ ifconfig && route -n
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.248  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 2a02:a317:c040:dd00:532e:aa4d:ce40:7c2f  prefixlen 64  scopeid 0×0<global>
        inet6 fe80::931f:6f53:3c6f:c60c  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:21:77:33  txqueuelen 1000  (Ethernet)
        RX packets 244  bytes 51423 (50.2 KiB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 67  bytes 31261 (30.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.255.255.100  netmask 255.255.255.0  broadcast 10.255.255.255
        inet6 fe80::85dc:7311:317f:3fac  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:21:77:3d  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 125  bytes 18840 (18.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 eth0
0.0.0.0         10.255.255.10   0.0.0.0         UG    101    0        0 eth1
10.255.255.0    0.0.0.0         255.255.255.0   U     101    0        0 eth1
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
```

2. Skanowanie całej podsieci na 24 bitach, w której znajdują się maszyny oraz inne urządzenia.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sn 192.168.0.0/24
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-13 04
:26 EST
Nmap scan report for 192.168.0.1
Host is up (0.0021s latency).
MAC Address: 38:43:7D:A5:3D:44 (Compal Broadband Network
s)
Nmap scan report for 192.168.0.2
Host is up (0.00049s latency).
MAC Address: 64:66:B3:64:F2:8E (TP-Link Technologies)
Nmap scan report for 192.168.0.19
Host is up (0.00035s latency).
MAC Address: B4:2E:99:15:FB:26 (Giga-byte Technology)
Nmap scan report for 192.168.0.45
Host is up (0.054s latency).
MAC Address: AA:9D:09:C3:7C:EB (Unknown)
Nmap scan report for 192.168.0.87
Host is up (0.0017s latency).
MAC Address: 6C:A6:04:75:29:03 (Arris Group)
Nmap scan report for 192.168.0.101
Host is up (0.00027s latency).
MAC Address: 00:0C:29:A2:9A:E8 (VMware)
Nmap scan report for 192.168.0.248
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.66
 seconds
```

Po przeskanowaniu sieci pojawiły się informacje o adresach IP urządzeń, które pracują w mojej sieci. Adres 192.168.0.19 (komputer stacjonarny z win10), 192.168.0.101 (win10 na maszynie wirtualnej) – nastąpiła zmiana adresu IP, 192.168.0.45 (laptop z win11) – nastąpiła zmiana adresu IP. No na koniec 192.168.0.248, czyli adres KaliLinuxa. Nastąpiła zmiana dzierżawy niektórych adresów przez DHCP.

3. Sprawdzenie czy maszyna wirtualna Windows 10 jest poprawnie rozpoznawana przez narzędzie Nmap.

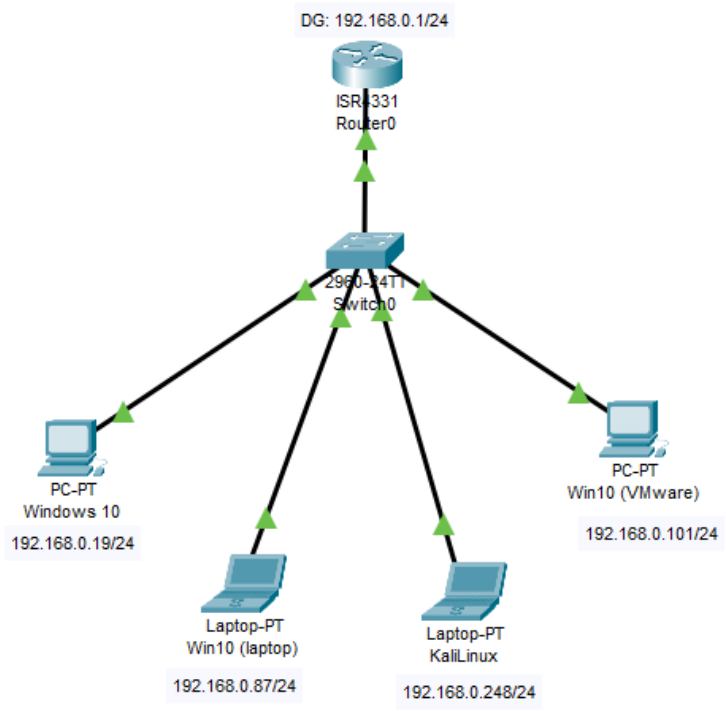
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -A 192.168.0.101
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-13 04:34 EST
Nmap scan report for 192.168.0.101
Host is up (0.00044s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
5357/tcp open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
MAC Address: 00:0C:29:A2:9A:E8 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|2019 (89%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3
Aggressive OS guesses: Microsoft Windows XP SP3 (89%), Microsoft Windows Server 2019 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE
HOP RTT     ADDRESS
1   0.44 ms 192.168.0.101

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.49 seconds
```

Odpowiedź jest: tak. Mamy tutaj informację np. o tym ile skoków należało wykonać, aby dostać się do maszyny wirtualnej z Win10. Udało się odczytać również adres MAC oraz otwarte porty (tj. 5357/tcp).

Aktualizacja rysunku technicznego



Zadanie 3. – Analiza ruchu sieciowego przy wykorzystaniu narzędzia TCPdump

1. Na systemie KaliLinux polecenie tcpdump jest domyślnie zainstalowane. Zatem przechodzę do wykonania polecenia sudo tcpdump -i eth0 -v

```
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:42:07.562282 IP (tos 0×0, ttl 1, id 61130, offset 0, flags [none], proto IGMP (2), length 32, options (RA))
    192.168.0.101 > 224.0.0.252: igmp v2 report 224.0.0.252
04:42:07.843291 IP (tos 0×0, ttl 64, id 54430, offset 0, flags [DF], proto UDP (17), length 70)
```

2. Wykonanie polecenia ping z systemu KaliLinux w kierunku bramy domyślnej i przefiltrowanie wykonanego pingu

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -v host 192.168.0.1
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:44:25.833235 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.116 tell compalhub.home, length 46
04:44:25.889339 IP (tos 0×0, ttl 64, id 26686, offset 0, flags [DF], proto UDP (17), length 72)
    192.168.0.248.52215 > compalhub.home.domain: 59704+ PTR? 116.0.168.192.in-addr.arpa. (44)
04:44:25.893495 IP (tos 0×0, ttl 64, id 30180, offset 0, flags [DF], proto UDP (17), length 72)
    compalhub.home.domain > 192.168.0.248.52215: 59704 NXDomain* 0/0/0 (44)
04:44:25.893698 IP (tos 0×0, ttl 64, id 53520, offset 0, flags [DF], proto UDP (17), length 70)
    192.168.0.248.59708 > compalhub.home.domain: 27182+ PTR? 1.0.168.192.in-addr.arpa. (42)
04:44:25.896769 IP (tos 0×0, ttl 64, id 30181, offset 0, flags [DF], proto UDP (17), length 98)
    compalhub.home.domain > 192.168.0.248.59708: 27182* 1/0/0 1.0.168.192.in-addr.arpa. PTR compalhub.home. (70)
04:44:25.992610 IP (tos 0×0, ttl 64, id 18346, offset 0, flags [DF], proto UDP (17), length 72)
    192.168.0.248.36266 > compalhub.home.domain: 42149+ PTR? 248.0.168.192.in-addr.arpa. (44)
04:44:25.996501 IP (tos 0×0, ttl 64, id 30182, offset 0, flags [DF], proto UDP (17), length 72)
    compalhub.home.domain > 192.168.0.248.36266: 42149 NXDomain* 0/0/0 (44)
04:44:26.832763 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.116 tell compalhub.home, length 46
04:44:27.833838 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.116 tell compalhub.home, length 46
04:44:29.071499 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.0.116 tell compalhub.home, length 46
```

Operacja przefiltrowania uzyskanych wyników zakończył się powodzeniem.

3. Zastosowanie filtrowania w programie tcpdump wskazując przy tym adres źródłowy i docelowy. Otwarcie przeglądarki i wyświetlenie jednej z witryn internatowych (tj. www.onet.pl)

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -v host www.onet.pl
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:49:44.325080 IP (tos 0×0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 1385)
    192.168.0.248.44019 > server-18-244-102-79.waw51.r.cloudfront.net.https: UDP, length 1357
04:49:44.325165 IP (tos 0×0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 425)
    192.168.0.248.44019 > server-18-244-102-79.waw51.r.cloudfront.net.https: UDP, length 397
04:49:44.346659 IP (tos 0×0, ttl 249, id 0, offset 0, flags [DF], proto UDP (17), length 586)
    server-18-244-102-79.waw51.r.cloudfront.net.https > 192.168.0.248.44019: UDP, length 558
04:49:44.368849 IP (tos 0×0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 72)
    192.168.0.248.44019 > server-18-244-102-79.waw51.r.cloudfront.net.https: UDP, length 44
04:49:44.386838 IP (tos 0×0, ttl 249, id 0, offset 0, flags [DF], proto UDP (17), length 71)
    server-18-244-102-79.waw51.r.cloudfront.net.https > 192.168.0.248.44019: UDP, length 43
04:49:46.574672 IP (tos 0×0, ttl 64, id 64441, offset 0, flags [DF], proto TCP (6), length 91)
```

Zastosowanie filtru przyniosło zamierzony efekt i udało się odnaleźć stronę www.onet.pl

4. Wykonanie filtrowania wskazując kolejno port 80 i 443

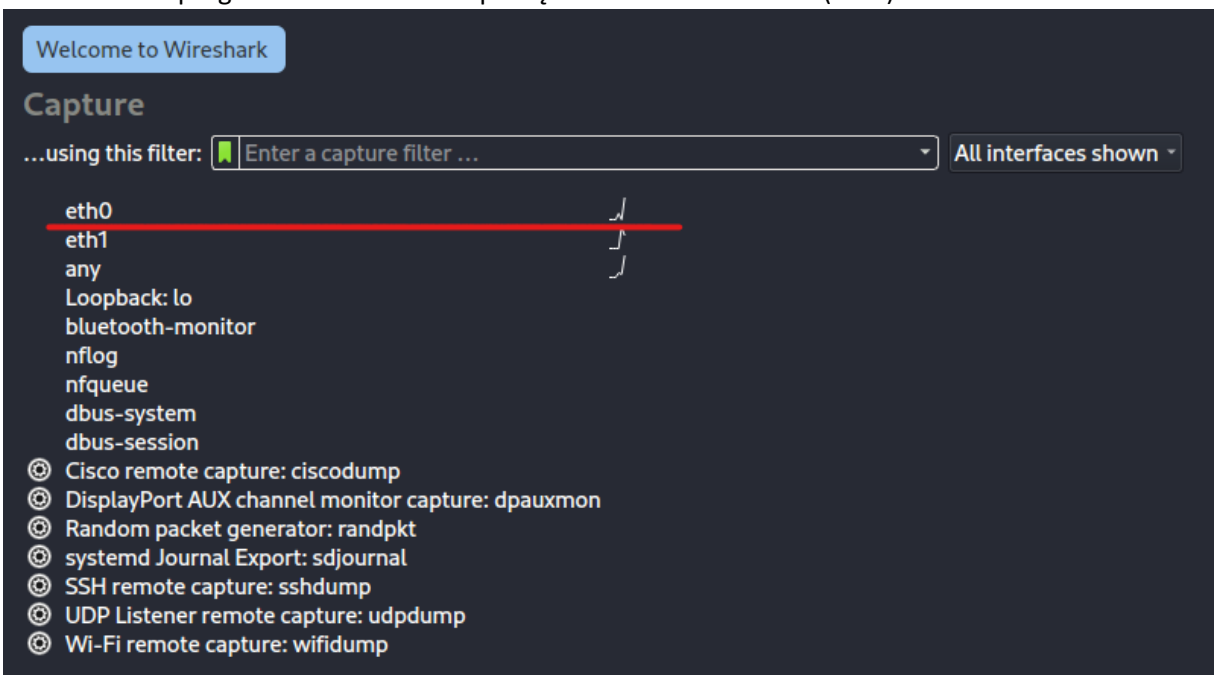
```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -v dst port 80
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:53:07.248287 IP (tos 0×0, ttl 64, id 3117, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.0.19.51591 > 591837941.ams.cdn77.com.http: Flags [S], cksum 0×1f7b (correct), seq 261049009, win 64240,
options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
04:53:07.293358 IP (tos 0×0, ttl 64, id 3118, offset 0, flags [DF], proto TCP (6), length 40)
    192.168.0.19.51591 > 591837941.ams.cdn77.com.http: Flags [.], cksum 0×566a (correct), ack 886492140, win 1026, l
ength 0
04:53:07.293423 IP (tos 0×0, ttl 64, id 3119, offset 0, flags [DF], proto TCP (6), length 139)
    192.168.0.19.51591 > 591837941.ams.cdn77.com.http: Flags [P.], cksum 0×5d63 (correct), seq 0:99, ack 1, win 1026
, length 99: HTTP, length: 99
        GET / HTTP/1.1
        Host: ping.urban-vpn.com
        User-Agent: Go-http-client/1.1
        Accept-Encoding: gzip
```

Port 443, wyświetlił nam się przykładowo Facebook.

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -v dst port 443
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:54:12.318505 IP6 (flowlabel 0×ec725, hlim 64, next-header TCP (6) payload length: 52) 2a02:a317:c040:dd00:3577:ee
8a:799e:242d.50124 > edge-star6-shv-01-waw1.facebook.com.https: Flags [P.], cksum 0×f0f2 (correct), seq 1881997718:1
881997750, ack 2549014247, win 1024, length 32
04:54:12.318506 IP6 (flowlabel 0×c305e, hlim 64, next-header TCP (6) payload length: 52) 2a02:a317:c040:dd00:3577:ee
8a:799e:242d.50107 > edge-star6-shv-01-waw1.facebook.com.https: Flags [P.], cksum 0×f3d6 (correct), seq 322357272:32
2357304, ack 3472047446, win 1026, length 32
04:54:12.318506 IP6 (flowlabel 0×75459, hlim 64, next-header TCP (6) payload length: 52) 2a02:a317:c040:dd00:3577:ee
8a:799e:242d.50106 > edge-star6-shv-01-waw1.facebook.com.https: Flags [P.], cksum 0×bc0d (correct), seq 3688409402:3
688409434, ack 1416470482, win 1026, length 32
04:54:12.412807 IP6 (flowlabel 0×ec725, hlim 64, next-header TCP (6) payload length: 20) 2a02:a317:c040:dd00:3577:ee
8a:799e:242d.50124 > edge-star6-shv-01-waw1.facebook.com.https: Flags [.], cksum 0×e3cb (correct), ack 29, win 1024,
 length 0
04:54:12.412808 IP6 (flowlabel 0×75459, hlim 64, next-header TCP (6) payload length: 20) 2a02:a317:c040:dd00:3577:ee
8a:799e:242d.50106 > edge-star6-shv-01-waw1.facebook.com.https: Flags [.], cksum 0×5323 (correct), ack 29, win 1025,
 length 0
04:54:12.412808 IP6 (flowlabel 0×c305e, hlim 64, next-header TCP (6) payload length: 20) 2a02:a317:c040:dd00:3577:ee
8a:799e:242d.50107 > edge-star6-shv-01-waw1.facebook.com.https: Flags [.], cksum 0×e4db (correct), ack 29, win 1026,
 length 0
```

Zadanie 4. – Analiza ruchu sieciowego przy wykorzystaniu programu Wireshark

1. Program Wireshark na systemie KaliLinux jest domyślnie zainstalowany
2. Otworzenie programu Wireshark i rozpoczęcie nasłuchiwania sieci (eth0)



3. Po uruchomieniu skanowania na Wireshark oraz wykonaniu polecenia sudo nmap -sS 192.168.1.101

Pierwsze polecenie jakie zobaczyłem:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	ARRISGro_75:29:03	Giga-Byt_15:fb:26	ARP	60	Who has 192.168.0.19? Tell 192.168.0.87
2	0.000000211	Giga-Byt_15:fb:26	ARRISGro_75:29:03	ARP	60	192.168.0.19 is at b4:2e:99:15:fb:26
3	1.025134189	192.168.0.19	104.17.138.37	TCP	60	50241 → 443 [ACK] Seq=1 Ack=1 Win=1028 Len=1 [TCP segment of a reassemb…
4	1.048368397	104.17.138.37	192.168.0.19	TCP	66	443 → 50241 [ACK] Seq=1 Ack=2 Win=8 Len=0 SLE=1 SRE=2
5	2.284261813	2a02:a317:c040:dd00…	2a03:2880:f016:8:fa…	TLSv1.2	103	Application Data
6	2.305353365	2a03:2880:f016:8:fa…	2a02:a317:c040:dd00…	TCP	74	443 → 51310 [ACK] Seq=1 Ack=30 Win=1643 Len=0
7	2.353515893	2a03:2880:f016:8:fa…	2a02:a317:c040:dd00…	TLSv1.2	99	Application Data
8	2.408182311	2a02:a317:c040:dd00…	2a03:2880:f016:8:fa…	TCP	74	51310 → 443 [ACK] Seq=30 Ack=26 Win=1025 Len=0
9	3.290255895	2a02:a317:c040:dd00…	2a03:2880:f016:b:fa…	TLSv1.2	106	Application Data
10	3.290256119	2a02:a317:c040:dd00…	2a03:2880:f016:b:fa…	TLSv1.2	106	Application Data

Polecenia TCP, których podczas skanowania sieci generowana jest olbrzymia ich ilość.

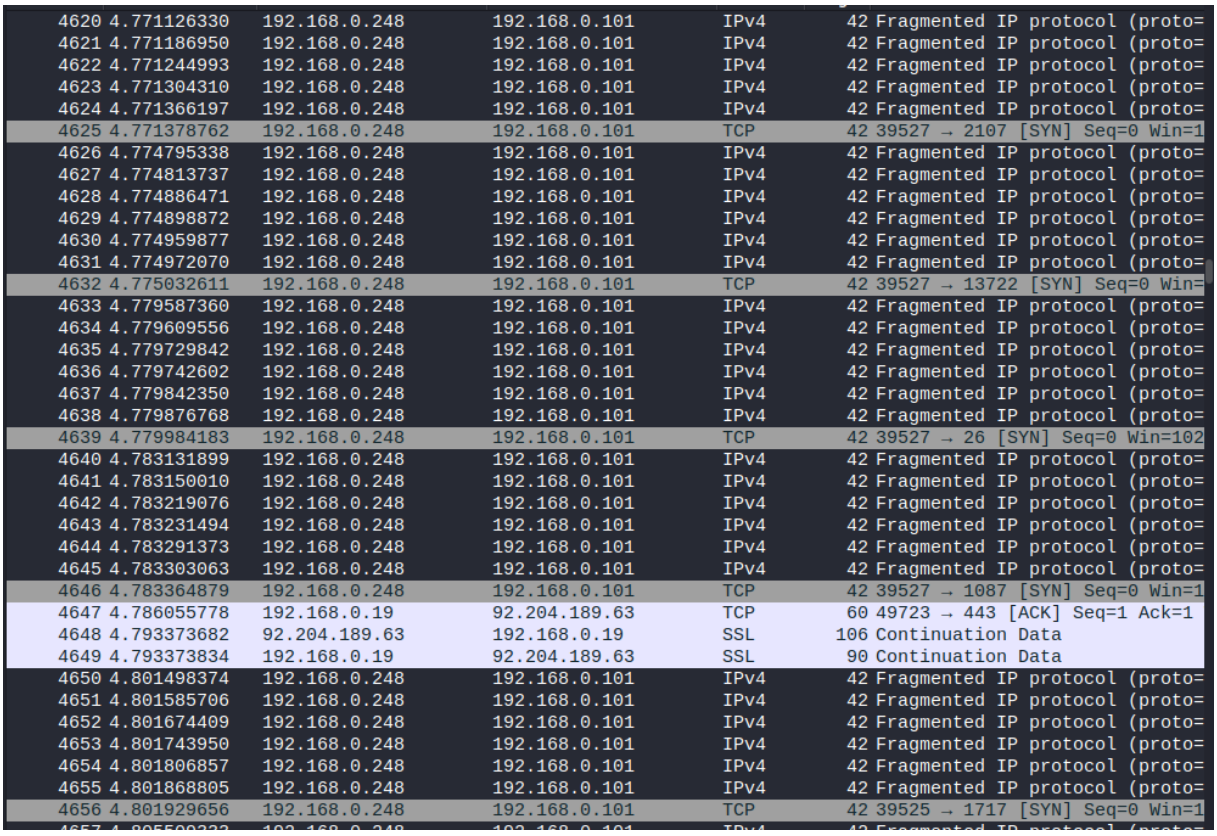
No.	Time	Source	Destination	Protocol	Length	Info
460	3.940501105	192.168.0.248	192.168.0.101	TCP	58	51370 → 1494 [SYN] Seq=0
461	3.940537289	192.168.0.248	192.168.0.101	TCP	58	51370 → 7402 [SYN] Seq=0
462	3.940555639	192.168.0.248	192.168.0.101	TCP	58	51370 → 146 [SYN] Seq=0
463	3.944757085	192.168.0.248	192.168.0.101	TCP	58	51370 → 1107 [SYN] Seq=0
464	3.947407735	192.168.0.248	192.168.0.101	TCP	58	51370 → 3869 [SYN] Seq=0
465	4.027235064	192.168.0.248	192.168.0.101	TCP	58	51372 → 5877 [SYN] Seq=0
466	4.027276147	192.168.0.248	192.168.0.101	TCP	58	51372 → 1600 [SYN] Seq=0
467	4.030205639	192.168.0.248	192.168.0.101	TCP	58	51372 → 3052 [SYN] Seq=0
468	4.034921767	192.168.0.248	192.168.0.101	TCP	58	51372 → 416 [SYN] Seq=0
469	4.038563610	192.168.0.248	192.168.0.101	TCP	58	51372 → 3351 [SYN] Seq=0
470	4.041587003	192.168.0.248	192.168.0.101	TCP	58	51372 → 146 [SYN] Seq=0
471	4.041628723	192.168.0.248	192.168.0.101	TCP	58	51372 → 7402 [SYN] Seq=0
472	4.041647275	192.168.0.248	192.168.0.101	TCP	58	51372 → 1494 [SYN] Seq=0
473	4.045606881	192.168.0.248	192.168.0.101	TCP	58	51372 → 1107 [SYN] Seq=0
474	4.048330928	192.168.0.248	192.168.0.101	TCP	58	51372 → 3869 [SYN] Seq=0
475	4.127614576	192.168.0.248	192.168.0.101	TCP	58	51370 → 425 [SYN] Seq=0
476	4.127688617	192.168.0.248	192.168.0.101	TCP	58	51370 → 687 [SYN] Seq=0
477	4.131235069	192.168.0.248	192.168.0.101	TCP	58	51370 → 2010 [SYN] Seq=0
478	4.135251833	192.168.0.248	192.168.0.101	TCP	58	51370 → 8291 [SYN] Seq=0
479	4.138870030	192.168.0.248	192.168.0.101	TCP	58	51370 → 8181 [SYN] Seq=0
480	4.141835343	192.168.0.248	192.168.0.101	TCP	58	51370 → 7512 [SYN] Seq=0
481	4.141882770	192.168.0.248	192.168.0.101	TCP	58	51370 → 7443 [SYN] Seq=0
482	4.141902287	192.168.0.248	192.168.0.101	TCP	58	51370 → 1052 [SYN] Seq=0
483	4.145952700	192.168.0.248	192.168.0.101	TCP	58	51370 → 9071 [SYN] Seq=0
484	4.149614019	192.168.0.248	192.168.0.101	TCP	58	51370 → 21571 [SYN] Seq=
485	4.228043260	192.168.0.248	192.168.0.101	TCP	58	51372 → 687 [SYN] Seq=0
486	4.228084325	192.168.0.248	192.168.0.101	TCP	58	51372 → 425 [SYN] Seq=0
487	4.232110752	192.168.0.248	192.168.0.101	TCP	58	51372 → 2010 [SYN] Seq=0
488	4.235986424	192.168.0.248	192.168.0.101	TCP	58	51372 → 8291 [SYN] Seq=0
489	4.239563153	192.168.0.248	192.168.0.101	TCP	58	51372 → 8181 [SYN] Seq=0
490	4.242286192	192.168.0.248	192.168.0.101	TCP	58	51372 → 1052 [SYN] Seq=0
491	4.242323693	192.168.0.248	192.168.0.101	TCP	58	51372 → 7443 [SYN] Seq=0
492	4.242342094	192.168.0.248	192.168.0.101	TCP	58	51372 → 7512 [SYN] Seq=0
493	4.246264542	192.168.0.248	192.168.0.101	TCP	58	51372 → 9071 [SYN] Seq=0
494	4.250721375	192.168.0.248	192.168.0.101	TCP	58	51372 → 21571 [SYN] Seq=
495	4.329079191	192.168.0.248	192.168.0.101	TCP	58	51370 → 99 [SYN] Seq=0 W
496	4.329119902	192.168.0.248	192.168.0.101	TCP	58	51370 → 5120 [SYN] Seq=0
497	4.333024151	192.168.0.248	192.168.0.101	TCP	58	51370 → 3880 [SYN] Seq=0
498	4.336045869	192.168.0.248	192.168.0.101	TCP	58	51370 → 7 [SYN] Seq=0 Wi

Łącznie udało się przechwycić blisko (1200 logów), na początku pojawiały się polecenia nie związane z protokołem TCP. Jak widać stroną pytającą, a więc źródłową był adres 192.168.0.248, czyli komputer z KaliLinux. Natomiast stroną odbierającą, komputer o adresie 192.168.0.101 (maszyna Vmware z win10)

1232	8.668033801	192.168.0.248	192.168.0.101	TCP
1233	8.668050743	192.168.0.248	192.168.0.101	TCP
1234	8.668067391	192.168.0.248	192.168.0.101	TCP
1235	8.668085348	192.168.0.248	192.168.0.101	TCP
1236	8.668102765	192.168.0.248	192.168.0.101	TCP
1237	8.668119649	192.168.0.248	192.168.0.101	TCP
1238	8.668136175	192.168.0.248	192.168.0.101	TCP

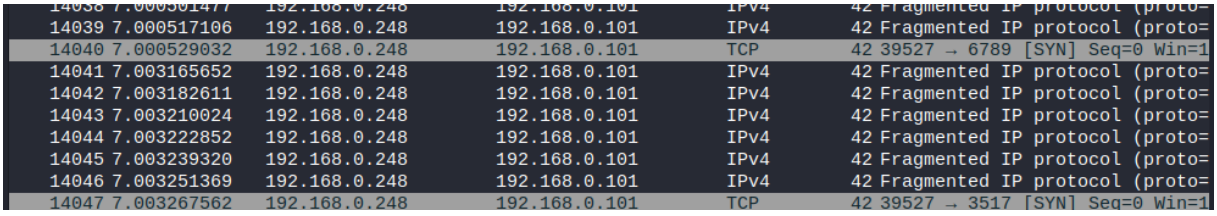
Osoba, która monitoruje sieć byłaby w stanie zorientować się, że sieć jest właśnie skanowana, z łatwością odczytałaby adres IP nadawcy (osoba skanująca sieć). W konsekwencji nie stanowiłoby to większego problemu, aby zablokować osobę przeprowadzającą skan sieci w taki sposób.

Teraz wykonałem inny sposób skanowania sieci, przy pomocy polecenia: sudo nmap 192.168.1.103 --data-length 32 -f -T5
Zapytań tych, po wykonaniu tego polecenia, jest znacznie więcej, ale są one podzielone, przez co automaty czy inne zautomatyzowane narzędzia nie byłyby w stanie zablokować skanowania sieci przeprowadzonego w ten sposób.



4620	4.771126330	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4621	4.771186950	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4622	4.771244993	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4623	4.771304310	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4624	4.771366197	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4625	4.771378762	192.168.0.248	192.168.0.101	TCP	42	39527 → 2107 [SYN] Seq=0 Win=1
4626	4.774795338	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4627	4.774813737	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4628	4.774886471	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4629	4.774898872	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4630	4.774959877	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4631	4.774972070	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4632	4.775032611	192.168.0.248	192.168.0.101	TCP	42	39527 → 13722 [SYN] Seq=0 Win=
4633	4.779587360	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4634	4.779609556	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4635	4.779729842	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4636	4.779742602	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4637	4.779842350	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4638	4.779876768	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4639	4.779984183	192.168.0.248	192.168.0.101	TCP	42	39527 → 26 [SYN] Seq=0 Win=102
4640	4.783131899	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4641	4.783150010	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4642	4.783219076	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4643	4.783231494	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4644	4.783291373	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4645	4.783303063	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4646	4.783364879	192.168.0.248	192.168.0.101	TCP	42	39527 → 1087 [SYN] Seq=0 Win=1
4647	4.786055778	192.168.0.19	92.204.189.63	TCP	60	49723 → 443 [ACK] Seq=1 Ack=1
4648	4.793373682	92.204.189.63	192.168.0.19	SSL	106	Continuation Data
4649	4.793373834	192.168.0.19	92.204.189.63	SSL	90	Continuation Data
4650	4.801498374	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4651	4.801585706	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4652	4.801674409	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4653	4.801743950	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4654	4.801806857	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4655	4.801868805	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
4656	4.801929656	192.168.0.248	192.168.0.101	TCP	42	39525 → 1717 [SYN] Seq=0 Win=1

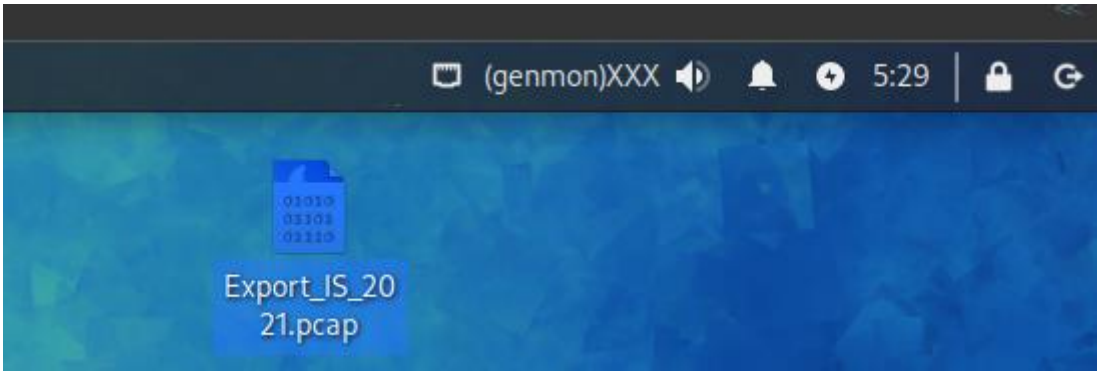
Łącznie podczas rej sesji udało się przechwycić blisko 14 tysięcy pakietów, były one generowane z równym odstępem czasu.



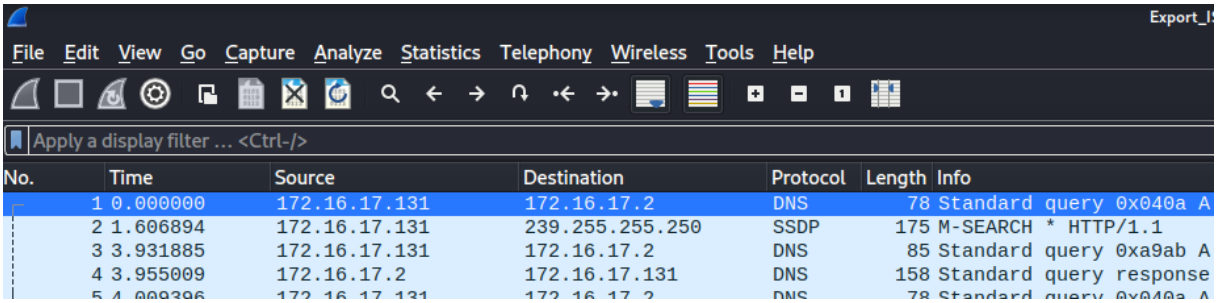
14039	7.000517106	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
14040	7.000529032	192.168.0.248	192.168.0.101	TCP	42	39527 → 6789 [SYN] Seq=0 Win=1
14041	7.003165652	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
14042	7.003182611	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
14043	7.003210024	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
14044	7.003222852	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
14045	7.003239320	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
14046	7.003251369	192.168.0.248	192.168.0.101	IPv4	42	Fragmented IP protocol (proto=
14047	7.003267562	192.168.0.248	192.168.0.101	TCP	42	39527 → 3517 [SYN] Seq=0 Win=1

Zadanie 5 - Analiza pliku zawierającego dane pakietów z zainfekowanego komputera

1. Pobranie z platformy UPEL pliku z ruchem sieciowym



Wczytanie go do programu Wireshark.

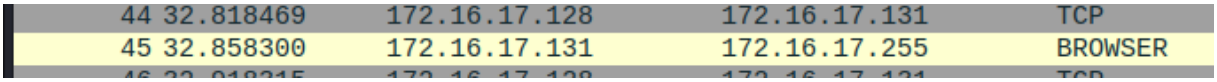


Plik zawiera zrzut pakietów sieciowych z ostatnich 8 minut działania zainfekowanego komputera. W trakcie tego czasu został przeprowadzony dodatkowo atak sieciowy wykorzystujący podatność systemową.

Odpowiedzi na pytania:

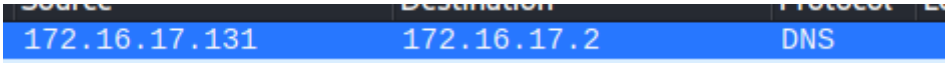
a) Podaj adres IP komputera, który został poddany analizie.

Adres IP komputera, który został poddany analizie posiada adres 172.16.17.131, jestem w stanie go zidentyfikować za pomocą protokołu BROWSER.



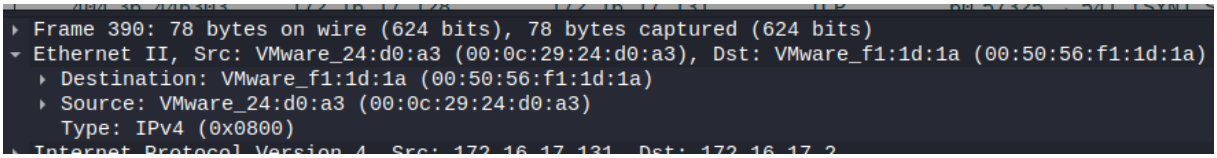
b) Podaj adres gatewaya tego komputera

GT, dla tego komputera wynosi 172.16.17.2



c) Czy przedstawione zdarzenie działo się w ramach wirtualnych maszyn?

TAK, zarówno GT jak i komputer to maszyny wirtualne



d) Czy w trakcie działania zainfekowanego komputera jesteśmy w stanie określić, czy stacja była skanowana w sieci w poszukiwaniu otwartych portów?
Oczywiście, że stacja była skanowana w poszukiwaniu otwartych portów, świadczy o tym ilość wysłanych pakietów TCP. Pakietów tych jest ok 2 tysięcy, pakiety są wysyłane na różne porty.

```
172.16.17.128    172.16.17.131    TCP    60 57325 → 1556
172.16.17.128    172.16.17.131    TCP    60 57325 → 255 [
172.16.17.128    172.16.17.131    TCP    60 57325 → 109 [
172.16.17.128    172.16.17.131    TCP    60 57325 → 2869
172.16.17.128    172.16.17.131    TCP    60 57325 → 1031
172.16.17.128    172.16.17.131    TCP    60 57325 → 2381
172.16.17.128    172.16.17.131    TCP    60 57325 → 10025
```

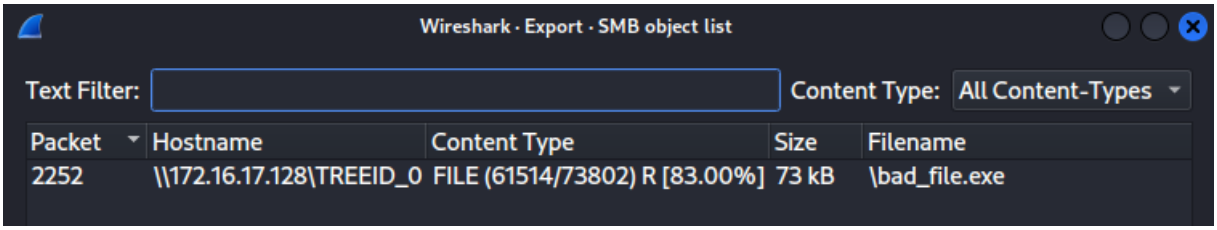
e) Jeśli tak, to przez kogo (IP sprawcy i jaką metodą), jeśli nie, to jakich informacji brakuje w badanym pliku?
IP sprawcy to 172.16.17.128, jest to adres źródłowy, z którego wykonywano zapytania TCP, w celu przeskanowania sieci.
Wydaje mi się, że sprawca mógł użyć polecenia sudo nmap -sS na adres IP komputera, który został poddany atakowi. Wynik w Wireshark jest bardzo podobny do tego, który zaprezentowałem wcześniej w tym raporcie.
f) W takcie działania zainfekowanego komputera został rozgłoszony ARP z adresem MAC (00:0c:29:ec:8a:14). Do kogo należy? Należy do naszego sprawcy, tj. 172.16.17.128

```
▸ Frame 25: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
▾ Ethernet II, Src: VMware_ec:8a:14 (00:0c:29:ec:8a:14), Dst: VMware_24:d0:a3 (0
  ▾ Destination: VMware_24:d0:a3 (00:0c:29:24:d0:a3)
      Address: VMware_24:d0:a3 (00:0c:29:24:d0:a3)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: VMware_ec:8a:14 (00:0c:29:ec:8a:14)
      Address: VMware_ec:8a:14 (00:0c:29:ec:8a:14)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
    Padding: 0000
▾ Internet Protocol Version 4, Src: 172.16.17.128, Dst: 172.16.17.131
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
```

g) Analizowane logi zawierają informacje o pliku wykonywalny exe. Sprawdź, kiedy został pobrany, z którego adresu i jak nazywa się plik?
Plik bad_file.exe został pobrany w szóstej minucie z adresu 172.16.17.128

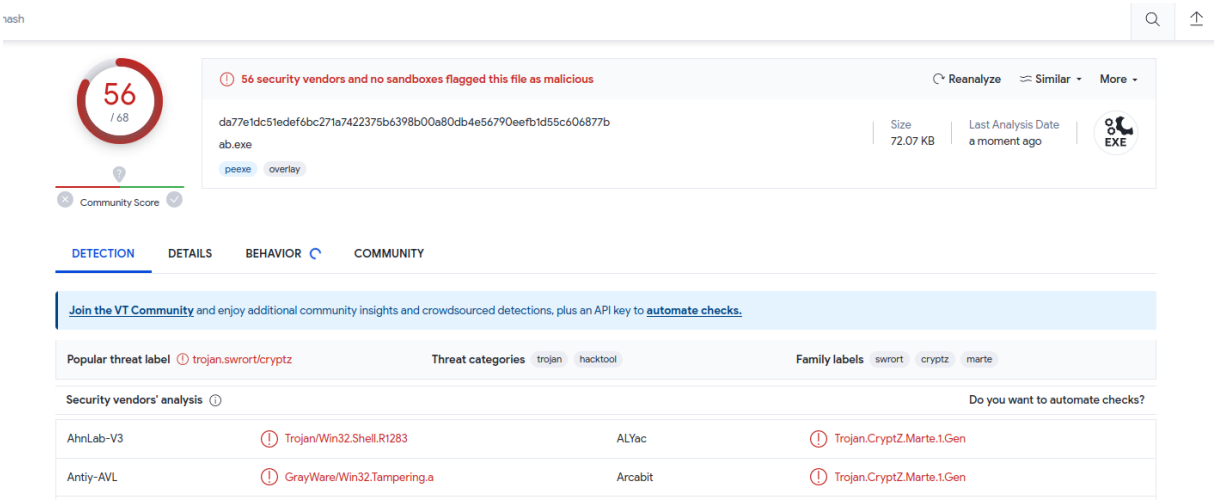
```
2176 360.690201  172.16.17.128  172.16.17.131  SMB   93 NT Create AndX Response, FID: 0x0000, Error: STATUS_OBJECT_NAME_NOT_FOUND
2188 362.248118  172.16.17.131  172.16.17.128  SMB  160 Trans2 Request, QUERY_PATH_INFO, Query File Basic Info, Path: \bad_file.exe
2189 362.251650  172.16.17.128  172.16.17.131  SMB  155 Trans2 Response, QUERY_PATH_INFO
2190 362.251732  172.16.17.131  172.16.17.128  SMB  160 Trans2 Request, QUERY_PATH_INFO, Query File Standard Info, Path: \bad_file.exe
2191 362.254977  172.16.17.128  172.16.17.131  SMB  137 Trans2 Response, QUERY_PATH_INFO
2192 362.255391  172.16.17.131  172.16.17.128  SMB  128 Trans2 Request, QUERY_FS_INFO, Query FS Attribute Info
2193 362.258165  172.16.17.128  172.16.17.131  SMB   93 Trans2 Response, QUERY_FS_INFO, Error: STATUS_NOT_FOUND
2194 362.258840  172.16.17.131  172.16.17.128  SMB  158 NT Create AndX Request, Path: \srvsvc
2195 362.261573  172.16.17.128  172.16.17.131  SMB   93 NT Create AndX Response, FID: 0x0000, Error: STATUS_OBJECT_NAME_NOT_FOUND
2196 362.262190  172.16.17.131  172.16.17.128  SMB  158 NT Create AndX Request, Path: \srvsvc
2197 362.265856  172.16.17.128  172.16.17.131  SMB   93 NT Create AndX Response, FID: 0x0000, Error: STATUS_OBJECT_NAME_NOT_FOUND
2213 363.717785  172.16.17.131  172.16.17.128  SMB  168 Trans2 Request, GET_DFS_REFERRAL, File: \172.16.17.128\SQCL
2214 363.720622  172.16.17.128  172.16.17.131  SMB   93 Trans2 Response, GET_DFS_REFERRAL, Error: STATUS_NOT_FOUND
2215 363.721455  172.16.17.131  172.16.17.128  SMB  168 Trans2 Request, GET_DFS_REFERRAL, File: \172.16.17.128\SQCL
2216 363.724759  172.16.17.128  172.16.17.131  SMB   93 Trans2 Response, GET_DFS_REFERRAL, Error: STATUS_NOT_FOUND
2217 363.725049  172.16.17.131  172.16.17.128  SMB  158 NT Create AndX Request, Path: \srvsvc
2218 363.727961  172.16.17.128  172.16.17.131  SMB   93 NT Create AndX Response, FID: 0x0000, Error: STATUS_OBJECT_NAME_NOT_FOUND
2219 363.728465  172.16.17.131  172.16.17.128  SMB  158 NT Create AndX Request, Path: \srvsvc
```

h) Przy użyciu opcji z Wireshark „Extract Object” wyciągnij odnaleziony plik, zapisz go w nowym folderze i przy pomocy narzędzia md5sum sprawdź jego sumę kontrolną.



Obliczenie sumy kontrolnej za pomocą narzędzia m5sum

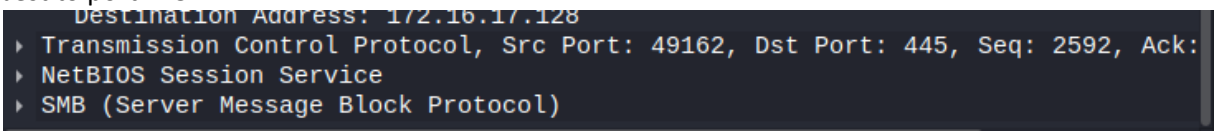
i) Pozyskaną sumę kontrolną wklej na stronie https://www.virustotal.com w zakładce search. Przedstaw i opisz wynik analizy.



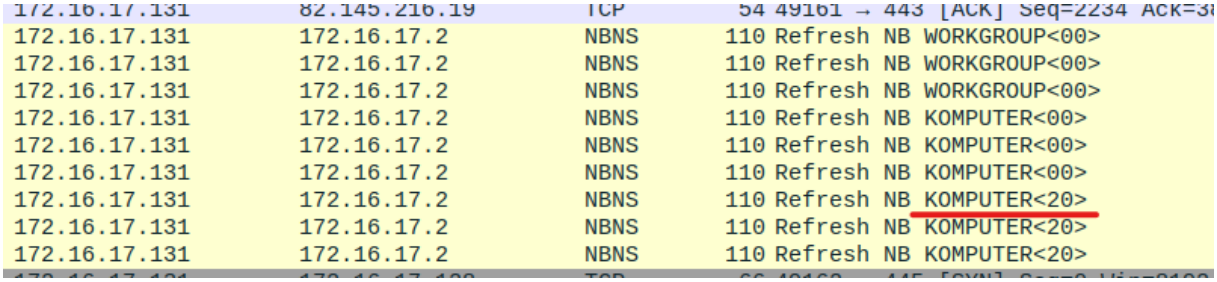
Jak widać na załączonym screenie plik ten jest bardzo niebezpieczny. Jest opisany w VirusTotal jako trojan.

j) Który z portów był wykorzystany do przesyłania danych pochodzących z ataku?

Jest to port 445.



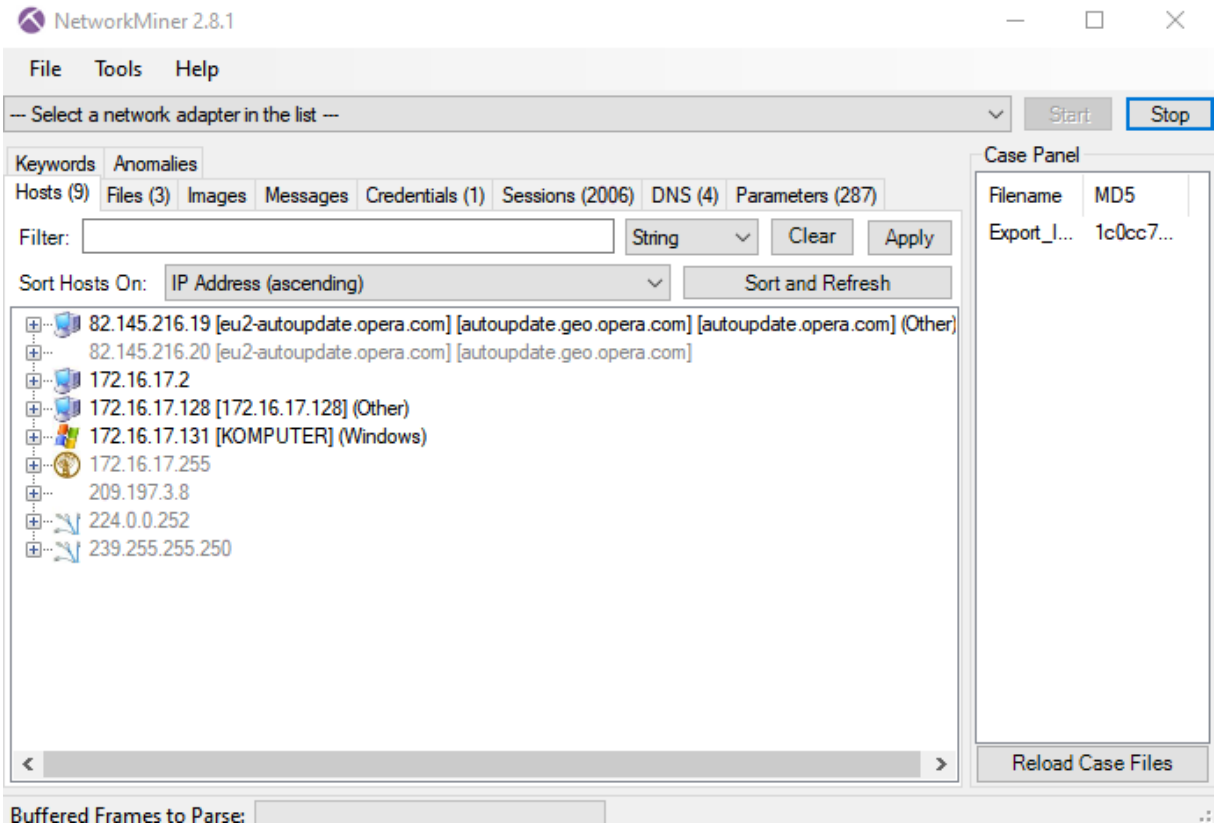
k) Podaj nazwę komputera, który został zaatakowany.



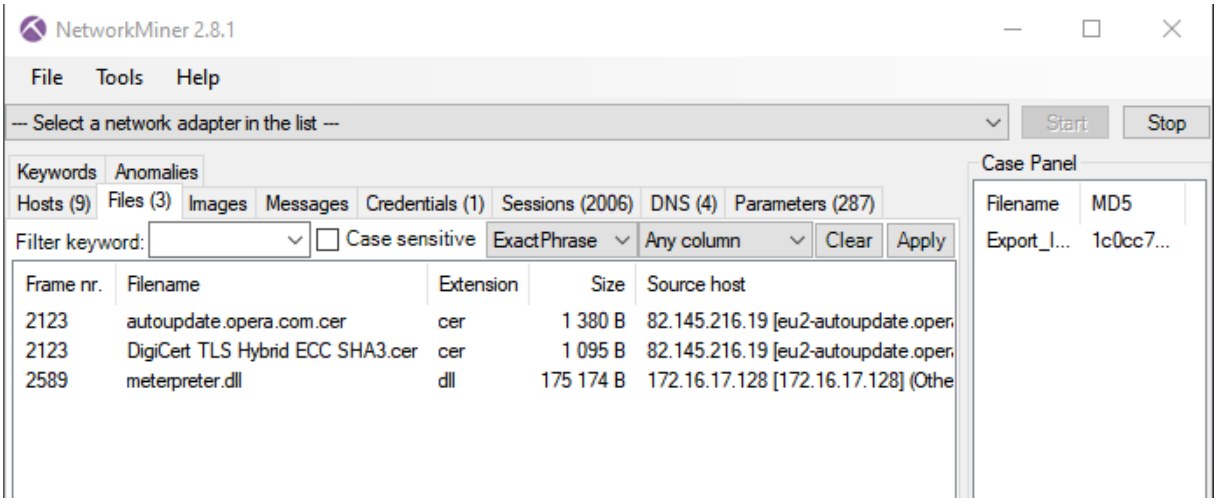
komputer ma bardzo oryginalną nazwę tj. KOMPUTER.

Zadanie 6. - NetworkMiner jako alternatywny program do analizy ruchu sieciowego

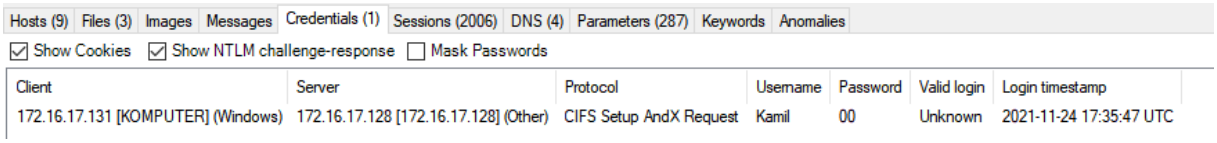
1. Zainstalowanie NetworkMiner i otwarcie pliku .pcap



W zakładce file (w plikach) nie widzimy pliku bad_file.exe



Jednak zakładka „Credentials" przedstawia bardzo ciekawe informacje



Mowa tutaj o loginie i haśle do serwera 172.16.17.128.

Według mnie pliku nie da się w odnaleźć/ Uważam, że może być to spowodowane tym, że był pobierany w kawałkach. Użyłem zakładki „Keywords” do odnalezienia szczątkowych informacji o pliku bad_file.exe.

610064005F00660069006...	.(.D.........+......V.........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..............................b.a.d._.f.i.l.e...e.x.e.	172.16.17.128 [172.16.17.128] (Other) TCP 445
610064005F00660069006...	.(.D.........+......V.........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..............................b.a.d._.f.i.l.e...e.x.e.	172.16.17.128 [172.16.17.128] (Other) TCP 445
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@......../..\.b.a.d._.f.i.l.e...e.x.e...M....	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	.(.D.........+......V.........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..............................b.a.d._.f.i.l.e...e.x.e.	172.16.17.128 [172.16.17.128] (Other) TCP 445
610064005F00660069006...	.......B.D.........E..........\.b.a.d._.f.i.l.e...e.x.e.:.Z....	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	.......".D.........%..........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	.......".D.........%..........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	..................@...........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	.(.D.........+......V.........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162
610064005F00660069006...	$.A.................$...........b.a.d._.f.i.l.e...e.x.e.	172.16.17.128 [172.16.17.128] (Other) TCP 445
610064005F00660069006...	.(.D.........+......V.........\.b.a.d._.f.i.l.e...e.x.e...	172.16.17.131 [KOMPUTER] (Windows) TCP 49162