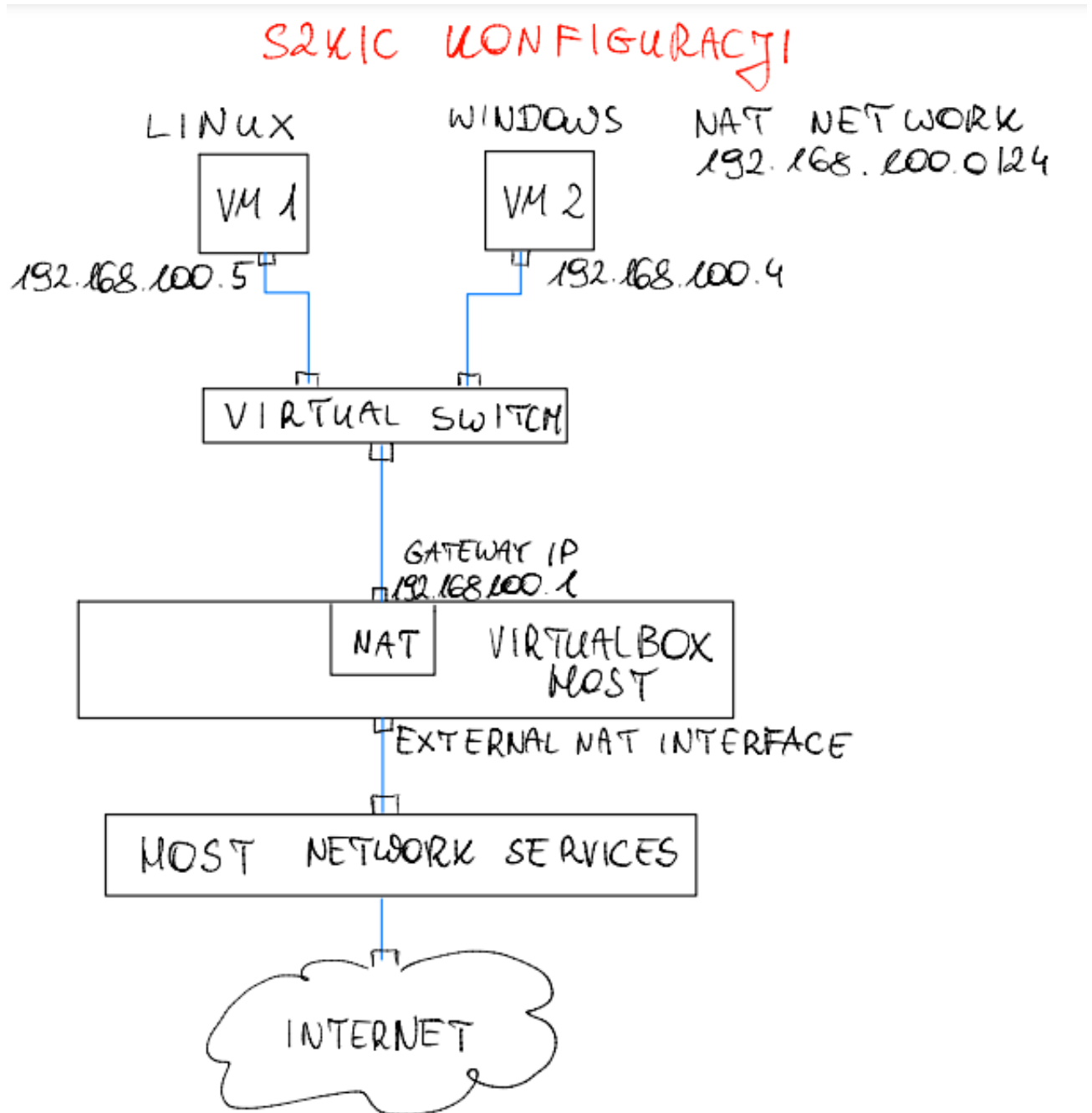


Informatyka śledcza Laboratorium nr 7

Raport – Nikodem Jakubowski

Zadanie 1 – Przygotowanie środowiska testowego.

Przedstawię tutaj szkic środowiska, które ostatecznie wykorzystałem w laboratorium. Początkowo wszystko miałem zrobione w domyślnej instancji NAT i zrezygnowałem z tego, bo urządzenia się nie widziały. Ostateczny szkic.



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

Użycie ifconfig na maszynie z linuxem.

```
(user@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe28:cbbb prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:28:cb:bb txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 590 (590.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 2910 (2.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Sprawdzenie tablicy routingu.

```
(user@kali)-[~]
$ route -n
Kernel IP routing table
Destination        Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0            10.0.2.2       0.0.0.0         UG    100    0      0 eth0
10.0.2.0           0.0.0.0        255.255.255.0   U     100    0      0 eth0
```

VirtualBox maszynie z Windowsem przydziela taki sam adres, obie maszyny korzystają z tego samego gateway. Jak przeczytałem w dokumentacji – to normalne, VirtualBox podobno ma kilka osobnych adapterów dla każdej maszyny...

```
C:\Users\user>ipconfig

Windows IP Configuration

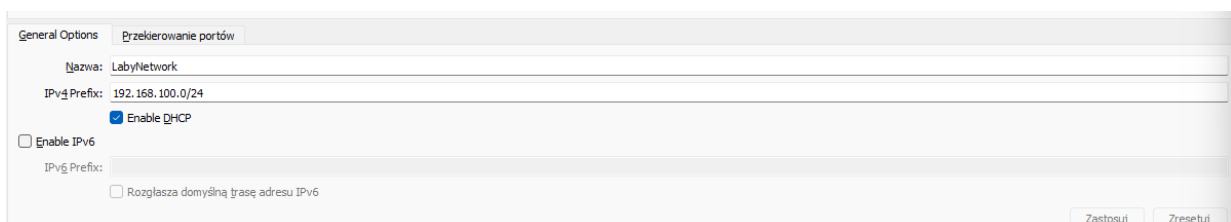
Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : home
    Link-local IPv6 Address . . . . . : fe80::78e0:a1fd:d163:84b2%7
    IPv4 Address. . . . . : 10.0.2.15
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.2.2
```

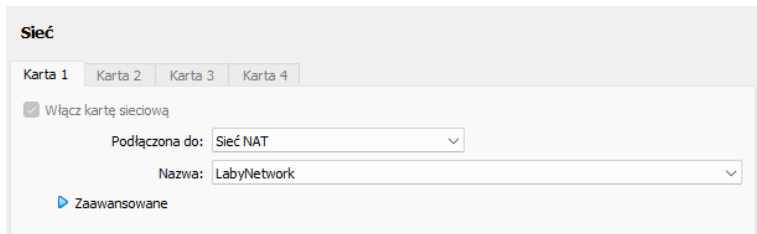
W tej sytuacji maszyny na pewno nie będą się widziały, zmiana planów. Poszukałem w dokumentacji i jest opcja stworzenia NAT Network.

Aby maszyny się widziały, stworzę osobną sieć NAT w VirtualBox.

Konfiguracja sieci, oczywiście wybieramy adres z puli prywatnej. Poniżej nowa sieć NAT stworzona przeze mnie. Nazwa to LabyNetwork, a adres 192.168.100.0/24.



Następnie na obu maszynach w ustawieniach karty zmieniamy na sieć NAT, konkretnie - LabyNetwork.



Nowy adres maszyny z Windows.

```
C:\Users\user>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : home
    Link-local IPv6 Address . . . . . : fe80::78e0:a1fd:d163:84b2%7
    IPv4 Address. . . . . : 192.168.100.4
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.100.1
```

Po reboot maszyna wirtualna dostała adres.

```
(user@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.5 netmask 255.255.255.0 broadcast 192.168.100.255
    inet6 fe80::a00:27ff:fe28:cbbb prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:28:cb:bb txqueuelen 1000 (Ethernet)
    RX packets 30 bytes 6502 (6.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 3700 (3.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Skanuję przy pomocy nmap. Dla porównania, przy pomocy sudo w ogóle coś widać!

```
(user@kali)-[~]
$ nmap -sn 192.168.100.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-09 16:30 CET
Nmap scan report for 192.168.100.1
Host is up (0.00081s latency).
Nmap scan report for 192.168.100.5
Host is up (0.00039s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 3.42 seconds

(user@kali)-[~]
$ sudo nmap -sn 192.168.100.0/24
[sudo] password for user:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-09 16:30 CET
Nmap scan report for 192.168.100.1
Host is up (0.00047s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.100.2
Host is up (0.00032s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.100.3
Host is up (0.00031s latency).
MAC Address: 08:00:27:31:9F:7B (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.100.4
Host is up (0.0011s latency).
MAC Address: 08:00:27:2C:F5:60 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.100.5
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.38 seconds
```

Zadanie 3 – Analiza ruchu sieciowego przy wykorzystaniu narzędzia TCPdump.

Prześląłem ping od maszyny z Windows do maszyny z Linuxem i przechwyciłem garść informacji.

```
(user@kali)-[~]
$ sudo tcpdump -i eth0 -v
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:33:26.310637 IP (tos 0x0, ttl 128, id 45538, offset 0, flags [none], proto ICMP (1), length 60)
    192.168.100.4 > 192.168.100.5: ICMP echo request, id 1, seq 28, length 40
16:33:26.310689 IP (tos 0x0, ttl 64, id 41050, offset 0, flags [none], proto ICMP (1), length 60)
    192.168.100.5 > 192.168.100.4: ICMP echo reply, id 1, seq 28, length 40
16:33:26.344353 IP (tos 0x0, ttl 64, id 42809, offset 0, flags [DF], proto UDP (17), length 72)
    192.168.100.5.59125 > funbox.home.domain: 4357+ PTR? 5.100.168.192.in-addr.arpa. (44)
16:33:26.348731 IP (tos 0x0, ttl 255, id 2456, offset 0, flags [none], proto UDP (17), length 72)
    funbox.home.domain > 192.168.100.5.59125: 4357 NXDomain* 0/0/0 (44)
16:33:26.348811 IP (tos 0x0, ttl 64, id 2068, offset 0, flags [DF], proto UDP (17), length 72)
    192.168.100.5.60186 > funbox.home.domain: 57879+ PTR? 4.100.168.192.in-addr.arpa. (44)
16:33:26.353255 IP (tos 0x0, ttl 255, id 2457, offset 0, flags [none], proto UDP (17), length 72)
    funbox.home.domain > 192.168.100.5.60186: 57879 NXDomain* 0/0/0 (44)
16:33:26.438254 IP (tos 0x0, ttl 64, id 21566, offset 0, flags [DF], proto UDP (17), length 70)
    192.168.100.5.33567 > funbox.home.domain: 38644+ PTR? 1.1.168.192.in-addr.arpa. (42)
16:33:26.441688 IP (tos 0x0, ttl 255, id 2458, offset 0, flags [none], proto UDP (17), length 95)
    funbox.home.domain > 192.168.100.5.33567: 38644* 1/0/0 1.1.168.192.in-addr.arpa. PTR funbox.home. (67)
16:33:27.355426 IP (tos 0x0, ttl 128, id 45539, offset 0, flags [none], proto ICMP (1), length 60)
```

```
61 packets captured
61 packets received by filter
0 packets dropped by kernel
```

Udało się przechwycić oczywiście zapytanie ARP, które idzie przez broadcast, protokół ICMP odpowiedzialny za ping, adres wysyłającego oraz router, z którym komunikuje się urządzenie (funbox).

Pinguję bramę domyślną, wszystko jest w porządku.

```
(user@kali)-[~]
$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 192.168.100.1 0.0.0.0 UG 100 0 0 eth0
192.168.100.0 0.0.0.0 255.255.255.0 U 100 0 0 eth0

(user@kali)-[~]
$ ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data:
64 bytes from 192.168.100.1: icmp_seq=1 ttl=255 time=0.553 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=255 time=1.58 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=255 time=0.664 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=255 time=2.04 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=255 time=0.882 ms
64 bytes from 192.168.100.1: icmp_seq=6 ttl=255 time=0.712 ms
64 bytes from 192.168.100.1: icmp_seq=7 ttl=255 time=4.01 ms
64 bytes from 192.168.100.1: icmp_seq=8 ttl=255 time=1.20 ms
^C
— 192.168.100.1 ping statistics —
8 packets transmitted, 8 received, 0% packet loss, time 7100ms
rtt min/avg/max/mdev = 0.553/1.453/4.007/1.075 ms
```

Otworzyłem sobie dwa terminale naraz, żeby przefiltrować ping.

```
(user@kali)-[~]
$ ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=255 time=1.89 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=255 time=0.539 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=255 time=0.978 ms
^C
— 192.168.100.1 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2019ms
rtt min/avg/max/mdev = 0.539/1.135/1.889/0.562 ms

user@kali: ~
File Actions Edit View Help
(user@kali)-[~]
$ sudo tcpdump -i eth0 -v host 192.168.100.1
[sudo] password for user:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:41:35.790476 IP (tos 0x0, ttl 64, id 8067, offset 0, flags [DF], proto ICMP (1), length 84)
192.168.100.5 > 192.168.100.1: ICMP echo request, id 39379, seq 1, length 64
16:41:35.792345 IP (tos 0x0, ttl 255, id 8067, offset 0, flags [DF], proto ICMP (1), length 84)
192.168.100.1 > 192.168.100.5: ICMP echo reply, id 39379, seq 1, length 64
16:41:36.792342 IP (tos 0x0, ttl 64, id 8150, offset 0, flags [DF], proto ICMP (1), length 84)
192.168.100.5 > 192.168.100.1: ICMP echo request, id 39379, seq 2, length 64
16:41:36.792858 IP (tos 0x0, ttl 255, id 8150, offset 0, flags [DF], proto ICMP (1), length 84)
192.168.100.1 > 192.168.100.5: ICMP echo reply, id 39379, seq 2, length 64
16:41:37.810031 IP (tos 0x0, ttl 64, id 8353, offset 0, flags [DF], proto ICMP (1), length 84)
192.168.100.5 > 192.168.100.1: ICMP echo request, id 39379, seq 3, length 64
16:41:37.810975 IP (tos 0x0, ttl 255, id 8353, offset 0, flags [DF], proto ICMP (1), length 84)
192.168.100.1 > 192.168.100.5: ICMP echo reply, id 39379, seq 3, length 64
```

Użyłem takiej komendy, żeby przechwycić wyświetlenie strony pudelek.pl. Przeskakuje na raz dużo linii, więc trzeba było szybko zatrzymać.

```
(user@kali)-[~]
$ sudo tcpdump -i eth0 dst port 80 or dst port 443
```

Efekt.

```
16:48:33.229345 IP 192.168.100.5.34230 > pudelek.pl.https: Flags [..], ack 79308, win 65535, length 0
16:48:33.229989 IP 192.168.100.5.34230 > pudelek.pl.https: Flags [..], ack 79521, win 65535, length 0
16:48:33.243130 IP 192.168.100.5.36678 > a2-16-110-67.deploy.static.akamaitechnologies.com.https: Flags [..], ack 7947,
16:48:33.249353 IP 192.168.100.5.36678 > a2-16-110-67.deploy.static.akamaitechnologies.com.https: Flags [..], ack 8691,
16:48:33.280082 IP 192.168.100.5.42420 > server-18-244-97-213.waw51.r.cloudfront.net.https: Flags [..], ack 7865, win 6
16:48:33.280936 IP 192.168.100.5.42420 > server-18-244-97-213.waw51.r.cloudfront.net.https: Flags [..], ack 8528, win 6
```


Zadanie 4 – Analiza ruchu sieciowego przy wykorzystaniu programu Wireshark.

Używam nmap -Ss.

```
(user@kali)-[~]
$ sudo nmap -sS 192.168.100.0/24
[sudo] password for user:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-09 17:03 CET
Nmap scan report for 192.168.100.1
Host is up (0.0018s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.100.2
Host is up (0.0029s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.100.3
Host is up (0.00028s latency).
All 1000 scanned ports on 192.168.100.3 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:31:9F:7B (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.100.4
Host is up (0.0099s latency).
All 1000 scanned ports on 192.168.100.4 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 08:00:27:2C:F5:60 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.100.5
Host is up (0.0000070s latency).
All 1000 scanned ports on 192.168.100.5 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (5 hosts up) scanned in 16.01 seconds
```

Po pierwsze widzimy, że host 192.168.100.5 (Linux) wysłał zapytania ARP, żeby rozpoznać się w sieci.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.5
2	0.000000	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.2? Tell 192.168.100.5
3	0.016576	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.3? Tell 192.168.100.5
4	0.016576	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.4? Tell 192.168.100.5
5	0.016576	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.6? Tell 192.168.100.5
6	0.016576	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.7? Tell 192.168.100.5
7	0.016576	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.8? Tell 192.168.100.5
8	0.016576	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.9? Tell 192.168.100.5
9	0.016576	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.10? Tell 192.168.100.5
10	0.016576	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.11? Tell 192.168.100.5
11	0.016606	PCSSystemtec_2c:f5:...	PCSSystemtec_28:cb:...	ARP	42	192.168.100.4 is at 08:00:27:2c:f5:60
12	0.048847	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.14? Tell 192.168.100.5
13	0.048847	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.15? Tell 192.168.100.5
14	0.048847	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.16? Tell 192.168.100.5
15	0.048847	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.17? Tell 192.168.100.5
16	0.048847	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.18? Tell 192.168.100.5
17	0.048847	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.19? Tell 192.168.100.5
18	0.048847	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.20? Tell 192.168.100.5
19	0.048847	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.21? Tell 192.168.100.5
20	0.104023	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.24? Tell 192.168.100.5
21	0.104419	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.25? Tell 192.168.100.5
22	0.104419	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.26? Tell 192.168.100.5
23	0.104927	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.27? Tell 192.168.100.5
24	0.104927	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.28? Tell 192.168.100.5
25	0.104927	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.29? Tell 192.168.100.5
26	0.157866	PCSSystemtec_28:cb:...	Broadcast	ARP	60	Who has 192.168.100.32? Tell 192.168.100.5

Następnie, co typowe dla wyszukiwania sS (stealth scan), host Linuxowy wysyła tylko inicjalizującą część „3-way handshake”, czyli SYN. Nasza maszyna odpowiada SYN-ACK (czego się domyślamy), a host Linuxowy w ostatniej chwili się rozmyśla i nie odpowiada ACK (nie nawiązuje pełnego połączenia).

Użycie nmap z fragmentacją.

```
(user@kali)-[~]
└─$ sudo nmap 192.168.100.0/24 -data-length 32 -f -T5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-09 17:20 CET
Nmap scan report for 192.168.100.1
Host is up (0.0031s latency).
All 1000 scanned ports on 192.168.100.1 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.100.2
Host is up (0.0028s latency).
All 1000 scanned ports on 192.168.100.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.100.3
Host is up (0.00069s latency).
All 1000 scanned ports on 192.168.100.3 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:31:9F:7B (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.100.4
Host is up (0.0018s latency).
All 1000 scanned ports on 192.168.100.4 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 08:00:27:2C:F5:60 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.100.5
Host is up (0.000062s latency).
All 1000 scanned ports on 192.168.100.5 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (5 hosts up) scanned in 11.75 seconds
```

Efekt.

No.	Time	Source	Destination	Protocol	Length	Info
655	34.209968	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=40, ID=de13) [Reas
656	34.209968	192.168.100.5	192.168.100.4	VNC	60	
657	34.210429	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=3741) [Reass
658	34.210429	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=3741) [Reass
659	34.210429	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=16, ID=3741) [Reas
660	34.210429	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=24, ID=3741) [Reas
661	34.210429	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=32, ID=3741) [Reas
662	34.210863	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=40, ID=3741) [Reas
663	34.210863	192.168.100.5	192.168.100.4	TCP	60	62581 → 80 [SYN] Seq=0 Win=1024 Len=32 MSS=1460
664	34.211208	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=44a5) [Reass
665	34.211208	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=44a5) [Reass
666	34.211208	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=16, ID=44a5) [Reas
667	34.211208	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=24, ID=44a5) [Reas
668	34.211623	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=32, ID=44a5) [Reas
669	34.211623	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=40, ID=44a5) [Reas
670	34.211623	192.168.100.5	192.168.100.4	TCP	60	62581 → 3389 [SYN] Seq=0 Win=1024 Len=32 MSS=1460
671	34.214934	192.168.100.5	192.168.100.4	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=f8c3) [Reass

Skanowanie w ten sposób bardzo rzuca się w oczy w wiresharku w momencie przechwytywania. Powoduje to ilość rekordów jak i ich kolor. Niektóre z tych pakietów są zniekształcone albo mają podejrzaną zawartość. Ta metoda mi się wydaje o wiele bardziej widoczna i łatwiejsza do wykrycia „na żywo”, ale gdyby ktoś grzebał w .pcap z takiego urządzenia po dłuższym czasie, to mógłby tego nawet nie zauważyć.

Zadanie 5 – Analiza pliku zawierającego dane pakietów z zainfekowanego komputera.

Poniżej dodaję odpowiedzi do odpowiednich podpunktów.

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

Myślę, że jest to adres 172.16.17.131. Pojawia się on bardzo często i wykonuje typowe czynności dla hosta jak zapytania DNS. Równie często pojawia się adres 172.16.17.128, ale on raczej dokonywał jakiegoś skanowania typu sS.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.17.131	172.16.17.2	DNS	78	Standard query 0x040a A isatap.localdomain
2	1.606894	172.16.17.131	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
3	3.931885	172.16.17.131	172.16.17.2	DNS	85	Standard query 0xa9ab A teredo.ipv6.microsoft.com
4	4.955009	172.16.17.2	172.16.17.131	DNS	158	Standard query response 0xa9ab No such name A teredo.ipv6
5	4.009396	172.16.17.131	172.16.17.2	DNS	78	Standard query 0x040a A isatap.localdomain
6	8.019457	172.16.17.131	172.16.17.2	DNS	78	Standard query 0xaf02 A isatap.localdomain
7	9.032584	172.16.17.131	172.16.17.2	DNS	78	Standard query 0xaf02 A isatap.localdomain
8	11.045871	172.16.17.131	172.16.17.2	DNS	78	Standard query 0xaf02 A isatap.localdomain

Dodatkowo mamy HostAnnouncement dla przeglądarki z adresu 172.16.17.131.

```

45 32,858398 172.10.17.131 172.10.17.255 BROWSER 243 Host Announcement COMPUTER, Workstation, Server, NT Workstation
60 57324 ~ 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
47 32,918484 172.10.17.128 172.16.17.131 TCP 60 57324 ~ 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
46 32,918617 172.10.17.131 172.16.17.131 TCP 60 57324 ~ 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
49 32,918735 172.10.17.128 172.16.17.131 TCP 60 57324 ~ 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
50 32,918841 172.10.17.128 172.16.17.131 TCP 60 57324 ~ 139 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
51 32,918952 172.10.17.128 172.16.17.131 TCP 60 57324 ~ 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 243: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits) on interface 0
Ethernet II, Src: VMware24:00:a3, (00:0c:29:24:00:a3), Dat: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 172.16.17.131, Dest: 172.16.17.255
User Datagram Protocol, Src Port: 138, Dest Port: 138
Netbios Datagram Service
SMB (Server Message Block Protocol)
SMB Mailslot Protocol
Mailslot Message Browser Protocol
Command: Host Announcement (0x01)
Update Count: 0
Update Periodicity: 2 minutes
Host Name: K0MPUTER
Windows Version: Windows 7 or Windows Server 2008 R2
OS Major Version: 6
OS Minor Version: 1
Server Type: 0x00001003, Workstation, Server, NT Workstation
Browser Protocol Major Version: 15
Browser Protocol Minor Version: 1
Signature: 0xa55
Host Comment:

```

- b. Podaj adres gatewaya tego komputera.

Gateway to 172.16.17.2.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.17.131	172.16.17.2	DNS	78	Standard query 0x0000 a isatap.localdomain
2	0.010005	172.16.17.131	172.16.17.2	DNS	95	Standard query 0x0000 a isatap.ipv6.microsoft.com
3	4.955009	172.16.17.2	172.16.17.131	DNS	158	Standard query response 0xa9ab No such name a isatap.ipv6.microsoft.com SOA ns1-04.azure-dns.com
4	0.000396	172.16.17.131	172.16.17.2	DNS	78	Standard query 0x000a a isatap.localdomain
6	0.019457	172.16.17.131	172.16.17.2	DNS	78	Standard query 0xaf02 a isatap.localdomain
7	0.032084	172.16.17.131	172.16.17.2	DNS	78	Standard query 0xaf02 a isatap.localdomain
8	11.045871	172.16.17.131	172.16.17.2	DNS	78	Standard query 0xaf02 a isatap.localdomain
9	12.230699	172.16.17.131	172.16.17.2	DNS	78	Standard query 0x0d04 a isatap.localdomain
10	23.244152	172.16.17.131	172.16.17.2	DNS	78	Standard query 0x0d04 a isatap.localdomain
11	23.258151	172.16.17.131	172.16.17.2	DNS	78	Standard query 0x0d04 a isatap.localdomain

Widać to po komunikacji DNS hosta z bramą.

- c. Czy przedstawione zdarzenie miało miejsce w ramach wirtualnych maszyn?

Na jakiej podstawie zostały wyciągnięte wnioski?

Tak zdecydowanie w obrębie maszyn wirtualnych. Można to zauważyć w pakietach ethernetowych, których opis podaję.

```

▶ Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
▶ Ethernet II, Src: VMware_24:d0:a3 (00:0c:29:24:d0:a3), Dst: VMware_f1:1d:1a (00:50:56:f1:1d:1a)
  ▶ Destination: VMware_f1:1d:1a (00:50:56:f1:1d:1a)
  ▶ Source: VMware_24:d0:a3 (00:0c:29:24:d0:a3)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 172.16.17.131, Dst: 172.16.17.2
▶ User Datagram Protocol, Src Port: 64538, Dst Port: 53
▶ Domain Name System (query)

```


d. Czy w trakcie działania zainfekowanego komputera jesteśmy w stanie określić, czy stacja była skanowana w sieci w poszukiwaniu otwartych portów?

Tak, zdecydowanie widać, że maszyna 172.16.17.128 wykonywała coś w rodzaju stealth scan i badała, na których portach dostanie odpowiedź. W kolumnie info na samym początku są różne porty docelowe.

No.	Time	Source	Destination	Protocol	Length	Info
25	31.715128	172.16.17.128	172.16.17.131	TCP	60	57324 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
26	31.715128	172.16.17.128	172.16.17.131	TCP	60	57324 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
27	31.715369	172.16.17.128	172.16.17.131	TCP	60	57324 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
28	31.715370	172.16.17.128	172.16.17.131	TCP	60	57324 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29	31.715370	172.16.17.128	172.16.17.131	TCP	60	57324 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
30	31.715560	172.16.17.128	172.16.17.131	TCP	60	57324 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
31	31.715698	172.16.17.128	172.16.17.131	TCP	60	57324 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
32	31.715699	172.16.17.128	172.16.17.131	TCP	60	57324 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33	31.715926	172.16.17.128	172.16.17.131	TCP	60	57324 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
34	31.715927	172.16.17.128	172.16.17.131	TCP	60	57324 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
35	32.817707	172.16.17.128	172.16.17.131	TCP	60	57325 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36	32.817708	172.16.17.128	172.16.17.131	TCP	60	57325 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
37	32.817709	172.16.17.128	172.16.17.131	TCP	60	57325 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
38	32.817879	172.16.17.128	172.16.17.131	TCP	60	57325 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
39	32.817986	172.16.17.128	172.16.17.131	TCP	60	57325 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
40	32.817987	172.16.17.128	172.16.17.131	TCP	60	57325 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
41	32.818188	172.16.17.128	172.16.17.131	TCP	60	57325 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
42	32.818329	172.16.17.128	172.16.17.131	TCP	60	57325 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
43	32.818329	172.16.17.128	172.16.17.131	TCP	60	57325 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
44	32.818469	172.16.17.128	172.16.17.131	TCP	60	57325 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
46	32.918315	172.16.17.128	172.16.17.131	TCP	60	57324 → 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

e. Jeśli tak, to przez kogo (IP sprawcy i jaką metodą), jeśli nie, to jakich informacji brakuje w badanym pliku?

Opisałem to już powyżej w rozważaniach. Adres hosta 172.16.17.128 od początku był podejrzany.

f. W trakcie działania zainfekowanego komputera został rozgłoszony ARP z adresem MAC (00:0c:29:ec:8a:14). Do kogo należy?

Należy on do hosta o adresie 172.16.17.128.

No.	Time	Source	Destination	Protocol	Length	Info
25	31.715128	172.16.17.128	172.16.17.131	TCP	60	57324 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
26	31.715128	172.16.17.128	172.16.17.131	TCP	60	57324 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
27	31.715369	172.16.17.128	172.16.17.131	TCP	60	57324 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
28	31.715370	172.16.17.128	172.16.17.131	TCP	60	57324 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29	31.715370	172.16.17.128	172.16.17.131	TCP	60	57324 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
30	31.715560	172.16.17.128	172.16.17.131	TCP	60	57324 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
31	31.715698	172.16.17.128	172.16.17.131	TCP	60	57324 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
32	31.715699	172.16.17.128	172.16.17.131	TCP	60	57324 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33	31.715926	172.16.17.128	172.16.17.131	TCP	60	57324 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
34	31.715927	172.16.17.128	172.16.17.131	TCP	60	57324 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
35	32.817707	172.16.17.128	172.16.17.131	TCP	60	57325 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36	32.817708	172.16.17.128	172.16.17.131	TCP	60	57325 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
37	32.817709	172.16.17.128	172.16.17.131	TCP	60	57325 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
38	32.817879	172.16.17.128	172.16.17.131	TCP	60	57325 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
39	32.817986	172.16.17.128	172.16.17.131	TCP	60	57325 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
40	32.817987	172.16.17.128	172.16.17.131	TCP	60	57325 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
41	32.818188	172.16.17.128	172.16.17.131	TCP	60	57325 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
42	32.818329	172.16.17.128	172.16.17.131	TCP	60	57325 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
43	32.818329	172.16.17.128	172.16.17.131	TCP	60	57325 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
44	32.818469	172.16.17.128	172.16.17.131	TCP	60	57325 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
46	32.918315	172.16.17.128	172.16.17.131	TCP	60	57324 → 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

eth.addr == 00:0c:29:ec:8a:14

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 (00:0c:29:24:d0:a3)

Destination: VMware_24:d0:a3 (00:0c:29:24:d0:a3)

Source: VMware_ec:8a:14 (00:0c:29:ec:8a:14)

Address: VMware_ec:8a:14 (00:0c:29:ec:8a:14)

... .. = LG bit: Globally unique address (factory default)

... .. = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

Padding: 0000

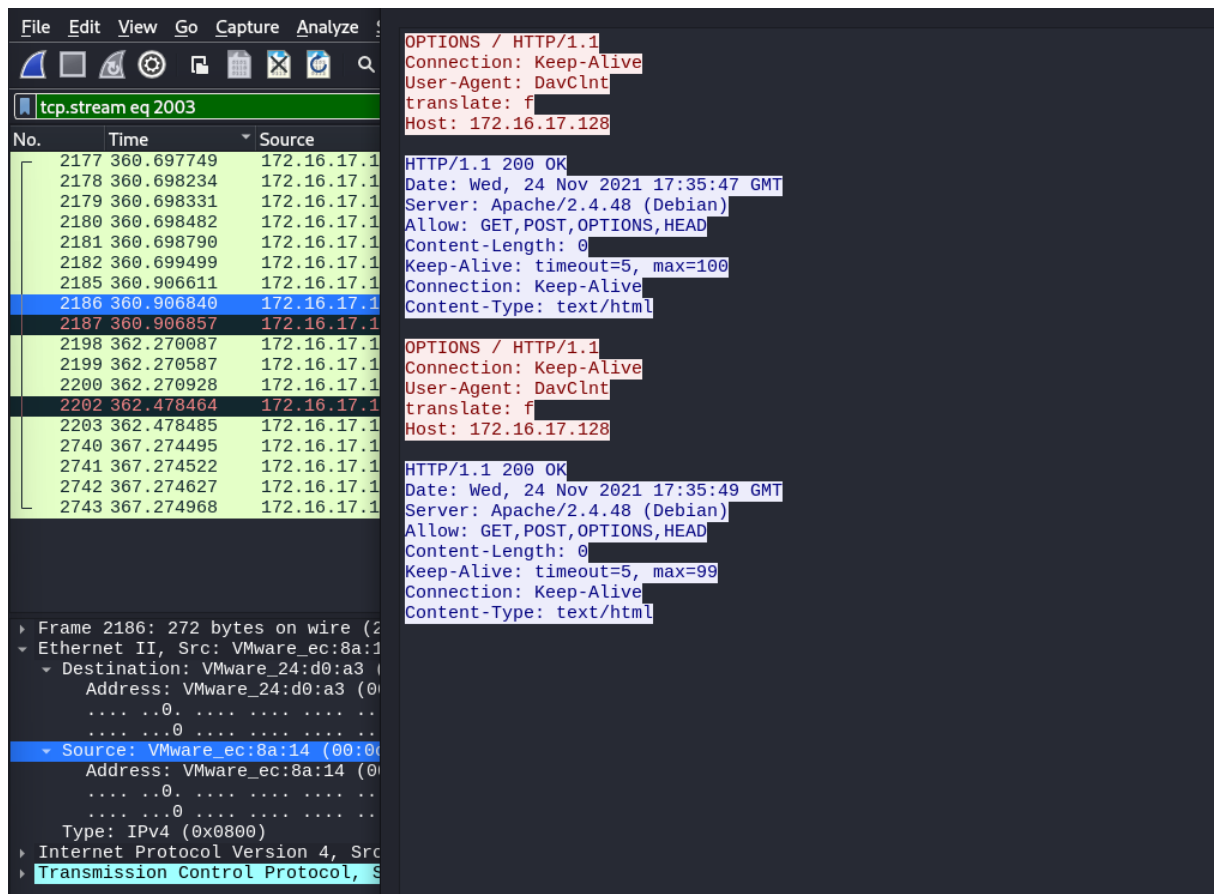
Internet Protocol Version 4, Src: 172.16.17.128, Dst: 172.16.17.131

Transmission Control Protocol, Src Port: 57324, Dst Port: 445, Seq: 0, Len: 0

0000 00 0c 29 24
0010 00 2c 6d d9
0020 11 83 df ec
0030 04 00 f2 c7

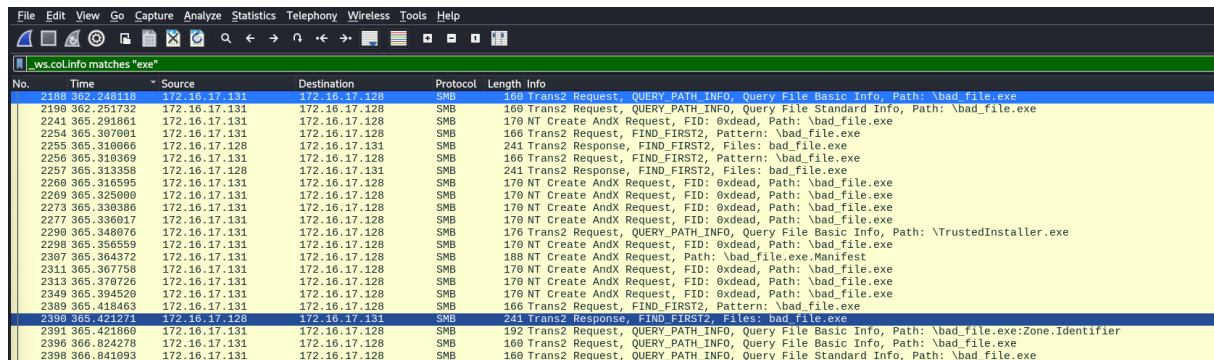
g. Analizowane logi zawierają informacje o pliku wykonywalny exe. Sprawdź, kiedy został pobrany, z którego adresu i jak nazywa się plik?

Najpierw szukałem po http response, a później podążałem strumieniem.



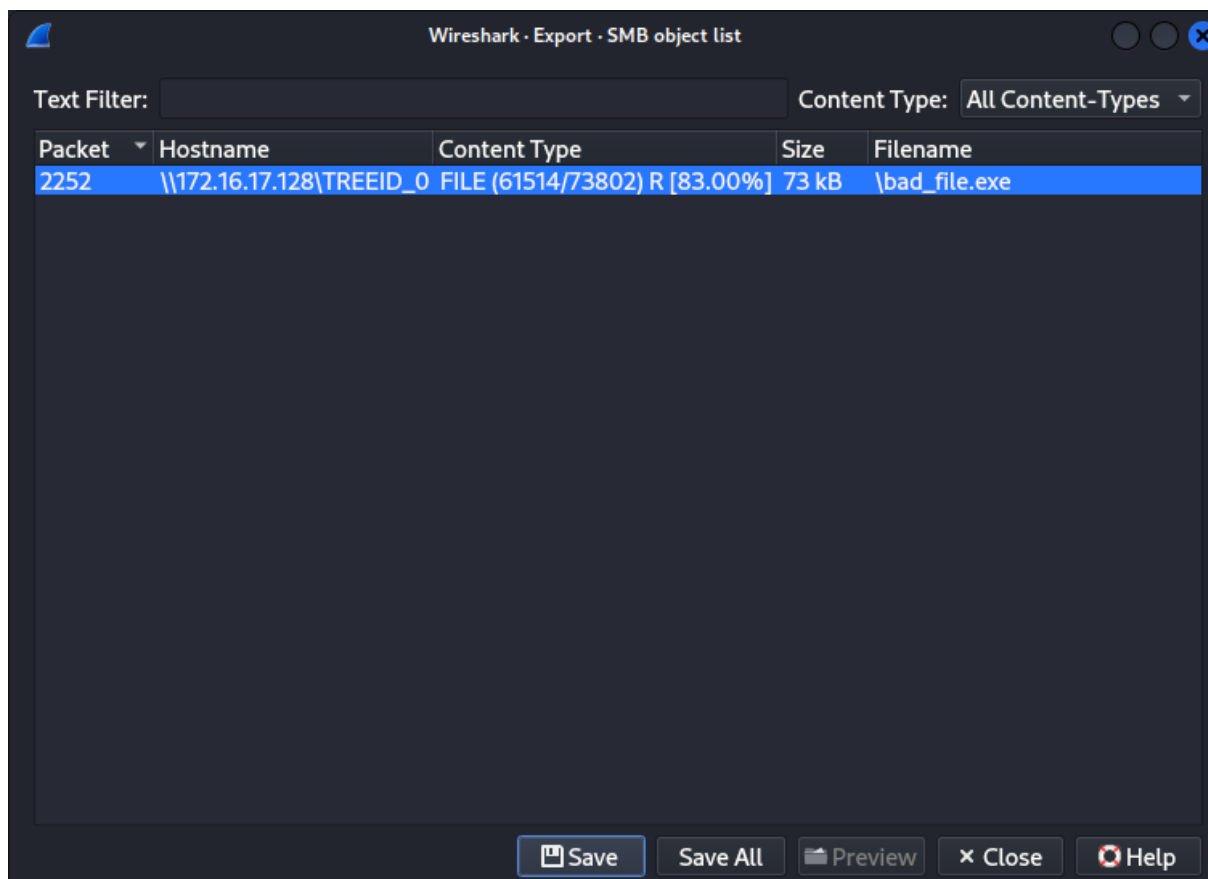
Nie znalazłem jednak tego, czego szukałem. Kombinowałem dalej.

Znalazłem w dokumentacji opcję filtrowania danych w danej kolumnie po nazwie.



Widzimy plik bad_file.exe.

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.

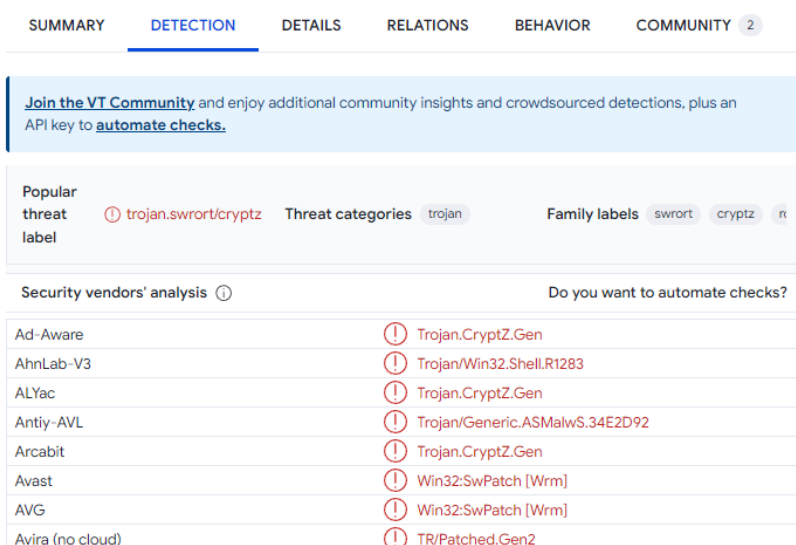


Pozyskuję sumę kontrolną.

```
(user@kali)-[~/Desktop]
$ md5sum %5cbad_file.exe
a8910c628418380eed87b6e58ee61019 %5cbad_file.exe
```

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

Widzimy, że jest to trojan, dużo sygnatur i zgłoszeń.



j. Który z portów był wykorzystywany do przesyłania danych pochodzących z ataku? Podaj nazwę komputera, który został zaatakowany.

Zaatakowany został komputer 172.16.17.131 o nazwie: Komputer\Kamil.

No.	Time	Source	Destination	Protocol	Length	Info
2151	360.666992	172.16.17.131	172.16.17.128	TCP	60	49162 → 445 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM
2152	360.666926	172.16.17.128	172.16.17.131	TCP	60	445 → 49162 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 SACK_PERM WS=128
2153	360.666457	172.16.17.131	172.16.17.128	TCP	54	49162 → 445 [ACK] Seq=1 Ack=1 Win=0 Len=0
2154	360.666595	172.16.17.131	172.16.17.128	SMB	213	Negotiate Protocol Request
2155	360.666848	172.16.17.128	172.16.17.131	TCP	60	445 → 49162 [ACK] Seq=1 Ack=169 Win=64128 Len=0
2156	360.666324	172.16.17.128	172.16.17.131	SMB	135	Negotiate Protocol Response
2157	360.666582	172.16.17.131	172.16.17.128	SMB	298	Session Setup AndX Request, User: Komputer\Kamil; Tree Connect AndX, Path: \\172.16.17.128\IPC\$
2158	360.667203	172.16.17.128	172.16.17.131	TCP	60	445 → 49162 [ACK] Seq=82 Ack=404 Win=64128 Len=0
2159	360.666907	172.16.17.128	172.16.17.131	SMB	184	Session Setup AndX Response; Tree Connect AndX
2160	360.669165	172.16.17.131	172.16.17.128	SMB	168	Trans2 Request, GET_DFS_REFERRAL, File: \\172.16.17.128\SQCL
2161	360.669578	172.16.17.128	172.16.17.131	TCP	60	445 → 49162 [ACK] Seq=212 Ack=518 Win=64128 Len=0
2162	360.672025	172.16.17.128	172.16.17.131	SMB	93	Trans2 Response, GET_DFS_REFERRAL, Error: STATUS_NOT_FOUND
2163	360.674154	172.16.17.131	172.16.17.128	SMB	298	Session Setup AndX Request, User: Komputer\Kamil; Tree Connect AndX, Path: \\172.16.17.128\SQCL
2164	360.674525	172.16.17.128	172.16.17.131	TCP	60	445 → 49162 [ACK] Seq=251 Ack=762 Win=64128 Len=0
2165	360.676109	172.16.17.128	172.16.17.131	SMB	184	Session Setup AndX Response; Tree Connect AndX
2166	360.676345	172.16.17.131	172.16.17.128	SMB	134	Trans2 Request, QUERY_PATH_INFO, Query File Basic Info, Path:
2167	360.676637	172.16.17.128	172.16.17.131	TCP	60	445 → 49162 [ACK] Seq=381 Ack=842 Win=64128 Len=0
2168	360.679212	172.16.17.128	172.16.17.131	SMB	155	Trans2 Response, QUERY_PATH_INFO
2169	360.679265	172.16.17.131	172.16.17.128	SMB	134	Trans2 Request, QUERY_PATH_INFO, Query File Standard Info, Path:
2170	360.679573	172.16.17.128	172.16.17.131	TCP	60	445 → 49162 [ACK] Seq=482 Ack=922 Win=64128 Len=0
2171	360.682094	172.16.17.128	172.16.17.131	SMB	137	Trans2 Response, QUERY_PATH_INFO
2172	360.682520	172.16.17.131	172.16.17.128	SMB	150	NT Create AndX Request, Path: \svsvc

Port, który był wykorzystany do przesyłania danych to 445. Czyli ten odpowiedzialny za SMB. Stary dobry i znany atak.

Transmission Control Protocol, Src Port: 49162, Dst Port: 445, Seq: 160, Ack: 82, Len: 244
NetBIOS Session Service

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

Zainstalowałem program i wgrałem plik .pcap.



Takie pliki są widoczne w files.

Frame nr.	Filename	Extension	Size	Source host	S. port	Destination host	D. port	Protocol
2123	autoupdate.opera.com.cer	cer	1 380 B	82.145.216.19 [eu2-autoupdate.opera.com] [autoupdate.g...	TCP 443	172.16.17.131 [KOMPUTER] (Windows)	TCP 49161	TlsCertificate
2123	DigiCert TLS Hybrid ECC SHA3.cer	cer	1 095 B	82.145.216.19 [eu2-autoupdate.opera.com] [autoupdate.g...	TCP 443	172.16.17.131 [KOMPUTER] (Windows)	TCP 49161	TlsCertificate
2589	meterpreter.dll	dll	175 174 B	172.16.17.128 [KOMPUTER] (Windows)	TCP 4444	172.16.17.131 [KOMPUTER] (Windows)	TCP 49165	Meterpreter

Niestety nie ma tutaj interesującego nas pliku. Należy zagłębić się trochę dalej.

Hosts (9) Files (3) Images Messages Credentials (1) Sessions (2006) DNS (4) Parameters (287) Keywords Anomalies					
Filter keyword: exe					
<input type="checkbox"/> Case sensitive ExactPhrase Any column Clear Apply					
Parameter name	Parameter value	Frame number	Source host	Source port	Destination host
SMB NT Create AndX Request 1344	\\bad_file.exe	2241	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 1856	\\bad_file.exe	2260	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 2048	\\bad_file.exe	2269	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 2176	\\bad_file.exe	2273	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 2304	\\bad_file.exe	2277	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 2880	\\bad_file.exe	2298	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 3072	\\bad_file.exe.Manifest	2307	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 3200	\\bad_file.exe	2311	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 3264	\\bad_file.exe	2313	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 3904	\\bad_file.exe	2349	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 4672	\\bad_file.exe	2400	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 5504	\\bad_file.exe	2432	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 6016	\\bad_file.exe.Manifest	2451	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 6144	\\bad_file.exe	2455	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 6208	\\bad_file.exe	2457	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 6656	\\bad_file.exe	2484	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 6784	\\bad_file.exe	2488	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 7168	\\bad_file.exe	2500	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 7296	\\bad_file.exe	2504	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 7424	\\bad_file.exe	2508	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 7552	\\bad_file.exe	2512	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 8128	\\bad_file.exe	2533	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 8192	\\bad_file.exe.Manifest	2535	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 8320	\\bad_file.exe	2539	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)
SMB NT Create AndX Request 8384	\\bad_file.exe	2541	172.16.17.131 [KOMPUTER] (Windows)	TCP 49162	172.16.17.128 [172.16.17.128] (Other)

Po odfiltrowaniu wszystko dobrze widać. Znajdujemy sygnatury pliku bad_file.exe. Port docelowy również się zgadza – 445.