

# Warsztaty 7

---

Dawid Dieu  
302052

## Zadanie dopuszczające do dalszych części

---

Uruchom dwie maszyny wirtualne Virbian1 i Virbian2. Każda powinna mieć dwie karty sieciowe nazwane `enp-local` i `enp-remote`. Karty `enp-local` powinny być zmostkowane ze sobą, zaś za pomocą interfejsów `enp-remote` maszyny powinny być połączone (przez NAT) z Internetem.

- Aktywuj oba interfejsy obu maszyn. Interfejsom `enp-local` nadaj adresy IP odpowiednio `192.168.1.1/24` i `192.168.1.2/24`, zaś interfejsy `enp-remote` skonfiguruj za pomocą protokołu DHCP.
- Na obu maszynach uruchom Wiresharka nasłuchującego na wszystkich interfejsach.

- V1:

```
sudo ip link set enp0s3 name enp-local
sudo ip link set up dev enp-local
sudo ip addr add 192.168.1.1/24 dev enp-local
sudo ip link set enp0s8 name enp-remote
sudo dhclient
```

- V2:

```
sudo ip link set enp0s3 name enp-local
sudo ip link set up dev enp-local
```

```
sudo ip addr add 192.168.1.2/24 dev enp-local  
sudo ip link set enp0s8 name enp-remote  
sudo dhclient
```

## Zadanie do zaprezentowania (2 pkt.)

Prostym sposobem zaszyfrowania połączenia jest wykorzystanie tunelowania strumienia danych w danych protokołu SSH.

- Na maszynie *Virbian2* utwórz tunel SSH łączący port 7777 lokalnej maszyny (*Virbian2*) z portem 7 maszyny *Virbian1*. W tym celu wykonaj polecenie  
`V2{{{html}}}  
V2{{{html}}}`gt; ssh -N -L 7777:localhost:7 user@192.168.1.1  
i pozostaw je uruchomione. Sprawdź, że po wpisaniu na maszynie *Virbian2*ip polecenia  
`V2{{{html}}}  
V2{{{html}}}`gt; telnet localhost 7777  
odpowiada serwer echa maszyny *Virbian1*.

The screenshot shows two windows. The top window is Wireshark, titled 'Capturing from enp-local'. It displays a packet capture table with the following data:

| No. | Time        | Source      | Destination     | Protocol | Length | Info                   |
|-----|-------------|-------------|-----------------|----------|--------|------------------------|
| 1   | 0.000000000 | 0.0.0.0     | 255.255.255.255 | DHCP     | 342    | DHCP Discover - Transa |
| 2   | 4.824856533 | 192.168.1.2 | 192.168.1.1     | SSH      | 110    | Client: Encrypted pack |
| 3   | 4.825335639 | 192.168.1.1 | 192.168.1.2     | SSH      | 110    | Server: Encrypted pack |
| 4   | 4.825346180 | 192.168.1.2 | 192.168.1.1     | TCP      | 66     | 36170 → 22 [ACK] Seq=4 |
| 5   | 6.551249203 | 0.0.0.0     | 255.255.255.255 | DHCP     | 342    | DHCP Discover - Transa |

The bottom window is a terminal titled 'user@virbian: ~'. It shows the following commands and output:

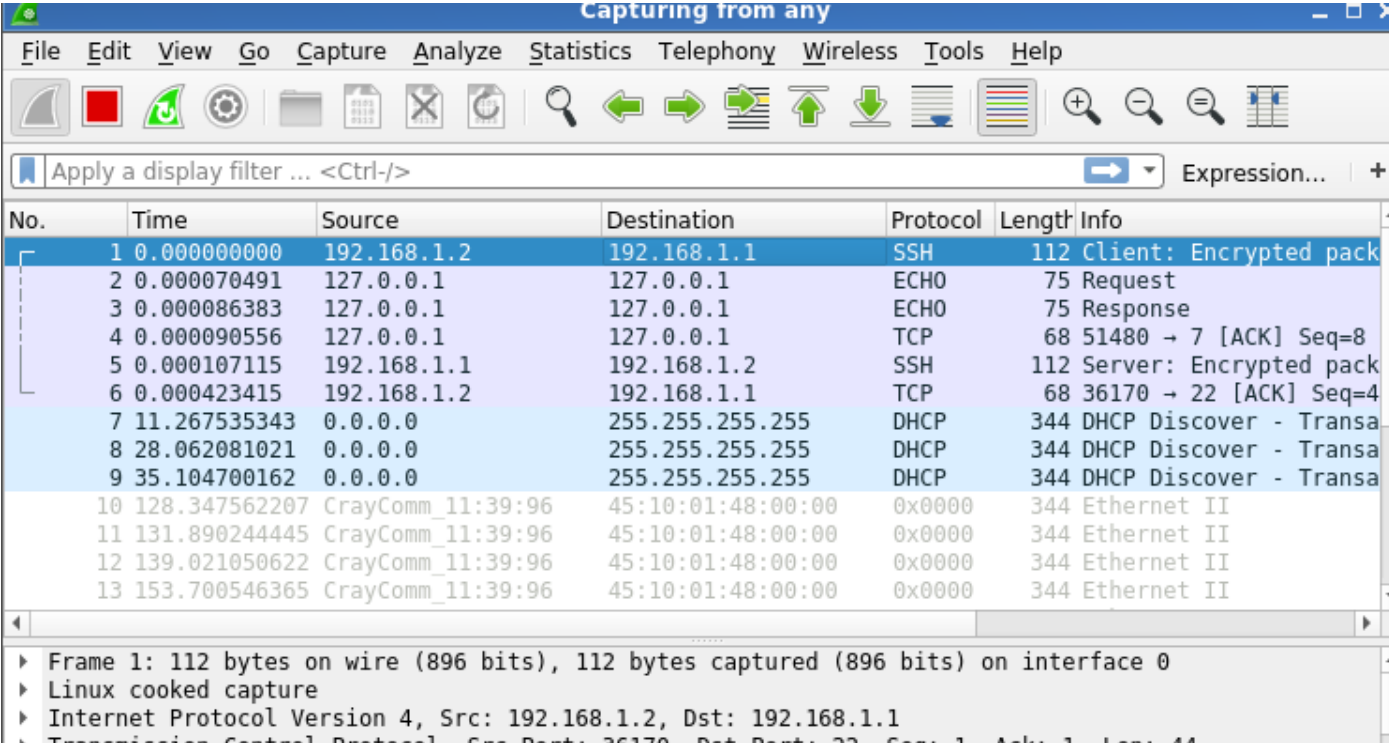
```
user@virbian:~$ telnet localhost 7777  
Trying ::1...  
Connected to localhost.  
Escape character is '^]'.  
echo  
echo  
echo  
echo
```

- Na podstawie Wiresharka odpowiedz na pytania (w każdym polu należy wpisać adres IP i port) dotyczące strumienia danych od maszyny Virbian2 do maszyny Virbian1.
  - Na maszynie Virbian2 strumień danych występuje w postaci niezaszyfrowanej jako pakiety przesyłane z :::1 port 42480 do :::1 port 7777 .
  - Pomiedzy maszyną Virbian2 a maszyną Virbian1 strumień danych występuje w postaci zaszyfrowanej jako pakiety przesyłane z 192.168.1.2 port 36170 do 192.168.1.1 port 22 .
  - Na maszynie Virbian1 strumień danych występuje w postaci niezaszyfrowanej jako pakiety przesyłane z 127.0.0.1 port 7 do 127.0.0.1 port 51480 .

Które z powyższych adresów IP są w wersji 4 a które w wersji 6?

  - :::1 - v6
  - reszta - v4

- V1



| No. | Time          | Source            | Destination       | Protocol | Length | Info                   |
|-----|---------------|-------------------|-------------------|----------|--------|------------------------|
| 1   | 0.000000000   | 192.168.1.2       | 192.168.1.1       | SSH      | 112    | Client: Encrypted pack |
| 2   | 0.000070491   | 127.0.0.1         | 127.0.0.1         | ECHO     | 75     | Request                |
| 3   | 0.000086383   | 127.0.0.1         | 127.0.0.1         | ECHO     | 75     | Response               |
| 4   | 0.000090556   | 127.0.0.1         | 127.0.0.1         | TCP      | 68     | 51480 → 7 [ACK] Seq=8  |
| 5   | 0.000107115   | 192.168.1.1       | 192.168.1.2       | SSH      | 112    | Server: Encrypted pack |
| 6   | 0.000423415   | 192.168.1.2       | 192.168.1.1       | TCP      | 68     | 36170 → 22 [ACK] Seq=4 |
| 7   | 11.267535343  | 0.0.0.0           | 255.255.255.255   | DHCP     | 344    | DHCP Discover - Transa |
| 8   | 28.062081021  | 0.0.0.0           | 255.255.255.255   | DHCP     | 344    | DHCP Discover - Transa |
| 9   | 35.104700162  | 0.0.0.0           | 255.255.255.255   | DHCP     | 344    | DHCP Discover - Transa |
| 10  | 128.347562207 | CrayComm_11:39:96 | 45:10:01:48:00:00 | 0x0000   | 344    | Ethernet II            |
| 11  | 131.890244445 | CrayComm_11:39:96 | 45:10:01:48:00:00 | 0x0000   | 344    | Ethernet II            |
| 12  | 139.021050622 | CrayComm_11:39:96 | 45:10:01:48:00:00 | 0x0000   | 344    | Ethernet II            |
| 13  | 153.700546365 | CrayComm_11:39:96 | 45:10:01:48:00:00 | 0x0000   | 344    | Ethernet II            |

Frame 1: 112 bytes on wire (896 bits), 112 bytes captured (896 bits) on interface 0  
 Linux cooked capture  
 Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.1  
 Transmission Control Protocol, Src Port: 36170, Dst Port: 22, Seq: 1, Ack: 1, Len: 44

- V2

| No. | Time         | Source            | Destination       | Protocol | Length | Info   |
|-----|--------------|-------------------|-------------------|----------|--------|--|
| 1   | 0.000000000  | :::1              | :::1              | TCP      | 95     | 42480 → 7777 [PSH, ACK] Seq=1 Ack=1 Win=512  |
| 2   | 0.000043551  | 192.168.1.2       | 192.168.1.1       | SSH      | 112    | Client: Encrypted packet (len=44)            |
| 3   | 0.000467737  | 192.168.1.1       | 192.168.1.2       | SSH      | 112    | Server: Encrypted packet (len=44)            |
| 4   | 0.000475402  | 192.168.1.2       | 192.168.1.1       | TCP      | 68     | 36170 → 22 [ACK] Seq=45 Ack=45 Win=501 Len=0 |
| 5   | 0.000512051  | :::1              | :::1              | TCP      | 95     | 7777 → 42480 [PSH, ACK] Seq=1 Ack=8 Win=512  |
| 6   | 0.000516779  | :::1              | :::1              | TCP      | 88     | 42480 → 7777 [ACK] Seq=8 Ack=8 Win=512 Len=0 |
| 7   | 11.267562506 | CrayComm 11:39:96 | 45:10:01:48:00:00 | 0x0000   | 344    | Ethernet II                                  |
| 8   | 25.428344259 | 10.0.3.15         | 172.16.0.1        | DNS      | 78     | Standard query 0x3b00 A www.facebook.com     |
| 9   | 25.428366620 | 10.0.3.15         | 172.16.0.1        | DNS      | 78     | Standard query 0xad06 AAAA www.facebook.com  |

## Zadanie do zaprezentowania (3 pkt.)

W tym zadaniu wygodnie jest myśleć, że maszyna Virbian1 należy do użytkownika *user1*, zaś maszyna Virbian2 do użytkownika *user2*.

- Zapisz klucz publiczny użytkownika *user1* z maszyny Virbian1 w czytelnej postaci do pliku

`user1-pgp-key` poleceniem

`V1{{{html}}}`gt; `gpg -a --export user1 > user1-pgp-key`

```
user@virbian:~$ gpg -a --export user1 > user1-pgp-key
user@virbian:~$ ls
user1-pgp-key
user@virbian:~$
```

- Na maszynie Virbian2 wygeneruj klucz prywatny i publiczny, jako użytkownika podając *user2* a jako adres email *user2@mail.example.com* (`mailto:user2@mail.example.com`). Wyeksportuj klucz publiczny do pliku `user2-pgp-key`.

```
user@virbian:~$ gpg -a --export user2 > user2-pgp-key
user@virbian:~$ ls
user2-pgp-key
user@virbian:~$
```

- Za pomocą SSH skopiuj plik `user1-pgp-key` na maszynę `Virbian2`, a następnie zaimportuj go do kluczy użytkownika `user2` za pomocą polecenia.

```
V2{{{html}}}gt; gpg --import < user1-pgp-key
```

Wejdź w tryb edycji tego klucza, upewnij się, że jego funkcja skrótu jest odpowiednia i podpisz go kluczem prywatnym użytkownika `user2`.

```
user@virbian:~$ scp 192.168.1.1:user1-pgp-key user1-pgp-key
user@192.168.1.1's password:
user1-pgp-key                               100% 2452      3.3MB/s   00:00
user@virbian:~$ ls
user1-pgp-key  user2-pgp-key
user@virbian:~$ 
user@virbian:~$ gpg --edit-key user1@mail.example.com
gpg (GnuPG) 2.2.12; Copyright (C) 2018 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid: 1  signed: 0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2022-06-08
sec  rsa3072/024438FE9BAC24F4
     created: 2020-06-08  expires: 2022-06-08  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa3072/23ACD2D1EC741FBD
     created: 2020-06-08  expires: 2022-06-08  usage: E
[ultimate] (1). user1 <user1@mail.example.com>

gpg> fpr
pub  rsa3072/024438FE9BAC24F4 2020-06-08 user1 <user1@mail.example.com>
   Primary key fingerprint: 6D2E 4ED9 6A94 958B 2ED9  0A31 0244 38FE 9BAC 24F4

gpg> sign
"user1 <user1@mail.example.com>" was already signed by key 024438FE9BAC24F4
Nothing to sign with key 024438FE9BAC24F4

gpg> quit
user@virbian:~$
```

- Wykonaj powyższy punkt, ale zamieniając role user1 i user2: w efekcie klucz użytkownika user2 powinien znaleźć się na maszynie Virbian1, zostać zaimportowany i podpisany kluczem użytkownika user1.

```
user@virbian:~$ scp 192.168.1.2:user2-pgp-key user2-pgp-key
The authenticity of host '192.168.1.2 (192.168.1.2)' can't be established.
ECDSA key fingerprint is SHA256:5BKg+T0tp10RbvQ0MKkMMf34MV5cNWJHAJ50kcSmVq8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.2' (ECDSA) to the list of known hosts.
user@192.168.1.2's password:
user2-pgp-key                               100% 2452      2.6MB/s   00:00
user@virbian:~$ ls
user1-pgp-key  user2-pgp-key
user@virbian:~$
```

- Na maszynie Virbian1 utwórz plik message umieść w nim jakąś treść. W celu podpisania wiadomości kluczem użytkownika user1 i zaszyfrowania jej kluczem publicznym użytkownika user2 wydaj polecenie

```
V1{{{html}}}gt; gpg -a -r user2 -se message
```

```
message user1-pgp-key user2-pgp-key
user@virbian:~$ gpg -a -r user2 -se message
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 1 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: depth: 1 valid: 1 signed: 0 trust: 1-, 0q, 0n, 0m, 0f, 0u
gpg: next trustdb check due at 2022-06-08
user@virbian:~$ ls
message message.asc user1-pgp-key user2-pgp-key
user@virbian:~$
```

Szyfrogram zostanie zapisany do pliku message.asc, który należy skopiować za pomocą SSH na komputer Virbian2.

- Na maszynie Virbian2 otrzymany plik message.asc należy odszyfrować kluczem prywatnym użytkownika user2 i zweryfikować prawdziwość podpisu poleceniem

```
V2{{{html}}}gt; gpg -d message.asc > deciphered message
```

```
gpg:
user@virbian:~$ scp 192.168.1.1:message.asc message.asc
user@192.168.1.1's password:
message.asc                               100% 1321      2.0MB/s   00:00
user@virbian:~$ ls
message.asc  user1-pgp-key  user2-pgp-key
user@virbian:~$ gpg -d message.asc > deciphered message
usage: gpg [options] --decrypt [filename]
user@virbian:~$ ls
deciphered message.asc  user1-pgp-key  user2-pgp-key
user@virbian:~$ gpg -d message.asc
gpg: encrypted with 3072-bit RSA key, ID 48ABFDD292B4720E, created 2020-06-08
      "user2 <user2@mail.example.com>"
123
gpg: Signature made Mon Jun  8 22:13:29 2020 CEST
gpg:          using RSA key 6D2E4ED96A94958B2ED90A31024438FE9BAC24F4
gpg: Good signature from "user1 <user1@mail.example.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 6D2E 4ED9 6A94 958B 2ED9  0A31 0244 38FE 9BAC 24F4
user@virbian:~$
```