

Security: IPsec

Instalacja oprogramowania

Zainstalować ipsec-tools oraz strongswan

Polecenia do instalacji ipsec:

```
wget http://launchpadlibrarian.net/234435245/ipsec-  
tools_0.8.2+20140711-5_amd64.deb  
sudo apt install ./ipsec-tools_0.8.2+20140711-5_amd64.deb
```

Polecenie do instalacji strongswan:

```
sudo apt install strongswan
```

Konfiguracja IPsec za pomocą setkey

Modyfikacja plików konfiguracyjnych

[setkey.conf_a_](#)

```
#!/usr/sbin/setkey -f  
  
## konfiguracja dla maszyny A  
  
## flush SAD and SPD  
flush;  
spdflush;  
  
## add SAs in SAD  
add 172.20.253.237 172.20.252.198 esp 0x1000 -E aes-cbc  
0xaa223344556677889900aabbccddeeff;  
add 172.20.252.198 172.20.253.237 esp 0x2000 -E aes-cbc  
0xbb223344556677889900aabbccddeeff;  
  
## add SPs in SPD  
spdadd 172.20.253.237 172.20.252.198 any -P out ipsec  
esp/transport//require;  
spdadd 172.20.252.198 172.20.253.237 any -P in ipsec  
esp/transport//require;
```

[setkey.conf_b_](#)

```
#!/usr/sbin/setkey -f

## konfiguracja dla maszyny B

## flush SAD i SPD
flush;
spdflush;

## add SAs in SAD
add 172.20.253.237 172.20.252.198 esp 0x1000 -E aes-cbc
0xaa223344556677889900aabbccddeeff;
add 172.20.252.198 172.20.253.237 esp 0x2000 -E aes-cbc
0xbb223344556677889900aabbccddeeff;

## add SPs in SPD
spdadd 172.20.253.237 172.20.252.198 any -P in ipsec
esp/transport//require;
spdadd 172.20.252.198 172.20.253.237 any -P out ipsec
esp/transport//require;
```

Załadowanie konfiguracji

Na maszynie a:

```
administrator@ipsec-A:~$ sudo setkey -f setkey.conf_a_
administrator@ipsec-A:~$ sudo setkey -D
172.20.252.198 172.20.253.237
  esp mode=transport spi=8192(0x00002000) reqid=0(0x00000000)
  E: aes-cbc bb223344 55667788 9900aabb ccddeeff
  seq=0x00000000 replay=0 flags=0x00000000 state=mature
  created: Jun 14 19:29:57 2025   current: Jun 14 19:30:01 2025
  diff: 4(s)   hard: 0(s)   soft: 0(s)
  last:
  current: 0(bytes)   hard: 0(bytes)   soft: 0(bytes)
  allocated: 0   hard: 0   soft: 0
  sadb_seq=1 pid=13464 refcnt=0
172.20.253.237 172.20.252.198
  esp mode=transport spi=4096(0x00001000) reqid=0(0x00000000)
  E: aes-cbc aa223344 55667788 9900aabb ccddeeff
  seq=0x00000000 replay=0 flags=0x00000000 state=mature
  created: Jun 14 19:29:57 2025   current: Jun 14 19:30:01 2025
  diff: 4(s)   hard: 0(s)   soft: 0(s)
  last:
  current: 0(bytes)   hard: 0(bytes)   soft: 0(bytes)
  allocated: 0   hard: 0   soft: 0
  sadb_seq=0 pid=13464 refcnt=0
administrator@ipsec-A:~$ sudo setkey -DP
172.20.252.198[any] 172.20.253.237[any] 255
```

```
fwd prio def ipsec
esp/transport//require
created: Jun 14 19:29:57 2025  lastused:
lifetime: 0(s) validtime: 0(s)
spid=82 seq=1 pid=13511
refcnt=1
172.20.252.198[any] 172.20.253.237[any] 255
in prio def ipsec
esp/transport//require
created: Jun 14 19:29:57 2025  lastused:
lifetime: 0(s) validtime: 0(s)
spid=72 seq=2 pid=13511
refcnt=1
172.20.253.237[any] 172.20.252.198[any] 255
out prio def ipsec
esp/transport//require
created: Jun 14 19:29:57 2025  lastused:
lifetime: 0(s) validtime: 0(s)
spid=65 seq=3 pid=13511
refcnt=1
(per-socket policy)
in(socket) none
created: Jun 14 19:18:30 2025  lastused:
lifetime: 0(s) validtime: 0(s)
spid=59 seq=4 pid=13511
refcnt=1
(per-socket policy)
out(socket) none
created: Jun 14 19:18:30 2025  lastused:
lifetime: 0(s) validtime: 0(s)
spid=52 seq=5 pid=13511
refcnt=1
(per-socket policy)
in(socket) none
created: Jun 14 19:18:30 2025  lastused:
lifetime: 0(s) validtime: 0(s)
spid=43 seq=6 pid=13511
refcnt=1
(per-socket policy)
out(socket) none
created: Jun 14 19:18:30 2025  lastused:
lifetime: 0(s) validtime: 0(s)
spid=36 seq=7 pid=13511
refcnt=1
(per-socket policy)
in(socket) none
created: Jun 14 19:18:30 2025  lastused:
lifetime: 0(s) validtime: 0(s)
spid=27 seq=8 pid=13511
refcnt=1
(per-socket policy)
```

```

    out(socket) none
    created: Jun 14 19:18:30 2025  lastused:
    lifetime: 0(s) validtime: 0(s)
    spid=20 seq=9 pid=13511
    refcnt=1
(per-socket policy)
    in(socket) none
    created: Jun 14 19:18:30 2025  lastused:
    lifetime: 0(s) validtime: 0(s)
    spid=11 seq=10 pid=13511
    refcnt=1
(per-socket policy)
    out(socket) none
    created: Jun 14 19:18:30 2025  lastused:
    lifetime: 0(s) validtime: 0(s)
    spid=4 seq=0 pid=13511
    refcnt=1
administrator@ipsec-A:~$

```

Na maszynie b:

```

administrator@ipsec-B:~$ sudo setkey -f setkey.conf_b_
administrator@ipsec-B:~$ sudo setkey -D
172.20.252.198 172.20.253.237
    esp mode=transport spi=8192(0x00002000) reqid=0(0x00000000)
    E: aes-cbc bb223344 55667788 9900aabb ccddeeff
    seq=0x00000000 replay=0 flags=0x00000000 state=mature
    created: Jun 14 19:31:00 2025  current: Jun 14 19:31:15 2025
    diff: 15(s)      hard: 0(s)      soft: 0(s)
    last:
    current: 0(bytes)      hard: 0(bytes) soft: 0(bytes)
    allocated: 0      hard: 0 soft: 0
    sadb_seq=1 pid=32009 refcnt=0
172.20.253.237 172.20.252.198
    esp mode=transport spi=4096(0x00001000) reqid=0(0x00000000)
    E: aes-cbc aa223344 55667788 9900aabb ccddeeff
    seq=0x00000000 replay=0 flags=0x00000000 state=mature
    created: Jun 14 19:31:00 2025  current: Jun 14 19:31:15 2025
    diff: 15(s)      hard: 0(s)      soft: 0(s)
    last:
    current: 0(bytes)      hard: 0(bytes) soft: 0(bytes)
    allocated: 0      hard: 0 soft: 0
    sadb_seq=0 pid=32009 refcnt=0
administrator@ipsec-B:~$ sudo setkey -DP
172.20.252.198[any] 172.20.253.237[any] 255
    out prio def ipsec
    esp/transport//require
    created: Jun 14 19:31:00 2025  lastused:
    lifetime: 0(s) validtime: 0(s)
    spid=17 seq=1 pid=32078
    refcnt=1

```

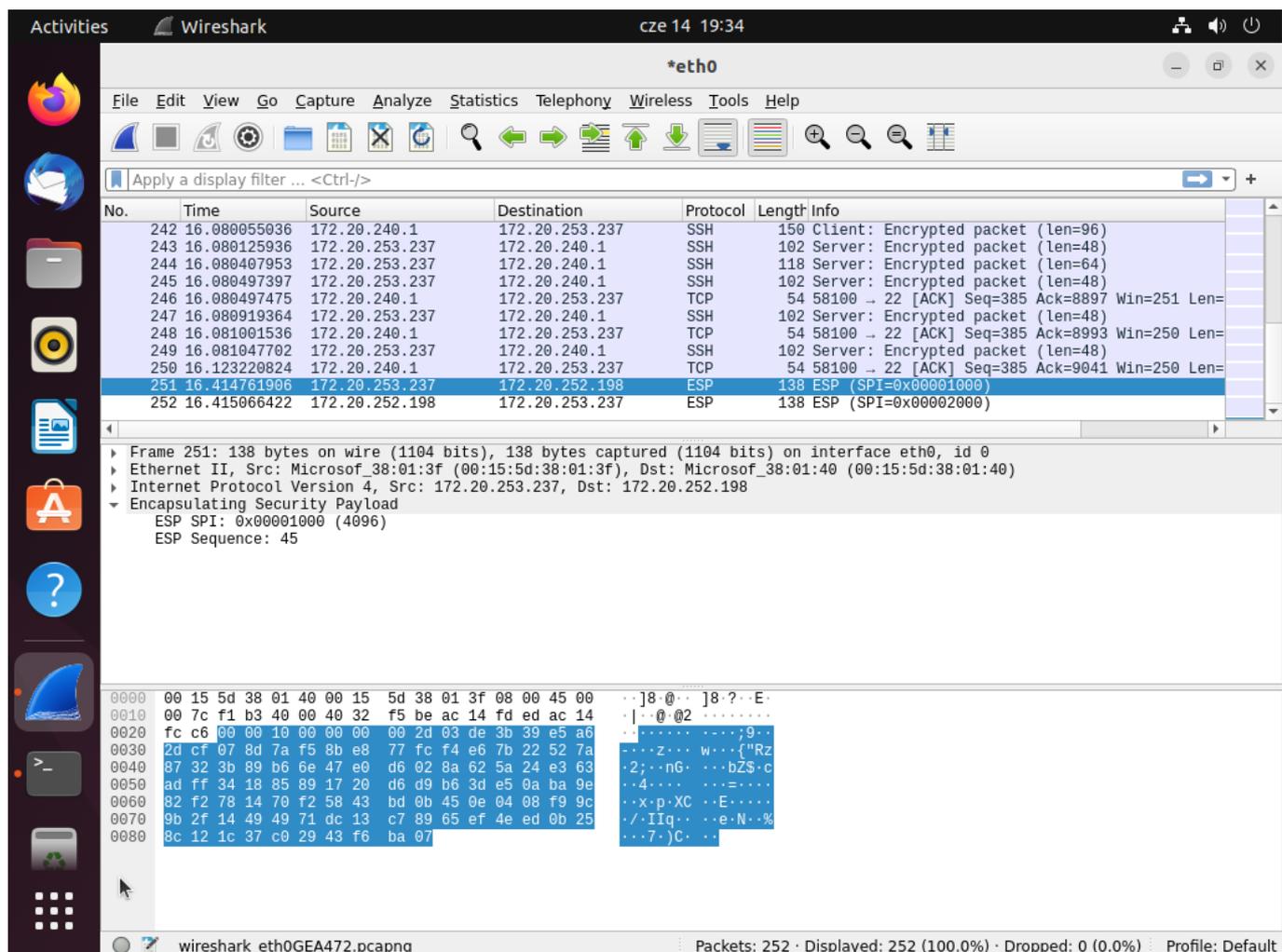
```

172.20.253.237[any] 172.20.252.198[any] 255
  fwd prio def ipsec
  esp/transport//require
  created: Jun 14 19:31:00 2025  lastused:
  lifetime: 0(s) validtime: 0(s)
  spid=10 seq=2 pid=32078
  refcnt=1
172.20.253.237[any] 172.20.252.198[any] 255
  in prio def ipsec
  esp/transport//require
  created: Jun 14 19:31:00 2025  lastused:
  lifetime: 0(s) validtime: 0(s)
  spid=8 seq=0 pid=32078
  refcnt=1
administrator@ipsec-B:~$

```

Test Działania tunelu

Wykonanie ping z maszyny A do B i obserwacja nagłówka ESP



Pytania

Jaki ruch zabezpieczony jest tunelem IPsec?

Ruch IP pomiędzy maszynami A (172.20.253.237) i B (172.20.252.198) – każdy protokół (any) zdefiniowany w SPD.

Jaki protokół IPsec jest używany? Jakie algorytmy?

- Protokół: ESP
- Algorytm szyfrowania: AES-CBC
- Usługi: Szyfrowanie (brak autentykacji w tym przykładzie)

Rola SAD i SPD:

- SAD: Przechowuje parametry SA (klucze, algorytmy, SPI)
- SPD: Przechowuje polityki bezpieczeństwa (który ruch ma być zabezpieczony i jak)

Które komendy są przetwarzane dla ruchu wychodzącego/ przychodzącego?

- Dla pakietów wychodzących – `spdadd ... -P out`
- Dla pakietów przychodzących – `spdadd ... -P in`

Dlaczego potrzebne są dwie SA?

SA są jednokierunkowe – jedna dla $A \rightarrow B$, druga dla $B \rightarrow A$, aby zabezpieczyć dwukierunkowy kanał.

Znaczenie komend "setkey -D" i "setkey -DP"

- -D – pokazuje aktywne asocjacje bezpieczeństwa (SA)
- -DP – pokazuje aktywne polityki bezpieczeństwa (SPD)

Jakie jest przeznaczenie pola SPI w nagłówku IPsec?

SPI (Security Parameter Index) to unikalny identyfikator asocjacji bezpieczeństwa (SA), który pozwala odbiorcy zidentyfikować, jakich parametrów użyć do odszyfrowania i uwierzytelnienia pakietu.

Jakie jest przeznaczenie pola Sequence Number w nagłówku IPsec?

Sequence Number służy do zapobiegania atakom typu replay. Numer rośnie monotonicznie dla każdego pakietu, co pozwala sprawdzić, czy pakiety nie zostały powtórzone.

Modyfikacja polityk IPsec i SA

ESP z AES-CBC + HMAC-SHA1

[setkey.conf_a_](#)

```
#!/usr/sbin/setkey -f

## maszyna A (172.20.253.237)
flush;
spdflush;

## SA: A->B
add 172.20.253.237 172.20.252.198 esp 0x1000 -E aes-cbc
0xaa223344556677889900aabbccddeeff -A hmac-sha1
0x00112233445566778899aabbccddeeff00112233;
## SA: B->A
add 172.20.252.198 172.20.253.237 esp 0x2000 -E aes-cbc
0xbb223344556677889900aabbccddeeff -A hmac-sha1
0x11223344556677889900aabbccddeeff00112233;

## SPD: zabezpiecz wszystkie pakiety ESP w trybie transportowym
spdadd 172.20.253.237 172.20.252.198 any -P out ipsec
esp/transport//require;
spdadd 172.20.252.198 172.20.253.237 any -P in ipsec
esp/transport//require;
```

[setkey.conf_b_](#)

```
#!/usr/sbin/setkey -f

## maszyna B (172.20.252.198)
flush;
spdflush;

## SA: B->A
add 172.20.252.198 172.20.253.237 esp 0x2000 -E aes-cbc
0xbb223344556677889900aabbccddeeff -A hmac-sha1
0x00112233445566778899aabbccddeeff00112233;
## SA: A->B
add 172.20.253.237 172.20.252.198 esp 0x1000 -E aes-cbc
0xaa223344556677889900aabbccddeeff -A hmac-sha1
0x00112233445566778899aabbccddeeff00112233;

## SPD: zabezpiecz wszystkie pakiety ESP w trybie transportowym
spdadd 172.20.252.198 172.20.253.237 any -P out ipsec
esp/transport//require;
spdadd 172.20.253.237 172.20.252.198 any -P in ipsec
```

esp/transport//require;

The screenshot shows the Wireshark interface with a packet capture on the eth0 interface. The packet list pane shows several packets, with packet 236 selected. The packet details pane shows the structure of an Encapsulating Security Payload (ESP) packet. The raw packet bytes pane shows the hexadecimal and ASCII representation of the packet data.

No.	Time	Source	Destination	Protocol	Length	Info
226	14.069346321	172.20.253.237	172.20.240.1	SSH	134	Server: Encrypted packet (len=80)
227	14.069455062	172.20.253.237	172.20.240.1	SSH	102	Server: Encrypted packet (len=48)
228	14.069548092	172.20.240.1	172.20.253.237	TCP	54	58100 → 22 [ACK] Seq=289 Ack=7585 Win=253 Len=
229	14.070255704	172.20.253.237	172.20.240.1	SSH	118	Server: Encrypted packet (len=64)
230	14.070378813	172.20.253.237	172.20.240.1	SSH	102	Server: Encrypted packet (len=48)
231	14.070465348	172.20.240.1	172.20.253.237	TCP	54	58100 → 22 [ACK] Seq=289 Ack=7697 Win=253 Len=
232	14.070897085	172.20.253.237	172.20.240.1	SSH	102	Server: Encrypted packet (len=48)
233	14.070995807	172.20.253.237	172.20.240.1	SSH	102	Server: Encrypted packet (len=48)
234	14.071090232	172.20.240.1	172.20.253.237	TCP	54	58100 → 22 [ACK] Seq=289 Ack=7793 Win=252 Len=
235	14.375167485	172.20.253.237	172.20.252.198	ESP	150	ESP (SPI=0x00001000)
236	14.375468057	172.20.252.198	172.20.253.237	ESP	150	ESP (SPI=0x00002000)

Frame 134: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface eth0, id 0
 Ethernet II, Src: Microsof_38:01:40 (00:15:5d:38:01:40), Dst: Microsof_38:01:3f (00:15:5d:38:01:3f)
 Internet Protocol Version 4, Src: 172.20.252.198, Dst: 172.20.253.237
 Encapsulating Security Payload
 ESP SPI: 0x00002000 (8192)
 ESP Sequence: 37

AH z HMAC-SHA1

[setkey.conf_a](#)

```
#!/usr/sbin/setkey -f

## maszyna A (172.20.253.237)
flush;
spdf flush;

## SA: A->B
add 172.20.253.237 172.20.252.198 ah 0x1000 -A hmac-sha1
0x00112233445566778899aabbccddeeff00112233;
## SA: B->A
add 172.20.252.198 172.20.253.237 ah 0x2000 -A hmac-sha1
0x11223344556677889900aabbccddeeff00112233;

## SPD: zabezpiecz wszystkie pakiety AH w trybie transportowym
spdadd 172.20.253.237 172.20.252.198 any -P out ipsec
```

```
ah/transport//require;  
spdadd 172.20.252.198 172.20.253.237 any -P in ipsec  
ah/transport//require;
```

[setkey.conf_b_](#)

```
#!/usr/sbin/setkey -f  
  
## maszyna B (172.20.252.198)  
flush;  
spdflush;  
  
## SA: B->A  
add 172.20.252.198 172.20.253.237 ah 0x2000 -A hmac-sha1  
0x11223344556677889900aabbccddeeff00112233;  
## SA: A->B  
add 172.20.253.237 172.20.252.198 ah 0x1000 -A hmac-sha1  
0x00112233445566778899aabbccddeeff00112233;  
  
## SPD: zabezpiecz wszystkie pakiety AH w trybie transportowym  
spdadd 172.20.252.198 172.20.253.237 any -P out ipsec  
ah/transport//require;  
spdadd 172.20.253.237 172.20.252.198 any -P in ipsec  
ah/transport//require;
```

The screenshot shows a Wireshark capture on the eth0 interface. The packet list pane displays several ICMP Echo (ping) requests and replies. The selected packet (No. 1084) is an Echo (ping) reply from 172.20.253.237 to 172.20.252.198. The packet details pane shows the following structure:

- Frame 1084: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface eth0, id 0
- Ethernet II, Src: Microsof_38:01:40 (00:15:5d:38:01:40), Dst: Microsof_38:01:3f (00:15:5d:38:01:3f)
- Internet Protocol Version 4, Src: 172.20.252.198, Dst: 172.20.253.237
- Authentication Header
 - Next header: ICMP (1)
 - Length: 4 (24 bytes)
 - Reserved: 0000
 - AH SPI: 0x00002000
 - AH Sequence: 81
 - AH ICV: 919b3cc7a651f7ddeeada695
- Internet Control Message Protocol

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```

0000 00 15 5d 38 01 3f 00 15 5d 38 01 40 08 00 45 00  ..]8?..]8.0..E.
0010 00 6c e4 c7 00 00 40 33 42 ba ac 14 fc c6 ac 14  .l...@3 B.....
0020 fd ed 01 04 00 00 00 00 20 00 00 00 00 51 91 9b  .....Q.....
0030 3c c7 a6 51 f7 dd ee ad a6 95 00 00 ff 27 00 08  <..Q.....'
0040 00 51 5c bd 4d 68 00 00 00 00 93 86 04 00 00 00  .Qv.Mh.....
0050 00 00 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d  .....
0060 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d  ..!"#$%&'()*+,-
0070 2e 2f 30 31 32 33 34 35 36 37  .-/012345 67
  
```

ESP (AES-CBC) + AH (HMAC-SHA1)

setkey.conf_a_

```

#!/usr/sbin/setkey -f

## maszyna A (172.20.253.237)
flush;
spdf flush;

## SA ESP: A->B
add 172.20.253.237 172.20.252.198 esp 0x3000 -E aes-cbc
0xaa223344556677889900aabbccddeeff;
## SA ESP: B->A
add 172.20.252.198 172.20.253.237 esp 0x4000 -E aes-cbc
0xbb223344556677889900aabbccddeeff;

## SA AH: A->B
add 172.20.253.237 172.20.252.198 ah 0x5000 -A hmac-sha1
0x00112233445566778899aabbccddeeff00112233;
## SA AH: B->A
add 172.20.252.198 172.20.253.237 ah 0x6000 -A hmac-sha1
  
```

```
0x00112233445566778899aabbccddeeff00112233;  
  
## SPD: zabezpiecz pakiety ESP i AH w trybie transportowym  
spdadd 172.20.253.237 172.20.252.198 any -P out ipsec  
esp/transport//require ah/transport//require;  
spdadd 172.20.252.198 172.20.253.237 any -P in ipsec  
esp/transport//require ah/transport//require;
```

setkey.conf_b_

```
#!/usr/sbin/setkey -f  
  
## maszyna B (172.20.252.198)  
flush;  
spdflush;  
  
## SA ESP: B->A  
add 172.20.252.198 172.20.253.237 esp 0x4000 -E aes-cbc  
0xbb223344556677889900aabbccddeeff;  
## SA ESP: A->B  
add 172.20.253.237 172.20.252.198 esp 0x3000 -E aes-cbc  
0xaa223344556677889900aabbccddeeff;  
  
## SA AH: B->A  
add 172.20.252.198 172.20.253.237 ah 0x6000 -A hmac-sha1  
0x00112233445566778899aabbccddeeff00112233;  
## SA AH: A->B  
add 172.20.253.237 172.20.252.198 ah 0x5000 -A hmac-sha1  
0x00112233445566778899aabbccddeeff00112233;  
  
## SPD: zabezpiecz pakiety ESP i AH w trybie transportowym  
spdadd 172.20.252.198 172.20.253.237 any -P out ipsec  
esp/transport//require ah/transport//require;  
spdadd 172.20.253.237 172.20.252.198 any -P in ipsec  
esp/transport//require ah/transport//require;
```

The screenshot shows the Wireshark interface with the following details for the selected packet (No. 31):

- Frame 31:** 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface eth0, id 0
- Ethernet II:** Src: Microsoft_38:01:3f (00:15:5d:38:01:3f), Dst: Microsoft_38:01:40 (00:15:5d:38:01:40)
- Internet Protocol Version 4:** Src: 172.20.253.237, Dst: 172.20.252.198
- Authentication Header:**
 - Next header: Encap Security Payload (50)
 - Length: 4 (24 bytes)
 - Reserved: 0000
 - AH SPI: 0x00005000
 - AH Sequence: 27
 - AH ICV: e2193db7d7ca5e790f22afc0
- Encapsulating Security Payload:**
 - ESP SPI: 0x00003000 (12288)
 - ESP Sequence: 27

The packet bytes pane shows the following hex and ASCII data:

```

0000 00 15 5d 38 01 40 00 15 5d 38 01 3f 08 00 45 00  ..]8-@...8-?-E-
0010 00 94 dd bb 40 00 40 33 09 9e ac 14 fd ed ac 14  ...@_@3 .....
0020 fc c6 32 04 00 00 00 50 00 00 00 00 1b e2 19  =2.....P.....
0030 3d b7 d7 ca 5e 79 0f 22 af c0 00 00 30 00 00 00  =...Ay."...0...
0040 00 1b 4b 27 ac 9d 2b 28 39 c9 a7 55 95 7d 23 36  =.K'++(9..U.)#6
0050 17 d3 4d ee e8 6c 37 e7 d6 f8 19 7c bb e9 ae d6  ..M..17...|....
  
```

Pytania

Różnica pomiędzy konfiguracjami ESP a ESP+AH

- ESP chroni dane (szyfruje), ale nie chroni całego nagłówka IP.
- AH nie szyfruje, ale chroni integralność całego pakietu IP (z nagłówkiem).
- ESP+AH zapewnia poufność i integralność nagłówka i danych.

Zastosowanie AH

Przy połączeniach, gdzie ważna jest integralność całego nagłówka IP (np. dla systemów audytu lub gdzie niemożliwe jest użycie szyfrowania).

Protokół IKE i strongSwan (PSK)

Czyszczenie SAD i SPD

Na maszynie A i B:

```
sudo setkey -F      # usuń wszystkie SA
sudo setkey -FP     # usuń wszystkie SP
```

Konfiguracje

Maszyna A

/etc/ipsec.conf:

```
config setup
    charondebug = "ike 1, knl 1, cfg 1"

conn host-host
    keyexchange=ikev2
    authby=psk
    left=172.20.253.237      # lokalny endpoint
    leftsubnet=172.20.253.237/32
    right=172.20.252.198    # zdalny endpoint
    rightsubnet=172.20.252.198/32
    ike=aes256-sha1-modp1024
    esp=aes256-sha1
    auto=add
```

/etc/ipsec.secrets:

```
172.20.253.237 172.20.252.198 : PSK "SuperTajnyPSK"
```

Maszyna B

/etc/ipsec.conf:

```
config setup
    charondebug = "ike 1, knl 1, cfg 1"

conn host-host
    keyexchange=ikev2
    authby=psk
    left=172.20.252.198
    leftsubnet=172.20.252.198/32
    right=172.20.253.237
    rightsubnet=172.20.253.237/32
    ike=aes256-sha1-modp1024
    esp=aes256-sha1
    auto=add
```

/etc/ipsec.secrets:

```
172.20.252.198 172.20.253.237 : PSK "SuperTajnyPSK"
```

Uruchomienie strongSwan i nawiązanie tunelu

```
administrator@ipsec-A:~$ sudo ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.9.5 IPsec [starter]...
administrator@ipsec-A:~$ sudo ipsec up host-host
initiating IKE_SA host-host[1] to 172.20.252.198
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP)
N(FRAG_SUP) N(HASH_ALG) N(REDIR_SUP) ]
sending packet: from 172.20.253.237[500] to 172.20.252.198[500] (1044 bytes)
received packet: from 172.20.252.198[500] to 172.20.253.237[500] (344 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP)
N(FRAG_SUP) N(HASH_ALG) N(CHDLESS_SUP) N(MULT_AUTH) ]
selected proposal: IKE:AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
authentication of '172.20.253.237' (myself) with pre-shared key
establishing CHILD_SA host-host{1}
generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) IDr AUTH SA TSi TSr
N(MOBIKE_SUP) N(NO_ADD_ADDR) N(MULT_AUTH) N(EAP_ONLY) N(MSG_ID_SYN_SUP) ]
sending packet: from 172.20.253.237[4500] to 172.20.252.198[4500] (412
bytes)
received packet: from 172.20.252.198[4500] to 172.20.253.237[4500] (220
bytes)
parsed IKE_AUTH response 1 [ IDr AUTH SA TSi TSr N(MOBIKE_SUP)
N(NO_ADD_ADDR) ]
authentication of '172.20.252.198' with pre-shared key successful
IKE_SA host-host[1] established between
172.20.253.237[172.20.253.237]...172.20.252.198[172.20.252.198]
scheduling reauthentication in 9788s
maximum IKE_SA lifetime 10328s
selected proposal: ESP:AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ
CHILD_SA host-host{1} established with SPIs c4b20310_i c04d47dc_o and TS
172.20.253.237/32 === 172.20.252.198/32
peer supports MOBIKE
connection 'host-host' established successfully
administrator@ipsec-A:~$ sudo setkey -D
172.20.253.237 172.20.252.198
    esp mode=tunnel spi=3226290140(0xc04d47dc) reqid=1(0x00000001)
    E: aes-cbc 7ed41201 e80a0bf4 fb871600 4cfcbfab 1e6e4fb3 c346376a
429d1575 1d58fc17
    A: hmac-sha1 7a7a8e7d 7beaad8e 5f045ba1 45b42d7f bcc4de08
    seq=0x00000000 replay=0 flags=0x00000000 state=mature
    created: Jun 14 20:37:14 2025    current: Jun 14 20:37:52 2025
    diff: 38(s)    hard: 3600(s)    soft: 2765(s)
    last: Jun 14 20:37:14 2025    hard: 0(s)    soft: 0(s)
    current: 3192(bytes)    hard: 0(bytes)    soft: 0(bytes)
    allocated: 38    hard: 0    soft: 0
    sadb_seq=1 pid=28477 refcnt=0
```

```

172.20.252.198 172.20.253.237
  esp mode=tunnel spi=3300000528(0xc4b20310) reqid=1(0x00000001)
  E: aes-cbc b834b5fd 2412b9cf 9e4ac726 29160d9e afbb91b0 7b2aafd5
07fc4c67 052b7958
  A: hmac-sha1 31564ba0 bfdb79ff 872d5e0b fc3079d4 ea43a976
  seq=0x00000000 replay=32 flags=0x00000000 state=mature
  created: Jun 14 20:37:14 2025    current: Jun 14 20:37:52 2025
  diff: 38(s)    hard: 3600(s)    soft: 2819(s)
  last: Jun 14 20:37:14 2025    hard: 0(s)    soft: 0(s)
  current: 3192(bytes)    hard: 0(bytes)    soft: 0(bytes)
  allocated: 38    hard: 0    soft: 0
  sadb_seq=0 pid=28477 refcnt=0

```

```

administrator@ipsec-A:~$ sudo setkey -DP

```

```

172.20.253.237[any] 172.20.252.198[any] 255
  out prio high + 1073374593 ipsec
  esp/tunnel/172.20.253.237-172.20.252.198/unique:1
  created: Jun 14 20:37:14 2025    lastused: Jun 14 20:37:56 2025
  lifetime: 0(s) validtime: 0(s)
  spid=265 seq=1 pid=28510
  refcnt=1

```

```

172.20.252.198[any] 172.20.253.237[any] 255
  fwd prio high + 1073374593 ipsec
  esp/tunnel/172.20.252.198-172.20.253.237/unique:1
  created: Jun 14 20:37:14 2025    lastused:
  lifetime: 0(s) validtime: 0(s)
  spid=258 seq=2 pid=28510
  refcnt=1

```

```

172.20.252.198[any] 172.20.253.237[any] 255
  in prio high + 1073374593 ipsec
  esp/tunnel/172.20.252.198-172.20.253.237/unique:1
  created: Jun 14 20:37:14 2025    lastused: Jun 14 20:37:56 2025
  lifetime: 0(s) validtime: 0(s)
  spid=248 seq=3 pid=28510
  refcnt=1

```

```

(per-socket policy)
  in(socket) none
  created: Jun 14 20:34:15 2025    lastused: Jun 14 20:37:14 2025
  lifetime: 0(s) validtime: 0(s)
  spid=243 seq=4 pid=28510
  refcnt=1

```

```

(per-socket policy)
  out(socket) none
  created: Jun 14 20:34:15 2025    lastused: Jun 14 20:37:14 2025
  lifetime: 0(s) validtime: 0(s)
  spid=236 seq=5 pid=28510
  refcnt=1

```

```

(per-socket policy)
  in(socket) none
  created: Jun 14 20:34:15 2025    lastused: Jun 14 20:37:14 2025
  lifetime: 0(s) validtime: 0(s)
  spid=227 seq=6 pid=28510

```

```
    refcnt=1
(per-socket policy)
  out(socket) none
  created: Jun 14 20:34:15 2025  lastused: Jun 14 20:37:14 2025
  lifetime: 0(s) validtime: 0(s)
  spid=220 seq=7 pid=28510
  refcnt=1
(per-socket policy)
  in(socket) none
  created: Jun 14 20:34:15 2025  lastused:
  lifetime: 0(s) validtime: 0(s)
  spid=211 seq=8 pid=28510
  refcnt=1
(per-socket policy)
  out(socket) none
  created: Jun 14 20:34:15 2025  lastused:
  lifetime: 0(s) validtime: 0(s)
  spid=204 seq=9 pid=28510
  refcnt=1
(per-socket policy)
  in(socket) none
  created: Jun 14 20:34:15 2025  lastused:
  lifetime: 0(s) validtime: 0(s)
  spid=195 seq=10 pid=28510
  refcnt=1
(per-socket policy)
  out(socket) none
  created: Jun 14 20:34:15 2025  lastused:
  lifetime: 0(s) validtime: 0(s)
  spid=188 seq=0 pid=28510
  refcnt=1
administrator@ipsec-A:~$
```

Pytania

Ile asocjacji SA zostało wynegocjowanych?

W wyniku setkey -D widać dwie asocjacje ESP w bazie SAD (SPI 0xc04d47dc i 0xc4b20310) – po jednej dla każdego kierunku ruchu.

Jaki rodzaj nagłówka IPsec i jaki tryb pracy został użyty dla stworzonych asocjacji SA? Jakie algorytmy zostały użyte?

Zgodnie z logiem ipsec up host-host, wynegocjowano ESP w trybie tunelowym (ESP:AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ).

Algorytmy: AES-CBC 256-bit dla szyfrowania oraz HMAC-SHA1-96 (96-bitowy tag) dla uwierzytelnienia.

Jaka jest długość użytego klucza dla poszczególnych algorytmów?

AES-CBC-256 używa klucza o długości 256 bitów.

HMAC-SHA1-96 wykorzystuje tag o długości 96 bitów (12 bajtów), choć sam klucz HMAC może być dłuższy (typowo 160 bitów), w logu widzimy 20 bajtów klucza, ale tag jest obcięty do 12 bajtów.

W której fazie protokołu IKE użyty został współdzielony klucz PSK?

Klucz PSK został użyty w fazie 1 (IKE_SA_INIT oraz IKE_AUTH) do uwierzytelnienia endpointów.

Jakie polityki SPD zostały stworzone przez proces IKE?

setkey -DP pokazuje cztery wpisy:

dwa wpisy out i in dla tunelu ESP w trybie tunelowym (po jednej każda dla ruchu A→B i B→A) z reqid = 1 oraz flagą unique.

dotychczasowy wpis fwd związany z przekazywaniem pakietów.

Polityki dotyczą ruchu 172.20.253.237[any] ↔ 172.20.252.198[any] i mają priorytet „high”, operację ipsec esp/tunnel/.../unique:1.

Analiza ruchu IKE w Wireshark

Filtr: udp.port == 500 or udp.port == 4500.

W pakietach IKE_SA_INIT widać wymianę ofert SA (AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024) oraz kluczy DH.

The image shows a Wireshark network traffic analysis window titled '*eth0'. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar with various icons for packet capture and analysis. A filter is applied: 'udp.port == 500 or udp.port == 4500'. The packet list pane shows several ISAKMP packets, with packet 2056 selected. The packet details pane shows the structure of the IKE SA INIT message, including fields like Initiator SPI, Responder SPI, Exchange type (IKE_SA_INIT), and several Notify payloads. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1236	77.119649765	172.20.253.237	172.20.252.198	ISAKMP	394	CREATE_CHILD_SA MID=02 Initiator Request
1239	77.120492164	172.20.252.198	172.20.253.237	ISAKMP	250	CREATE_CHILD_SA MID=02 Responder Response
1337	80.785390066	172.20.253.237	172.20.252.198	ISAKMP	122	INFORMATIONAL MID=03 Initiator Request
1338	80.786029732	172.20.252.198	172.20.253.237	ISAKMP	122	INFORMATIONAL MID=03 Responder Response
2056	126.470803934	172.20.253.237	172.20.252.198	ISAKMP	1086	IKE_SA_INIT MID=00 Initiator Request
2059	126.472340783	172.20.252.198	172.20.253.237	ISAKMP	386	IKE_SA_INIT MID=00 Responder Response
2069	126.473956500	172.20.253.237	172.20.252.198	ISAKMP	458	IKE_AUTH MID=01 Initiator Request
2071	126.474719901	172.20.252.198	172.20.253.237	ISAKMP	266	IKE_AUTH MID=01 Responder Response

Destination Port: 500
Length: 1052
Checksum: 0x570b [unverified]
[Checksum Status: Unverified]
[Stream index: 4]
[Timestamps]
UDP payload (1044 bytes)
Internet Security Association and Key Management Protocol
Initiator SPI: 63a14f6441a91db6
Responder SPI: 0000000000000000
Next payload: Security Association (33)
Version: 2.0
Exchange type: IKE_SA_INIT (34)
Flags: 0x08 (Initiator, No higher version, Request)
Message ID: 0x00000000
Length: 1044
Payload: Security Association (33)
Payload: Key Exchange (34)
Payload: Nonce (40)
Payload: Notify (41) - NAT_DETECTION_SOURCE_IP
Payload: Notify (41) - NAT_DETECTION_DESTINATION_IP
Payload: Notify (41) - IKEV2_FRAGMENTATION_SUPPORTED
Payload: Notify (41) - SIGNATURE_HASH_ALGORITHMS
Payload: Notify (41) - REDIRECT_SUPPORTED

```
0000 00 15 5d 38 01 40 00 15 5d 38 01 3f 08 00 45 00  ..]8-@..]8-?-E-
0010 04 30 20 d3 40 00 40 11 c3 0c ac 14 fd ed ac 14  .0-@-@-.....
0020 fc c6 01 f4 01 f4 04 1c 57 0b 63 a1 4f 64 41 a9  ....W-cOdA-
0030 1d b6 00 00 00 00 00 00 00 00 21 20 22 08 00 00  ..!"....
0040 00 00 00 00 04 14 22 00 02 f4 02 00 00 2c 01 01  ....",...
0050 00 04 03 00 00 0c 01 00 00 0c 80 0e 01 00 03 00  ....
```

W IKE_AUTH pojawiają się IDi, IDr oraz AUTH payload z uwierzytelnieniem za pomocą PSK.

Activities Wireshark cze 14 20:44

*eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp.port == 500 or udp.port == 4500

No.	Time	Source	Destination	Protocol	Length	Info
1236	77.119649765	172.20.253.237	172.20.252.198	ISAKMP	394	CREATE_CHILD_SA MID=02 Initiator Request
1239	77.120492164	172.20.252.198	172.20.253.237	ISAKMP	250	CREATE_CHILD_SA MID=02 Responder Response
1337	80.785390066	172.20.253.237	172.20.252.198	ISAKMP	122	INFORMATIONAL MID=03 Initiator Request
1338	80.786029732	172.20.252.198	172.20.253.237	ISAKMP	122	INFORMATIONAL MID=03 Responder Response
2056	126.470803934	172.20.253.237	172.20.252.198	ISAKMP	1086	IKE_SA_INIT MID=00 Initiator Request
2059	126.472340783	172.20.252.198	172.20.253.237	ISAKMP	386	IKE_SA_INIT MID=00 Responder Response
2069	126.473956500	172.20.253.237	172.20.252.198	ISAKMP	458	IKE_AUTH MID=01 Initiator Request
2071	126.474719901	172.20.252.198	172.20.253.237	ISAKMP	266	IKE_AUTH MID=01 Responder Response

Frame 2069: 458 bytes on wire (3664 bits), 458 bytes captured (3664 bits) on interface eth0, id 0
 Ethernet II, Src: Microsof_38:01:3f (00:15:5d:38:01:3f), Dst: Microsof_38:01:40 (00:15:5d:38:01:40)
 Internet Protocol Version 4, Src: 172.20.253.237, Dst: 172.20.252.198
 User Datagram Protocol, Src Port: 4500, Dst Port: 4500
 Source Port: 4500
 Destination Port: 4500
 Length: 424
 Checksum: 0x5497 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 2]
 [Timestamps]
 UDP payload (416 bytes)
 UDP Encapsulation of IPsec Packets
 Non-ESP Marker
 Internet Security Association and Key Management Protocol
 Initiator SPI: 63a14f6441a91db6
 Responder SPI: 6a3bb1402e3a03ac
 Next payload: Encrypted and Authenticated (46)
 Version: 2.0
 Exchange type: IKE_AUTH (35)
 Flags: 0x08 (Initiator, No higher version, Request)
 Message ID: 0x00000001
 Length: 412
 Payload: Encrypted and Authenticated (46)

```

0000  00 15 5d 38 01 40 00 15 5d 38 01 3f 08 00 45 00  ..]8.@..]8?..E.
0010  01 bc 20 d4 40 00 40 11 c5 7f ac 14 fd ed ac 14  ..@.@.....
0020  fc c6 11 94 11 94 01 a8 54 97 00 00 00 00 63 a1  ....T....c.
0030  4f 64 41 a9 1d b6 6a 3b b1 40 2e 3a 03 ac 2e 20  OdA...j; @.:.
0040  23 08 00 00 00 01 00 00 01 9c 23 00 01 80 04 ac  #.....#....
0050  ef f9 23 d1 7e c6 79 7e 7f 31 60 c9 09 87 38 76  -#...y~.1...8v
  
```

wireshark_eth04ZA72.pcapng Packets: 4097 · Displayed: 8 (0.2%) Profile: Default