

# Security: IPsec

## Software installation

Install ipsec-tools and strongswan

Commands to install 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
```

Command to install strongswan:

```
sudo apt install strongswan
```

## IPSec configuration via setkey

### Modification of configuration files

[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;
```

## Configuration loading

On the machine 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:~$

```

On machine 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

```

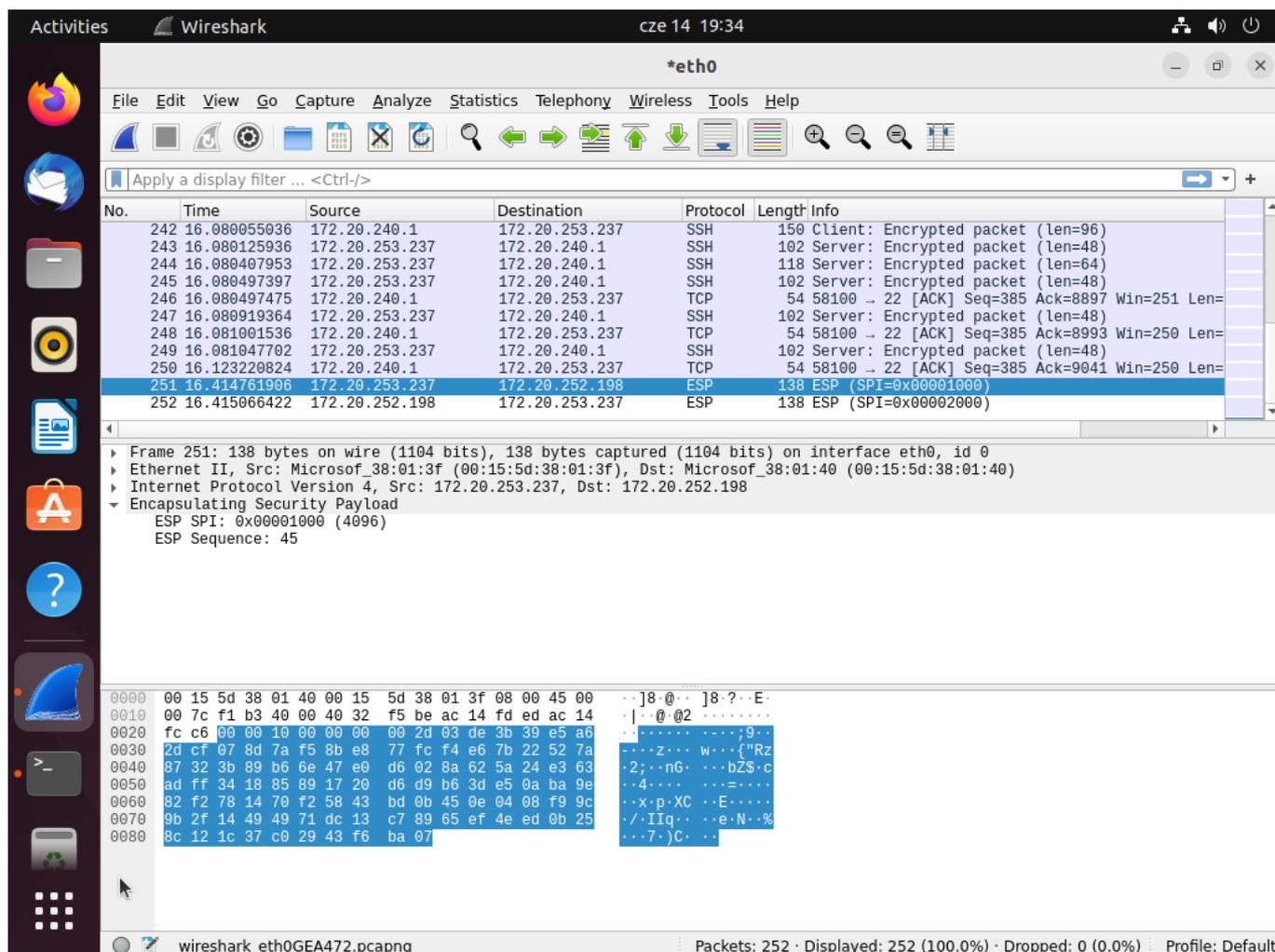
```

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:~$

```

## Tunnel Action Test

### Performing a ping from machine A to B and observing the ESP header



## Questions

### What traffic is secured by an IPsec tunnel?

IP traffic between machines A (172.20.253.237) and B (172.20.252.198) - any protocol (any) defined in the SPD.

### What IPsec protocol is used? Which algorithms?

- Protocol: ESP
- Encryption algorithm: AES-CBC
- Services: Encryption (no authentication in this example)

### Role of SAD and SPD:

- SAD: Stores SA parameters (keys, algorithms, SPI).
- SPD: Stores the security policies (which traffic is to be secured and how)

### Which commands are processed for outbound/inbound traffic?

- For outgoing packets -. `spdadd ... -P out`
- For incoming packets - `spdadd ... -P in`

### Why do you need two SAs?

SAs are unidirectional - one for  $A \rightarrow B$ , the other for  $B \rightarrow A$  to secure a two-way channel.

### Meaning of commands "setkey -D" i "setkey -DP"

- -D - shows active security associations (SA)
- -DP - shows active safety policies (SPD)

### What is the purpose of the SPI field in the IPsec header?

The SPI (Security Parameter Index) is a unique security association (SA) identifier that allows the recipient to identify which parameters to use to decrypt and authenticate the packet.

### What is the purpose of the Sequence Number field in the IPsec header?

The Sequence Number is used to prevent replay attacks. The number increases monotonically for each packet, allowing you to check that packets have not been replayed.

# Modifying IPsec and SA policies

## ESP with 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 list and a packet details pane. The packet list shows various protocols including SSH, TCP, and ESP. The selected packet (No. 134) is an Encapsulating Security Payload (ESP) packet. The details pane shows the following information:

- 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

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

### AH with 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 interface eth0. The packet list pane displays several ICMP Echo (ping) requests and replies. Packet 1084 is selected, and its details pane shows an Authentication Header (AH) and an Internet Control Message Protocol (ICMP) section. The AH section includes fields for Next header (ICMP), Length (4 bytes), Reserved (0000), AH SPI (0x00002000), AH Sequence (81), and AH ICV (919b3cc7a651f7ddeeada695). The ICMP section shows an Echo (ping) reply with ID 0x0000 and sequence number 81/20736.

No.	Time	Source	Destination	Protocol	Length	Info
1076	166.336281532	Microsof_38:01:40	Microsof_38:01:3f	ARP	42	172.20.252.198 is at 00:15:5d:38:01:40
1077	166.368019376	172.20.253.237	172.20.252.198	ICMP	122	Echo (ping) request id=0x0000, seq=78/19968
1078	166.368377204	172.20.252.198	172.20.253.237	ICMP	122	Echo (ping) reply id=0x0000, seq=78/19968
1079	167.392044628	172.20.253.237	172.20.252.198	ICMP	122	Echo (ping) request id=0x0000, seq=79/20224
1080	167.392370502	172.20.252.198	172.20.253.237	ICMP	122	Echo (ping) reply id=0x0000, seq=79/20224
1081	168.416047888	172.20.253.237	172.20.252.198	ICMP	122	Echo (ping) request id=0x0000, seq=80/20480
1082	168.416434035	172.20.252.198	172.20.253.237	ICMP	122	Echo (ping) reply id=0x0000, seq=80/20480
1083	169.440010480	172.20.253.237	172.20.252.198	ICMP	122	Echo (ping) request id=0x0000, seq=81/20736
1084	169.440342692	172.20.252.198	172.20.253.237	ICMP	122	Echo (ping) reply id=0x0000, seq=81/20736
1085	170.464242124	172.20.253.237	172.20.252.198	ICMP	122	Echo (ping) request id=0x0000, seq=82/20992
1086	170.464768431	172.20.252.198	172.20.253.237	ICMP	122	Echo (ping) reply id=0x0000, seq=82/20992

### 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 a Wireshark interface with a packet capture on the eth0 interface. The packet list pane shows several packets, with packet 31 selected. The packet details pane shows the structure of the selected packet:

- Frame 31: 162 bytes on wire (1296 bits), 162 bytes captured (1296 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
- 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 raw hex and ASCII data for the selected packet, including the AH and ESP fields.

## Questions

### Difference between ESP and ESP+AH configurations

- ESP protects the data (encrypts), but does not protect the entire IP header.
- AH does not encrypt, but protects the integrity of the entire IP packet (with header).
- ESP+AH ensures the confidentiality and integrity of the header and data.

### Application of AH

For connections where the integrity of the entire IP header is important (e.g. for auditing systems or where it is not possible to use encryption).

## IKE and strongSwan protocol (PSK)

### SAD and SPD cleansing

On bush A and B:

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

## Configurations

### Machine 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"
```

### Machine 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"
```

## Starting strongSwan and establishing a tunnel

```
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:~$
```

## Questions

### How many SA associations have been negotiated?

The setkey -D result shows two ESP associations in the SAD database (SPI 0xc04d47dc and 0xc4b20310) - one for each direction of movement.

### What type of IPSec header and mode was used for the SA associations created? What algorithms were used?

According to the ipsec log up host-host, ESP was negotiated in tunnel mode (ESP:AES\_CBC\_256/HMAC\_SHA1\_96/NO\_EXT\_SEQ).

Algorithms: AES-CBC 256-bit for encryption and HMAC-SHA1-96 (96-bit tag) for authentication.

## What is the key length used for each algorithm?

AES-CBC-256 uses a key length of 256 bits.

HMAC-SHA1-96 uses a tag of 96 bits (12 bytes), although the HMAC key itself can be longer (typically 160 bits), in the log we see 20 bytes of key, but the tag is truncated to 12 bytes.

## In which phase of the IKE protocol was the shared PSK key used?

The PSK key was used in phase 1 (IKE\_SA\_INIT and IKE\_AUTH) to authenticate the endpoints.

## What SPD policies were created by the IKE process?

setkey -DP shows four entries:

two out and in entries for ESP tunnel mode (one each for A→B and B→A traffic) with reqid = 1 and unique flag.

an additional fwd entry related to packet forwarding.

Policies apply to traffic 172.20.253.237[any]. ↔ 172.20.252.198[any] and have priority 'high', operation ipsec esp/tunnel/.../unique:1.

## Analysis of IKE traffic in Wireshark

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

The IKE\_SA\_INIT packets show the exchange of SA offers (AES\_CBC\_256/HMAC\_SHA1\_96/PRF\_HMAC\_SHA1/MODP\_1024) and DH keys.

Activities Wireshark cze 14 20:43

\*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

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  ....
  
```

Terminal

wireshark\_eth04ZAQ72.pcapng Packets: 3739 · Displayed: 8 (0.2%) Profile: Default

In IKE\_AUTH, IDi, IDr and AUTH payload with PSK authentication appear.

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