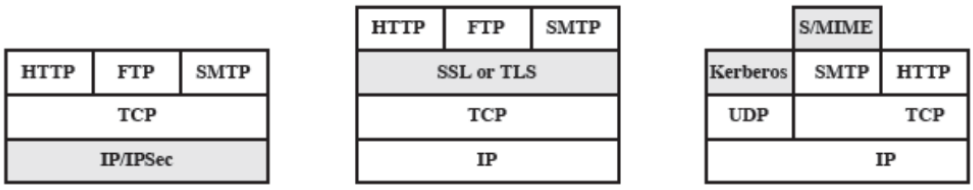
TTKS0600, Encryption Techniques and Systems, Lecture assignment 4

Lehosvuo Timo

Bordi Tuukka

1 Review questions

1.1



A) The advantage of using IPSec(Figure a) is that it is transparent to end users and applications and provides a general-purpose solution. Further, IPSec includes a filtering capability so that only selected traffic need incur the overhead of IPSec processing.

B) The advantage of using TLS (Figure b) is that it makes use of the reliability and flow control mechanisms of TCP.

C) The advantage of application-specific security services (Figure c) is that the service can be tailored to the specific needs of a given application.

1.2
- the Handshake Protocol
- the Change Cipher Spec Protocol
- the Alert Protocol

1.3 TLS connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For TLS, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

TLS session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

1.4 The phases of operation are Discovery, Authentication, Key Management, Protected data transfer and Connection termination

1.5 For key certification the difference is that S/MIME uses X.509 certificates while In OpenPGP, users generate their own OpenPGP public and private keys. X.509 certificates are trusted if there's a valid PKI chain to the trusted root but OpenPGP has a so-called Web-of-Trust where the trust is formed when the public key is signed by someone the recipient can trust.

Key distribution is also different. OpenPGP does not include the sender's public key with each message, while in S/MIME it is.

1.6
- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

1.7 Security Parameter Index or SPI, IP Destination Address and Security Protocol Identifier.

1.8 Transport mode provides protection mainly for upper-layer protocols (in OSI model) while Tunnel mode provides protection to the entire IP packet. Mainly, transport mode does not encrypt the IP header (or only partially encrypts it in AH mode) while tunnel mode encrypts also the IP header (and parts of the outer IP header in AH mode) as well as the payload. Payload is encrypted in both modes.

1.9 It consists of identification step (Presenting an identifier to the security system) and verification step (Presenting or generating authentication information that corroborates the binding between the entity and the identifier).

1.10 Kerberos realm is basically defined by the database: all nodes that share the same Kerberos database are in the same Kerberos realm

2 Hacking the WLAN encryption

2.1

WEP

Alkutilanne:

Lataa Kali Linux osoitteesta: https://www.kali.org/downloads/ tai käynnistä se, jos se on jo olemassa. Tätä konetta käytettään hyökkäyksessä.

Reititin

- Käytä Wifi reititintä, joka tukee "WEP" tilaa

- Mene reitittimen verkkosivulle (Selaimen osoite palkkiin reitittimen ip-osoite).
- Laita langattoman verkon nimeksi (SSID=service set identifier) "WEP-Wifi" (voit myös valita oman nimen, jos haluat)
- Mene turvallisuus asetuksiin ja laita "wireless security mode" "WEP" tilaan
  ja valitse ensimmäinen "transmit key" (tämänki voit vaihtaa haluamaasi, jos nii haluat)
- Anna tunnussana (passphrase) ja luo avaimet
- tallenna ja ota asetukset käyttöön.

Client

- Tarvitset toisen koneen (client-kone) millä yhdistät wifi-reitittimeen.
- Yhdistä verkkoon "WEP-wifi" ja pysy verkossa kiinni hyökkäyksen ajan.
- Tämä on tärkeää, koska muuten et verkon nimeä (SSID) ei voi löytää ilman brute-forcea.

ATTACK

Hyökkääjän koneella (kali) käynnistettiin ohjelma ”airmon—ng” komennolla:

”sudo airmon-ng”.

Komento näyttää koneen kaikki langattomat verkkorajapinnat. Komento sallii monitorointi/hallinnointi tilan verkkorajapinnoille. Listasta valittiin monitoroitavaksi rajapinta ”wlan0”. Seuraavaksi ajettiin komento:

”sudo rfkill list”.

Komento näyttää kaikkien langattomien verkkorajapintojen tilan, eli ovatko ne päällä vai ei. Tällä komennolla tarkistettiin, onko langaton LAN estetty vai ei. Jos LAN on estetty, se voidaan laittaa päälle komennolla:

”sudo rfkill unblock wifi”.

Tämän jälkeen sammutettiin langattoman rajapinnan ”wlan0” monitorointi tila komennolla:

”sudo airmon-ng stop wlan0”,
“sudo airmon-ng stop wlan0mon (airmon-ng).

Samalla myös sammutettiin koko rajapinta kirjoittamalla:

”sudo ifconfig wlan0 down”.

Komennolla ”ifconfig” konfiguroidaan verkkorajapintoja ja komennon kohta ”wlan0” kuvaa rajapinnan nimeä ja ”down” tilaa sammuta (up/down=päällä/sammunut). Kun rajapinta oli sammutettu, käynnistettiin airmon-ng ohjelma uudelleen monitoroimaan rajapintaa wlan0:

”sudo airmon-ng start wlan0”.

Parametri “start” kertoo, että halutaan käynnistää rajapinta ja ”wlan0” kuvaa rajapinnan nimeä. Komento luo monitorointi rajapinnan nimellä ”wlan0mon”. Tämä mahdollistaa kaikkien pakettien kuuntelemisen verkossa. Rajapinta pitää myös sallia käyttäen komentoa:

”sudo iwconfig”.

”iwconfig” on hyvin samanlainen kuin aikaisemmin käytetty ”ifconfig”, mutta se on suunnattu vain langattomiin rajapintoihin. Komentoa käytetään, kun halutaan asettaa parametrejä, jotka koskevat langattomia verkkorajapintoja. Tällä voi myös nähdä käytettävät parametrit ja statistiikkaa. Aikaisemmin käytetty komento ”sudo airmon-ng start wlan0” kertoi mahdollisista haitallisista prosesseista ja nämä ”tapettiin” komennolla:

”sudo airmon-ng check kill”.

Seuraavaksi monitoroitiin langattomia verkkoja käyttämällä langatonta verkkorajapintaa ”wlan0mon”:

”sudo airodump-ng wlan0mon”.

“airodump-ng” käytetään ”raakojen” 802.11 pakettien sieppaamiseen ja on erityisen hyvä WEP:tä varten. Komennolla nähdään kaikki saatavilla olevat langattomat verkot. Näistä etsittiin verkko, joka vastaa aikaisemmin konfiguroitua wifi reititintä (BSSID vastaa reitittimen MAC-osoitetta, salaustila on ”WEP” ja ESSID on ”WEP-Wifi”). Oikean verkon löydyttyä lopetettiin verkon monitorointi (ctrl + c) ja konfiguroitiin airodump-ng monitoroimaan löydettyä verkkoa ja laittamaan kaikki data kansioon:

”sudo airodump-ng -c(channel) -w (capture filenamel) –bssid (BSSID) wlan0mon”

- Parametri “-c” tarkoittaa kanavaa (channel). Oletuksena airodump-ng kuuntelee kaikkia 2,4GHz kanavia
- Parametri ”-w” kuvaa tiedoston nimeä mihin data tallennetaan.
- Parametri ”—bssid” seuloo AP:ita BSSID arvolla eli reitittimen MAC-osoitteella.
- Viimeinen parametri on verkkorajapinnan nimi

Seuraavaksi käynnistettiin ”aireplay-ng” sovellus ja aloitettiin väärennetyjen autentikaatioiden lähettäminen reitittimen liityntäpisteelle (access point(AP)). Aireplay-ng:tä käytetään langattoman verkon ”freimien” injektoimiseen, ohjelmassa on monta eri hyökkäys tapaa. Komento:

”sudo aireplay-ng -1 6000 -o 1 -q 10 -e (ESSID) -a (BSSID) -h (attacker’s MAC) wlan0mon”

Parametrit:

- “-1” tällä valitaan “attack mode” (0-9) eli tässä tapauksessa ”1” tarkoittaa “fake authentication with ap”.
- luku “6000” kertoo, kuinka usein tehdään uudelleen autentikointi (re-authenticate). Luku kuvaa aikaa sekunneissa.
- “-o” kertoo Kuinka monta paketia lähetetään ”burstissa”.
- ”-q” kertoo kuinka monta sekunttia on TTL viestien välissä.

- ”-e” tähän laitetaan kohde verkon nimi (WEP-wifi
- ”-a” AP:n MAC-osoite eli reitittimen
- ”-h” Hyökkäjän MAC-osoite. Voi jättää myös pois, käyttää silloin koneen MAC-osoitetta.
- Viimeiseksi parametriksi käytettävän verkkorajapinnan nimi eli tässä tapauksessa ”wlan0mon”.

Tämän jälkeen käytettiin ”aireplay-ng” sovellusta luomaan suuri määrä dataa verkkoon. Tämän tarkoituksena oli saada selville salattu salasana. Aireplay-ng lähettää paljon ARP-paketteja uusilla IV:llä (initialization vector):

”sudo aireplay-ng -3 -b (BSSID) -h (attacker’s MAC) (interface)”.

Parametrit:
- “-3” kertoo hyökkäys tavan, tässä tapauksessa ”standard ARP-request replay”.
- ”-b” AP:n MAC-osoite.
- ”-h” Hyökkääjän MAC-osoite. Voi jättää myös pois, käyttää silloin koneen MAC-osoitetta.
- Viimeiseksi rajapinnan nimi.

Ohjelman pyöriessä pakettien ja ARP-kyselyiden määrän pitäisi kasvaa. Tämän jälkeen odota 5 – 10 minuuttia ja tarkista oletko saanut viestin ”Got a deauth/disassoc packet. Is the source mac associated?” tämä tarkoittaa, että ole menettäny ”liitännän” (association) AP:n kanssa. Kaikki injektoidun paketit sivuutetaan, mutta autentikointi pyörii luupilla. Jos et saa viestiä niin voit myös pakottaa kättelyn komennolla:

”sudo aireplay-ng -0 1 -a (BSSID) -c (client’s MAC) (interface)”.

Parametrit:
- “-0” kertoo hyökkäys tavan, tässä tapauksessa ”deauthenticate 1 or all stations”.
- ”-a” AP:n MAC-osoite.
- “-c” Clientin MAC-osoite.
- “interface” rajapinnan nimi.

Tämän seurauksena ”clientti” kirjautuu takaisin sisään, jos automaattinen uudelleen kirjautuminen on päällä. Tämän voit myös tehdä itse, jos kokeilet harjoitusta omiin laitteisiisi.

Kun ohjelma oli tuottanut riittävästi dataa, pystyttiin WEP murtamaan ja avain löytämään. Avain saatiin, kun ajettiin komento:

”sudo aircrack-ng -b (BSSID) (capture filename-01.cap)

Parametrit:
- “-b” AP:n MAC-osoite
- ”capture filename-01.cap” kertoo tiedoston nimen, eli filename-01.cap

Tästä tiedostosta löytyy avain, jos dataan kerätty riittävästi. Jos tiedostossa lukee ”FAILED” tarkoittaa se, että dataa ei ollut kerätty riittävästi ja joudut ajamaan komennot uudelleen (ARP ja

aircrack). Saanulla avaimella ja kohteen verkon nimellä (SSID) voidaan yhdistää kyseiseen verkkoon.

WEP:n haavoittuvuus

WEP käyttää salaamiseen RC4 salausalgoritmia ja RC4 vaatii, että alustamis vektorit (initialization vectors) ovat satunnaisia. RC4 WEP:ssä toistaa tämän prosessin joka 6000 freimi ja kun näitä freimejä saa riittävästi haltuunsa pystyy avaimen salauksen purkamaan. IV:n ongelmia WEP:ssä

- IV on lyhy, vain 24 bittiä pitkä
- IV lähetetään viestin selkoteksti osuudessa
- Samaa IV:tä käytetään uudestaan (“produces identical key streams for the protection of data”)
- 802.11 standardi ei spesifioi miten IV:t asetetaan tai muokataan.

Eli ohjelmat käyttävät tätä haavoittuvuutta hyödykseen. Pakottamalla deauthentication ja keräämällä riitävästi dataa (ARP-paketit) voidaan avain saada selville juuri tämän IV toistuvuuden avulla.

WPA2

ALKUHUOMIO: yritimme tehdä tätä tehtävää livenä, mutta jostain syystä annetut komennot eivät pelanneet toivotulla tavalla tai ne antoivat ei-odotettuja tuloksia (kuten sen, että airomon-ng ei löytänyt yhtään verkkoa, jolla olisi ESSID <length: 8> vaikka sen varmasti asetimme oikein. Joten teemme nyt käytännön työn lisäksi selostuksen, joka on lisännyt tämän tehtävän työmäärää meille huomattavasti.

Alkutilanne:

Lataa Kali Linux osoitteesta: https://www.kali.org/downloads/ tai käynnistä se, jos se on jo olemassa. Tätä konetta käytettään hyökkäyksessä.

Reititin

- Käytä Wifi reititintä, joka tukee "WPA2" tilaa

- Mene reitittimen verkkosivulle (Selaimen osoite palkkiin reitittimen ip-osoite).

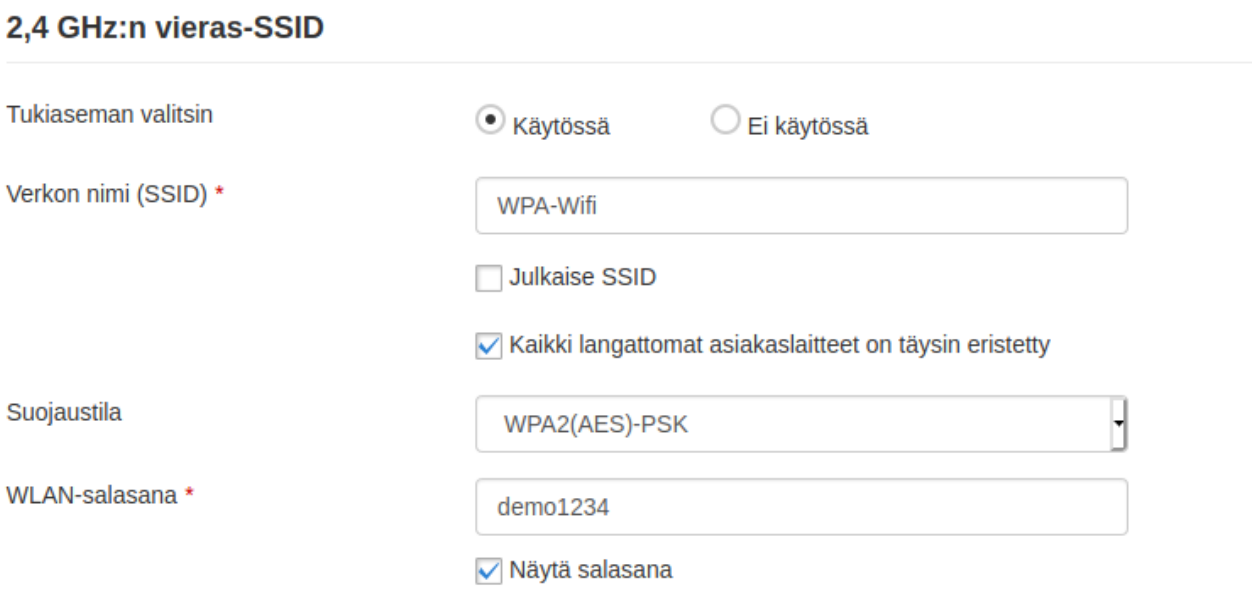
- Laita langattoman verkon nimeksi (SSID=service set identifier) "WPA-Wifi" (voit myös valita oman nimen, jos haluat)

- Mene turvallisuus asetuksiin ja laita "wireless security mode" "WPA" tilaan

- Aseta salasana (demo123)

- tallenna ja ota asetukset käyttöön.

Kuva setupista:



Client

- Tarvitset toisen koneen (client-kone) millä yhdistät wifi-reitittimeen.

- Yhdistä salattuun verkkoon "WPA-wifi" ja pysy verkossa kiinni hyökkäyksen ajan.

- Joudut yhdistämään client-koneella verkkoon jossain vaiheessa, niin että handshake tallentuu Kalille

- Tämä on tärkeää, koska muuten et pysty kaappaamaan handshakea

Käytä hyökkäyksessä salasanalistaa koska se helpottaa salasana löytymistä. Kali Linuxissa on valmiita salasanalistoja ja valitse niistä fasttrack.txt joka löytyy kansiosta /usr/share/wordlists/. Kopioi tiedosto fasttrack.txt komennolla:

“cp /usr/share/wordlists/fasttrack.txt <uusi tiedostonimi>”.

Komento kopioi (cp=copy) tiedoston uudella nimellä kansioon ”wordlists”. Tämän jälkeen

Tarkista onko salasana "demo1234" tiedoston sisällä komennolla:

"grep demo1234 fasttrack.txt".

komento "grep" etsii tekstiä annetun tiedoston sisältä eli tässä tapauksessa komento etsii tekstiä "demo1234" tiedoston fasttrack.txt sisältä. Syntaksi on seuraavanlainen:

”grep "etsittävä teksit" "tiedosto mistä etsitään".

Seuraavaksi tiedosto pitää muuttaa wpa sanastoksi eli salasanojen pitää olla vähintään 8 merkkiä pitkiä. Tämä toteutetaan putkittamalla komennot:

“cat fasttrack.txt | sort | uniq | pw-inspector -m 8 -M 63 > wpa.txt”.

komento "cat" lukee dataa tiedostosta ja tulostaa tiedoston sisällön "outputtina", komento "sort" lajitteleen tiedoston sisällön, jonka logiikka lajittelussa on seuraava:

1. Numerolla alkavata rivit näkyvät ennen kirjaimella alkavia.

2. Rivit mitkä alkavat kirjaimella, joka on aakkosissa ensimmäisenä näkyvät ennen rivejä, jossa kirjain on aakkosissa kauempana.

3. Rivit, jotka alkavat pienellä kirjaimella näkyvä ennen isolla/isoilla kirjaimilla kirjoitettuja.

komento "uniq" poistaa tai huomaa tiedostosta kaikki "duplikaatit" ja komento "pw-inspector" on työkalu, jolla voidaan muokata salasanoja. Pw-inspector lukee salasanoja ja "tulostaa" salasanat, jotka täyttävät vaatimukset. PW-inspectorissa olevat parametrit "-m" ja "-M" tarkoittavata salasanan minimipituutta (-m) ja maksimipituutta (-M). Eli pw-inspectori tulostaa kaikki salasanat, jotka ovat vähintään 8 merkkiä pitkiä ja enintään 63 merkkiä pitkiä. Tulostus palautta numeron kuinka monta salasanaa parametreillä löytyi. Viimeiseksi parametri ”> wpa.txt” tekee tiedoston ”wpa.txt” tai vie tulosteen tiedostoon ”wpa.txt” jos se on jo olemassa. Komentojen välissä olevat putki "|" tarkoittaa putkittamista eli komento syöttää vasemmalla olevat komennon outputin oikealla olevaan komentoon. Eli komennon "cat fasttrack" output menee "sort" komentoon ja "sort" komennon output menee komentoon "uniq" jne... Lopputuloksena on että, komento:

"cat fasttrack.txt | sort | uniq | pw-inspector -m 8 -M 63 > wpa.txt".

Sama kuvana:

```
kali@kali:~$ cp /usr/share/wordlists/fasttrack.txt .
kali@kali:~$
kali@kali:~$ grep demo1234 fasttrack.txt
kali@kali:~$ echo demo1234 >> fasttrack.txt
kali@kali:~$ grep demo1234 fasttrack.txt
demo1234
kali@kali:~$ cat fasttrack.txt | sort | uniq | pw-inspector -m 8 -M 63 > wpa.txt
kali@kali:~$
```

Tekee tiedoston ”wpa.txt” missä on kaikki salasanat fasttrack tiedostosta, jotka ovat pituudeltaa vähintää 8 merkkiä ja enintää 63 merkkiä sekä lajittelee ja poistaa ”duplikaatit”. Seuraavaksi valmistaudumme hyökkäykseen. Komennolla rfkill list voimme listata kaikki koneen verkkorajapinnat ja sen, onko rajapinta “blokattu”. Blokattu tarkoittaa tässä tapauksessa sitä, että blokattuna rajapinta ei lähetä tai vastaanota mitään, eli se ei kuluta virtaa. Komento antoi meille 5 eri rajapintaa, kuten seuraavassa kuvassa näkyy. Mikään verkko ei onneksi meillä ollut blokattu, joten jatkoimme eteenpäin. Jos rajapinta wlan0 olisi ollut blokattu, olisi meidän pitänyt suorittaa seuraava komento:

”sudo rfkill unblock wifi ”.

Tämä komento olisi sallinut rajapinnan taas lähettää ja vastaanottaa radiomagneettisen säteilyn avulla. Seuraavaksi täytyy estää wlan0-rajapinnan menemistä monitorointi-tilaan, joka tehdään komennolla. Monitorointi-tila on verkkorajapinnan erityinen tila, jolla rajapinta seuraa verkon liikennettä.

”sudo airmon-ng stop wlan0“

“sudo airmon-ng stop wlan0mon”

Viimeiseksi viedään koko rajapinta alas komennolla sudo ifconfig wlan0 down. Tämä tehdään siksi, että rajapinnan MAC-osoitetta voidaan muuttaa, mikä tehdään seuraavaksi. Komento, jolla vaihdetaan rajapinnan MAC-osoite on:

”sudo macchanger--mac 00:11:22:33:44:55 wlan0 ”

Jossa –mac lipulla annetaan uusi MAC-osoite, jonka perään pitää tulla rajapinnan nimi (tässä wlan0). Seuraavaksi alamme nuuskimaan “sniffing”, verkkoa. Nämä tehdään seuraavilla komennoilla:

”sudo airmon-ng start wlan0 ”

”sudo iwconfig“

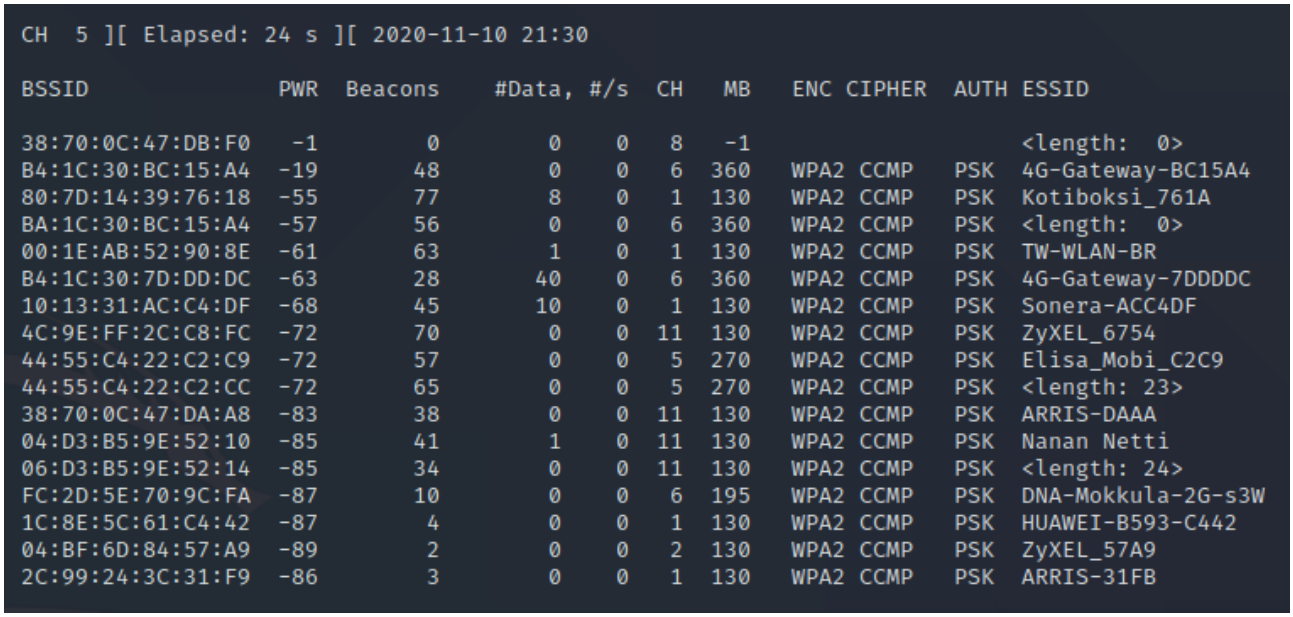
Näistä komennoista ensimmäinen käynnistää nuuskimissovelluksen ja toinen määrittää valmiiksi langattoman rajapinna. Iwconfig ei toimi muille, kuin langattomille rajapinnoille. Viimeiseksi, että monitorointi ei häriintyny, käsketään airmon tuhoamaan kaikki prosessit, jotka voisivat häiritä tätä operaatiota

”sudo airmon-ng check kill”

Viimein käynnistetään kaappaus komennolla

”sudo airodump-ng wlan0mon”

Joka aloittaa langattomassa verkossa kulkevan liikenteen monitoroinnin. Huomattavaa airodump-ng työkalussa on se, että se on varta vasten tehty kaappaamaan liikennettä sellaisessa muodossa, jota aircrack-ng –sovellus voi käyttää. Aircrack-ng on sovellus, jolla voi murtautua langattomiin verkkoihin. Tekemämme kaappaus on seuraava. Emme löytäneet yhtään verkkoa, jossa olisi tehtävänannon mukainen ESSID <length: 8>. Totesimme, että verkko (WPA-Wifi) on kuitenkin olemassa yhdistämällä siihen client-koneella.

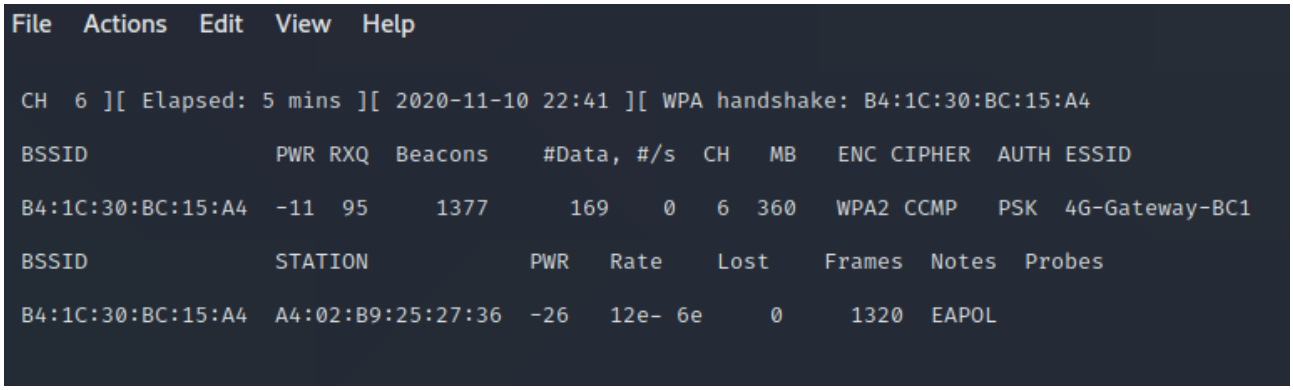


```
CH  5 ][ Elapsed: 24 s ][ 2020-11-10 21:30

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

38:70:0C:47:DB:F0   -1        0        0    0   8   -1                    <length:  0>
B4:1C:30:BC:15:A4  -19       48        0    0   6  360   WPA2 CCMP   PSK  4G-Gateway-BC15A4
80:7D:14:39:76:18  -55       77        8    0   1  130   WPA2 CCMP   PSK  Kotiboksi_761A
BA:1C:30:BC:15:A4  -57       56        0    0   6  360   WPA2 CCMP   PSK  <length:  0>
00:1E:AB:52:90:8E  -61       63        1    0   1  130   WPA2 CCMP   PSK  TW-WLAN-BR
B4:1C:30:7D:DD:DC  -63       28       40    0   6  360   WPA2 CCMP   PSK  4G-Gateway-7DDDDC
10:13:31:AC:C4:DF  -68       45       10    0   1  130   WPA2 CCMP   PSK  Sonera-ACC4DF
4C:9E:FF:2C:C8:FC  -72       70        0    0  11  130   WPA2 CCMP   PSK  ZyXEL_6754
44:55:C4:22:C2:C9  -72       57        0    0   5  270   WPA2 CCMP   PSK  Elisa_Mobi_C2C9
44:55:C4:22:C2:CC  -72       65        0    0   5  270   WPA2 CCMP   PSK  <length: 23>
38:70:0C:47:DA:A8  -83       38        0    0  11  130   WPA2 CCMP   PSK  ARRIS-DAAA
04:D3:B5:9E:52:10  -85       41        1    0  11  130   WPA2 CCMP   PSK  Nanan Netti
06:D3:B5:9E:52:14  -85       34        0    0  11  130   WPA2 CCMP   PSK  <length: 24>
FC:2D:5E:70:9C:FA  -87       10        0    0   6  195   WPA2 CCMP   PSK  DNA-Mokkula-2G-s3W
1C:8E:5C:61:C4:42  -87        4        0    0   1  130   WPA2 CCMP   PSK  HUAWEI-B593-C442
04:BF:6D:84:57:A9  -89        2        0    0   2  130   WPA2 CCMP   PSK  ZyXEL_57A9
2C:99:24:3C:31:F9  -86        3        0    0   1  130   WPA2 CCMP   PSK  ARRIS-31FB
```

Koska tämä komento antoi odotetun vastaisia tuloksia, päätimme kaapata liikenteen julkiselta rajapinnalta, jota reitittimemme jakoi. Tämän rajapinnan ESSID on tuo yllä olevassa kuvassa näkyvä “4G-Gateway-BC15A4". Kaappasimme liikenteen ja autentikaation tästä, jonka jälkeen jatkoimme eteenpäin.

Seuraava komento, jolla on tarkoitus saada talteen handshake-tapahtuma perustui vahvasti edelliseen komentoon. Ensinnäkin tarvittiin langattoman tukiaseman BSSID, jonka saa edellisestä kuvasta. Toisekseen tarvittiin client-koneen MAC-osoite, joka löytyi myös edellisen komennon avulla (ei kuvassa). Komento oli seuraava ja antoi seuraavan tuloksen (kuva):

“sudo airodump-ng -c 6 -w capture.txt --bssid B4:1C:30:BC:15:A4 wlan0 “



```
File   Actions   Edit   View   Help

CH  6 ][ Elapsed: 5 mins ][ 2020-11-10 22:41 ][ WPA handshake: B4:1C:30:BC:15:A4

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

B4:1C:30:BC:15:A4  -11  95     1377      169    0   6  360   WPA2 CCMP   PSK  4G-Gateway-BC1

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

B4:1C:30:BC:15:A4  A4:02:B9:25:27:36  -26   12e- 6e     0     1320  EAPOL
```

Tämä komento ja sen lopputulos käytännössä antoivat meille kaappauksen, jota voimme käyttää hyväksi. Komennon liput merkitsevät tätä:

”-c” kertoo WLAN-kanavan, jota kuunnellaan (WLAN-kanavia on yhteensä 32 [ainakin omassa reitittimessä])

”-w” kertoo käytännössä tiedostonimen-mihin tallennus tehdään. Huomioitavaa on se, että airodump luo monia tiedostoja, joilla on sama tiedoston etuliite, kuin tämän flagin parametrilla

”—bssid” kertoo yksittäisen aseman BSSID:n jota halutaan erityisesti tarkkailla.

Eli tämä komento tarkkailee juuri yhtä tukiasemaa, jotta sen kommunikaatiota voidaan käyttää hyökkäyksiin. Saimme tästä halutut tiedostot ulos, ja seuraavaksi on hyökkäyksen aika. Sueraava komento on ensimmäinen osa hyökkäyksestä:

“sudo aireplay-ng -0 1 -a B4:1C:30:BC:15:A4 -c A4:02:B9:25:27:36 “

Jossa –a on BSSID ja –c clientin MAC-osoite. Komento toistaa wpa-2 tukiaseman handshaken niin, että joitakin osia kommunikaatiosta voidaan purkaa, joka altistaa yhteyden sanakirjahyökkäykselle. Meillä tämä komento ei kuitenkaan onnistunut, ehkä siksi, että käytämämme läppäri on liian vanha? (8-10 vuotta). Tämä tuloksen komento antoi, ja jouduimme luopumaan hands-on kokeilusta tämän takia.

```
kali@kali:~$ sudo aireplay-ng -0 1 -a b4:1c:30:bc:15:a4 -c A4:02:B9:25:27:36 wlan0
21:33:56  Waiting for beacon frame (BSSID: B4:1C:30:BC:15:A4) on channel 11
21:33:57  wlan0 is on channel 11, but the AP uses channel 6
kali@kali:~$ sudo aireplay-ng -0 1 -a b4:1c:30:bc:15:a4 -c A4:02:B9:25:27:36 wlan0
21:36:30  Waiting for beacon frame (BSSID: B4:1C:30:BC:15:A4) on channel 11
21:36:36  wlan0 is on channel 11, but the AP uses channel 6
kali@kali:~$
```

Viimeinen osio hyökkäyksessä on murtaa salasana. Tähän käytämme aircrack-ng:n salasananmurtotoimintoa, jossa voi käyttää hyödyksi sanakirjahyökkäystä.

“sudo aircrack-ng –a2 -b B4:1C:30:BC:15:A4 -w fasttrack.txt capture.txt “

”-b” on tässä komennossa tukiaseman BSSID,
”-w” nimeää sanakirjatiedoston ja
”-a2” kertoo, että murtaudumme WPA2-verkkoon

Tämän komennon jälkeen sinulla pitäisi olla salausavaimet ja salasana taskussasi!

2.3 Sanakirjahyökkäyksen laajentaminen

Seuraavan tehtävän tarkoitus on opetella käyttämään sanakirjatiedostoja sillä tavalla, että voidaan kokeille myös sanojen yhdistelmiä sekä yleisien lisämerkkien lisäystä sanakirjan salasanoihin. Alkuasetelma tässä tehtävässä on se, että meillä ei olisikaan täydellisen valmista salasanaa fasttrack.txt tiedostossa, ja ensimmäinen tehtävä onkin poistaa se sanakirjasta. Tässä tehtävässä oletetaan, että meillä on sanakirjatiedosto “wpa.txt”, johon on lisätty salasana “demo12”. Pelkästään tällä salasanalla ei saisi murtauduttua wlan-tukiasemaan niin kuin aiemmin.

Käytämme ilmeisesti John the Ripperiä hyödykseen, sillä sanakirjahyökkäysasetuksia muokataan sijainnissa /etc/john/john.conf. Tehtävänanto pyytää etsimään osion [List.Rules:Wordlist] ja lisäämään sen loppuun merkkisarjan $[0-9]$[0-9]. Tämä sarja muistuttaa kovasti monen ohjelmointikielen regular expression –syntaksia. Käytännössä tämä rimpsu kertoo lisäämään kaksi numeroa jokaisen salasanan loppuun. Kokeilluksi tulee kaikki numerot 00-99. Näiden uusien

sääntöjen avulla luodaan dynaamisesti uusi sanakirja seuraavan komennon avulla:

”sudo john --wordlist=wpa.txt --rules --stdout >> wpa2.txt “

Komennon input on –wordlist, joka kertoo mikä sanakirja on pohjana uusien sääntöjen perustalta luotavalla sanakirjalle. --Rules kertoo, että tulemme käyttämään conf-tiedoston sääntöjä ja –stdout kertoo, että putkitamme kaiken outputin stdout:iin, josta viimeiseksi syntaksilla >> viemme jo valmiin sanakirjan loppuun jo olemassa olevien salasanojen kaveriksi. Jos tuota stdout –parametria ei olisi, emme voisi käyttää >> operaattoria, joka lukee sitä, mitä stdoutissa on.

Seuraavaksi teemme jo aiemmin selostetun hyökkäyksen uudestaan vähän eri parametreilla:

“sudo aircrack-ng –e “4G-Gateway-BC15A4" -w wpa2.txt “

Komennon –e on ESSID (eli ei BSSID enää!) ja –w on sama kuin aiemmin, eli sanakirjatiedosto (tässä laajennettu sanakirja wpa2.txt).

Toinen mahdollisuus sanakirjahyökkäykseen on käyttää ohjelmaa nimeltä cowpatty, joka ei välttämättä löydy oletuksena kalista. Cowpatty on tehtävänannon mukaan enemmän automatisoitu kuin aircrack-ng. Komento on seuraava:

“sudo john --wordlist=wpa2.txt --rules --stdout | cowpatty-s “4G-Gateway-BC15A4" -f --r capture.txt”

Komento putkittaa john:in outputin cowpattylle, joka suorittaa hyökkäyksen automatisoidusti.

2.2 WEP-salauksen initialization vector (IV) on huomattavan pieni. Tämän takia IV:t loppuvat kesken aika nopeasti (noin 7 tunnissa) WEP-salatussa verkossa, joten samoja IV:tä joudutaan käyttämään uudestaan. Tämä altistaa liikenteen analysointihyökkäykselle. Toinen heikkous on, että rc4-salausalgoritmi, jota WEP käyttää, lähettää IV:n RC4-avaimen mukana plaintextinä. Tämä helpottaa huomattavasti edellämainitun analysointihyökkäyksen toteuttamista.
IV-arvojen muodostuksessa on myös ongelma.WEPin käyttämä RC4-algoritmi ei anna erityisiä ohjeita siitä, miten IV generoidaan. Tämä tarkoittaa sitä, että monella laitevalmistajalla voi olla käytössä hyvin yksinkertainen ja jopa deterministinen IV:n luontialgoritmi, joka altistaa kaikki laitteet, jotka käyttävät samaa IV:n luontialgoritmia hyökkäykselle, jos yksi sen valmistajan laitteen IV:n luontialgoritmi on selvitetty

2.3 WPA3 paikkaa monia haavoittuvuuksia, joita WPA2:ssa on. WPA2 on altis bruteforce-hyökkäyksille (niin kuin tehtävässä 2.1 voi todeta, tämä hyökkäys on nimeltään “KRACK attack”), jonka WPA3 korjaa.

Toisekseen WPA2 mahdollisti muiden käyttäjien liikenteen seuraamisen, jos sinulla on tiedossa WPA-verkon salasana ja muita käyttäjiä on liittynyt samaan verkkoon. WPA3 korjaa myös tämän (sessioavaimet ovat WPA:ssa yksityisiä). Tämä muiden käyttäjien liikenteen näkeminen on altistanut WPA2:n myös monille muille hyökkäyksille (esim “hole196”)

Toki yksi etu WPA:ssa (nnin kuin aina uudemmissa turvallisuusratkaisuissa) on pidemmät avaimenpituudet ja modernimmat kryptografiset ominaisuudet, jotka tekevät verkkon murtautumista vaikeampaa, varsinkin, kun tässä tehtävässä mainitut haavoittuvuudet on korjattu.

Kolmas haavoittuvuus, joka on WPA2:ssa mutta ei enää WPA3:ssa on heikon salasanan ongelma: koska käyttäjän salasana on häshätty yhdistämällä verkon SSID ja salasana, on olemassa hashitaulukoita (rainbow tableja), jossa on yleisimmät hotspot-salasanayhdistelmät. Näitä voi käyttää sanakirjahyökkäyksissä.

Viimeisenä haavoittuvuutena on se, että WPA2-enterprisen MS-CHAPv2 tunnistumisessa on todettu haavoittuvuuksia käytännössä serverin canonical namen tunnistuksessa, kun radius server on käytössä. Tätäkään haavoittuvuutta ei ole enää WPA3:ssa

2.4

3 IPSec

Käytetyissä conffeissa left tarkoittaa localia ja right ulkoista. Conn %default on kaikkien yhteyksien yhteiset asetukset. Yhteys ”rw” tarkoittaa PSK yhteyttä ja ”pki-rw” tarkoittaa PKI yhteyttä.

Figure 1: ipsec-server conf

Verkkoyhteys on mahdollista ”leftsubnet” ja ”leftfirewall” asetuksien avulla. Leftsubnet pitää olla 0.0.0.0/0 jotta liikenne kaikkiin verkkoihin sallitaan. Leftfirewall luo automaattisesti palomuuri asetukset serverille ja vastaavasti myös clientille. Authby=secrets tarkoittaa että käytämmä ipsec.secret tunnistetietoja jota tarvitaan PSK yhteydessä.

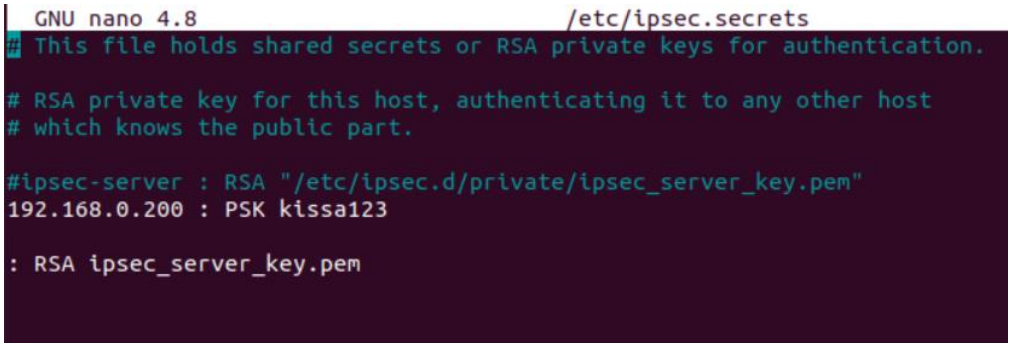


Figure 2: ipsec-server secrets

Conffin yhteys “home” on sama kuin serverin ”rw” ja ”pki-home” on sama kuin serverin ”pki-rw”.

```
  GNU nano 4.8                                    /etc/ipsec.conf
config setup

conn %default
        ikelifetime=60m
        keylife=20m
        rekeymargin=3m
        keyingtries=1
        keyexchange=ikev2
        #leftdns=8.8.8.8,8.8.4.4

conn home
        left=192.168.0.200
        leftfirewall=yes
        right=192.168.0.1
        rightsubnet=0.0.0.0/0
        auto=add
        authby=secret

conn pki-home
        left=192.168.0.200
        leftfirewall=yes
        leftcert=ipsec_pki_client_cert.pem
        leftid="C=FI, O=JAMK, CN=ipsec-pki-client"
        right=192.168.0.1
        rightsubnet=0.0.0.0/0
        rightid="C=FI, O=JAMK, CN=ipsec-server"
        auto=add
```

Figure 3: ipsec-client conf

```
  GNU nano 4.8                                    /etc/ipsec.secrets
# This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host
# which knows the public part.

192.168.0.200 : PSK kissa123
: RSA ipsec_client_key.pem
```

Figure 4: ipsec-client secrets

```
user@ipsec-pki-client:~/Desktop$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.8.2, Linux 5.4.0-52-generic, x86_64):
  uptime: 23 minutes, since Nov 10 17:51:39 2020
  malloc: sbrk 2129920, mmap 0, used 1162480, free 967440
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 9
  loaded plugins: charon test-vectors ldap pkcs11 tpm aesni aes rc2 sha2 sha1 md5 mgf1 rdrand ran
dom nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem opens
sl gcrypt af-alg fips-prf gmp curve25519 agent chapoly xcbc cmac hmac ctr ccm gcm ntru drbg curl
attr kernel-netlink resolve socket-default connmark farp stroke vici updown eap-identity eap-aka
eap-md5 eap-gtc eap-mschapv2 eap-dynamic eap-radius eap-tls eap-ttls eap-peap eap-tnc xauth-gener
ic xauth-eap xauth-pam tnc-tnccs dhcp lookip error-notify certexpire led addrblock unity counters
Listening IP addresses:
  192.168.0.200
Connections:
        home:  192.168.0.200...192.168.0.1  IKEv2
        home:   local:  [192.168.0.200] uses pre-shared key authentication
        home:   remote: [192.168.0.1] uses pre-shared key authentication
        home:   child:  dynamic === 0.0.0.0/0 TUNNEL
    pki-home:  192.168.0.200...192.168.0.1  IKEv2
    pki-home:   local:  [C=FI, O=JAMK, CN=ipsec-pki-client] uses public key authentication
    pki-home:    cert:  "C=FI, O=JAMK, CN=ipsec-pki-client"
    pki-home:   remote: [C=FI, O=JAMK, CN=ipsec-server] uses public key authentication
    pki-home:   child:  dynamic === 0.0.0.0/0 TUNNEL
Security Associations (1 up, 0 connecting):
        home[3]: ESTABLISHED 13 minutes ago, 192.168.0.200[192.168.0.200]...192.168.0.1[192.168.0
.1]
        home[3]: IKEv2 SPIs: 2cd0742c35c111fd_i* 7169efd3ed30c334_r, pre-shared key reauthenticat
ion in 40 minutes
        home[3]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_AES128_XCBC/ECP_256
        home{3}:  INSTALLED, TUNNEL, reqid 3, ESP SPIs: ceedc61a_i cb1c428e_o
        home{3}:  AES_CBC_128/HMAC_SHA2_256_128, 2564755 bytes_i (2490 pkts, 9s ago), 201518 byte
s_o (1722 pkts, 4s ago), rekeying in 2 minutes
        home{3}:   192.168.0.200/32 === 0.0.0.0/0
user@ipsec-pki-client:~/Desktop$
```

Figure 5: ipsec-client statusall

Alla olevassa kuvassa olevat yhteydet ”rw_pki”, ”rw_psk” ja ”net” ovat ilmeisesti vanhoja yrityksiä ja ne eivät toimi. Toimivat yhteydet ovat rw ja pki-rw

```
user@ipsec-server:~/Desktop$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.8.2, Linux 5.4.0-52-generic, x86_64):
  uptime: 24 minutes, since Nov 10 17:51:44 2020
  malloc: sbrk 2007040, mmap 0, used 1226528, free 780512
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 6
  loaded plugins: charon test-vectors ldap pkcs11 tpm aesni aes rc2 sha2 sha1 md5 mgf1 rdrand ran
dom nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem opens
sl gcrypt af-alg fips-prf gmp curve25519 agent chapoly xcbc cmac hmac ctr ccm gcm ntru drbg curl
attr kernel-netlink resolve socket-default connmark farp stroke vici updown eap-identity eap-aka
eap-md5 eap-gtc eap-mschapv2 eap-dynamic eap-radius eap-tls eap-ttls eap-peap eap-tnc xauth-gener
ic xauth-eap xauth-pam tnc-tnccs dhcp lookip error-notify certexpire led addrblock unity counters
Listening IP addresses:
  192.168.0.1
  10.0.2.7
Connections:
         rw:  192.168.0.1...%any  IKEv2
         rw:   local:  [192.168.0.1] uses pre-shared key authentication
         rw:   remote: uses pre-shared key authentication
         rw:   child:  0.0.0.0/0 === dynamic TUNNEL
      pki-rw:  192.168.0.1...%any  IKEv2
      pki-rw:   local:  [C=FI, O=JAMK, CN=ipsec-server] uses public key authentication
      pki-rw:    cert:  "C=FI, O=JAMK, CN=ipsec-server"
      pki-rw:   remote: uses public key authentication
      pki-rw:   child:  0.0.0.0/0 === dynamic TUNNEL
      rw_pki:  192.168.0.1...%any  IKEv2
      rw_pki:   local:  [ipsec_server@jamk.fi] uses public key authentication
      rw_pki:    cert:  "C=FI, O=JAMK University of Applied Sciences, CN=ipsec_server@jamk.fi"
      rw_pki:   remote: uses public key authentication
         net:   child:  10.0.2.0/24 === dynamic TUNNEL
      rw_psk:  192.168.0.1...%any  IKEv2
      rw_psk:   local:  uses pre-shared key authentication
      rw_psk:   remote: uses pre-shared key authentication
         net:   child:  10.0.2.0/24 === dynamic TUNNEL
Security Associations (1 up, 0 connecting):
         rw[3]: ESTABLISHED 14 minutes ago, 192.168.0.1[192.168.0.1]...192.168.0.200[192.168.0.2
00]
         rw[3]: IKEv2 SPIs: 2cd0742c35c111fd_i 7169efd3ed30c334_r*, pre-shared key reauthenticat
ion in 42 minutes
         rw[3]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_AES128_XCBC/ECP_256
         rw{4}:  INSTALLED, TUNNEL, reqid 3, ESP SPIs: c5c6346e_i c7e16466_o
         rw{4}:  AES_CBC_128/HMAC_SHA2_256_128, 0 bytes_i, 0 bytes_o, rekeying in 15 minutes
         rw{4}:   0.0.0.0/0 === 192.168.0.200/32
user@ipsec-server:~/Desktop$
```

Figure 6: ipsec-server statusall

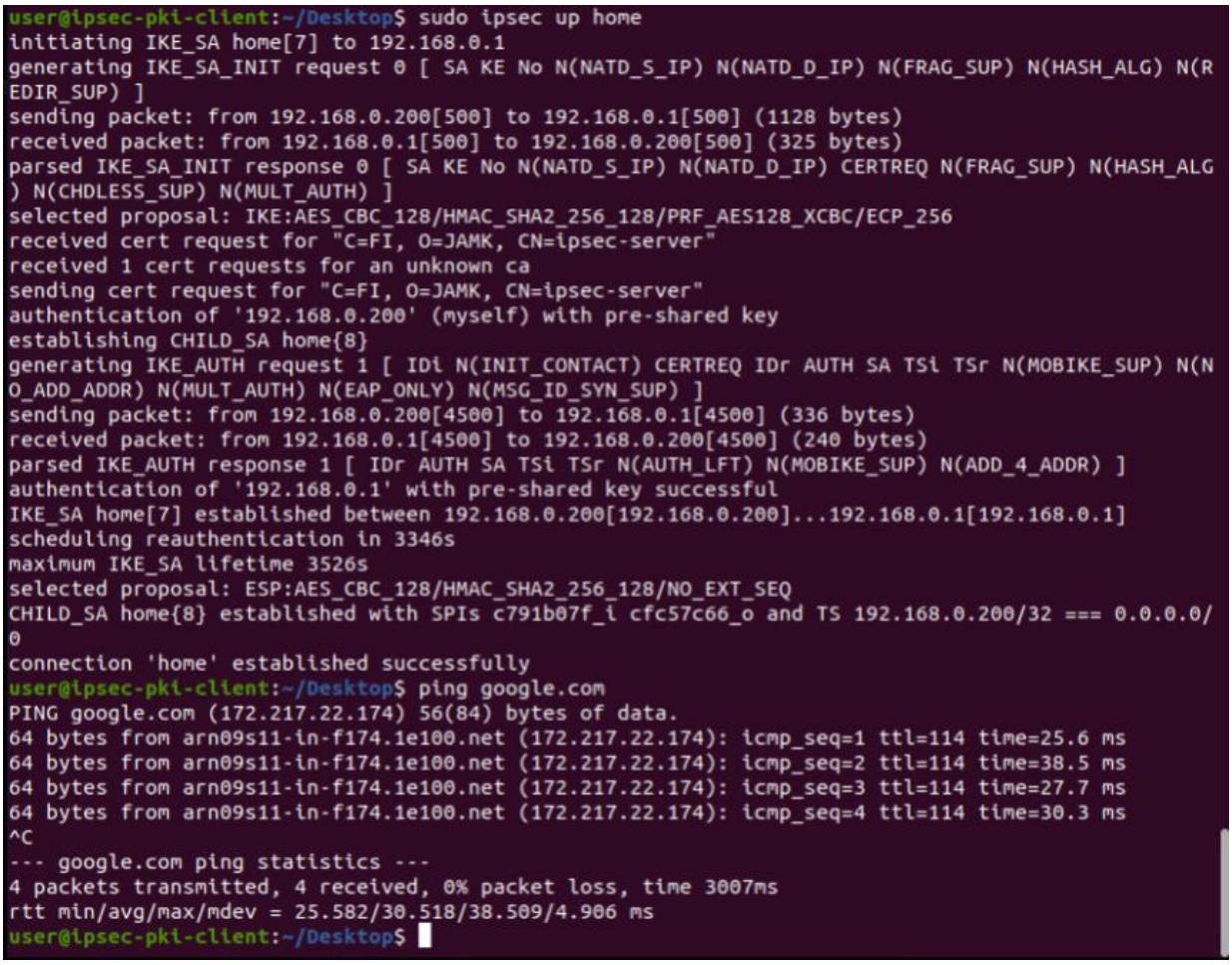


Figure 7: ipsec-client psk connection and ping

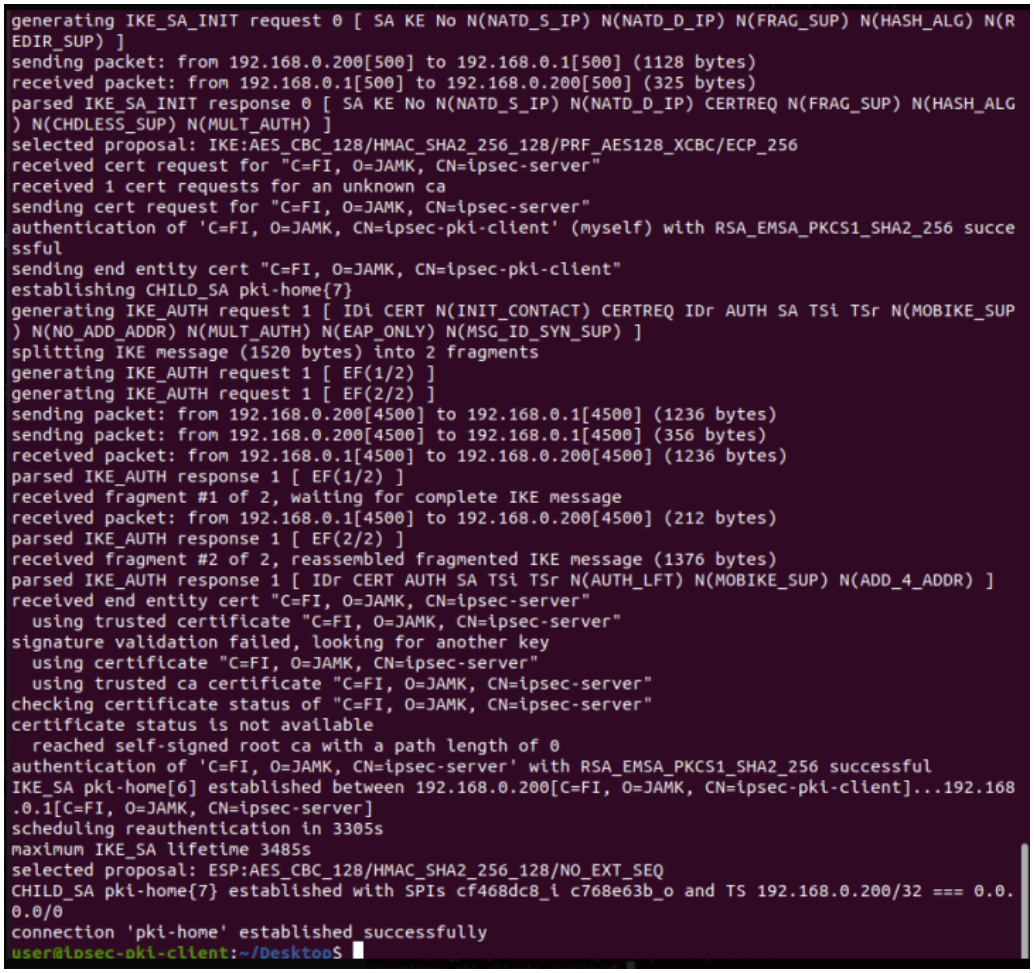


Figure 8: ipsec-client pki connection

yllä oleva pki yhteys fragmentoitui ja siitä tuli niin pitkä, ettei komento mahtunut kuvakaappaukseen, mutta käytimme komentoa ”sudo ipsec up pki-home”

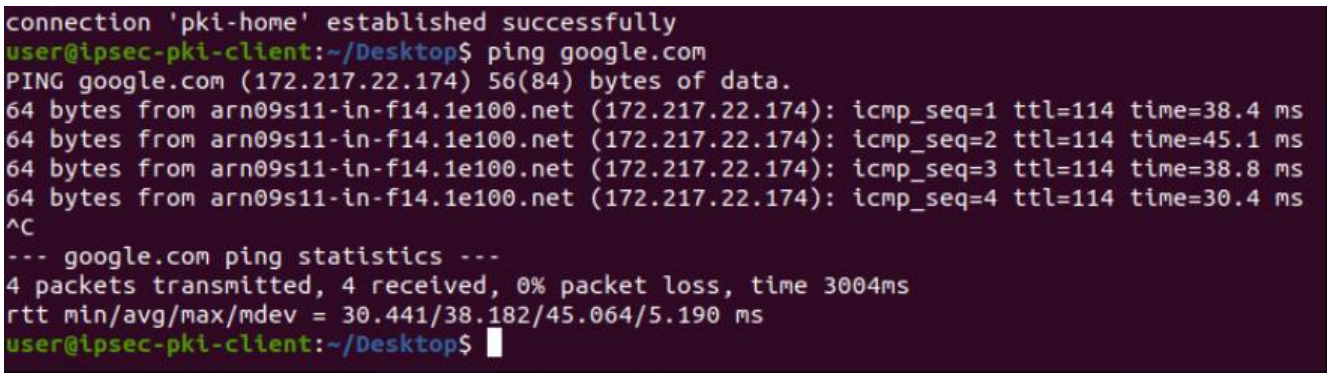


Figure 9: ipsec-client pki ping

Figure 10: ipsec-server sysctl.conf

Ilman sysctl asetuksia client ei voi päästä internettiin. Alla oleva strongswan conffi on oletusasetuksilla eli emme muuttaneet sitä.

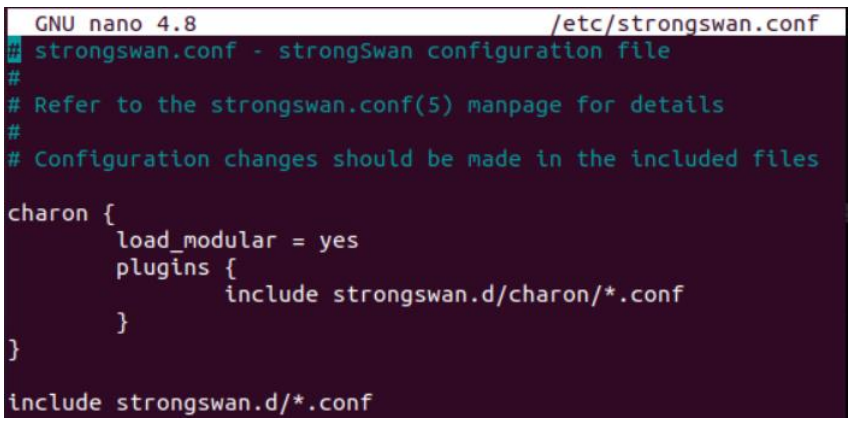


Figure 11: ipsec-server & client strongswan.conf (oletusasetukset)

Emme saaneet yhteyttä internettiin clientillä, ennekuin laitoimme seuraavat palomuurisäännöt.

```
user@ipsec-server:~/Desktop$ sudo iptables -t nat -A POSTROUTING -s 192.168.0.1/
24 -o enp0s8 -j MASQUERADE
user@ipsec-server:~/Desktop$ sudo iptables -t nat -A POSTROUTING -s 192.168.0.1/
24 -o enp0s8 -j MASQUERADE
user@ipsec-server:~/Desktop$
```

Figure 12: iptables sääntöjä

General comments about assignment

- Meni todella paljon aikaa
- Pitkä ja raskas tehtävä
- Meni viime tinkaan koska vei paljon aikaa
- Suomennokset eivät välttämättä ole 100% oikein
- Työnjako oli n. 50/50