LASSI LINTULA

Verkkoliikenteen analysointi ja Wireshark

TIETOJENKÄSITTELYN TUTKINTO-OHJELMA
2024

TIIVISTELMÄ

Lintula Lassi Verkkoliikenteen analysointi ja Wireshark
Opinnäytetyö, AMK
Tradenomi, tietojenkäsittely
Maaliskuu 2024
Sivumäärä: 42

Tämä opinnäytetyö käy läpi verkkoliikenteen kaappausmenetelmiä, verkkopakettien rakennetta ja ohjelmistoa nimeltä Wireshark.

Ensimmäisessä johdannon jälkeisessä luvussa kerrotaan tapoja, joilla kaappausympäristö luodaan yrityksissä, sekä sitä kuka saa ja millä luvalla kaapata verkkoliikennettä. Seuraavassa luvussa kerrotaan, miten verkossa liikkuvien pakettien rakenne koostuu. Mitä nämä paketit sisältävät ja miten ne liikkuvat laitteelta toiselle. Samassa luvussa esitellään myös muutamia yleisimpiä protokollia, joita kyseiset paketit sisältävät.

Viimeisessä luvussa ennen loppusanoja, kerrotaan ohjelmiston nimeltä Wireshark toiminnasta ja kuinka sitä käytetään. Saman luvun lopussa on muutamia esimerkkejä siitä, miten joitakin hyökkäyksiä voidaan havaita käyttämällä kyseistä ohjelmistoa.

Abstract

Lintula Lassi Network traffic analysis and Wireshark
Bachelor's thesis
Bachelor of Business Administration
March 2024
Number of pages: 42

This thesis reviews network traffic capture methods, the structure of network packets and software called Wireshark.

In the first chapter after the introduction, it is explained how the interception environment is created in companies, as well as who can intercept network traffic and with what permission. The next chapter explains the structure of packets traveling in the network. What do these packages contain and how do the transfer of packets from one device to another is done. It is also described how some of the most common protocols work in these packets.

The last chapter before the closing words, tells how the software called Wireshark functions and how to use it. At the end of the same chapter, there are a few examples of how some attacks can be detected by using the software in question.

SISÄLLYS

1 JOHDANTO

Tämä opinnäytetyö käsittelee verkkoliikenteen kaappaamista ja siihen käytettävää ohjelmistoa nimeltä Wireshark. Kuten kaikki tiedämme, internet on iso osa jokapäiväistä elämäämme ja sen käyttö on kasvanut todella paljon 2000-luvulla. Kasvava käyttäjämäärä ja verkon laajuus tuo mukanaan myös paljon erityyppistä verkkoliikennettä ja tämä lisää myös uhkia. Suurissa sisäverkoissa uhkien ja liikenteen tunnistaminen on ehdottoman tärkeää. Tähän tehtävään on kehitetty useita erilaisia tekniikoita ja ohjelmistoja, joilla voidaan kaapata ja analysoida verkossa kulkevia datapaketteja. Datapaketit liikkuvat ympäri maailman käyttäen useita erityyppisiä protokollia ja tulen kertomaan enemmän muutamista yleisimmistä.

Opinnäytetyössäni käyn ensiksi läpi mitä on verkkoliikenteen kaappaaminen ja minkä tyyppisillä fyysisillä ratkaisuilla verkossa liikkuvaa dataa voidaan kaapata analysointia varten. Käyn läpi verkkopakettien rakenteen ja yleisiä protokollia. Lopuksi käsittelen verkkoliikenteen kaappaamista käyttäen yhtä monista tähän tarkoitukseen rakennetuista ohjelmistoista nimeltä Wireshark. Kerron kuinka Wireshark toimii, miten sitä käytetään ja esimerkkejä mitä sillä voidaan tehdä.

2 VERKKOLIIKENTEEN ANALYSOINTI JA KAAPPAAMINEN

Verkkoliikenteen analysoinnilla tarkoitetaan verkkoliikenteen kaappaamista, monitorointia ja analysointia. Se on välttämätöntä muun muassa monissa analyysi- ja rikosteknillisissä tarpeissa, kuten suorituskyvyn seurannassa sekä tietoturvauhkien havaitsemisessa ja tutkimisessa. (Timeless technology, 2022)

Verkkoliikenteen analysoinnin yleisiä käyttötarkoituksia ovat esimerkiksi reaaliaikaisen ja menneen ajan liikenteen kerääminen ja tutkiminen, jotta tiedetään mitä verkossamme tapahtuu tai on tapahtunut. Kerääminen tapahtuu kaappaamalla verkkoliikennettä. Kaapattujen tietojen avulla voidaan havaita haittaohjelmia ja käynnissä olevia hyökkäyksiä. Tällä voidaan myös havaita haavoittuvia protokollia ja sanomia sekä tehdä vianmääritystä siitä miksi verkko toimii hitaasti. (Rapid7, n.d.)

Verkkoliikenteen kaappaamiseen ja analysoimiseen on useita erilaisia ohjelmistoja, joilla jokaisella on oma käyttökohteensa. Tässä työssä kuitenkin keskitytään kuitenkin yhteen käytetyimmistä ohjelmistoista nimeltään Wireshark.

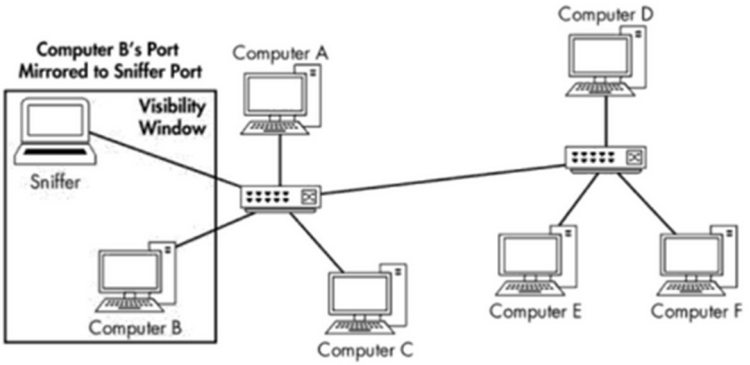
2.1 Verkkoliikenteen kaappaamiseen käytettyjä tekniikoita

Käytännössä liikenteen kaappaamiseen suurissa sisäverkoissa on neljä erityyppistä tapaa, kytkinportin peilaus, keskittimen käyttö (hubbing out), Tap:n (Test access point) käyttö ja ARP-välimuistin myrkytys, englanniksi tätä kutsutaan nimillä: ARP cache poisoning tai ARP spoofing. Verkkoliikennettä voidaan tarkastella myös omassa lähiverkossa siten, että liitetään kone verkkoon ja käynnistetään kaappausohjelma, jolloin oman laitteen liikennettä pystytään tarkkailemaan.

2.1.1 Kytkinportin peilaaminen

Kytkinportin peilaus aikaisemmin mainituista ehkä helpoin tapa kaapata liikennettä sisäverkossa, jossa on kytkin. Tämän tyyppisessä asetelmassa kaappaajalla on oltava pääsy kytkimen hallintapaneeliin komentokehotteen avulla tai verkkoliittymän kautta, jotta halutun portin liikenne voidaan pakottaa kulkemaan myös toisen portin läpi, johon kaappaamiseen käytettävä laite on kytketty. Esimerkiksi jos kohdelaite, jonka liikennettä halutaan kaapata, on kytketty porttiin numero 3 ja kaappaukseen käytetty laite on kytketty porttiin 4, niin portin 3 liikenne pakotetaan kulkemaan myös portin 4 kautta peilaamalla liikenne. (Sanders, 2017, s.21)

Seuraavassa kuvassa näkyy miten kytkentä tehdään, jos käytetään kytkinportin peilaamista liikenteen kaappaamiseen. Kuvassa oleva ”Sniffer” on kaappaukseen käytetty laite ja Computer B:n liikennettä halutaan kaapata.



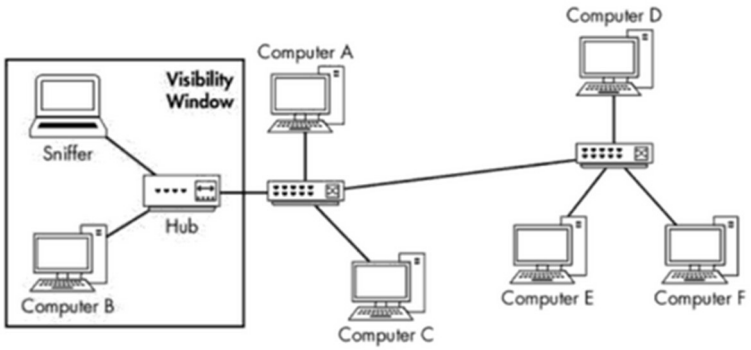
Kuva 1 Kytkinportin peilauskytkentä. (Sanders, 2017, s.21).

On kuitenkin huomattava portin peilaamisessa, että jotkut laitevalmistajat antavat peilata useita portteja yhteen porttiin. Tämä tarkoittaa käytännössä sitä, että jos käytössä on 24-porttinen kytkin, jonka yhden portin siirtonopeus on 100Mbps ja peilaamme 23 molemmin suuntaista liikennettä tukevan portin liikenteen yhteen porttiin. Tällöin tämän portin läpi liikkuu mahdollisesti 4600Mbps, joka on fyysisesti mahdotonta tälle portille ja aiheuttaa siksi pakettien kadottamista tai verkon hidastumista. Tämän tyyppisissä tilanteissa tiedetään kytkinten kadottavan kokonaan ylimääräiset paketit ja tämä saattaa

jopa pysäyttää koko kytkimen toiminnan. Vaikka kytkinportin peilaaminen saattaa kuulostaa helpolta ratkaisulta, ei sitä kuitenkaan kannata käyttää suurissa sisäverkoissa. Parempi vaihtoehto on Tapin käyttö, josta lisää myöhemmin. (Sanders, 2017, s.22)

2.1.2 Keskittimen käyttäminen

Keskittimen (hub) käyttö saattaa olla yksi ainoista keinoista joissakin sisäverkoissa, missä verkossa oleva kytkin ei tue portin peilaamista. Tällä tekniikalla kytkimeen liitetty laite, josta halutaan kaapata paketteja, irroitetaan kytkimestä ja liitetään keskittimeen, joka taas yhdistetään kytkimeen. Kaappaukseen käytettävä kone pitää olla liitetty samaan keskittimeen. Kaikki liikenne mitä liikkuu keskittimen kautta näkyy kaappaajalle, koska keskitin lähettää saapuvat ja lähtevät paketit käyttäen yleislähetysviestiä (broadcast message). (Sanders, 2017, s.23)



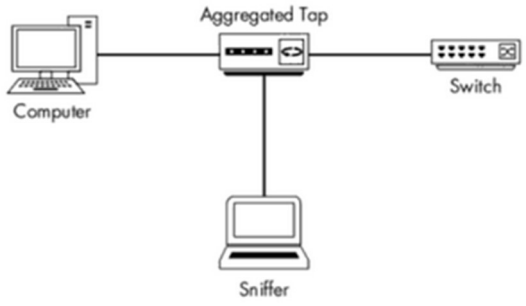
Kuva 2 Keskittimen kytkentä liikenteen kaappausta varten (Sanders, 2017, s.23).

On tiedostettava, että keskittimien käyttö on nykyään todella vähäistä ja tämän tyyppisen laitteen hankinta saattaa olla hankalaa. Usein nykymarkkinoilla myytävistä keskittimistä suurin osa on kytkimiä, joita markkinoidaan keskittiminä. Tämän tyyppiset keskittimet, jotka ovat todellisuudessa kytkimiä eivät tuota aikaisemmin mainittua toimintaa kaappausympäristössä.

2.1.3 Tap:n käyttäminen

Tap:n (Test access point) käyttäminen on lähes samanlaista kuin keskittimen käyttö. Tap on laite mikä kytketään kahden kaapeleilla kytketyn laitteen väliin, jotta voidaan kaapata kyseisen yhteyden läpi liikkuvia paketteja. Ero keskittimen käyttöön on se, että Tap on rakennettu juuri tätä pakettien kaappausta ajatellen. Tap-tyyppejä on kahta erilaista päätyyppiä, aggregoituja ja aggregoimattomia. Erona näissä on se, että aggregoimattomassa on neljä porttia ja aggregoidussa kolme. Tarkoittaen sitä, että aggregoimattomassa tarvitaan molemminpuolisen liikenteen kaappaamiseen kaksi käyttöliittymää ja niihin kytkennät, kun taas aggregoidussa tarvitsemme vain yhden käyttöliittymän. Aggregoitu Tap on helpompi käyttöinen, koska tarvitsemme vain yhden kaappaukseen käytettävän käyttöliittymän ja tavallisen verkkokortin sisältävän laitteen voidaksemme kaapata liikennettä molempiin suuntiin. Aggregoimattomassa on taas oma etuna se, että voimme katsoa toisella käyttöliittymällä liikennettä, joka saapuu ja toisella lähtevää liikennettä. Kun käytämme kahta käyttöliittymää yhdellä kaappauslaitteella, tarvitsee laite verkkokortin, jossa on vähintään kaksi verkkoliitäntää. (Sanders, 2017, s.24-26.)

Seuraavassa kuvassa on aggregoidun Tap:n kytkentä verkossa.



Kuva 3 Aggregoitu Tap. (Sanders, 2017, s.25)

Tap on hyvä verkkoliikenteen kaappaamiseen, koska se ei vaikuta tai hidasta verkkoliikennettä vaan luo verkossa liikkuvista paketeista kopion ja jakaa sen koneelle, jossa kaappausohjelmaa käytetään. Tämä tarkoittaa, että jos laite esimerkiksi vikaantuu niin verkkoliikenne ei kuitenkaan katkea. Tap:sta on

olemassa useita erityyppisiä alatyyppejä eri käyttötarkoituksiin ja erikokoisiin verkkoihin. (Network Critical, n.d.)

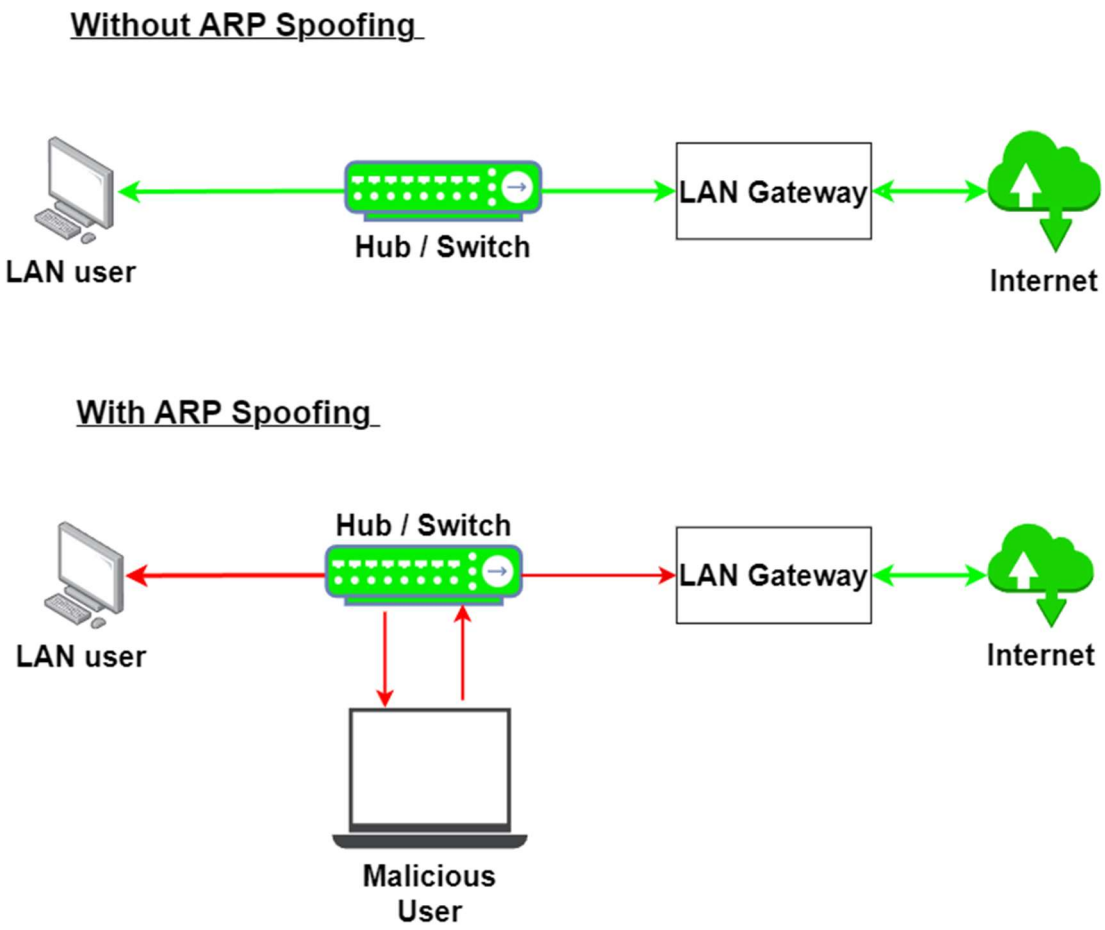
2.1.4 ARP-välimuistin myrkyttäminen

ARP (Address Resolution Protocol) on protokolla, joka toimii OSI-mallin toisessa kerroksessa. Protokollasta itsestään kerron myöhemmin tässä työssä. Otan kuitenkin nyt esiin sen, miten tätä kyseistä protokollaa hyväksi käyttäen voidaan kaapata verkkoliikennettä.

ARP-välimuistiin tallennetaan laitteen IP-osoite ja kyseistä laitetta vastaava fyysinen MAC-osoite. ARP-välimuistin tehtävänä on kertoa laitteelle mikä IP-osoite vastaa mitäkin laitetta paikallisverkossa. Se myös kertoo laitteelle yhdyskäytävän (gateway) IP-osoitteen, jotta laite pystyy lähettämään paketteja verkkoon. (Rouse, 2016).

ARP-välimuistin myrkyttäminen on haavoittuvuus, jota käytetään niin sanotun ”Man in the middle” hyökkäyksen tekemiseen. Hyökkäyksen idea on se, että hyökkääjä naamioi oman laitteensa kertomaan muille lähiverkon laitteille olevansa yhdyskäytävä ja täten kaikki verkkopaketit kulkevat hyökkääjän laitteen läpi. Jos hyökkääjä käyttää kaappausohjelmistoa niin hän voi nähdä kaikki lähiverkossa liikkuvat verkkopaketit ja niiden sisällön omalla koneellaan. (Imperva, n.d.)

ARP-välimuistin myrkyttäminen tapahtuu käytännössä niin että kaappauslaite lähettää kertoo reitittimelle olevansa käyttäjä ja käyttäjälle olevansa reititin, näin ollen käyttäjä lähettää kaikki verkkopaketit kaappaajalle ja kaappaaja lähettää ne eteenpäin reitittimelle ja toisinpäin. Kaappaaja, joka on myrkyttänyt ARP-välimuistin voi tehdä monenlaista kiusaa kyseisessä verkossa kuten palvelunestohyökkäyksiä tai vakoilua. Sitä voidaan myös käyttää tilanteissa, joissa ei ole mahdollista käyttää aikaisemmin mainittuja tapoja liikenteen kaappaamiseen.



Kuva 4 ARP Spoofing / ARP Poisoning Diagram (GeeksforGeeks, 2023).

2.2 Verkkoliikenteen kaappauksen laillisuus

Verkkoliikenteen kaappaaminen ei ole aina laillista ja sen tekemiseen on oltava verkonhaltijan lupa. Liikenteen kaappaajalla on aina rikosoikeudellinen vastuu tekemisistään. Suomen rikoslain luvussa 38 pykälässä 3 viestintäsalaisuuden loukkaus todetaan seuraavasti:

Joka oikeudettomasti
1) avaa toiselle osoitetun kirjeen tai muun suljetun viestin taikka suojauksen murtaen hankkii tiedon sähköisesti tai muulla vastaavalla teknisellä keinolla tallennetusta, ulkopuoliselta suojatusta viestistä taikka

2) hankkii tiedon televerkossa tai tietojärjestelmässä välitettävänä olevan puhelun, sähkeen, tekstin-, kuvan- tai datasiirron taikka muun vastaavan tele-

viestin sisällöstä taikka tällaisen viestin lähettämisestä tai vastaanottamisesta,

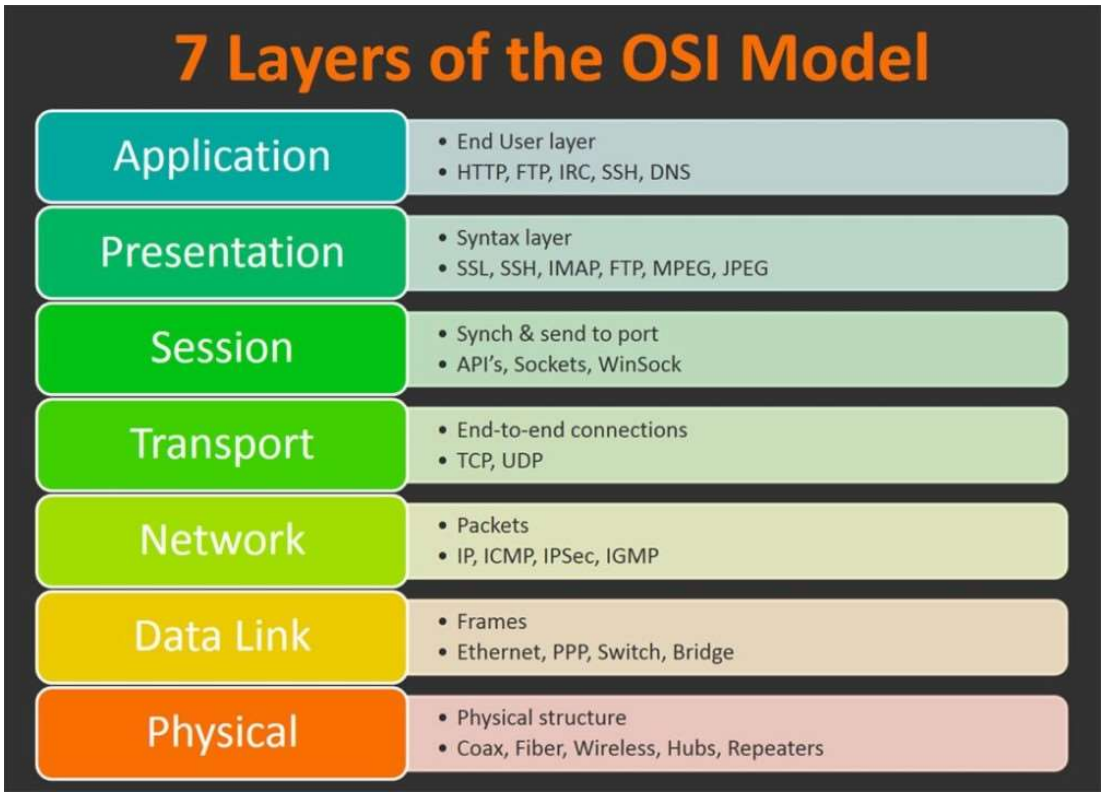
on tuomittava viestintäsalaisuuden loukkauksesta sakkoon tai vankeuteen enintään kahdeksi vuodeksi. Yritys on rangaistava.
(Finlex, 1995, luku 38 § 3.)

3 VERKKOPAKETIN RAKENNE

Kun kaappaamme paketteja verkossa, niin on hyvä ymmärtää miten paketit koostuvat ja miten ne liikkuvat. Tämä kappale kertoo miten nämä paketit rakentuvat.

3.1 OSI-malli

OSI-malli (Open Systems Interconnection) on kansainvälisen standardointiorganisaation, ISO:n (International Organization for Standartization) 80-luvulla kehittelemä malli, jonka ajateltiin ratkaisevan erilaisten verkkotekniikoiden yhteensopivuusongelmat ja tulevaisuudessa yhdistäisi kaikki laitteet toisiinsa. OSI-malli jäi TCP/IP-protokollapinon varjoon käytettävyydessä ja käytön määrässä. OSI-mallia kuitenkin käytetään seitsenkerroksisena niin sanottuna referenssipinona. OSI-malli on referenssiluonteestaan huolimatta verkon ymmärtämisen kannalta hyvin tärkeää. (Anttila, 2001, s.30-31).



Kuva 5 OSI-mallin seitsemän kerrosta. (Jasrotia, 2021)

Sovelluskerroksen (Application Layer) tehtävänä on tarjota verkkopalveluita eri sovelluksille. Esimerkiksi tiedostojen avaus, sulkeminen, kirjoittaminen ja lukeminen, sähköpostin siirtäminen ja etäsuorituksen välittäminen. (Anttila, 2001, s.32).

Esitystapakerroksella (Presentation layer) määritellään siirtyvän datan muoto. Erilaisia esitystapoja ovat esimerkiksi JPEG-kuvakoodaus (Jointed Pictures Expert Group) ja ASCII-merkistö (American Standard Code for Information Interchange). Se toimii myös tulkkina kuten vaikka kääntäen ASCII-merkistöä käyttävän sanoman ISO-8859-1-merkistöksi. (Anttila, 2001, s.33).

Istuntokerroksen (Session layer) tehtävänä on sovellusten toimintojen koordinointi, kuten vaikkapa lähetyksen käynnistäminen ja pysäyttäminen. Istuntokerros huolehtii myös, että data välittyy oikeassa järjestyksessä. Esimerkkejä istuntokerroksen protokollista ovat NFS (Network File System), RPC (Remote Procedure Call) ja NetBIOS (Network Binary Input/Output System). (Anttila, 2001, s.33).

Kuljetuskerroksen (Transport layer) tehtävänä on ylemmiltä kerroksillta tulevan datan pilkkominen sopivan kokoisiin palasiin eli segmentteihin ja niiden välitys vastaanottajalle. Se voi toimia joko yhteydellisesti, jolloin kommunikoivat laitteet muodostavat yhteyden ennen lähetystä, tai yhteydettömänä, jolloin data lähetetään verkkoon toivoen, että se menee perille. Esimerkkejä kuljetuskerroksen protokollista ovat TCP (Transmission Control Protocol) ja UDP (User Datagram Protocol). (Anttila, 2001, s.34).

Verkkokerroksen (Network layer) tehtävänä on pakata kuljetuskerrokselta tuleva data paketteihin ja välittää ne vastaanottajan verkkokerroksen osoitteeseen. Tätä prosessia kutsutaan reitittämiseksi ja se on tyypillisesti riippumaton aiempien kerrosten tekniikoista. Välitys voi tapahtua lähiverkon sisällä tai siihen voi liittyä useampia reitittimiä. Esimerkkiprotokollia ovat IP (Internet protocol) ja IPX (Internetwork Packet Exchange). (Anttila, 2001, s.34).

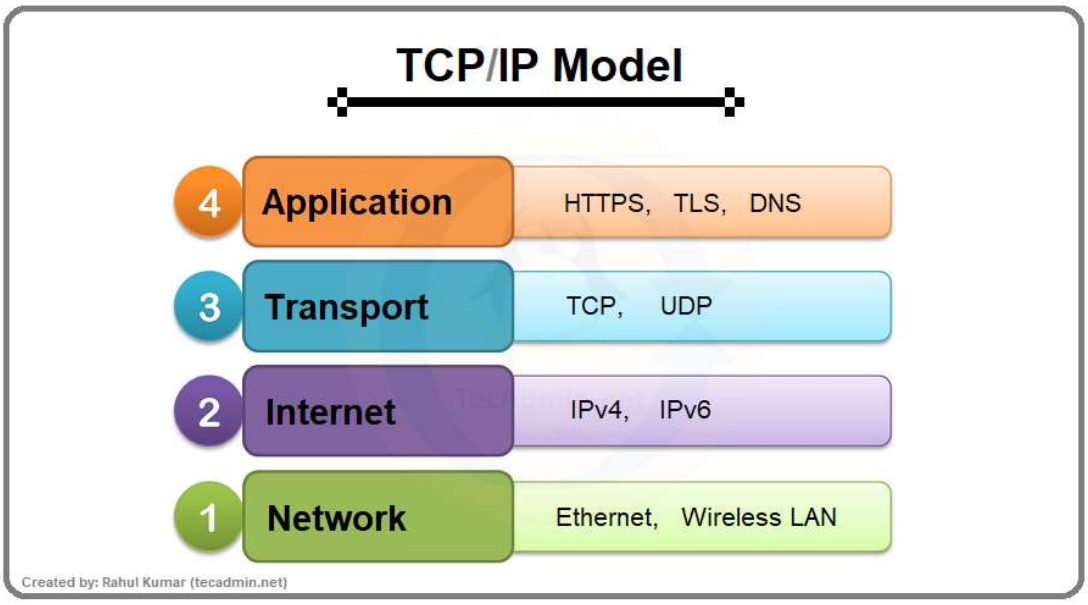
Siirtokerroksen (Data Link layer) tehtävänä on rakentaa kehys, johon pakataan verkkokerrokselta saatu data. Siirtokerros lisää kehykseen otsikot, joihin kuuluu muun muassa vastaanottajan ja lähettäjän siirtokerros osoite. Esimerkkejä siirtokerroksen kehystavoista ovat HDLC (High-Level Data Link Control) ja IEEE:n (Institute of Electrical and Electronics Engineers) lähiverkkoihin määrittelemät kaksi osaa; LLC (Logical Link Control) ja MAC (Media Access Control). Siirtokerros on yleensä voimakkaasti riippuvainen fyysisestä kerroksesta. (Anttila, 2001, s.34).

Fyysinen kerros (Physical layer) määrittelee kullekin verkkotekniikalle ominaisia bittien välittämiseen liittyviä asioita. Näitä ovat muun muassa liittimien tyyppi, käytettävät koodausmenetelmät ja jännitetasot. (Anttila, 2001, s.34).

3.2 TCP/IP-malli

Kuten aikaisemmin todettiin TCP/IP-mallista tuli käytetympi kuin OSI-mallista, vaikka edelleen OSI-mallia käytetään referenssinä asioiden ymmärtämiseen.

Tämä johtui siitä, että laitteet ja ohjelmistot poikkesivat toisistaan ja olivat myyjäkohtaisia. Tarvittiin yleinen protokolla eri valmistajien laitteiden väliseen kommunikaatioon. Tämä johti TCP:n ja IP:n luomiseen. Myös kasvava laitevaatimusten määrä ajoi useiden muidenkin protokollien luomiseen. Tämä johti uuteen referenssimalliin nimeltä TCP/IP. TCP/IP sisältää vain neljä kerrosta, jotka ovat yhdistelmiä OSI-mallin kerroksista. (Shanmugan ym., 2002, s.21).



Kuva 6 TCP/IP-malli. (Rahul, 2023).

3.2.1 Sovelluskerros

Sovelluskerros (Application Layer) antaa käyttäjälle pääsyn verkkoon antamalla käyttöön useita palveluita, kuten esimerkiksi SMTP:n (Simple Mail Transfer Protocol) sähköpostien lähettämiseen tai FTP:n (File Transfer Protocol) tiedostojen lähettämiseen. Myös yleiset verkkoselaukseen liittyvät protokollat kuten esimerkiksi HTTP (Hypertext Transfer Protocol) toimivat tässä kerroksessa. (Shanmugan ym., 2002, s.22).

3.2.2 Siirtokerros

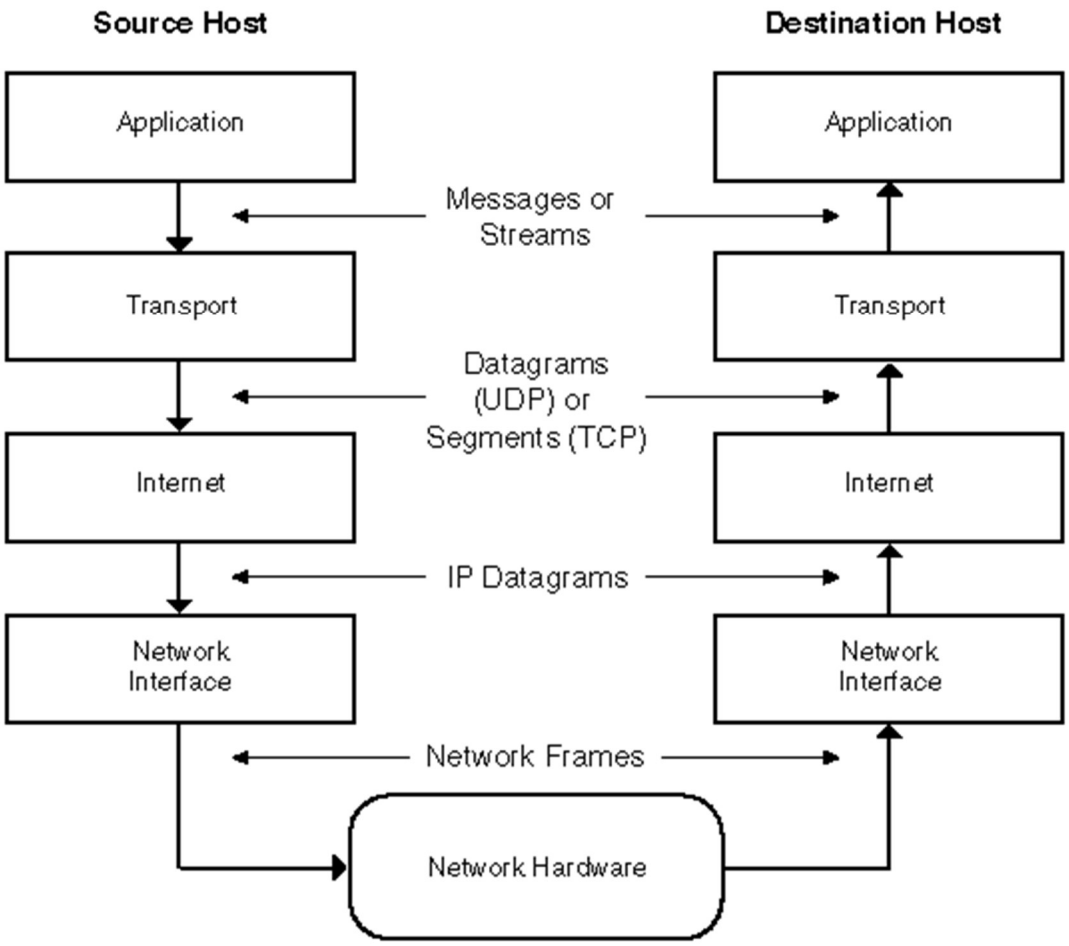
Siirtokerros (Transport layer) käsittelee tietokoneiden välistä kommunikaatiota käyttäen esimerkiksi TCP- (Transmission Control Protocol) tai UDP- (User Datagram Protocol) protokollia hyödykseen. Tämä kerros jakaa datan loogisiksi yksiköiksi, joita kutsutaan paketeiksi, ennen niiden lähettämistä. (Shanmugan ym., 2002, s.22).

3.2.3 Verkkokerros

Verkkokerros (Internet layer) on vastuussa datapakettien reitittämisestä oikeaan osoitteeseen käyttäen IP-protokollaa, joka on vastuussa datan päätymisestä oikeaan osoitteeseen. Protokolla ei kuitenkaan ole vastuussa datan eheydestä. IP on vuorovaikutuksessa ARP:n (Address Resolution Protocol) ja RARP:n (Reverse Address Resolution Protocol) kanssa selvittääkseen oikean osoitteen. IP toimii ARP:n kanssa selvittääkseen MAC- (Media Access Control) osoitteen ja siirtoyhteyskerros käyttää tätä osoitetta välittääkseen datan fyysistä linjaa pitkin. (Shanmugan ym., 2002, s.22).

3.2.4 Siirtoyhteyskerros

Siirtoyhteyskerros (Network Interface layer) on OSI-mallin siirtokerroksen ja fyysisen kerroksen yhdistelmä. Tämä kerros on vastuussa datan jakamisesta loogisiin joukkoihin, joita kutsutaan kehyksiksi. Riippuen siitä millaista yhteyttä käytetään, tämä kerros lisää kehyksiin oikeat otsikot. Otsikko voi sisältää kehysten määrän ja niiden oikean järjestyksen ja miten kehykset pitää järjestää päätepisteessä. Vastaanottavassa päässä tämä kerros kokoaa kaikki kehykset ja lähettää ne ylemmille kerroksille. Tämä kerros takaa sen, että kaikki kehykset vastaanotetaan oikein ja oikeassa järjestyksessä. (Shanmugan ym., 2002, s.22).



Kuva 7 Miten paketti kulkee lähteestä määränpäähän (Novell n.d.).

3.3 Yleisiä verkkoprotokollia

Verkossa on useita erilaisia protokollia. Verkkoliikenteen analysoinnin kannalta on tärkeää ymmärtää miten yleisimmät protokollat toimivat, jotta liikenteen tutkiminen olisi sujuvampaa, koska pelkän protokollan avulla pystytään tietämään minkä tyyppisestä liikenteestä on kyse. Alaluvuissa avataan näistä muutamia yleisimpiä.

3.3.1 HTTP (Hypertext Transfer Protocol)

HTTP on sovelluskerroksen protokolla. HTTP-pyyntö on tapa, jolla esimerkiksi verkkoselain pyytää tietoja, joita se tarvitsee verkkosivun lataamiseen. Jokainen HTTP-pyyntö, joka tehdään, kuljettaa mukanaan sarjan koodattua dataa, joka kantaa mukanaan erityyppisiä tietoja. Yleisimmät HTTP-metodit

ovat ”GET” ja ”POST”. GET-metodi pyytää tietoja käyttäjälle yleensä esimerkiksi verkkosivulta, kun taas POST-metodi lähettää tietoja palvelimelle kuten esimerkiksi käyttäjänimen ja salasanan. HTTP:n tilakoodit ovat kolmenumeroisia koodeja, jotka kertovat onnistuiko haluttu HTTP-pyyntö. Numerolla 2 alkavat koodit kertovat, että pyyntö onnistui. Esimerkiksi jos käyttäjä pyytää verkkosivua, niin useimmiten saadaan takaisin koodi ”200 OK” tarkoittaen, että pyyntö onnistui. Numeroilla 4 ja 5 alkavat koodit kertovat virheistä. Numerolla 4 alkavat kertovat käyttäjäpuolen virheestä kuten virheellisestä verkko-osoitteesta, kun taas numerolla 5 alkavat koodit kertovat palvelinpuolen ongelmasta. (Cloudflare, n.d.-b)

3.3.2 TLS (Transport Layer Security)

TLS-salausprotokolla on liikenteen suojaukseen käytetty protokolla, joka auttaa suojaamaan tiedot käyttäjän laitteen ja verkkosivun välillä. Se käyttää salaukseen symmetristä ja epäsymmetristä salausta. Symmetrisessä salauksessa käytetään jaettua salaista avainta ja epäsymmetrisessä julkista ja yksityistä avainta. TLS-salauksessa tiedot salataan ensin symmetrisellä avaimella, jonka jälkeen symmetrinen avain salataan vastaanottajan epäsymmetrisellä julkisella avaimella. Vastaanottaja voi sitten purkaa symmetrisen avaimen salauksen yksityisellä avaimellaan. Tällä prosessilla varmistetaan, että vain tarkoitettu vastaanottaja voi lukea tiedot. Verkkosivustot, jotka käyttävät tätä salausta, toimivat HTTPS-protokollalla. Eli jos osoiterivillä ensimmäisenä lukee HTTPS:// tiedät että kyseinen sivusto käyttää tätä protokollaa tai jotakin vastaavaa tietojen salaamiseen. (Internetopas, 2022)

3.3.3 DNS (Domain Name System)

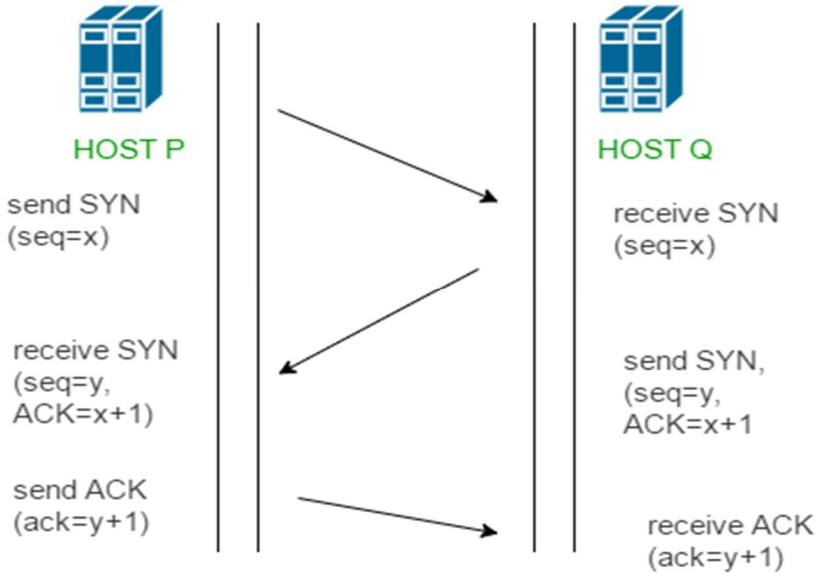
DNS on järjestelmä, jonka tehtävänä on muuntaa verkkotunnus IP-osoitteeksi, jotta käyttäjän ei tarvitse tietää IP-osoitetta vaan pystyy löytämään sivuston pelkän verkkotunnuksen avulla. Tämä järjestelmä koostuu useasta palvelimesta, joissa sijaitsevat tiedot IP-osoitteista, joiden avulla

verkkosivut saadaan haettua käyttäjän selaimelle. DNS-kysely on käyttäjälle huomaamaton. Kun verkkosivun osoite, esimerkiksi www.esimerkki.com, kirjoitetaan selaimeen, lähtee tämä osoite ensiksi rekursiiviselle DNS-palvelimelle, joka on yleensä palveluntarjoajan palvelin. Jos tällä palvelimella ei ole välimuistissan kyseisen verkkotunnuksen IP-osoitetta, siirtyy kysely juurinimipalvelimelle, joka selvittää missä esimerkiksi .com osoitteet sijaitsevat. Juurinimipalvelin lähettää tämän jälkeen kyselyn ylätasonpalvelimelle, jossa .com verkkotunnukset sijaitsevat. Tämän jälkeen ylätasonpalvelin lähettää kyselyn valtuutetulle palvelimelle, missä sijaitsee esimerkki.com ja selvittää tämän IP-osoitteen ja palauttaa sen käyttäjälle. Käyttäjällä on nyt pääsy halutulle verkkosivulle. (Cloudflare, n.d.-a)

3.3.4 TCP (Transmission Control Protocol)

TCP-verkkoprotokollaa käytetään, kun siirretään dataa internetissä. TCP on suosituin verkkoprotokolla luotettavuutensa ansiosta. TCP on yhteyspohjainen eli se muodostaa yhteyden lähettäjän ja vastaanottajan välille ja pitää sitä yllä, kun dataa lähetetään ja vastaanotetaan. Se on luotettava siksi, koska kun dataa vastaanottava osapuoli saa paketin niin siitä lähetetään takaisin kuittaus yhteydenmuodostuspaketin saapumisesta. Jos kuittaus ei saavu takaisin lähettävälle osapuolelle niin sama paketti lähetetään uudestaan, kunnes kuittaus saapuu. Tämä takaa, että kaikki paketit saapuvat perille, joten dataa ei katoa siirron yhteydessä. Tätä protokollaa käytetään esimerkiksi verkonselaamiseen ja sähköpostien lähetykseen ja kaikkeen missä halutaan varmistaa, että kaikki data saapuu perille. (Zieniūtė, 2022)

TCP luo lähettäjän ja vastaanottajan välille yhteyden käyttämällä kolmivaiheista kättelyä. Tämä kättely tapahtuu niin, että lähettävä osapuoli lähettää vastaanottajalle SYN (synchronization) viestin, johon vastaanottaja vastaa SYN/ACK (synchronization/acknowledgement) viestillä, jos aikaisempi viesti on päässyt perille. Tämän jälkeen lähettävä osapuoli lähettää vastaanottajalle ACK viestin ja yhteys on täten muodostettu. (Browne, 2023).



Kuva 8 TCP SYN/ACK viestien lähetys (GeeksforGeeks, 2022).

3.3.5 UDP (User Datagram Protocol)

UDP on myös paljon käytetty protokolla. Tässä protokollassa tärkein ominaisuus on sen nopeus. UDP on yhteydetön protokolla. Se ei muodosta yhteyttä lähettäjän ja vastaanottajan välille kuten TCP, joten se ei tarvitse kuittausta saapuneesta paketista. Lähettävän osapuolen ei tarvitse odottaa vaan voi lähettää seuraavan paketin heti. Tämän takia jotkut paketit eivät välttämättä saavu perille koskaan. Tätä protokollaa käytetään usein suoratoistamisessa tai videopelien pelaamisessa, missä pakettien siirtonopeus on tärkeämpää kuin se että saapuvatko kaikki paketit perille. (Zieniūtė, 2022)

3.3.6 IP (Internet Protocol)

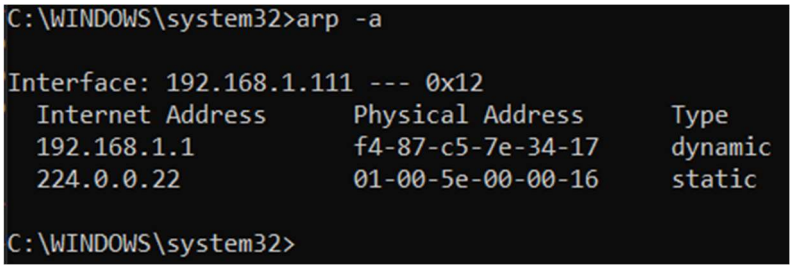
IP on protokolla, jonka tehtävänä on reitittää ja antaa osoitteita datapaketeille, jotka liikkuvat ympäri verkkoa. IP-tiedot lisätään jokaiseen pakettiin ja tämä tieto auttaa reitittimiä lähettämään ne oikeaan paikkaan. Jokainen verkossa oleva laite saa oman IP-osoitteen. Verkkopakettiin liitetty IP-osoite ohjaa paketin siihen IP-osoitteeseen, johon se on tarkoitettu. Kun paketti saa-

puu osoitteeseensa, sitä käsitellään riippuen siitä mitä siirtokerroksen protokollaa on käytetty. Kuten esimerkiksi TCP- tai UDP-protokollaa. Jokainen IP-osoite on sarja merkkejä, kuten esimerkiksi: ”192.168.1.1”. IP-osoitteita on olemassa kahta käytössä olevaa versiota, IPv4 ja IPv6, joista aikaisempi näyttää samalta kuin edeltävässä esimerkissä ja jälkimmäinen näyttää tältä: ”2001:0db8:85a3:0000:0000:8a2e:0370:7334”. IPv4-osoitteet ovat kehitetty 1983 ja silloin ei vielä tiedetty kuinka paljon osoitteita tarvitaan, joten kun huomattiin että osoitteet ovat käymässä vähiin kehitettiin uudempi enemmän merkkejä sisältävä IPv6, joka mahdollistaa enemmän uniikkeja osoitteita. IP-otsikko lisätään datapakettiin ennen kuin se lähetetään. IP-otsikko sisältää useita erilaisia tietoja paketista, kuten lähettäjän ja vastaanottajan IP-osoitteet, otsikon pituuden ja paketin koon bitteinä ja niin sanotun TTL (Time To Live) arvon, joka tarkoittaa kuinka monta ”hyppäystä” paketti voi tehdä verkossa ennen kuin se kadotetaan. Se sisältää myös tiedon siitä mitä siirtokerroksen protokollaa käytetään, kuten TCP tai UDP. (Cloudflare, n.d.-c)

3.3.7 ARP (Address Resolution Protocol)

ARP on protokolla, joka yhdistää alati muuttuvan IP-osoitteen laitteen fyysiseen osoitteeseen, jota kutsutaan nimellä MAC-osoite (Media Access Controll). Tämä proseduuri on tärkeä, koska MAC-osoite ja IP-osoite ovat eri pituisia. Usein käytetyt IPv4-osoitteet (Internet Protocol versio 4) ovat pituudeltaan 32 bittiä kun taas fyysisten laitteiden MAC-osoitteet ovat 48 bittiä pitkiä.
ARP toimii muuntaen 32-bittiä pitkän osoitteen 48-bittiseksi ja toisinpäin. Suurin osa käyttöjärjestelmistä tallentaa nämä osoitteet niin sanottuun ARP-tauluun. (Fortinet, n.d.)

ARP-taulusta voimme selvittää mikä IP-osoite on tallennettu millekin MAC-osoitteelle. Seuraavassa kuvassa näemme ARP-taulun Windows koneella ja komennon sen saamiseksi esiin.



Kuva 9 ARP-taulu sekä komento ”arp -a”, jolla saamme kyseisen taulun Windows käyttöjärjestelmällä näkyviin.

Kuvassa näkyvä osoite 192.168.1.111 tarkoittaa tätä tietokonetta. Osoite 192.168.1.1 tarkoittaa IP-osoitetta reitittimelle ja sen perässä on MAC-osoite. 224.0.0.22 on IGMP-osoite (Internet Group Management Protocol), joka on oma protokollansa paikallisverkkoliikenteen ohjaukseen.

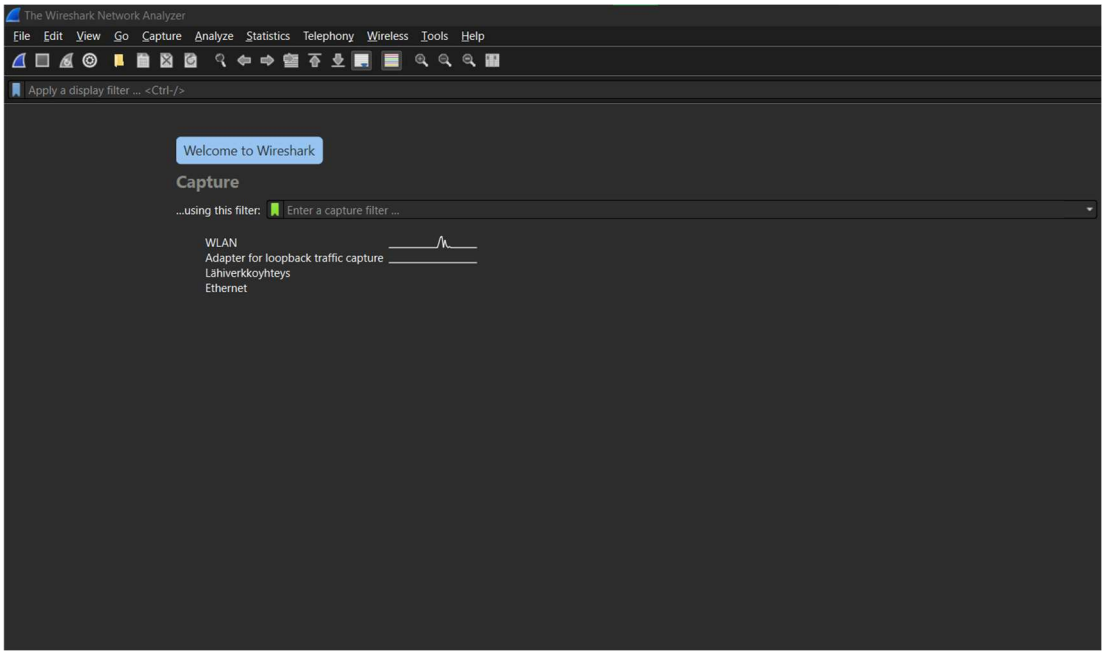
4 WIRESHARK

Wireshark, entiseltä nimeltään Ethereal, on tehokas verkon analysointiin keskittyvä työkalu. Tällä työkalulla voi kaapata tai ”haistella” verkkoliikennettä. Kaappauksen jälkeen kaapatut paketit voidaan analysoida, vaikka siksi, että etsitään hitaita sovelluksia, epäilyttävää liikennettä tai laiterikkoja.

Wireshark on avoimen lähdekoodin ohjelmisto, joten sen voi kuka tahansa ladata sen omilta kotisivuilta. Wiresharkin asennus on suhteellisen helppoa Windows-koneella, koska se tapahtuu lähes samalla tavalla kuin minkä tahansa muun ohjelmiston asennus. Asennuksen yhteydessä on kuitenkin muistettava sallia WinPcapin asennus, jotta reaaliaikainen pakettien kaappaus onnistuu. Wireshark kaappaustiedostot ovat nykyään normaalisti pcapng formaatissa, mutta Wiresharkilla on mahdollista lukea myös muissa formaateissa olevia tiedostoja kuten esimerkiksi cap. Käytän esimerkkikuvissani omalla koneellani Wiresharkista ottamiani kuvakaappauksia käyttäen omassa sisäverkossani liikkuvaa dataa sekä tiedostoja, joita olen ladannut

sivustolta, joka niitä jakaa tutkimuskäyttöön. Linkki sivustoon on seuraava: https://wiki.wireshark.org/SampleCaptures.

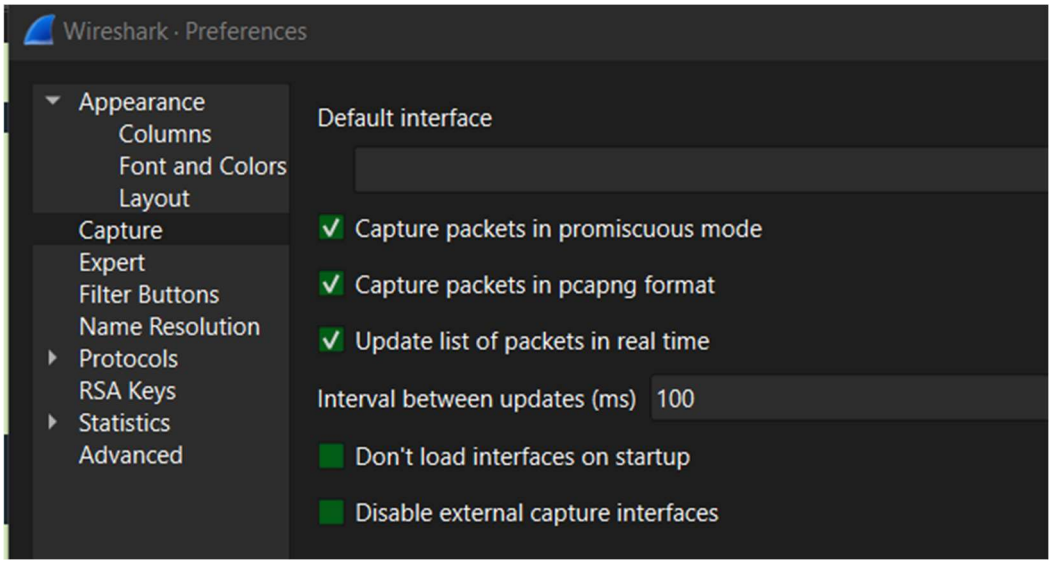
Kun Wireshark käynnistetään, avautuu seuraavassa kuvassa näkyvä ikkuna.



Kuva 10 Wireshark aloitusikkuna.

Ikkunassa näkyy käytössä olevat verkot ja verkon nimen vieressä näkyvä viiva kertoo liikenteestä kyseisessä verkossa. Kuten kuvassa näkyvä WLAN ja sen jälkeen viiva, missä on pieni hyppy, tarkoittaa, että WLAN yhteys on käytössä ja siellä on jotain liikennettä. Näkymä saattaa poiketa kuvasta hieman eri laitteilla ja asetuksilla.

Kun kaappaus ollaan aloittamassa, kannattaa käydä laittamassa niin sanottu ”promiscuous mode” päälle. Tämä asetus takaa sen, että Wireshark löytää kaikki verkkopaketit mitä verkkokortille saapuu.



Kuva 11 Asetus mistä saadaan "promiscuous mode päälle".

4.1 Pakettien kaappausnäkymä

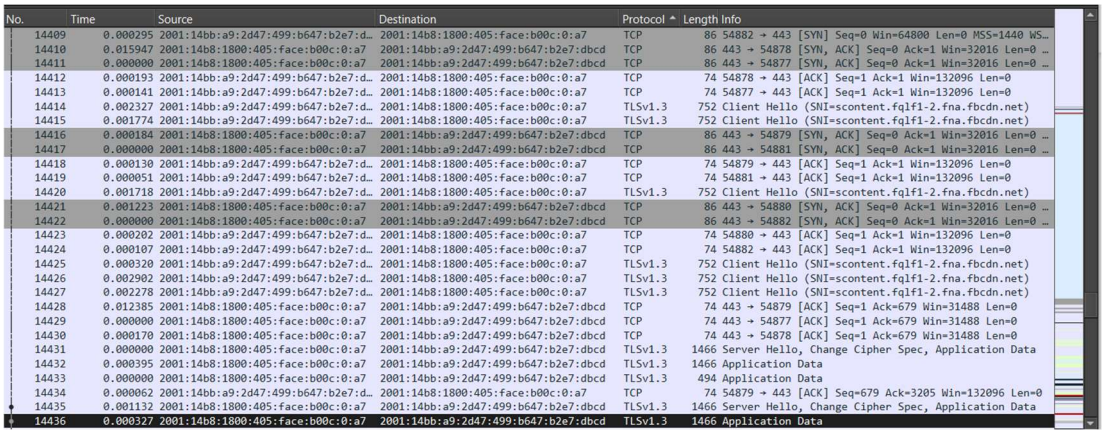
Pakettien kaappaus aloitetaan painamalla haluttua yhteyttä kaksi kertaa, valitsemalla haluttu yhteys ja napsauttamalla ylhäällä vasemmalla olevaa sinistä nappulaa tai painamalla capture ja start. Wireshark aloittaa pakettien kaappaamisen heti ja jatkaa sitä, kunnes kaappaaminen keskeytetään.

Pakettien kaappaus on reaaliaikaista ja aluksi näkymä saattaa olla hankalaa tulkita ja sekavaa, koska paketit liikkuvat reaaliajassa pakettilistaruudussa. On järkevää tehdä kaappaus ja tallentaa kaapattu osuus ennen pakettien analysointia. Kaappausnäkymässä on oletuksena kolme eri ruutua.

4.1.1 Pakettilistaruutu.

Pakettilistaruudussa oleva näkymä liikkuu koko ajan kaappaushetkellä, koska paketteja lähetetään tai niitä saapuu. Tämä näkymä koostuu riveistä ja sarakkeista, joista jokainen rivi on yksi verkkopaketti ja jokaisella sarakkeella näkyy eri tietoja kyseisestä paketista, kuten järjestysnumero, lähteen ja määränpään IP-osoitteet, käytetty protokolla, paketin koko tavuina ja lisäinfoa kuten porttinumero tai vaikka HTTP-protokollan paluuarvo (esimerkiksi: 200 OK). Sarakkeita pystytään lisäämään ja poistamaan. Voidaan lisätä esimerk-

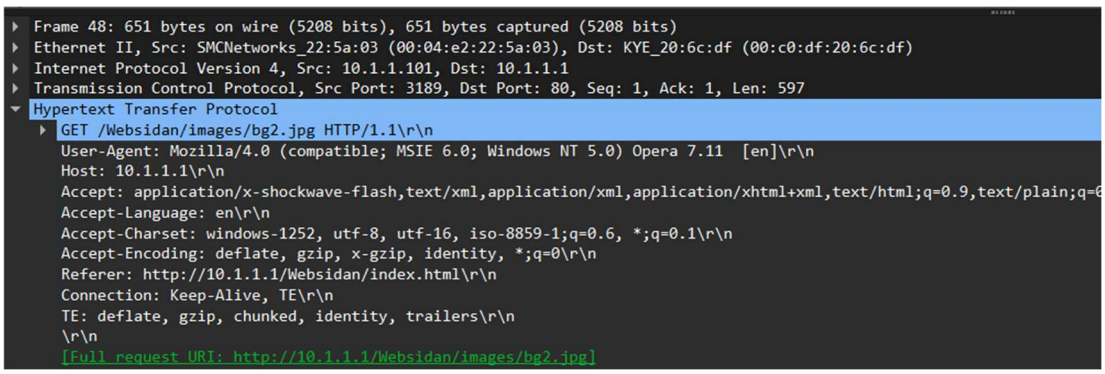
kinä sarake ”host”, joka näyttää pakettiin mahdollisesti liitetyn verkkotunnuksen esimerkiksi: www.wireshark.org tai voimme lisätä sarakkeen, joka näyttää paketin TTL-arvon.



Kuva 12 Pakettilistaruutu.

4.1.2 Paketin tiedot ruutu

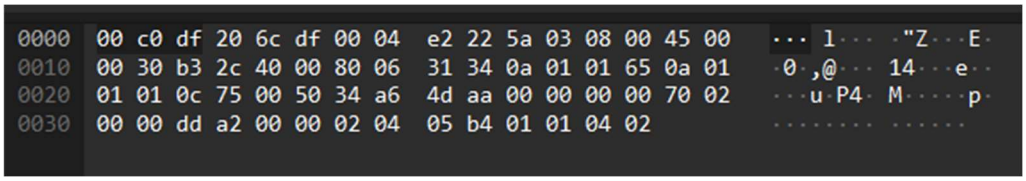
Paketin tiedot ruutu näyttää tarkemmin kaikki paketin sisältämät tiedot.
Kun katsotaan tätä ruutua, voidaan nopeasti huomata, että se noudattaa ja näyttää sisällön samoin kuin TCP/IP-mallissa.



Kuva 13 Paketin tiedot ruutu. HTTP-paketti, jonka sisällä on jpg-kuva.

4.1.3 Paketin tavut-ruutu.

Paketin tavut-ruudussa nähdään kyseisen paketin sisältö HEXA-koodina ja tavuina.



Kuva 14 Paketin tavut-ruutu, jossa on nähtävissä TCP-synkronisointipyynnön tavut.

4.2 Pakettien suodatus

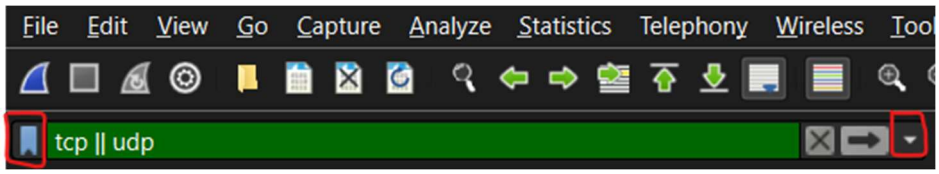
Kaapattuja paketteja on kuitenkin yleensä melkoinen määrä, joten tarkastelua varten on järkevää suodattaa paketteja, jotta nähdään esimerkiksi pelkkää TCP-liikennettä tai muuta vastaavaa.

Suodattimen tai ”filtterin” voi kirjoittaa ylhäällä olevaan kenttään. Oletuksena Wireshark maalaa kentän punaiseksi, silloin kun suodatin ei ole toimiva. Keltaiseksi, jolloin Wireshark ei osaa tunnistaa suodatinta oikeaksi, mutta se saattaa toimia silti tai vihreäksi, jolloin se on toimiva suodatin.

Suodattimien kirjottaminen noudattaa ohjelmoinnistakin tuttuja operaattoreita kuten suurempi kuin (>), yhtäsuuri (==), pienempi kuin (<) ja kuten AND (&&), OR (||) ja NOT (!). Esimerkkinä jos kirjoitamme suodattimeksi: ”tcp || udp” niin silloin Wireshark näyttää pelkästään TCP- tai UDP-liikennettä. Suodattimessa ei ole pakko käyttää edellä mainittuja merkkejä kuten ” || ”, vaan on mahdollista myös käyttää kirjoitettuja sanoja pienillä kirjaimilla kirjoitettuna, esimerkiksi ” tcp or udp”.

Suodattimien tekeminen valmiiksi on myös mahdollista ja se on suotavaa, koska se nopeuttaa työskentelyä huomattavasti. Suodattimet voidaan tallentaa suodatinrivin vasemmassa reunassa olevasta nappulasta, josta löytyy myös jo tallennetut suodattimet. Wireshark myös tallentaa viimeksi käytetyt

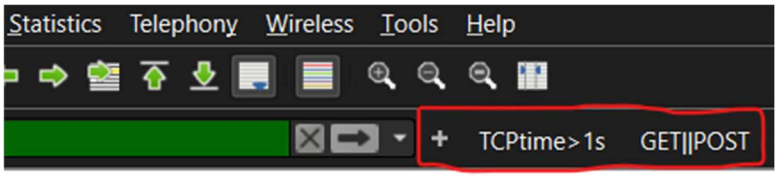
suodattimet välimuistiinsa, joten jos halutaan nopeasti käyttää aikaisempaa suodatinta, voidaan se valita suodatin rivin oikeasta reunasta löytyvästä alasvetolaatikosta. Suodattimet tallentuvat profiiliin.



Kuva 15 Suodatin ja tallennetut sekä viimeisimmät suodattimet.

Suodatinta voidaan käyttää myös esimerkiksi siihen, jos halutaan katsoa tietyn IP-osoitteen liikennettä kirjottamalla suodatinriville: ”ip.addr == (IP-osoite kuten 192.168.1.1)”. On myös mahdollista etsiä sisältääkö paketti jonkin tietyn merkkijonon tai osan merkkijonoa kirjoittamalla esimerkiksi: ”http contains "https://www.wireshark.org"". Tällä suodattimella etsitään pakettia, jossa on HTTP-sanoma, johon on liitetty kyseinen osoite.

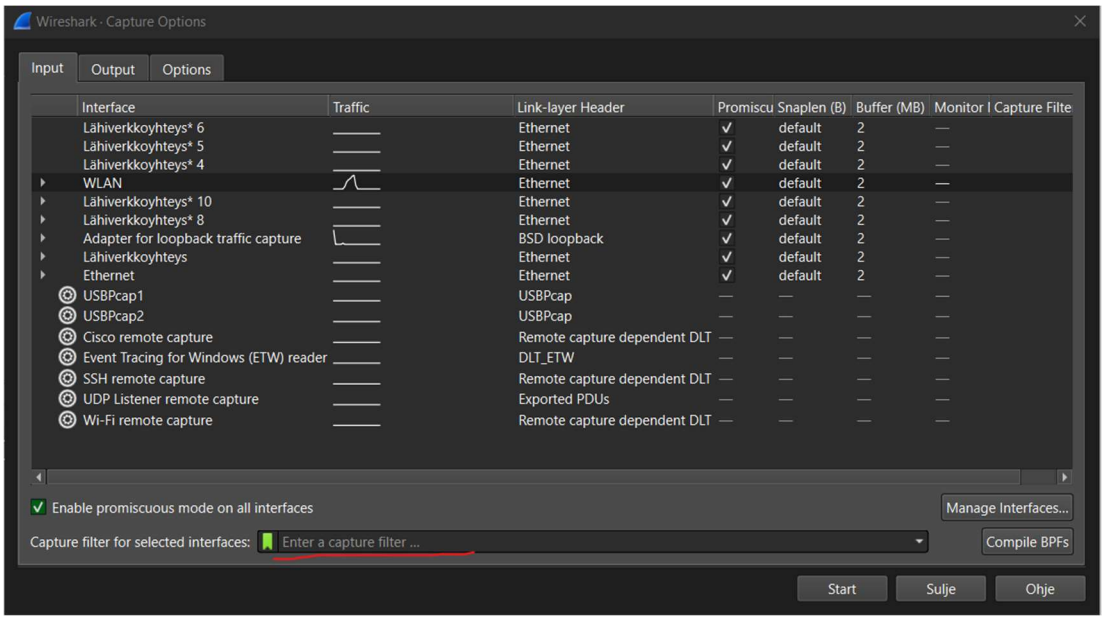
Suodattimia on myös mahdollista tallentaa niin, että luodaan valmis painike suodattimelle, jolloin saadaan yhdellä klikkauksella nopeasti suodatettua liikennettä. Seuraavassa kuvassa näemme kaksi nappulaa, joista ensimmäisestä painettaessa saamme katsottua kaikki paketit, joiden lähetys tai saapuminen on kestänyt yli sekunnin, ja toisesta painettaessa näemme kaikki http-protokollan GET- tai POST-kutsut.



Kuva 16 Suodatinnappuloita ja painike "+", josta niitä voi tehdä.

Suodattimen voi myös asettaa ennen pakettien kaappausta, jolloin kaappaustiedostoon ei tallennu kuin halutun suodatusehdon täyttäviä paketteja. Kaappaussuodattimen voi asettaa Wiresharkin aloitussivulla suoraan tai painamalla ylhäältä Capture, jonka jälkeen Options.

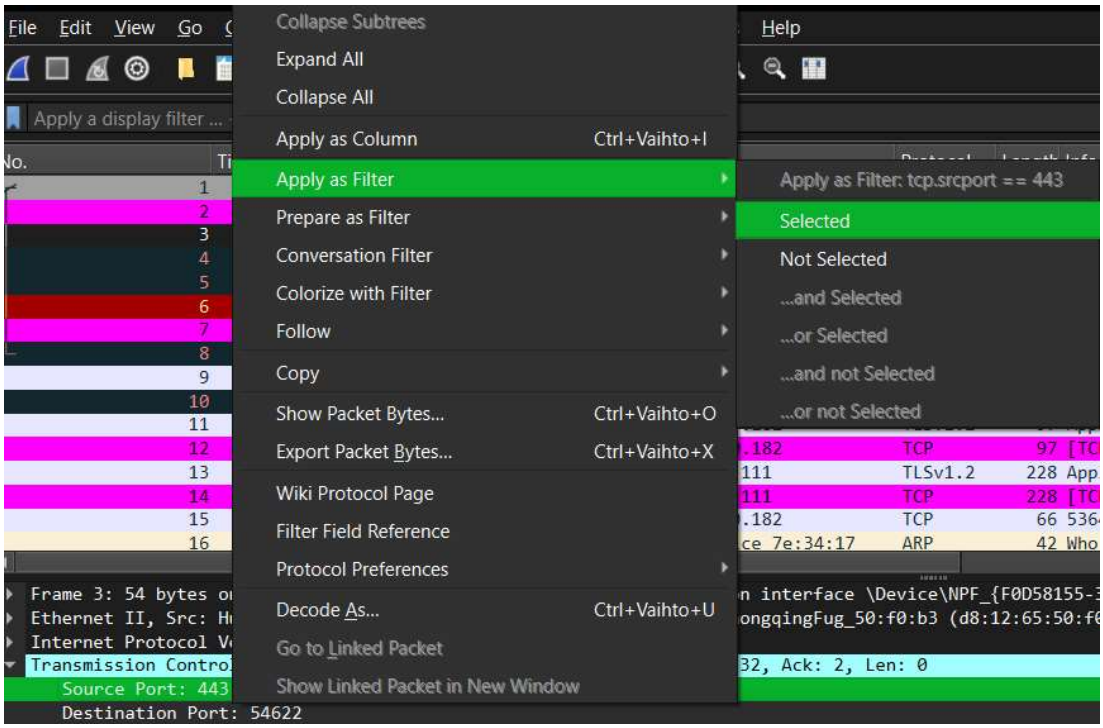
Kuva 17 Kohta, mitä painettaessa avautuu seuraavan kuvan näkymä.



Kuva 18 Näkymä, jossa voidaan asettaa suodatin ennen kaappausta.

On huomioitava, että tähän suodatinriville ei voi kirjoittaa kaikkia samoja suodattimia kuin valmiiksi kaapatun tiedoston suodatinriville. Esimerkiksi aikaisemmin mainitsemani ip.addr == 192.168.1.1 ei ole toimiva tällä tavalla, vaan se on korvattava kirjoittamalla host 192.168.1.1, joka tarkoittaa käytännössä sitä, että kaapataan ainoastaan kyseisen IP-osoitteen liikennettä. Suodattimia on useita erilaisia ja niiden kanssa täytyy olla tarkkaavainen.

Pakettien suodatus voidaan tehdä helposti painamalla hiiren kakkospainikkeella haluttua asiaa kuten esimerkiksi protokollaa, porttinumeroa tai IP-osoitetta. Seuraavassa kuvassa olen painanut oikealla hiiren painikkeella paketin tiedotruudussa kohtaa Source port:443 ja siitä avautuvasta valikosta pystytään valitsemaan se suodattimeksi painamalla ”Apply as Filter” ja ”Selected”, jolloin suodattimeksi valikoituu tcp.srcport == 443 ja Wireshark näyttää heti kaikki 443 portista lähtevän liikenteen. Jos valitaan ”Prepare as Filter”, niin silloin Wireshark kirjoittaa suodattimen, mutta ei heti ota sitä käyttöön.



Kuva 19 Oikealla hiiren painikkeella painettaessa avautuva valikko.

4.3 Värikoodit

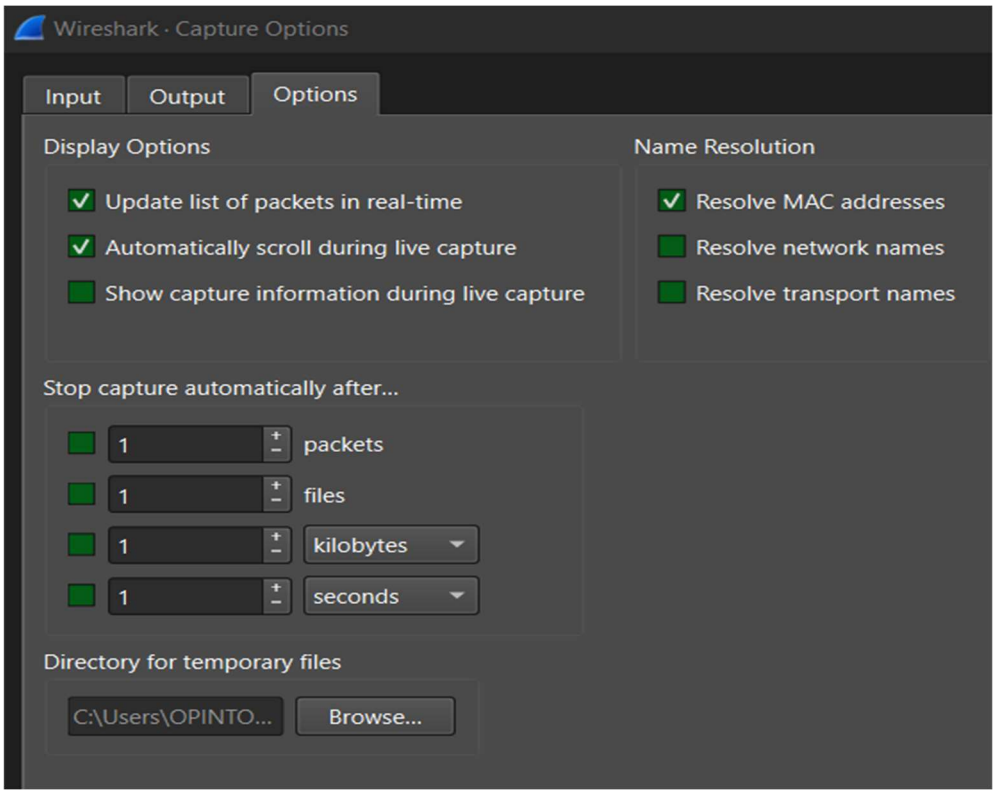
Wiresharkin pakettilistaruudussa näkyy paljon eri värillä olevia rivejä eli paketteja. Onkin hyvä tapa tehdä omia värityssääntöjä tietyille paketeille, jotta voidaan nopeasti arvioida mitä, milloinkin tapahtuu. Näitä värityssääntöjä voidaan tehdä menemällä kohtaan ”View”, josta valitaan kohta coloring rules. Värityssäännöt tallentuvat profiiliin, joka on käytössä.

4.4 Profiili

Wiresharkkiin on mahdollista tallentaa profiileja. Profiilia voidaan vaihtaa oikeasta alakulmasta. Profiiliin tallentuu värikoodiasetukset, suodattimet ja asetukset. On suotavaa luoda profiileita eri käyttötarkoituksiin, kuten esimerkiksi virheiden tarkasteluun. Samasta kohdasta, josta profiili vaihdetaan, voidaan myös luoda uusia painamalla hiiren kakkospainiketta.

4.5 Kaapatun tiedoston tallentaminen.

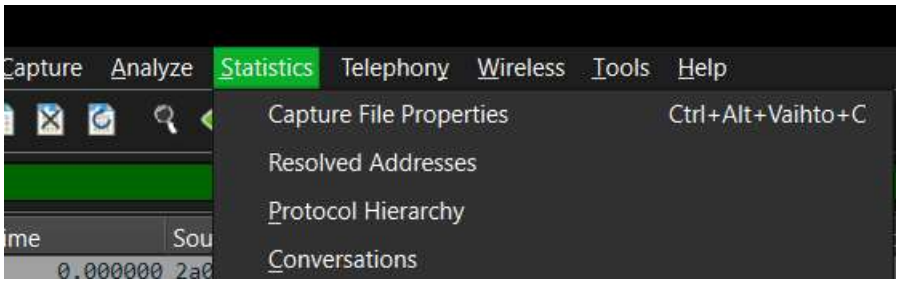
Kun paketteja on kaapattu ja kaappaus keskeytetään, Wireshark tallentaa kaappaustiedoston välimuistiin, mutta ei varsinaiselle kiintolevylle. Tähän on olemassa asetus, joka tallentaa tiedoston tai useamman automaattisesti käyttäen ennalta määriteltyjä parametreja kuten tiedoston koko, pakettien määrä tai kauanko kaappaus on kestänyt. Tiedoston kokoon liittyvä parametri on hyvä muistaa, koska jos pakettien kaappaus kestää pitkään, niin tiedoston koko saattaa kasvaa todella suureksi ja tiedoston tarkastelun yhteydessä Wireshark on hidas käsittelemään yli 100 megabitin tiedostoja, joten pitkäkestoiseen kaappaukseen on hyvä lisätä automaattinen tallennus, joka tallentaa tiedoston, kun ennalta määritelty koko täyttyy ja jatkaa kaappausta tämän jälkeen uudessa tiedostossa. Tämä asetus tehdään Capture options-valikossa, samassa, jossa voidaan asettaa kaappaukseen suodatin ennen kaappauksen aloittamista. Tästä valikosta voimme myös valita minne tiedostot tallennetaan. Tiedosto voidaan tallentaa suoraan myös CSV-tiedostoksi, joka voidaan avata esimerkiksi Excelissä.



Kuva 20 Capture options-valikko, josta voidaan valita, koska kaappaus keskeytetään.

4.6 Statistiikka-välilehti

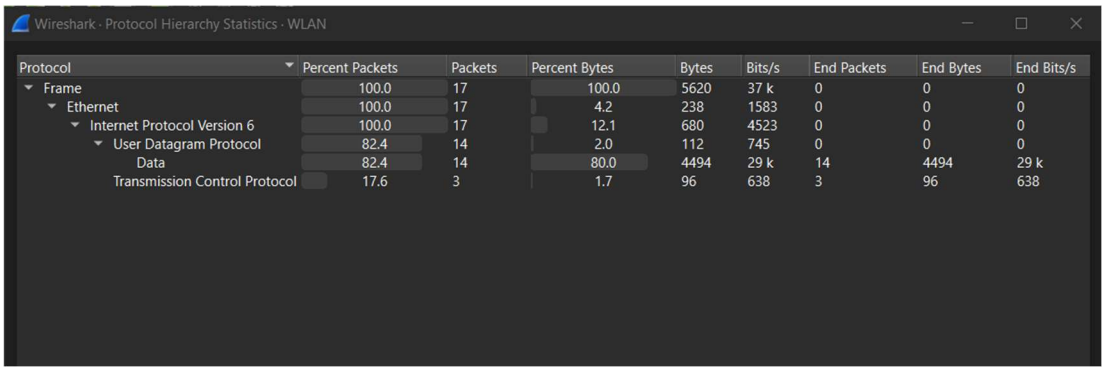
Statistiikka-välilehdeltä (Statistics) löytyy paljon analysoinnin kannalta hyödyllisiä asioita kuten kaikki IP-osoitteet ja protokollat kaappaustiedostossa. Lista on pitkä, joten käyn läpi tärkeimmät.



Kuva 21 Statistiikkavälilehti.

Ensimmäinen ”Capture File Properties”-kohta näyttää kaappaustiedoston nimen, koon, päivämäärän, kellonajan ja kaappauksen keston. Kaappauslaitteen prosessorin, verkkokortin, käyttöjärjestelmän ja ohjelmiston, sekä kaapattujen pakettien määrän ja muuta informaatiota itse kaappaustiedostosta.

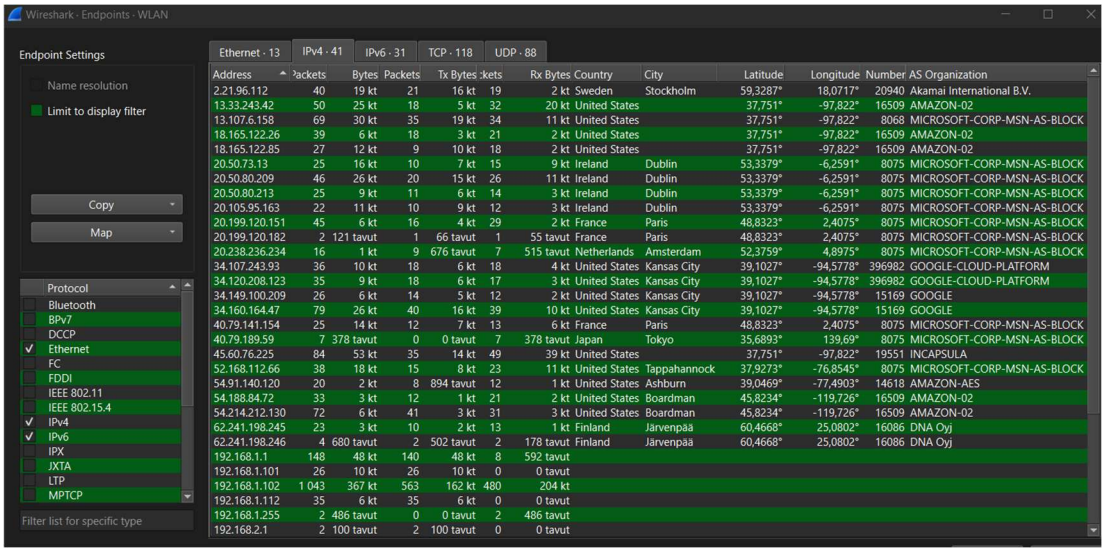
Protocol hierarchy-ikkunassa näemme kaikki protokollat kaappaustiedostossa ja niiden lähettämät tavujen määrät.



Kuva 22 Protocol hierarcy-ikkuna, jossa näkyy hieman UDP-liikennettä.

Endpoints valikko näyttää niin sanotut päätepisteet mihin paketit ovat menneet tai mistä niitä on saapunut. Jos MaxMind-nimisen yhtiön tietokanta ja lisäosa nimeltä GeoIP, josta lisää myöhemmin, on asennettu ja lisätty Wi-

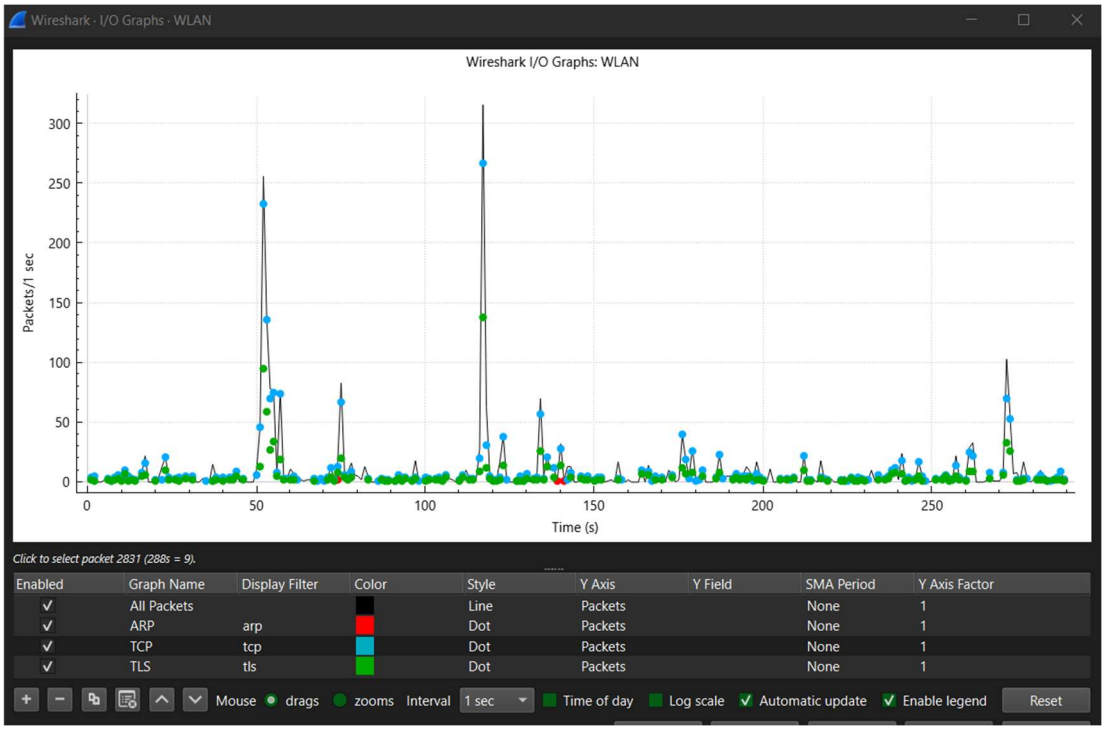
resharkkiin, niin silloin meillä on tässä ikkunassa näkyvissä myös paikannustiedot kyseisistä päätepisteistä maailmalla. Seuraavassa kuvassa kaappausta on tehty noin 5 minuuttia omassa kotiverkossani olevalla koneella. Kaappaustiedostossa on tavallista Windows-käyttöjärjestelmän ja verkkoselauksen tuottamaa liikennettä.



Kuva 23 Endpoints-ikkuna.

Endpoints-ikkunassa on välilehtiä kuten IPv4, IPv6, TCP ja UDP, joista voimme katsoa kyseisen protokollan päätepisteet. Jos olemme asettaneet suodattimen ennen kuin avaamme tämän ikkunan, niin silloin ikkunassa näkyy vain suodatettujen pakettien tiedot.

I/O graphs-ikkunassa voidaan tehdä diagrammeja liikenteestä. Ikkunan alareunasta voidaan lisätä mitä piirretään. Voimme asettaa suodattimia samoin kuin kaappausnäkymässä, jolloin Wireshark piirtää diagrammin kyseistä suodatinta käyttäen. Diagrammin piirtämiseen ja tarkasteluun on useita mahdollisuuksia. Seuraavassa kuvassa näkyy diagrammi, joka näyttää kaikkien pakettien määrän suhteutettuna aikaan mustalla viivalla, ARP-kyselyt punaisina pisteinä, TCP-paketit sinisinä pisteinä ja TLS-salausta käyttävät paketit vihreinä pisteinä. Tämä diagrammi on tehty samasta kaappaustiedostosta kuin aikaisempi kuva.



Kuva 24 I/O graphs-ikkuna.

4.7 GeoIP

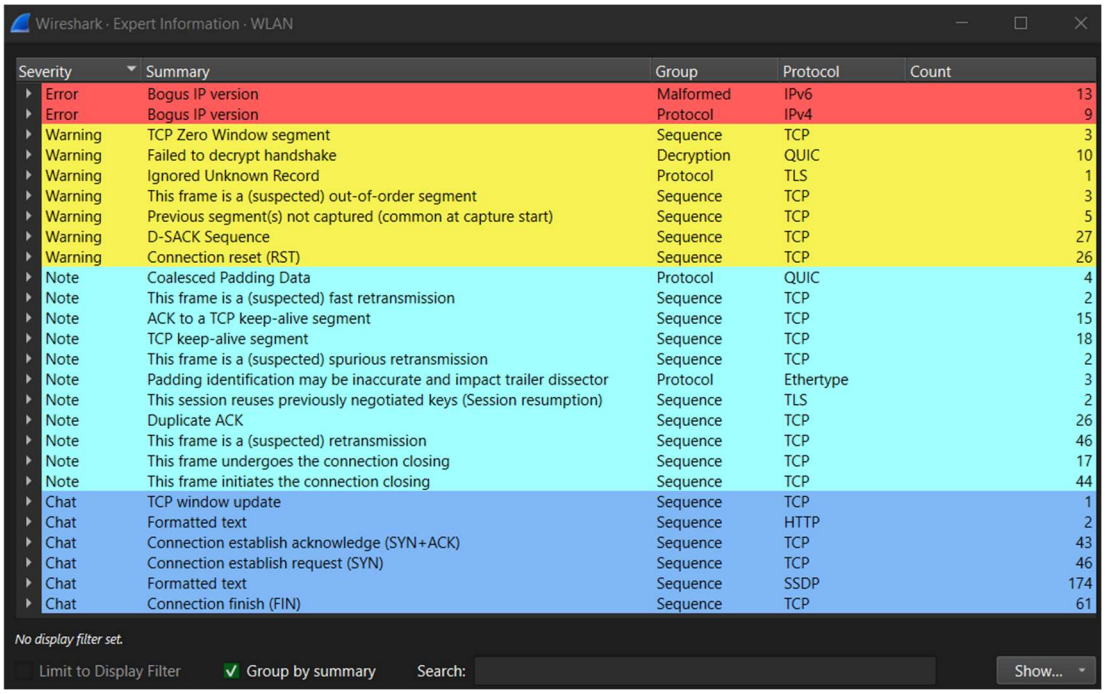
GeoIP on MaxMind-nimisen yhtiön tekemä tietokanta Wiresharkkiin. Tietokanta sisältää tiedon IP-osoitteen maantieteellisestä sijainnista. Lisäosasta on kaksi versiota maksullinen GeoIP2 ja maksuton Geolite2. Erona näissä on se, että Geolite2 on epätarkempi ja IP-osoitteet päivittyvät harvemmin kuin GeoIP2:ssa. (MaxMind, n.d.)

Kun tätä lisäosaa käytetään, endpoints-valikkoon ilmestyy mahdollisuus katsoa mistä maasta ja kaupungista paketti on saapunut tai minne se on lähetetty. Lisäosaa käyttämällä pystytään nopeasti tunnistamaan mistä maasta tai yrityksestä liikennettä saapuu ja minne sitä lähtee. Kun lisäosa on asennettu, endpoints-valikkoon ilmestyy nappula ”map”, josta voidaan avata kaappaustiedoston päätepisteistä luodun kartan. Seuraava kuva näyttää kyseisen kartan. Kuvassa on käytetty samaa kaappaustiedostoa kuin aikaisemmassa kuvassa.

Kuva 25 GeoIP:n tuottama kartta päätepisteistä.

4.8 Expert information

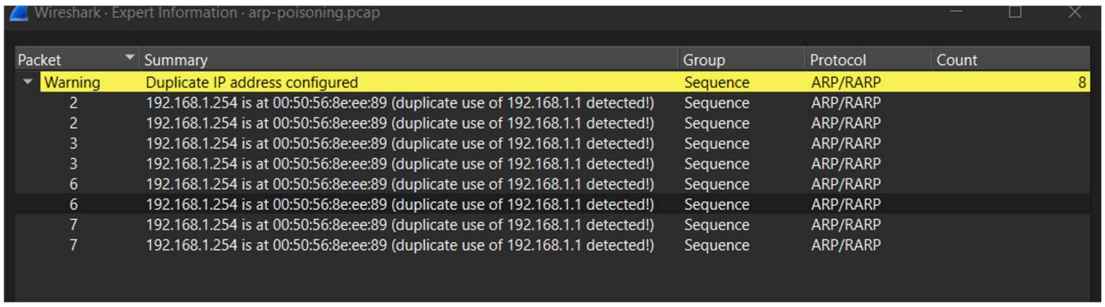
Expert information on liikenteen tarkastelun kannalta hyvin tärkeä. Kyseinen ikkuna aukeaa vasemmasta alakulmasta olevasta pallosta painamalla. Ikkunassa on nähtävissä virheet, varoitukset ja huomautukset kaappauksesta. Katsomalla tätä ikkunaa pystytään nopeasti huomaamaan sisältääkö kaappaustiedosto jotakin tarkastelemisen arvoista. Seuraavassa kuvassa on varoitukset ja virheet aikaisempien kuvien kaappaustiedostosta.



Kuva 26 Expert information-ikkuna.

4.9 ARP-välimuistin myrkyttämisen ja palvelunestohyökkäyksen havaitseminen Wiresharkilla

Expert information-ikkunasta voimme nopeasti huomata onko ARP-välimuisti myrkytetty. Wireshark kertoo tästä seuraavalla varoituksella: ”Duplicate IP address configured” Seuraavassa kuvassa on näkymä expert information-ikkunasta samasta asiasta.



Kuva 27 Wiresharkin kautta voidaan huomata, että ARP-välimuisti on myrkytetty.

Palvelunestohyökkäyksen tunnistaminen Wiresharkilla on myös melko suoraviivaista. Palvelunestohyökkäyksen tapahtuessa suuri määrä TCP SYN-paketteja saapuu palvelimelle. Palvelin yrittää saada kolmivaiheisen kättelyn suoritettua siinä onnistumatta. Palvelin menee tukkoon yrittäessään muodostaa yhteyttä. Wiresharkissa tämä näyttää samalta kuin seuraavassa kuvassa.

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	160.161.74.108	10.10.10.10	TCP	60 41885 → 25565 [SYN] Seq=0 Win=0 Len=0
2	0.000629	80.24.71.108	10.10.10.10	TCP	60 301 → 25565 [SYN] Seq=0 Win=0 Len=0
3	0.000557	64.93.51.108	10.10.10.10	TCP	60 3517 → 25565 [SYN] Seq=0 Win=0 Len=0
4	0.000108	190.238.52.108	10.10.10.10	TCP	60 12707 → 25565 [SYN] Seq=0 Win=0 Len=0
5	0.000570	112.124.85.108	10.10.10.10	TCP	60 49869 → 25565 [SYN] Seq=0 Win=0 Len=0
6	0.000353	38.19.55.108	10.10.10.10	TCP	60 49467 → 25565 [SYN] Seq=0 Win=0 Len=0
7	0.000074	22.13.62.108	10.10.10.10	TCP	60 30283 → 25565 [SYN] Seq=0 Win=0 Len=0
8	0.000354	108.233.74.108	10.10.10.10	TCP	60 60265 → 25565 [SYN] Seq=0 Win=0 Len=0
9	0.000596	122.60.73.108	10.10.10.10	TCP	60 47055 → 25565 [SYN] Seq=0 Win=0 Len=0
10	0.000082	202.197.75.108	10.10.10.10	TCP	60 23103 → 25565 [SYN] Seq=0 Win=0 Len=0
11	0.000211	130.153.72.108	10.10.10.10	TCP	60 39015 → 25565 [SYN] Seq=0 Win=0 Len=0
12	0.000606	84.121.78.108	10.10.10.10	TCP	60 47457 → 25565 [SYN] Seq=0 Win=0 Len=0
13	0.000144	194.204.81.108	10.10.10.10	TCP	60 29479 → 25565 [SYN] Seq=0 Win=0 Len=0
14	0.001861	62.150.70.108	10.10.10.10	TCP	60 11043 → 25565 [SYN] Seq=0 Win=0 Len=0
15	0.000796	26.140.178.108	10.10.10.10	TCP	60 29423 → 25565 [SYN] Seq=0 Win=0 Len=0
16	0.001859	76.14.170.108	10.10.10.10	TCP	60 33097 → 25565 [SYN] Seq=0 Win=0 Len=0
17	0.000334	212.7.169.108	10.10.10.10	TCP	60 4321 → 25565 [SYN] Seq=0 Win=0 Len=0
18	0.000790	94.237.230.108	10.10.10.10	TCP	60 14147 → 25565 [SYN] Seq=0 Win=0 Len=0
19	0.000491	146.59.230.108	10.10.10.10	TCP	60 2199 → 25565 [SYN] Seq=0 Win=0 Len=0
20	0.000201	78.100.228.108	10.10.10.10	TCP	60 39763 → 25565 [SYN] Seq=0 Win=0 Len=0
21	0.000411	164.64.236.108	10.10.10.10	TCP	60 8977 → 25565 [SYN] Seq=0 Win=0 Len=0
22	0.000442	118.137.232.108	10.10.10.10	TCP	60 65323 → 25565 [SYN] Seq=0 Win=0 Len=0
23	0.000606	116.201.165.108	10.10.10.10	TCP	60 29825 → 25565 [SYN] Seq=0 Win=0 Len=0
24	0.000360	96.150.248.108	10.10.10.10	TCP	60 27357 → 25565 [SYN] Seq=0 Win=0 Len=0
25	0.000001	174.162.13.108	10.10.10.10	TCP	60 55731 → 25565 [SYN] Seq=0 Win=0 Len=0
26	0.000249	138.186.249.108	10.10.10.10	TCP	60 8575 → 25565 [SYN] Seq=0 Win=0 Len=0
27	0.001463	84.66.5.108	10.10.10.10	TCP	60 44129 → 25565 [SYN] Seq=0 Win=0 Len=0
28	0.000721	178.224.6.108	10.10.10.10	TCP	60 6967 → 25565 [SYN] Seq=0 Win=0 Len=0

Kuva 28 TCP SYN-paketteja palvelunestohyökkäyksen aikana.

Tälläisessä tilanteessa, hyökkäyksen tunnistamiseen voidaan käyttää suodatinta ”tcp.flags.syn == 1 and tcp.flags.ack == 0” nähdäksemme kaikki SYN-paketit ja vertaamalla niiden määrää SYN/ACK-pakettien määrään, joka saadaan näkyviin tällä suodattimella ”tcp.flags.syn == 1 and tcp.flags.ack == 1”. Jos molempia on yhtä paljon, palvelin toimii oikein, mutta jos SYN-paketteja on huomattavasti enemmän kuin SYN/ACK-paketteja, palvelin ei pysty käsittelemään suurta määrää pyyntöjä. Tästä tiedetään, että hyökkäys on käynnissä. Tämän tyyppistä palvelunestohyökkäystä kutsutaan TCP SYN-tulvaksi.

Wiresharkilla on mahdollisuus tunnistaa monia käynnissä olevia hyökkäyksiä tai haavoittuvuuksia. Se sisältää monia muitakin ominaisuuksia ja mahdollisuuksia uhkien havaitsemiseksi.

5 LOPPUSANAT

Tämän työn tarkoituksena on tunnistaa ja tiedostaa verkkopakettien kaappaamiseen käytettyjä tekniikoita ja Wireshark-ohjelmiston käyttöä sekä verk-

kopakettien rakennetta. Verkkopaketit sisältävät paljon erityyppisiä protokollia ja tekniikoita, joiden avulla verkkopaketit liikkuvat paikasta toiseen. Näiden pakettien rakenne on verkkoliikenteen analysoinnin kannalta todella tärkeää, koska ilman ymmärrystä niistä, analysoiminen olisi mahdotonta. Pakettien kaappausmenetelmät, joita tässä työssä on esitelty, ovat tarkoitettu suurten sisäverkkojen liikenteen kaappaamiseen ja analysointiin. Vaikka Wireshark-ohjelmistoa käyttäessäni käytinkin omaa sisäverkkoani, niin samoin voidaan käyttää tätä ohjelmistoa suuremmissa verkoissa. Oman sisäverkon tarkastelussa ei näy yhtä mielenkiintoista liikennettä kuin esimerkiksi suurissa organisaatioiden hallitsemissa sisäverkoissa ja siksi käytinkin joissakin esimerkeissäni verkosta löytämiäni julkisia kaappaustiedostoja. On aina muistettava, että verkkoliikenteen kaappaamiseen on oltava verkonhaltijan lupa.

Opinnäytetyötäni tehdessä olen oppinut todella paljon verkkoliikenteestä ja sen toiminnasta. Moni artikkeli tai kirja, joita luin työtä tehdessäni, ei koskaan päätynyt itse työhön. Huomasin, että verkkoliikenteen rakenne on pysynyt lähes muuttumattomana 1980-luvulta asti. Suurimpina muutoksina ovat olleet turvallisuuden parantuminen ja salaustekniikoiden laaja käyttöönotto. Itse Wireshark on todella tehokas ohjelmisto ja siinä on paljon ominaisuuksia erilaisiin käyttötarkoituksiin kuten teleliikenteen tutkimiseen. Uhkien havaitseminen Wiresharkkia käyttäen on melko suoraviivaista, jos tiedetään miten esimerkiksi palvelunestohyökkäykset toimivat. Wiresharkilla ei kuitenkaan pystytä ennustamaan onko uhka vaanimassa kulman takana. Wireshark päivittyy useasti ja opinnäytetyön kirjoittamisen aikana päivityksiä ilmestyi useita. Mielestäni tämä kertoo siitä, että ohjelmisto on suosittu ja tulee varmasti kehittymään tulevaisuudessa entistä tehokkaammaksi työkaluksi. Suosittelenkin kaikkien verkkoturvallisuudesta tai verkkoliikenteen analysoinnista kiinnostuneiden ottamaan tämä ohjelmisto haltuun vähintäänkin siksi, että vaikka kaappaus olisi suoritettu toisella ohjelmistolla, voimme avata toisen ohjelmiston tuottaman tiedoston Wiresharkilla tarkastelua varten. Työssäni läpi käyty ARP-välimuistin myrkyttäminen on haavoittuvuus, jonka avulla hyökkääjä pystyy tarkastelemaan verkossa olevien käyttäjien liikennettä. Tätä hyökkäystä käytetään usein suojaamattomissa sisäverkoissa kuten julkisissa langattomissa verkoissa esimerkiksi hotelleissa tai lentokentillä. Verkon turvalli-

suus on mielestäni tärkeä asia ja aion opiskella lisää aiheesta tulevaisuudessa.

LÄHTEET

Anttila A. (2001) TCP/IP-tekniikka. Helsinki Media.

Browne A. (18.6.2023) SYN/ACK in the TCP protocol. https://www.baeldung.com/cs/tcp-protocol-syn-ack

Cloudflare (n.d.-a) What is DNS? Haettu 19.3.2024 osoitteesta https://www.cloudflare.com/learning/dns/what-is-dns/

Cloudflare (n.d.-b) What is HTTP? Haettu 4.12.2023 osoitteesta https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/

Cloudflare (n.d.-c) What is the internet protocol? Haettu 20.12.2023 osoitteesta https://www.cloudflare.com/learning/network-layer/internet-protocol/

Finflex (21.4.1995) Tieto- ja viestintärikoslaki. Haettu 17.11.2023 osoitteesta https://www.finlex.fi/fi/laki/ajantasa/1889/18890039001#L38

Fortinet (n.d.) What is address resolution protocol (ARP)? Haettu 2.2.2024 osoitteesta https://www.fortinet.com/resources/cyberglossary/what-is-arp

GeeksforGeeks (11.9.2023) What is ARP-spoofing? https://www.geeksforgeeks.org/what-is-arp-spoofing-arp-poisoning-attack/

GeeksforGeeks (8.9.2022) What is TCP-ACK Scanning? https://www.geeksforgeeks.org/what-is-tcp-ack-scanning/

Internetopas.com (26.10.2022) Mikä on TLS-salaus ja miten se pitää tietosi turvassa? Mikä on TLS-salaus ja miten se pitää tietosi turvassa? - Interneto-pas.com

Imperva (n.d.) ARP spoofing. Haettu 4.3.2024 osoitteesta https://www.imperva.com/learn/application-security/arp-spoofing/

Jasrotia R. (8.11.2021) The OSI model explained and how to easily remember its 7 layers. https://www.linkedin.com/pulse/osi-model-explained-how-easily-remember-its-7-layers-rahul-jasrotia

MaxMind (n.d.) GeoLite2 Free Geolocation Data. Haettu 23.3.2024 osoitteesta https://dev.maxmind.com/geoip/geolite2-free-geolocation-data

Network Critical (n.d.) Network Optimization Whitepaper. Haettu 20.2.2024 osoitteesta https://www.networkcritical.com/resources

Novell (n.d.) NetWare TCP/IP Administration Guide. Haettu 3.1.2024 osoitteesta https://www.novell.com/documentation/nw6p/?page=/documentation/nw6p/tcpipenu/data/hbnuubtt.html

Rahul (7.6.2023) Understanding the TCP/IP Model. https://tecadmin.net/tcp-ip-model/

Rapid7 (n.d.) What is network traffic analysis. Haettu 20.1.2024 osoitteesta https://www.rapid7.com/fundamentals/network-traffic-analysis/

Rouse M. (21.12.2016) What Does Address Resolution Protocol Cache Mean? Haettu 5.2.2024 osoitteesta https://www.techopedia.com/definition/27470/address-resolution-protocol-cache-arp-cache

Sanders C. (30.3.2017) Practical Packet Analysis, 3rd Edition: Using Wireshark to Solve Real-World Network Problems. No Starch Press.

Shanmugan R., Padmini R., Nivedita S. (2002) Using TCP/IP. Que publishing.

Timeless technology (03.08.2022) Verkkoliikenteen kaappaus-, monitorointi- ja analysointiratkaisut. https://timeless.fi/uutiset/profitap/verkkoliikenteen-sieppaus-monitorointi-ja-analysointiratkaisut/

Zieniūtė U. (06.11.2022) Mitä TCP ja UDP ovat? Yksinkertainen selitys. https://nordvpn.com/fi/blog/tcp-udp-protokolla/