Tampereen yliopisto

Mikko Vaurio

VERKKOLIIKENTEEN VUOTIEDON KERÄÄMINEN JA MALLINNUS

Informaatioteknologian ja viestinnän tiedekunta
Kandidaatintyö
Joulukuu 2019

TIIVISTELMÄ

Mikko Vaurio: Verkkoliikenteen vuotiedon kerääminen ja mallinnus (Collecting flow-based network traffic information and using information for network modeling)

Kandidaatintyö

Tampereen yliopisto

Tieto- ja sähkötekniikan TkK-tutkinto-ohjelma

Joulukuu 2019

Tässä työssä tarkastellaan verkkoliikenteen vuotiedon keräämistä ja sen hyödyntämistä liikenteen mallintamiseen. Lisäksi työssä käydään läpi eri tasoisten mallien hyödyntämismahdollisuuksia. Työssä vastataan kahteen tutkimuskysymykseen, jotka liittyvät edellä esitettyihin aihealueisiin: ”Minkälaisilla tekniikoilla voidaan kerätä vuotietoa verkossa kulkevasta tietoliikenteestä?” ja ”Miten tietoliikenteen vuotietoa on mahdollista hyödyntää?”. Työ on toteutettu kirjallisuusselvityksenä.

Tietoliikenteen vuotiedon keräämiseen on vakiintunut IPFIX standardi, jota eri tietoliikennelaitteiden valmistajat tukevat laajasti. Standardi perustuu Cisco Systemsin NetFlow määrittelyyn, joka on ollut jo ennen yhteisiä standardeja ylivoimaisesti yleisin ja tuetuin keräysmenetelmä. Muilla laitevalmistajilla on omia menetelmiään, mutta niiden merkitys on vähäistä.

Tietoliikenteestä kerätyn vuotiedon avulla on mahdollista tehdä malleja niin yksittäisen sovelluksen, yksittäisen tietoliikennelaitteen kuin verkkosegmentinkin liikenteestä. Malleja voidaan hyödyntää erilaisten tietoturvapoikkeamien havaitsemiseen, verkon kapasiteetin riittävyyden arviointiin sekä verkon palveluiden toimivuuden analysointiin.

Avainsanat: tietoliikenne, liikennevuo, mallinnus

SISÄLLYSLUETTELO

LYHENTEET JA MERKINNÄT

ANIDS	Anomaly-based network intrusion detection system
IaaS	Infrastructure as a service
IANA	Internet Assigned Numbers Authority
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IPS	Intrusion Prevention System
MIB	Management Information Base
PaaS	Platform as a service
RFC	Request for comments
SaaS	Software as a service
SCTP	Stream Control Transport Protocol
SIEM	Security incident and event management
SNMP	Simple network management protocol
SOC	Security Operations Center
UDP	User datagram protocol

.

1. JOHDANTO

Tietoverkkojen toiminnasta on tullut yhteiskunnallisesti merkittävää ja verkkojen varassa toimivat nykypäivänä esimerkiksi pankit ja sairaalat. Verkoissa on nykyään myös kasvava määrä laitteita, jotka kommunikoivat keskenään ja toimivat ilman ihmisten tekemää työtä. Jotta verkkojen toimivuus voidaan pitää riittävällä tasolla, pitää niiden toimintaan ja niissä kulkevan liikenteen analysointiin olla riittävät työkalut.

Tietoliikenneverkkojen liikenteestä saadaan tietoa useilla eri menetelmillä. Tietoliikennelaitteet tallentavat tietoa laite- tai verkkorajapintakohtaisiin liikennelaskureihin, joita voidaan lukea esimerkiksi SNMP-protokollalla. Näiden laskurien tarjoama tieto on hyödyllistä esimerkiksi liikennemäärien trendejä ja verkon kuormitusta analysoitaessa. Tiedon kerääminen on suoraviivaista eikä vaadi paljoa esimerkiksi tallennustilaa, koska kysymyksessä on vain yksittäinen lukuarvo kutakin laskuria kohti. Tietoturvalaitteet puolestaan tarjoavat lokeja ja rajapintoja käsittelemänsä verkkoliikenteen sisältöön sovellus- ja sisältönäkökulmasta. Reaaliaikainen tietoliikenteen sisällön käsittely on myös raskasta ja kallista, joten tyypillisesti kaikki verkkoliikenne ei kulje tietoturvalaitteiden kautta.

Tässä työssä tarkastellaan vuopohjaisen liikennetiedon keräämistä lähiverkkoympäristössä. Vuopohjaisessa tiedonkeruussa tietoliikennelaite analysoi liikennettä otsikkotasolla ja tietoliikennepaketit yhdistetään lähde- ja kohdeosoitteiden sekä porttien perusteella omaan liikennevuohonsa. Työssä tarkastellaan myös kerätyn tiedon analysointia ja tiedoista muodostettujen verkkoliikennemallien käyttöä.

Luvussa 2 taustoitetaan tietoliikenteen analysoinnin menetelmien kehitystä ja historiaa. Siihen on kerätty myös eri menetelmien haasteita ja hyötyjä. Luvussa 3 tarkastellaan tarkemmin vuotiedon keräämiseen tarvittavia menetelmiä ja osapuolia. Tiedon keruumenetelmistä käydään läpi niin teollisuuden valmistajakohtaiset standardit kuin IETF:n standardoimat menetelmät. Lisäksi tutustutaan tiedon keruuseen ja tiedon hyödynnettävyyteen liittyviin haasteisiin ja ongelmiin. Luvussa 4 keskitytään kerätyn tiedon hyödyntämismahdollisuuksiin verkon liikennettä mallinnettaessa ja havainnollistettaessa. Luvussa 5 on yhteenveto tehdyistä havainnoista ja kuvattujen mallintamisten käyttömahdollisuuksista.

2. TIETOLIIKENTEEN ANALYSOINTI

2.1 Taustaa

Internet-liikenteen valvonta ja tarkkailu ovat koko verkon historian ajan olleet vaikeita kokonaisuuksia. Teknisten haasteiden lisäksi on pitänyt huomioida verkon avoimuuteen ja yksilöiden tietosuojaan liittyvät yksityiskohdat. Lähtökohtaisesti Internet-verkko on ollut avoin ja vapaa tarkkailulta. Kaikenlainen kontrollointi on ollut verkon käyttöä rajoittavaa ja jopa käyttäjien yksityisyyden kannalta vahingollista. [1]

TCP/IP-liikenteen mittaukseen ja analysointiin kehitetyt standardit olivat 1900-luvun puolella vielä yleisimmin pakettilaskureita, jotka kertoivat arvokasta tietoa liikenteen määristä ja käytetyistä protokollista. RFC 1067 [3] määritteli jo 1988 yksinkertaiset menetelmät tiedon tallennukseen ja hallintaan SNMP-standardissa. Standardissa määriteltiin kerättävän tiedon rakenne ja hierarkia eli MIB. Määriteltynä oli myös operaatiot, joilla tietoja voitiin lukea ja kirjoittaa. SNMP ei siis ollut tarkoitettu vain tiedon keräämiseen, vaan sitä oli mahdollista käyttää myös verkkolaitteiden hallintaan.

Ensimmäinen versio SNMP-protokollasta oli tehty mahdollisimman yksinkertaiseksi ja siitä puuttuivat esimerkiksi tunnistautuminen ja käyttöoikeuksien hallinta kokonaan. Vuodesta 1993 eteenpäin julkaistuissa SNMP-standardin versiossa 2 [4] kokonaisuutta laajennettiin muun muassa laajemmilla tietomalleilla ja käyttöoikeuksien hallinnalla. Nykypäivän mittareilla esimerkiksi selväkielinen salasanatunnistautuminen ja kaksitasoinen hallintamalli eivät ole riittäviä, joten standardista on myös versio 3, jossa on mukana nykyaikaiset tunnistautumis- ja yhteysmekanismit.

Nykyaikaisten verkkojen liikenteen analysointi vaatii kuitenkin tarkempaa tietoa kuin pelkät liikennelaskurit. Verkot ovat nopeampia ja niissä kuljetettava sisältö on paljon moninaisempaa. Liikenteen määrien lisäksi tarvitaan tietoa verkon palveluiden käytöstä ja toimivuudesta. Käytännössä tällaista tietoa saadaan keräämällä ja yhdistelemällä tietoa verkon läpi eri mittauspisteissäkulkevista yhteen kuuluvista pakettivirroista. Näin saatua vuokohtaista tietoa voidaan käyttää paljon monipuolisemmin kuin yksittäisten pakettien kulkua tarkkailtaessa.

Vuopohjaisen tiedon kerääminen oli vuoteen 2013 asti valmistajakohtaisiin standardeihin perustuvaa. Myös ennen IPFIX-standardia [2] eri valmistajat tukivat laitteissaan toisten valmistajien standardeja.

2.2 Tavoitteet

Tietoliikenteen analysointia tehdään moniin eri käyttötarkoituksiin. Tärkeää nykyaikaisessa verkossa on esimerkiksi verkkopalveluiden laadun varmistaminen ja todentaminen. Yksinkertaisimmillaan laaduntarkkailu voi tarkoittaa liikenteen määrien mittaamista ja vertailua verkon kapasiteettiin. Jos halutaan parempaa tietoa käyttäjän kokemasta palvelunlaadusta, pitää mitata myös palvelukohtaisia mittareita, kuten palveluiden vasteaikoja.

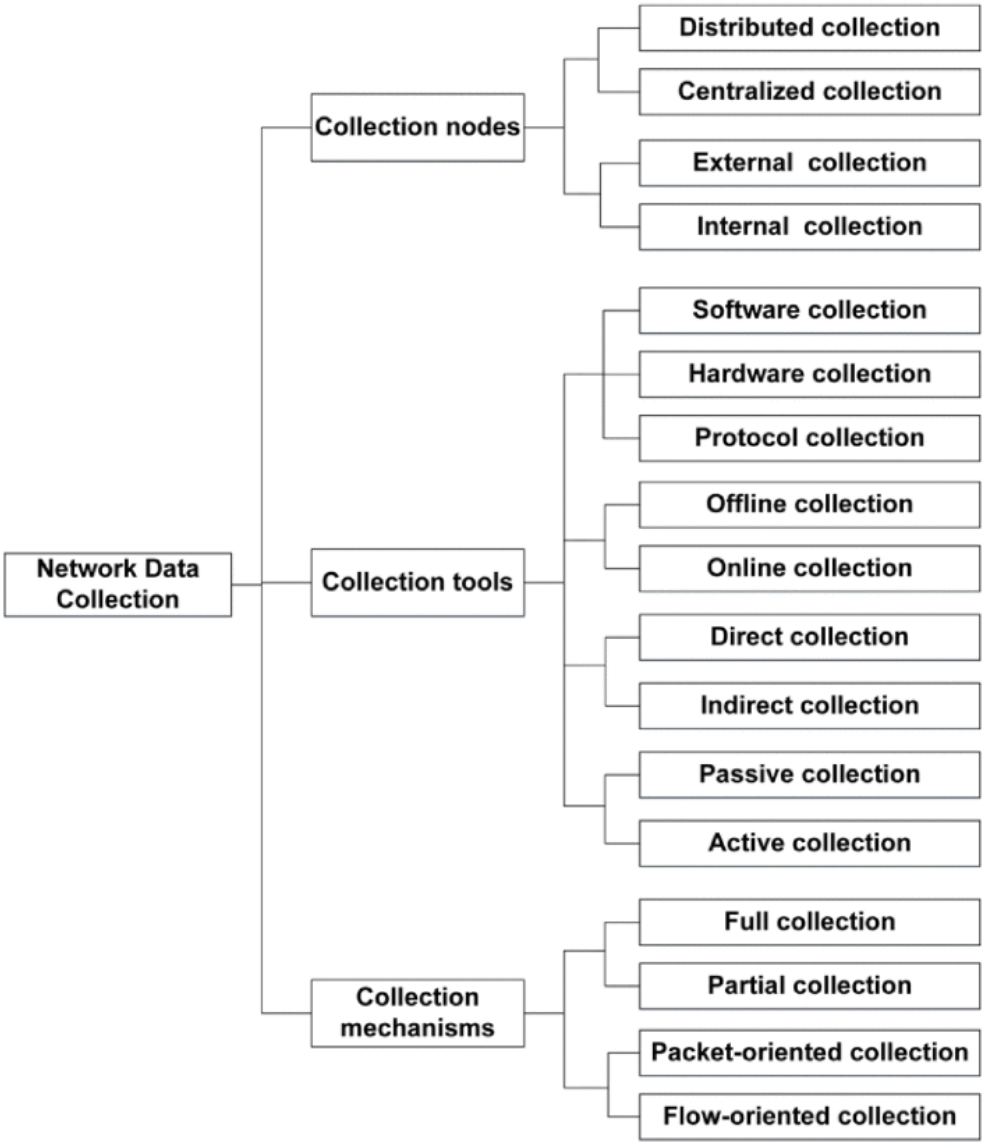
Mikäli palveluiden laadussa on ongelmia, tieto verkon nykytilanteesta ja historiasta myös erittäin hyödyllistä ongelmien syitä selvitettäessä. Vertailemalla nykytilannetta historiatietoon voidaan helpommin löytää muuttuneet ja ongelmia aiheuttavat komponentit.

Verkkoliikennettä voidaan mitata ja analysoida myös esimerkiksi palvelulupausten todentamiseen. Verkon käytettävyys ei tällöin ole esimerkiksi yksittäisen osoitteen tai palvelun tavoitettavuuden mittaamisen varassa. Vaikka verkossa kulkisikin liikennettä ei palvelulupaus täyty, jos liikenteelle määritellyt palveluparametrit ylittyvät. Mittauksilla voidaan myös tarjota tietoa verkon käyttömääristä ja tietoa voidaan käyttää esimerkiksi laskutusperusteena.

Nykypäivänä tietoturvan rooli korostuu verkoissa yhä enemmän ja analysointia halutaan tehdä niin organisaatioiden ulkorajoilla kuin sisäisessä liikenteessä. Verkkoihin on liitettynä monenlaisia laitteita, joilla analysoidaan tietoliikenteen sisältöä ja sieltä mahdollisesti nousevia uhkia.

2.3 Erilaisten tavoitteiden asettamat vaatimukset

Erilaisten tarpeiden täyttämiseksi verkkoliikennettä voidaan mitata monenlaisista laitteista ja monin eri tavoin. Myös mitatun tiedon keräämiseen on useita vaihtoehtoja. Kuvassa 1 on esitetty erilaisten tiedonkeruutapojen yhdistelmiä.



Kuva 1. *Verkkoliikenteen tiedon keräyksen taksonomia mukaillen lähdettä [5]*

Seuraavassa luvuissa tutustutaan tarkemmin vuopohjaisen tiedonkeruun toimintaan ja standardeihin.

3. VUOTIEDON KERÄÄMINEN

3.1 Menetelmät tiedon keruuseen

Verkkoliikenteestä erotettavan vuon määritelmä ei ole yksikäsitteinen. Yleisesti vuopohjaisessa tiedonkeruussa kuitenkin analysoidaan pakettijoukkoa kokonaisuutena, kun taas pakettipohjaisessa yksittäisen paketin ominaisuuksia.

Perinteinen IP -liikenteen vuon määritelmä on ollut kiinteästi määritelty yhdistelmä. Tyypillinen vuo on pakettivirta, jossa tunnisteena on lähde- ja kohdeosoitteiden sekä protokollan yhdistelmä. [6]

IPFIX standardissa määritelmä on joustavampi. Määritelmässä vuon muodostavat paketit, jotka ohittavat tarkkailupisteen määritetyllä aikavälillä ja joilla on määritellyt yhteiset ominaisuudet. Ominaisuudet voivat liittyä paketin otsikkokentän tietoihin kuten yleisesti käytetyssä määritelmässä. Edellä mainitun lisäksi ominaisuudet voivat olla myös pakettiin itseensä tai verkkolaitteessa tapahtuvaan paketin käsittelyyn liittyviä. [6]

Kuten kuvassa 1 esitettiin, tiedon keräämiseen on erilaisia lähestymistapoja. Vuotietoa voidaan esimerkiksi kerätä joko analysoimalla kaikki mittauspisteen liikenne tai käyttämällä liikenteestä otettuja näytteitä. Kaikki liikenne analysoimalla saadaan tarkempi mittaustulos, mutta tällöin mittaamisen aiheuttama kuormitus on suurempi. Riippuu tiedon käyttötarkoituksesta, onko näytteenottoon perustuva mittaus riittävä.

Myös tietojen keräystavat vaikuttavat tiedon hyödynnettävyyteen ja sen aiheuttamaan kuormitukseen. Tietoja voidaan kerätä hajautetusti kaikissa mittauspisteissä tai ne voidaan kerätä keskitettyyn tietovarastoon. Hajallaan olevan mittaustiedon hyödyntäminen ja hallinta on hankalampaa. Tietojen tallentaminen myös kuormittaa mittausta tekeviä järjestelmiä. Toisaalta tietojen keräilyyn ei tarvita erillisiä laitteistoja ja tiedon kerääminen ei itsessään aiheuta verkkoliikennettä. Keskitetyssä tiedonkeruussa tiedon tallennusympäristö on voitu suunnitella tätä käyttötarkoitusta varten. Mikäli keskitetty järjestelmä ei jonain ajanhetkenä ole käytettävissä, pitää keräyspisteissä olla valmius mittaustiedon puskurointiin tai tiedot jäävät puutteellisiksi.

Myös mittauspisteet voivat sijaita erilaisissa paikoissa. Tietoa voidaan kerätä esimerkiksi verkkolaitteista järjestelmien väliltä. Jos mitattava liikenne on esimerkiksi loppukäyttäjän päätelaitteen ja palvelua tuottavan järjestelmän välillä, kattaa ulkoinen mittaus koko palveluun liittyvän liikenteen. Joskus voidaan tarvita myös mittauspisteitä järjestelmien si-

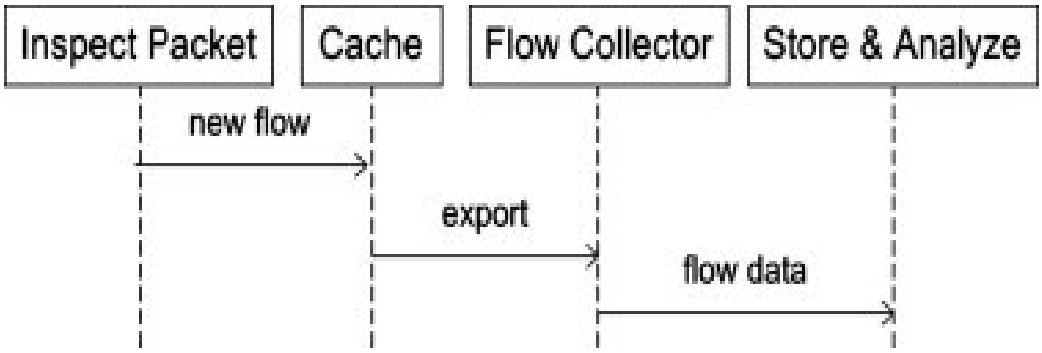
sälle, jotta kaikki kiinnostava verkkoliikenne saadaan mittausten piiriin. Joskus mittauksen tekeminen järjestelmän sisällä voi myös vähentää mittauspisteiden tarvetta ja yksinkertaistaa mittausten tekemistä. Aina mittaaminen järjestelmän ulkopuolella ei ole edes mahdollista, kuten esimerkiksi luvussa 3.5 tarkemmin käsiteltävissä pilvialustoissa.

3.2 Teollisuuden standardit

IETF työryhmissä oli jo 1990-luvulla hankkeita, joissa käsiteltiin tiedonkeruuta verkosta vuopohjaisella mallilla. Mikään näistä hankkeista ei kuitenkaan johtanut standardien kirjoittamiseen. Yhtenä syynä hankkeiden tuloksettomuuteen oli yleisesti liikenteen tallennusta ja valvontaa vastaan ollut asenne. Epäiltiin, että mittausmenetelmien kehittäminen johtaisi laskutuksen ja tarkkailun helpottamiseen, ja sitä kautta vapaan Internetin rajoituksiin. Toinen merkittävä syy oli laitevalmistajien puolelta puuttunut kiinnostus standardiratkaisuihin. [1]

Koska yhteiset standardit vuotiedon keräämiseen olivat pitkään puutteelliset, oli mittaamiseen tarjolla valmistajien omia ratkaisuja. Kytkin- ja reititinliiketoiminnassa Cisco on ollut koko Internetin historian ajan ylivoimaisesti merkittävin valmistaja ja hallitsee edelleen laskentatavasta riippuen arviolta puolta koko markkinasta [7]. On siis luonnollista, että Ciscon kehittämä NetFlow-standardi on ollut valmistajakohtaisista standardeista yleisin ja käytetyin. Muillakin valmistajilla on omat ratkaisunsa. Esimerkiksi Juniper on kehittänyt oman J-Flow-standardinsa ja Huawei NetStream-standardin. NetFlow on kuitenkin niin hallitseva teknologia, että sitä käytetään usein myös synonyyminä vuotiedon keräämiseen yleisesti.

NetFlow:n ensimmäisen version pohjana oleva patentti on haettu jo vuonna 1996 [8]. NetFlow-standardista on useita versioita, joista viimeisin on NetFlow versio 9. Suurimmat muutokset viimeisimmässä versiossa olivat versioiden 1-8 välillä vakioina pysyneiden tietuerakenteiden muuttaminen ja lisäys. Itse tiedon keräysmalli ja roolitus ovat pysyneet koko ajan kuvan 2 mukaisina.



Kuva 2. *NetFlow tiedon keräys ja lähetys [9]*

Vuotietoa keräävä laite luo välimuistiinsa jokaisesta vuosta tietueen, jonka tietoja päivitetään, kun vuohon liittyvää liikennettä saapuu laitteeseen. Kun joko laitteessa määritelty ajastin tai muutoin määritelty politiikka niin määrittää, kerätyt tietueet lähetetään eteenpäin. Lähetys tapahtuu joko UDP-paketteina tai SCTP-protokollalla. Mallissa vuotietoa voi kadota, mikäli esimerkiksi välimuisti täyttyy tai UDP-paketteja katoaa laitteen ja keräilijän välillä. Näytteiden käsittely myös kuormittaa laitteen suoritinta ja kuormituksen vähentämiseksi voidaan käyttää myös Sampled NetFlow versiota, jossa liikenteestä käsitellään vain kiinteästi määritelty tai satunnainen osa paketeista. [9]

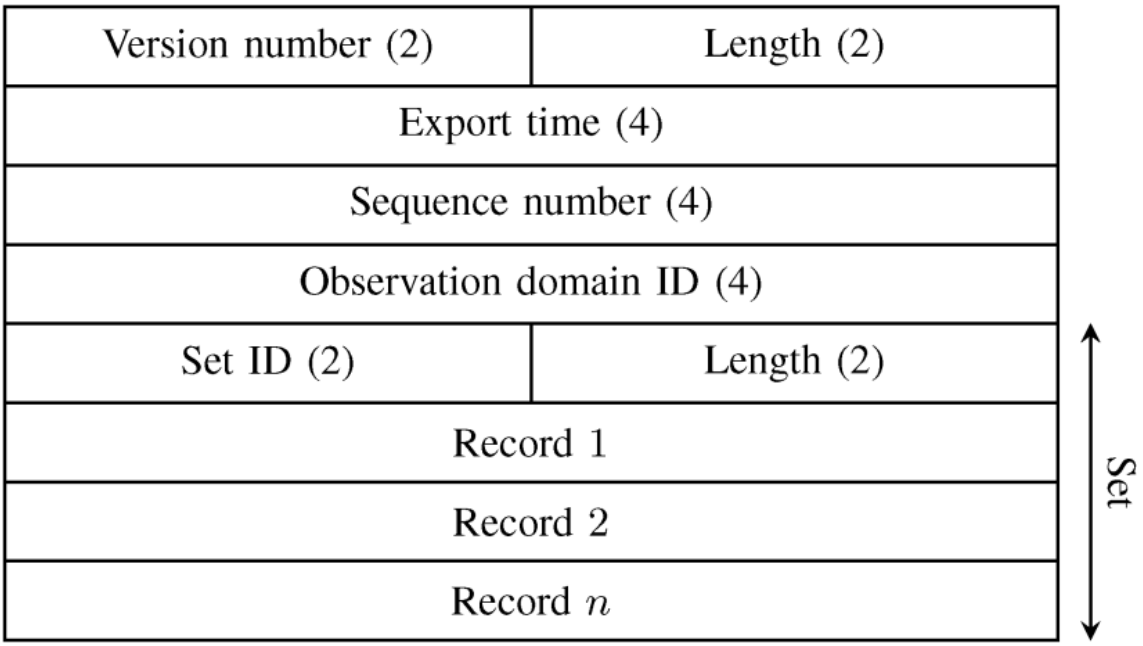
3.3 Standardit

IPFIX on IETF:n määrittelemä standardi vuotietueiden keräämiseen. Standardi perustuu NetFlow versioon 9, mutta IPFIX sisältää ominaisuuksia, joita NetFlow versio 9 ei sisällä. Kuten NetFlow, IPFIX määrittelee tiedon formaatit sekä tavat, joilla tiedot voidaan toimittaa edelleen käsiteltäviksi. Nimestään huolimatta IPFIX ei ole rajoittunut IP-protokollaan liittyvän tiedon keräämiseen. Tietueiden osina voidaan käyttää IANA:n ylläpitämässä listassa olevia tietoja. Tietoja on myös mahdollista täydentää organisaatiokohtaisilla laajennoksilla. Tietoja on mahdollista kerätä miltä tahansa tasolta aina sovellustasolta linkkitasolle. Yleisimmät toteutukset tukevat tiedon keruuta verkko- ja kuljetuskerroksilta. Kuvassa 3 esitetään kerätyn tietueen esimerkkirakenne.[1]

ID	Name	Description
152	flowStartMilliseconds	Timestamp of the flow's first packet.
153	flowEndMilliseconds	Timestamp of the flow's last packet.
8	sourceIPv4Address	IPv4 source address in the packet header.
12	destinationIPv4Address	IPv4 destination address in the packet header.
7	sourceTransportPort	Source port in the transport header.
11	destinationTransportPort	Destination port in the transport header.
4	protocolIdentifier	IP protocol number in the packet header.
2	packetDeltaCount	Number of packets for the flow.
1	octetDeltaCount	Number of octets for the flow.

Kuva 3. *Tyypilliset IPFIX tietueen minimitiedot lähteen [1] mukaan*

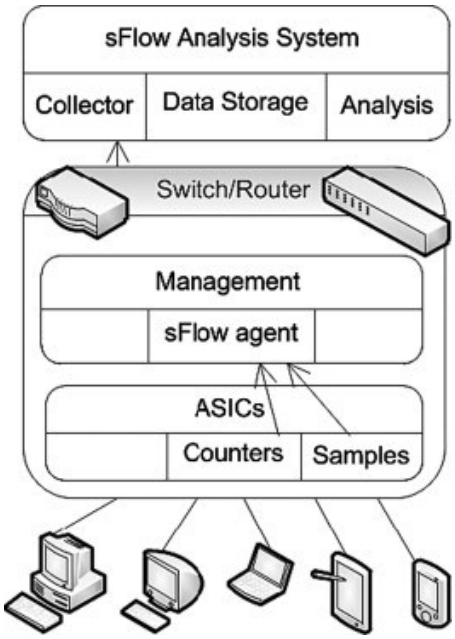
Siirtotiellä käytettäväksi protokollaksi IPFIX määrittelee joko SCTP-, UDP- tai TCP -protokollat. Näistä SCTP on määritelty pakolliseksi. [2]



Kuva 4. *IPFIX viestirakenne [1]*

Standardissa on määritelty myös viestirakenne lähetettäville tietueille. Kuvan 1 mukaisesti yhdessä sanomassa voidaan lähettää halutun suuruinen määrä tietueita kerralla.

sFlow [11] on myös standardi, joka tulee esille vuotiedon keräämisen yhteydessä. Nimestään huolimatta sFlow ei kuitenkaan kerää tietoa vuotasolla, vaan tieto koostetaan SNMP monitoroinnin tyyppisistä laskureista ja pakettinäytteistä. Tiedot myös lähetetään heti eteenpäin tietoja keräävältä laitteelta, joten paketeista kerättäviä tietoja ei voida laitteessa koostaa useiden pakettien muodostamaa vuota kuvaavaksi informaatioksi. sFlow tiedonkeruun rooleja ja tietovirtoja on kuvattu kuvassa 5.



Kuva 5. *SFlow tiedon keräys ja lähetys [9]*

sFlow ei myöskään mahdollista kaikkien pakettien tarkastelua. Kyse on aina siis näytteiden ottamisesta eikä koko verkkoliikenteen tarkastelusta. [11]

3.4 Tiedon keruun haasteet

Liikennettä mittaavat laitteet on yleensä suunniteltu jotain muuta käyttötarkoitusta varten. Verkkolaitteet, kuten reitittimet ja kytkimet, on optimoitu pakettien tehokkaaseen välitykseen. Tietoturvalaitteet, kuten palomuurit ja IDS/IPS, on optimoitu uhkien havaitsemiseen ja torjuntaan. Koska verkkoliikenne kulkee näiden laitteiden kautta, ne ovat kuitenkin luonnolliset mittauspisteet ja sisältävät tarvittavia ominaisuuksia, kuten tuki IPFIX-monitoroinnille. Koska mittausten tekeminen ei ole laitteiden suunniteltu päätehtävä, voi toteutuksissa olla rajoitteita. Esimerkiksi kytkimen tai reitittimen liikenteen välityskyky voi olla suurempi kuin sen kyky mitata vuotietoa tai laitteen mittauskyky voi rajoittaa kerättävän tiedon sisältöä. Liikenteen analysoinnissa voidaan siis joutua miettimään joko erillisten mittalaitteiden käyttöä tai rajoitteiden hyväksymistä. Rajoitteita voivat olla esimerkiksi:

- Tiedon keräys on mahdollista vain rajoitetulla näytetiheydellä.
- Mitataan vain rajoitettua joukkoa verkkolaitteen liikenteestä, kuten tietystä verkkolaitteen portista tai virtuaaliverkosta kulkeva liikennettä.
- Verkkolaitteen välityskyky on alentunut normaalista liikennettä mitattaessa.

Mikäli mittapisteissä käytetään erillisiä mittalaitteita, tulee haasteeksi kaiken tarvittavan liikenteen näkyvyys mittalaitteelle. Nykyaikaisessa verkossa palvelut ja verkkoliikenteen reitit saattavat muuttaa sijaintiaan automaattisesti ja katkottomasti, joten redundantissa verkossa mittauspisteitä on oltava joka tapauksessa useita.

Liikennemäärien kasvaessa kasvaa myös mittaustapahtumien määrä. Jos halutaan hyödyntää historiatietoja, joudutaan ennen pitkää tilanteeseen, jossa liikenteestä kerättyä tietoa joudutaan tiivistämään tai poistamaan. Tallennusongelmiin on mahdollista vaikuttaa myös käyttämällä laitteiden kuormitustakin vähentävää näytteenottomenetelmää.

Mittaustietoa useasta pisteestä kerättäessä voivat tiedot samasta vuosta tulla keräyspisteeseen useasta eri lähteestä. Keräysjärjestelmän tehtävänä on tunnistaa samaan vuohon liittyvät tiedot. Tunnistamisessa auttaa tiedoissa oleva aikaleima. Jotta tietojen yhdistely onnistuu, pitää eri tietolähteiden sisäisten kellojen olla samassa ajassa keskenään. Mikäli kellot eivät ole samassa ajassa, mittaustuloksien käsittely ei onnistu tai se tuottaa virheellisiä tuloksia.

3.5 Uudenlaiset IT-palveluympäristöt ja tiedon keräys

Moderneissa verkko- ja palveluympäristöissä palvelut tuotetaan usein eri tavoilla riippuen kunkin palvelun tarpeesta. Osa palveluista voidaan tuottaa organisaation omilla resursseilla, osa paikallisesti kumppanin palveluna ja lisäksi osa palveluista voidaan toimittaa suoraan palveluntarjoajan ympäristöistä. Palveluntarjoajan palvelu voi olla monentyyppistä. Joskus käyttötarkoituksena on saada mukautuvaa kapasiteettia IaaS-alustapalveluista, joissa käyttöön saadaan sovittu määrä kapasiteettia. Kapasiteetin rajoitteita voivat olla muun muassa sovittu suorittimen kellotaajuus, muistimäärä, tallennuskapasiteetti ja -nopeus sekä verkkoyhteyden nopeus ja siirrettävän tiedon määrä. Toisiin palveluihin käytetään PaaS-konseptia, jossa varsinaiseen sovelluksen käyttöön otetaan palveluntarjoajan komponentteja. Tyypillisiä PaaS-palveluita ovat WWW-palvelimet ja tietokantapalvelut. Pisimmälle viety palvelutyyppi on SaaS eli kokonaisen ohjelmiston käyttöönotto palveluympäristöstä.

Verkkoliikenteen analysoinnin kannalta haastava piirre alustapalveluissa on automaattinen mukautuminen. Muutokset voivat perustua esimerkiksi ympäristöjen kuormituksen,

käyttäjien maantieteellisen sijainnin tai yksittäisen komponentin vikaantumiseen. Aiheutuva muutos voi siis olla virtuaalipalvelimen tai palvelun siirtyminen palvelimesta tai konesalista toiseen. Yksittäinen tarkkailtava liikennevuo voi siis tarkkailun aikana siirtyä osittain tai kokonaan uuteen ympäristöön.

Jaettujen tai palveluntuottajan hallinnoimien ympäristöjen toinen merkittävä piirre on rajoitettu näkyvyys verkkotason tapahtumiin. Rajoitettu näkyvyys liikenteeseen on ymmärrettävä, koska samalla fyysisellä alustalla voi olla useiden asiakkaiden yhteyksiä. SaaS-palveluissa näkyvyys rajoittuu tyypillisesti palvelun käyttäjien puolella tehtävään liikenteen mittaamiseen. PaaS-palveluissa palveluntarjoaja voisi tarjota liikenteen analysointiin palveluita. Näkyvyyden tarjoaminen vain tarjotun palvelun osalle ei ole yksinkertaista esimerkiksi WWW-palvelinalustasta, jossa voi olla kymmenien eri asiakkaiden palveluita. IaaS-palveluissa näkyvyys on mahdollista toteuttaa joko asiakkaan toimesta käyttöjärjestelmätasolla tai käyttäen palveluntarjoajan tarjoamia rajapintoja palvelun virtuaalisiin verkkokomponentteihin.

4. LIIKENTEEN MALLINTAMINEN

4.1 Tietueiden sisältö ja tallennus

Edellisissä luvuissa on käsitelty erilaisia tapoja kerätä tietoa verkkoliikenteestä. Kun tieto on kerätty ja siirretty keskitettyyn keräyspisteeseen tallennettavaksi, voidaan tietoa hyödyntää verkon toiminnan ymmärtämiseen ja verkon toiminnassa havaittujen muutosten havaitsemiseen. Koska esimerkiksi IPFIX mahdollistaa tiedon keräämisen erilaisilla tarkkuuksilla ja sisällöillä, voidaan kerättävän tiedon sisältöä sovittaa haluttuun käyttötarkoitukseen.

Vuotiedon keräämisen etuna, esimerkiksi liikenteen sisällön kaappaamiseen verrattuna, on pienemmän datamäärän lisäksi datasisällön puuttuminen. Kerättävässä informaatiossa ei ole käyttäjien tunnistetietoja eikä verkossa välitettävää tietoa. IP-osoitteet voidaan kuitenkin tietosuojalakien piirissä määritelty yksilöiviksi tunnisteiksi, koska niiden perusteella on mahdollista tunnistaa kulloinkin osoitetta käyttävät laitteet ja käyttäjät. Jos IP-osoitteita ei tietosuojasyistä voida käyttää, voidaan osoitteet poistaa tai pseudonymisoida. IP-osoitetiedon poisto tekee tiedon käytettävyydestä rajallista, mutta se on toimiva tapa esimerkiksi vuomäärän sekä kunkin vuon ajallisen keston ja datamäärän keräämiseen tilastollista analyysiä varten. Myös pseudonymisoinnissa voidaan menettää tietoa, jonka avulla analysoinnin tuloksia voitaisiin käyttää verkon käytettävyyden tai tietoturvan parantamiseen. [1]

Merkittävä tekijä tiedon hyödynnettävyydessä on tiedon tallennustapa. Yleisesti käytettyjä tallennusvaihtoehtoja ovat tiedostopohjainen tallennus, relaatiotietokannat tai sarakkeiset tietokannat. Tallennustapaa valittaessa merkittäviä tekijäitä ovat tiedon tallennuksen nopeus, tiedon hakunopeus sekä tallennettavan tiedon määrä. Tiedostopohjaisen tallennuksen etuna voi joissakin käyttötarkoituksessa olla tiedostojen kronologinen vaihtuvuus. [1] Tietyltä aikaväliltä tehtävät haut on tällöin jo mahdollista rajoittaa suoraan tiettyihin tiedostoihin. Tiedostopohjaisessa tallennuksessa ei myöskään tarvita ylimääräistä tilaa tietokannan indekseille kuten relaatiotietokannoissa. Relaatiotietokannan ja tiedostotallennuksen suorituskykyä vertailtaessa, tiedostopohjainen tallennustapa oli myös hakuajoiltaan nopeampi kaikissa testitapauksissa [13]. Vastaavalla tavalla tiedostopohjaista ja sarakkeista tietokantaa verrattaessa [14], sarakepohjainen tietokanta osoittautui nopeammaksi kaikissa testitapauksissa. Nopeampi toiminta selittyy sillä, että

sarakepohjaisesta tietokannasta voidaan hakea vain halutut sarakkeet koko sisällön sijaan. Tarvittavien I/O operaatioiden määrä jää siis pienemmäksi kuin tiedostopohjaisessa tallennuksessa.

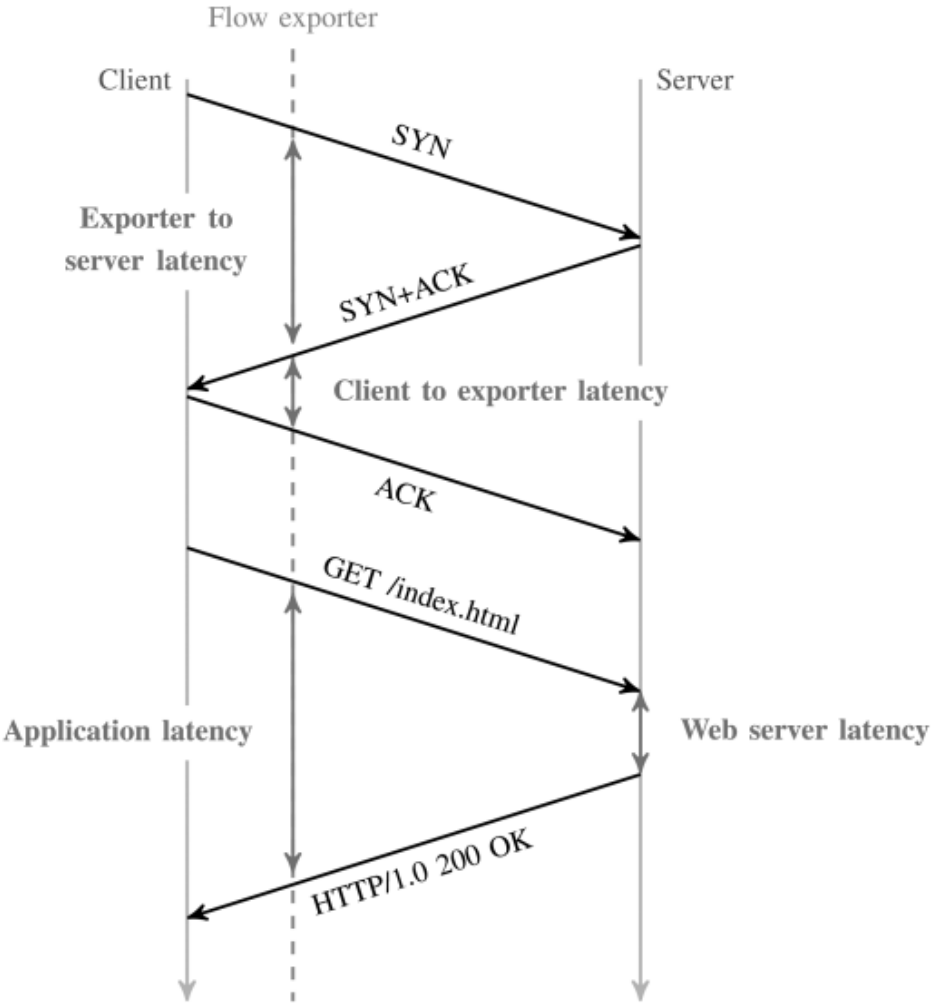
4.2 Mallinnuksella haetut hyödyt

Yksinkertaisimmillaan vuotiedon hyödyntäminen voi tarkoittaa verkossa kulkevan liikenteen visualisointia. Toiminnallisuuksia voivat olla esimerkiksi:

- Yksittäisten tietueiden selaus ja suodatus
- Tilastolliset liikennetiedot esimerkiksi protokolla ja palvelintasolla
- Yksittäisten osoitteiden tai palveluiden liikenneraportit
- Hälytykset esimerkiksi suuresta määrästä yhteyksiä

Ylläolevat ominaisuudet ovat yksinkertaisia toteuttaa ja niihin löytyy valmiina myös avoimen lähdekoodin toteutuksia. [1] Tulosten ymmärtäminen vaatii kuitenkin ymmärrystä tiedon keräyksestä ja keräyspisteestä. Esimerkiksi kytkinverkossa tyypillinen tiedon keräyspiste olisi runkokytkimessä tai reitittimessä, jonka läpi suurin osa verkkoliikenteestä kulkee. Tällöin ei kuitenkaan nähdä esimerkiksi samassa kokoojakytkimessä ja virtuaaliverkossa kiinni olevien laitteiden välistä liikennettä.

Toinen melko yksinkertainen, mutta hyödyllinen, tapa käyttää vuotietoa on palveluiden suorituskyvyn mittaus. Yksinkertaisessa mallissa verkkoviive mittauspisteestä palvelimelle voidaan mallintaa esimerkiksi TCP-yhteyden kättelyssä näkyvästä viiveestä. Tämän jälkeen voidaan palvelimelle suunnattujen pyyntöjen palvelimella käytettyä vasteaikaa arvioida palvelupyynnön ja verkkoviipeen erotuksena. Yksittäisten pyyntöjen ja vastausten vasteaikoja ei kuitenkaan ole käytössä, mikäli käytetään pelkästään standardin vaatimia ominaisuuksia toteuttavaa IPFIX tai NetFlow tiedon kerääjää. [1]



Kuva 6. *Esimerkki palvelimen vasteaikojen mittauksesta mukaillen lähdettä [15]*

Tyypillisiä verkkosovelluksia, joista vasteaikojen mittaus olisi hyödyllistä, olisivat WWW-palvelimet ja nimipalvelimet. Yksittäisten pyyntöjen tarkkailu johtaa mittauksessa käytettyjen oletusten takia helposti harhaan johtaviin tuloksiin. Esimerkiksi nimipalvelupyyntöjen kohdalla vastaus voi tulla joko suoraan nimipalvelimen välimuistista, saman organisaation toiselta nimipalvelimelta tai usean Internetin nimipalvelimille tehdyn rekursiivisen kyselyn tuloksena. Jos seurattavana kohteena on vasteaikojen keskiarvo, saadaan vertailukelpoisia arvoja esimerkiksi historiatietoon vertailtavaksi. Tietoja voidaan hyödyntää esimerkiksi verkkoon tai sen palveluihin tehtyjen muutoksien vaikutuksia arvioitaessa. Mikäli ympäristö tunnetaan riittävän hyvin, voidaan tietojen perusteella myös arvioida loppukäyttäjille toimitetun palvelun viiveiden kokonaismuodostusta ja ohjata parannuksia oikeisiin kohteisiin.

Kolmas sovellusalue vuotiedon hyödyntämisessä on tietoturvauhkien havaitseminen. Työkaluina uhkan havaitsemiseen voi tällöin olla esimerkiksi jokin seuraavista.

- Valmis tunniste yksittäiselle tunnetulle hyökkäykselle tai haavoittuvuudelle

- Määritelty palvelimeen kohdistuvien yhteyksien määrä
- Yksittäisestä laitteesta toisiin laitteisiin lähtevien yhteyksien määrä
- Verkkoon kuulumattoman protokollan havaitseminen
- Liikenne toisilleen normaalisti liikennöimättömien laitteiden välillä

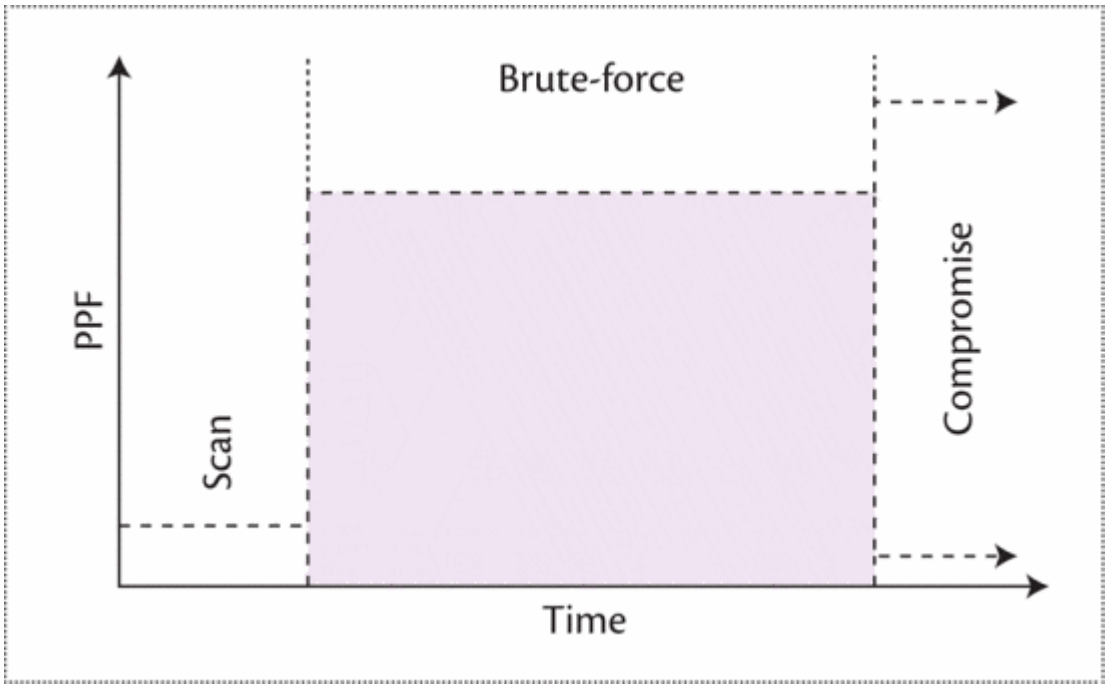
Tunnisteiden kautta tehtävä hyökkäysten havainnointi on kehittyneempää IDS/IPS laitteilla, joille on tarjottu koko liikenne analysoitavaksi. Nämä laitteet voivat olla joko erillisiä verkkokomponentteja tai palomuurien lisäominaisuuksia. Koska käytettävissä on paljon monipuolisempi näkymä analysoitavaan liikenteeseen, tunnisteiden tekeminen on suoraviivaisempaa ja monipuolisempaa kuin vuotiedon pohjalta tehtävissä tunnisteissa. [12]

Tilanteissa, joissa yksittäiset yhteydet ovat normaalia verkkoliikennettä, mutta yhdessä muodostavat uhan palvelulle, vuopohjainen tarkastelu on tehokkaampaa. Tällaisia tilanteita ovat ainakin palvelunestohyökkäykset tai verkon reunan suojauksesta läpi päässyt hyökkääjä kartoittamassa verkkoa etenemistä varten.

4.3 Sovelluksen ominaisuuksiin perustuva mallinnus

Yksinkertaisimmillaan liikenteen mallinnuksessa etsitään yksittäisten sovellusten toistuvia malleja. Yksi toimivaksi osoitettu esimerkki hyökkäyksen ja sen onnistumisen tunnistamisesta on SSH-palvelimeen kohdistetun brute-force-salasanahyökkäyksen havaitseminen. Hyökkäyksessä hyökkääjä käy ensin läpi verkkoja löytääkseen avoimia ssh-palvelimia. Kun ssh-palvelin on paikallistettu, hyökkääjä aloittaa tunnettujen käyttäjätunnusja salasanakombinaatioiden kokeilun palveluun kirjautumiseen. Hyökkäyksen lopputuloksena on joko hyökkäyksen päättyminen tuloksettomana tai hyökkääjän kirjautuminen palvelimelle. [12]

Koska ssh-yhteys on kryptografian keinoin suojattu, ei yhteyksien sisällön perusteella ole mahdollista havaita epäonnistuneita kirjautumisyrityksiä. Hyökkäyksen mallintaminen perustuu PPF (packets per flow) arvon tarkkailuun havaituissa ssh-yhteyksissä. Palvelua skannattaessa hyökkääjä lähettää vain SYN-paketteja tunnettuun ssh-porttiin ja odottaa vastausta. Kun brute-force-hyökkäys aloitetaan, tapahtuu yhteyden yli ensin yhteysavainten vaihto, jonka jälkeen hyökkääjä kokeilee maksimimäärän tunnistautumisia ennen yhteyden katkaisua. Mikäli tunnistautuminen onnistuu, tuloksena on pitempi yhteys, jolla on suurempi PPF arvo. Teoreettinen malli hyökkäyksen etenemisestä on kuvattu kuvassa 7. [12]



Kuva 7. *Teoreettinen malli SSH Brute-Force-hyökkäyksestä lähteestä [12]*

Vaikka malli itsessään on yksinkertainen, se vaatii toimiakseen hyvin toimivaa tiedonkeruujärjestelmää. Esimerkiksi jo yhden tietueen puuttuminen voi tarkoittaa hyökkäyksen onnistumisen jäämistä havaitsematta. Vastaavasti avoimien vuotietueiden puutteellinen poisto voi johtaa seuraavan yhteyden liikenteen laskemisen mukaan edelliseen yhteyteen. Tässä mallinnustavassa se tarkoittaisi, että hyökkäyksen todetaan onnistuneen, vaikka näin ei todellisuudessa olisi. [12]

Vastaavaa hyökkäyksentunnistusta on mahdollista tehdä myös Web Hosting-ympäristöissä. Hyökkäyksen tunnistamiseen käytettävä sekvenssi on jo paljon monimutkaisempi ja perustuu havaituissa yhteyksissä siirrettävän datan määrän mallintamiseen. Lisäksi pohjana käytetään riittävää määrää yrityksiä, jotta esimerkiksi hakukoneiden sivustoja läpikäyvät toiminnat eivät aiheuttaisi vikahälytyksiä. [16]

Molemmat edellisen esimerkin havainnoista voitaisiin tehdä myös hyökkäyksen kohteena olevan palvelimen lokeista. Tiedon käsittely edellyttää kuitenkin joko jokaiseen palvelimeen asennettua ohjelmistoa hyökkäyksien tunnistamiseen sekä niistä ilmoittamiseen tai keskitettyä lokienhallintajärjestelmää, johon lokitieto lähetetään analysoitavaksi. Vuotietoa käyttämällä esimerkiksi alustapalvelua tarjoava palveluntarjoaja voi kuitenkin havaita asiakkaan huonoista salasanakäytännöistä johtuvat ongelmat, jotka saattaisivat muuten vaarantaa koko ympäristön käytettävyyden, ilman pääsyä asiakkaan palvelimen käyttöjärjestelmätasolle.

4.4 Verkkosegmentin liikenteen mallinnus

Monimutkaisempia vuomalleja voidaan muodostaa yksittäisen verkkosegmentin tai mittauspisteen havaitusta liikenteestä. Mallinnuksen tavoitteena on näissä tapauksissa malli, jolla voidaan tarkkailla tietyn verkon osan toimintaa vertaamalla mittauksista tulevaa tietoa malliin. Tavoitteena olevat havainnot voidaan jakaa kahteen ryhmään poikkeamia, joita ollaan havaitsemassa. Ensimmäinen poikkeamatyyppi on verkon yleistä toimintaa mallintava malli, johon vertaamalla voidaan havaita poikkeamia esimerkiksi kuormituksen muutosten tai vikaantuneiden verkon laitteiden aiheuttamina [18]. Toisen poikkeamaryhmän muodostavat tietoturvaan liittyvät poikkeamat. [17]

Verkon yleistä toimintaa mallinnettaessa, poikkeamat ovat muutoksia, jotka tapahtuvat mallinnetussa liikenteessä. Yksinkertainen esimerkki mallinnuksesta on WWW-palvelin, jolle tulevien pyyntöjen määrä ja niiden keskimääräinen vasteaika on mallinnettu. Mikäli liikenteen määrä muuttuu mallissa ennustetusta, on kyseessä poikkeama. Samassa mallissa voidaan mallintaa myös palvelun taustalla olevan tietokannan liikennettä ja vasteaikoja. Tällaisella mallinnuksella on mahdollista havaita toiminnallisia poikkeamia, kuten taulukossa 1 esitetyt.

Poikkeama	**Mahdollinen syy**	**Toimenpiteet**
Kyselyiden lisääntyminen +20% mallista	Käytön lisääntyminen	Kapasiteetin lisäys ja mallin päivitys
Kyselyiden vasteajan pitenteminen	Palvelimen tai tietokannan vikatilanne	Palvelimen ja tietokannan vikaselvitys
Vastaamattomien yhteyksien määrä	Laitevika tai nimipalveluongelma	Yhteyksien sisällön tarkempi vikaselvitys
Yhteyksien väheneminen 10%:iin mallista	Ongelma Internet-liikenteen reitityksessä tai nimipalvelussa	Internet-liittymän vikaselvitys
Porttiin TCP443 tulevien yhteyksien lyhyt kesto	SSL-sertifikaatin ongelmat	SSL sertifikaatin voimassaolon ja nimien tarkistus

Taulukko 1: Esimerkkejä WWW-palvelun poikkeamista

Yksinkertaisessakin mallissa tulee sekä mallinnuksesta, että mallin sisällä määritellyistä poikkeamaksi luokiteltujen muutoksien määrittelystä monimutkainen kokonaisuus. Mallissa tulee huomioida esimerkiksi mallin ajalliset vaihtelut esimerkiksi vuorokaudenaikojen, viikonpäivien tai juhlapyhien mukaan.

Tietoturvapoikkeamien havainnoinnissa poikkeamat voivat olla joko muutoksia mallinnetun liikenteen sisällä tai kokonaan mallin ulkopuolista liikennettä. Tietoturvapoikkeamien havainnointia liikenteen mallinnuksen avulla tekeviä järjestelmiä kutsutaan ANIDS-järjestelmiksi (anomaly-based network intrusion detection system). [17]

ANIDS järjestelmissä verkon mallinnus voi tapahtua usealla erilaisella tavalla. Valvotussa (supervised) mallissa oletetaan, että kaikki mahdollinen normaali sekä erilaiset tietoturvapoikkeamat ovat mukana mallin pohjana olevassa esimerkkitiedossa. Kaikki mallin mukaan analysoitava liikenne on luokiteltavissa joko tiettyyn poikkeamaan tai normaalin liikenteen malliin. Osittain valvotussa (semi-supervised) mallinnetaan vain tiedossa oleva normaali liikenne ja kaikki malliin sopimaton liikenne tulkitaan poikkeamiksi. Valvomattomassa (unsupervised) mallissa järjestelmä ei tarvitse pohjatietoa liikenteestä ollenkaan, vaan mallintaminen perustuu yleisiin malleihin. ANIDS toteutukset voivat olla myös hybridimalleja, joissa hyödynnetään useita edellä mainituista menetelmistä. [17]

Poikkeamien etsinnässä voidaan käyttää apuna myös moderneja tiedon louhintamekanismeja (data mining). Tiedon louhinnan avulla voidaan saada erotettua tietoa esimerkiksi analysoitavien kohteiden ryhmittelystä ja samankaltaisuudesta, kohteiden luokittelusta ennalta määritetyn ryhmittelyn mukaan tai erityisistä kohteista, jotka poikkeavat kaikista muista.

5. YHTEENVETO

Vuotiedon kerääminen on hyvin standardoitu ja eri laitteissa tuettu menetelmä tiedon keräämiseen verkon toiminnasta. Tiedon kerääminen onnistuu niin laitevalmistajien omien toteutusten, kuin avoimena lähdekoodina saatavilla olevien kirjastojen avulla. Myös edelleen lähetetyn vuotiedon koostamiseen ja analysointiin löytyy monipuolisesti työkaluja niin kaupallisina tuotteina kuin avoimen lähdekoodin kirjastoinakin. Tuotteiden ominaisuudet ovat kuitenkin painottuneet liikenteen visualisointiin joko protokollatason jakaumina, liikennemäärinä tai osoitteiden välisinä yhteyskaavioina.

Vuotiedon kerääminen on kuitenkin tullut ajankohtaiseksi tavaksi tehdä myös tietoturvapoikkeamien havainnointia. Kasvaneet liikennemäärät suosivat monitorointia, jossa valvontatiedon määrä on pieni suhteessa liikenteen kokonaismäärään. Yleistynyt salauksen käyttö organisaatioiden sisäisessäkin liikenteessä tekee puolestaan itse datasisällön analysoinnista haastavampaa ja samalla tarkentuneet tietosuojakäytännöt rajoittavat tietojen käsittelyä. Verkot voivat myös olla maantieteellisesti hajallaan, jolloin liikenteen sisältöä analysoivaa toiminnallisuutta pitäisi olla joko päätelaitteissa itsessään tai verkkolaitteina aina verkon laidoilla asti.

Verkkojen mallinnukseen on viime vuosina tullut niin tutkimustietoa kuin valmiita teknisiä apukeinoja. Kehittyneet tiedonlouhinta- ja koneoppimismenetelmät mahdollistavat verkkojen mallintamisen ja eri lähteistä tulevan tiedon yhdistämisen. Järjestelmät mahdollistavat myös aivan uudenlaisien riippuvuuksien ja muutosten löytämisen automaattisesti luotujen mallien ja analysoitavan liikenteen välille. Mallinnuksen avulla saadaan nostettua suuresta tietomäärästä olennaisimmat asiat esiin ja tietoja voidaan yhdistellä esimerkiksi perinteisempien tietoturvakontrollien, kuten palomuurien ja päätelaitteiden tietoturvaohjelmistojen syötteiden kanssa.

Jatkotutkimuskohteena juuri mallinnuksien tulosten analysointi ja yhdistely muiden lähteiden kanssa olisi kiinnostava tutkimuskohde. Vaikka tiedon keräysmenetelmät ovat hyvällä tasolla ja mallinnukseen löytyy teoriatutkimusta, voitaisiin verkon tilatiedon jalostamista tietoturvatiedon ja -tapahtumien hallintajärjestelmiin (SIEM) kehittää edelleen. Näin saataisiin tietoturvallisuuden valvojille (SOC) mahdollisimman hyvä näkyvyys verkkoon myös silloin, kun tietoturvan ja verkon operointi on teknisesti ja hallinnollisesti erillään.

LÄHTEET

[1] HOFSTEDE, R., CELEDA, P., TRAMMELL, B., DRAGO, I., SADRE, R., SPEROTTO, A. and PRAS, A., 2014a. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. IEEE Communications Surveys & Tutorials, 16(4), pp. 2037-2064

[2] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, https://www.rfc-editor.org/info/rfc7011

[3] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1067, DOI 10.17487/RFC1067, August 1988, https://www.rfc-editor.org/info/rfc1067

[4] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to version 2 of the Internet-standard Network Management Framework", RFC 1441, DOI 10.17487/RFC1441, April 1993, https://www.rfc-editor.org/info/rfc1441

[5] H. LIN, Z. YAN, Y. CHEN and L. ZHANG, 2018. *A Survey on Network Security-Related Data Collection Technologies. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8320776*

[6] B. TRAMMELL and E. BOSCHI, 2011. An introduction to IP flow information export (IPFIX). IEEE Communications Magazine (Volume: 49 , Issue: 4 , April 2011)

[7] IDC, IDC press release 31.5.2019, https://www.idc.com/getdoc.jsp?containerId=prUS45119319, ladattu 27.11.2019

[8] Kerr DR, Bruins BL. Network flow switching and flow data export. US Patent Number US 6243667, 2001.

[9] Li, B., Springer, J., Bebis, G. & Hadi Gunes, M. 2013, "A survey of network flow applications", Journal of Network and Computer Applications, vol. 36, no. 2, pp. 567-581.

[10] TRAMMELL, B. and BOSCHI, E., 2011. An introduction to IP flow information export (IPFIX). *IEEE Communications Magazine,* **49**(4), pp. 89-95.

[11] P. Phaal, sFlow Version 5, 2004., https://sflow.org/sflow_version_5.txt

[12] HOFSTEDE, R., PRAS, A., SPEROTTO, A. and RODOSEK, G.D., 2018. Flow-Based Compromise Detection: Lessons Learned. *IEEE Security & Privacy,* **16**(1), pp. 82-89.

[13] R. Hofstede, A. Sperotto, T. Fioreze, A. Pras, "The network data handling war: MySQL vs NfDump", Proc. 16th EUNICE, vol. 6164, pp. 167-176, 2010.

[14] P. Velan, "Practical experience with IPFIX flow collectors", Proc. 13th IFIP/IEEE Int. Symp. IM, pp. 1015-1020, 2013.

[15] L. Deri, How to Monitor Latency Using nProbe, 2011., https://www.ntop.org/nprobe/how-to-monitor-latency-using-nprobenagios-world-conference-europe/, ladattu 27.11.2019

[16] HOFSTEDE, R., JONKER, M., SPEROTTO, A. and PRAS, A., 2017. Flow-Based Web Application Brute-Force Attack and Compromise Detection. *Journal of Network and Systems Management,* **25**(4), pp. 735-758.

[17] Bhuyan, M.H., Bhattacharyya, D.K. & Kalita, J.K. 2017, Network Traffic Anomaly Detection and Prevention : Concepts, Techniques, and Tools, Springer International Publishing AG, Cham.

[18] Thottan, M. & Ji, C. 2003, "Anomaly detection in IP networks", IEEE Transactions on Signal Processing, vol. 51, no. 8, pp. 2191-2204.