jamk.fi

Avoimen lähdekoodin työkalujen tutkiminen ja vertailu tilanne-tietoisuutta varten poikkeaman-hallinnassa

Jere Heimonen

Opinnäytetyö
Huhtikuu 2017
Tekniikan ja liikenteen ala
Insinööri (AMK), tietotekniikan (tietoverkkotekniikan) koulutusohjelma

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Heimonen, Jere	Opinnäytetyö, AMK	18.4.2017
	Sivumäärä 115	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: kyllä

Työn nimi

Avoimen lähdekoodin työkalujen tutkiminen ja vertailu tilannetietoisuutta varten poikkeamanhallinnassa

Tutkinto-ohjelma

Tietotekniikan koulutusohjelma

Työn ohjaaja(t)

Tero Kokkonen, Sampo Kotikoski

Toimeksiantaja(t)

Marko Vatanen, JYVSECTEC

Tiivistelmä

Opinnäytetyön toimeksiantajana toimi Jyväskylän ammattikorkeakoulun JYVSECTEC kyberturvallisuuden tutkimus-, kehitys- ja koulutuskeskus. JYVSECTEC tarjoaa kyberturvallisuuteen liittyvää harjoitustoimintaa, konsultointia, testausta, tutkimusta ja koulutusta.

Opinnäytetyön toimeksiantona oli vertailla kahta työkalua tietoturvapoikkeamien kirjaamiseen sekä perehtyä poikkeamanhallinnan toimintaan. Työkalut, joita vertailtiin, olivat TheHive ja FIR (Fast Incident Response). Näistä työkaluista tuli tutkia ja vertailla niiden ominaisuuksia sekä rajapintoja tiedon tuontiin järjestelmään ja tiedon vientiin järjestelmästä. Tässä oli tarkoituksena saada selville saako järjestelmään tuotua monitorointidataa ja saadaanko järjestelmästä vietyä dataa ulos järkevässä muodossa toisiin työkaluihin/järjestelmiin, tässä tapauksessa aikajana/visualisointityökaluun.

Opinnäytetyön tuloksina oli TheHive- ja FIR-järjestelmien esittely ja ominaisuuksien dokumentointi, oleellisten ominaisuuksien vertailu ja tiedon tuonnin ja viennin testaaminen järjestelmistä. Ominaisuuksien vertailussa kummassakin järjestelmässä oli hyödyllisiä ja oleellisia ominaisuuksia, joita ei toisessa järjestelmässä ollut. Tämän takia on hankala valita, kumpi näistä järjestelmistä on parempi. Järkevintä on valita käyttötilanteen mukaan parempi järjestelmä. Kuitenkin TheHive-järjestelmä vaikuttaa selkeämmältä ja havainnollisemmalta. Tiedon tuontiin ja vientiin ei ollut käyttöliittymässä toimintoja kummassakaan järjestelmässä, eikä tiedon tuonti onnistu kuin manuaalisesti tapaus ja havainto kerrallaan. Tiedon vienti testattiin TheHive:ssa API-rajapinnan kautta HTTP:n välityksellä ja FIR:ssä MySQL-tietokannan kautta PHP-skriptillä. Tiedon vienti onnistuu molemmista järjestelmistä sellaisessa muodossa, että sitä saa vietyä mahdolliseen aikajanatyökaluun.

Avainsanat (asiasanat)

Poikkeamanhallinta, Tilannetietoisuus, CSIRT (Poikkeamanhallintatiimi, Computer Security Incident Response Team), CIRT (Poikkeamanhallintatiimi) Computer Incident Response Team)

Muut tiedot

jamk.fi

Description

Author(s)	Type of publication	Date
Heimonen, Jere	Bachelor's thesis	18.4.2017
		Language of publication: Finnish
	Number of pages: 115	Permission for web publication: yes

Title of publication

Research and comparison of open source tools for situational awareness in incident response

Degree programme

Information Technology

Supervisor(s)

Kokkonen, Tero; Kotikoski, Sampo

Assigned by

Vatanen, Marko: JYVSECTEC

Abstract

The thesis was assigned by JYVSECTEC and implemented at JAMK University of Applied sciences. JYVSECTEC is a Cyber security research, development and training center. The services offered by JYVSECTEC are cyber security exercises, consulting, testing, research and training.

The assignment was to compare two systems for logging incidents and to explore incident response process. The compared tools were TheHive and FIR (Fast Incident Response). The goal was to compare the features of the systems. Another goal was to research the APIs for importing and exporting date to the systems. The purpose of this was to find out if one could import monitoring data to the systems and export data in a suitable format to, for example, a visualization/timeline tool.

The thesis resulted in introducing TheHive and FIR systems, documenting their features, comparing their essential features and testing importing and exporting data to both systems. While comparing the features, it was found out that both systems have essential and beneficial features that the others system does not have. This made it hard to decide which one of the systems is better; therefore, it is sensible to select the system based on which one performs better in the use case. TheHive system seemed to be clearer and more informative. Neither of the systems had data importing and exporting features in their user interface, and data can be input manually, only one incident case or observation at a time. In TheHive, exporting of the data was tested using TheHive's API interface through HTTP, and in FIR, the exporting of the data was tested through MySQL database using a PHP script. In both systems, it is possible to export data in a such format that enables its export to a possible timeline tool.

Keywords/tags (subjects)

Incident Response, Situational Awareness, CSIRT (Computer Security Incident Response Team), CIRT (Computer Incident Response Team)

Miscellaneous

Sisältö

Kuviot

Taulukot

Lyhenteet

API	Application Programming interface
CERT	Computer Emergency Response Team
CFT	Cross Functional Team
CIRT	Computer Incident Response Team
CSIRT	Computer Security Incident Response Team
DDoS	Dynamic Denial of Service
DNS	Domain Name System
DoS	Denial of Service
FIR	Fast Incident Response
FTK	Forensic Tool Kit
HTTP	Hyper Text Transfer Protocol
IDS	Intrusion Detection System
IOC	Indicator of Compromise
IP	Internet Protocol
IR	Incident Response
IRP	Incident Response plan
IT	Information Technology
JSON	JavaScript Object Notation
JYCSECTEC	Jyväskylä Security Technology
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OODA	Observe, Orient, Decide, Act
PHP	Hypertext Preprocessor
PR	Public Relations
RAM	Random Access Memory
RGCE	Realistic Global Cyber Environment
SA	Situational Awareness
SANS	Escal Institute of Advanced Technologies
SQL	Structured Query Language
TLP	Traffic Light Protocol
URL	Uniform Resource Locator

VPN	Virtual Private Network

1 Lähtökohdat ja toimeksianto

Tämän opinnäytetyön toimeksiantajana toimi JYVSECTEC (Jyväskylä Security Technology). JYVSECTEC on kyberturvallisuuden tutkimus-, kehitys- ja koulutuskeskus, joka toimii osana Jyväskylän ammattikorkeakoulun IT-instituuttia. JYVSECTEC:n palveluita ovat kyberturvallisuuteen liittyvä harjoitustoiminta, konsultointi, testaus, tutkimus ja koulutus. JYVSECTEC:llä on RGCE-ympäristö kyberturvallisuuden tutkimusta, kehitystä ja koulutusta varten. RGCE on lyhenne sanoista Realistic Global Cyber Environment. Tämä on eristetty ja kontrolloitu ympäristö, joka toimii kuin oikea internet ja sinne voidaan simuloida todellista verkkoliikennettä. (JYVSECTEC n.d.)

Alun perin toimeksiantona oli tutkia ja vertailla avoimen lähdekoodin aikajana-työkaluja poikkeamienhallinnan tueksi. Työkalua oli tarkoitus käyttää kyberturvallisuusharjoituksissa tilannetietoisuuden saavuttamiseen. Työssä piti myös tutustua Incident Response -toimintaan ja siitä tuleviin vaatimuksiin erilaisille järjestelmille, joista tutkittiin aikajana-työkaluja. Aikajana-työkalun lisäksi piti löytää järjestelmä tietoturvapoikkeamien kirjaamiseen. Tämä alkuperäinen suunnitelma kuitenkin muuttui matkan varrella, koska alustavan tutkimuksen perusteella incident responseen sopivia aikajanatyökaluja ei oikein ole saatavilla. Toimeksiantajan kanssa sovittiin, että muutetaan työn tavoitetta.

Työn lopullisena tavoitteena oli vertailla kahta järjestelmää tietoturvapoikkeamien kirjaamiseen. Vertailtavat järjestelmät olivat TheHive ja FIR (Fast Incident Response). Näistä järjestelmistä tutkittiin ominaisuudet sekä rajapinnat tiedon tuomiseen (import) järjestelmään ja tiedon viemiseen (export) järjestelmästä. Esimerkiksi miten monitorointidataa saadaan tuotua järjestelmään ja miten saadaan tietoa ulos esimerkiksi johonkin visualisointityökaluun. Työssä perehdyttiin myös incident responsen toimintaan teoria-osuudessa. Aikajana/visualisointityökalulle on JYVSECTEC:llä tarvetta kyberturvallisuusharjoitustoiminnassa. Tämä työkalu saatetaan ohjelmoida JYVSECTEC:n toimesta myöhemmin. Tässä työssä oli nyt oleellista tutkia, saadaanko järjestelmistä esimerkiksi niiden tietokannan kautta tietoa ulos järkevässä muodossa,

joka saataisiin tuotua johonkin muuhun työkaluun tai ohjelmaan (tässä tapauksessa käytännössä aikajana/visualisointityökalu) järkevästi.

2 Tutkimusasetelma ja tutkimusmenetelmät

Tutkimusongelmana opinnäytetyössä oli se, että JYVSECTEC:llä ei ollut järjestelmää, jolla pystyy tarkastelemaan eri tapahtumia aikajanalta. Tällaisesta aikajana-työkalusta on apua tilannekuvan ja tilannetietoisuuden muodostamisessa poikkeamanhallinnassa. Tavoitteena oli löytää vastaus seuraaviin kysymyksiin:

- Onko olemassa avoimen lähdekoodin aikajana-työkaluja, jotka sopivat tähän käyttötarkoitukseen?
- Jos sopivia työkaluja löytyy, mikä niistä on paras tähän käyttötarkoitukseen?
- Onko olemassa ”all-in-one” -työkalua, jolla voi sekä kirjata että tarkastella tapahtumia aikajanalla?
- Miten aikajana-työkalun saa liitettyä kirjaamisjärjestelmään?
- Mitä ominaisuuksia on TheHive ja FIR-järjestelmissä?
- Miten TheHive ja FIR järjestelmiin saa tuotua dataa ja vietyä dataa muihin järjestelmiin?
- Saako dataa ulos sellaisessa muodossa, että sen saa vietyä johonkin muuhun järjestelmään?

Kun tutkimuskysymykset on asetettu, voidaan määritellä käytettävät tutkimusmenetelmät. Yleinen tapa kuvata tutkimusmenetelmiä on käyttää kvantitatiivista ja kvalitatiivista tutkimusta.

Usein tutkimuksessa käytetään sekä kvantitatiivista että kvalitatiivista tutkimusmenetelmää. Kvantitatiivinen ja kvalitatiivinen lähestymistapa on käytännössä vaikea erottaa selkeästi toisistaan. Niitä ei tulisi nähdä kilpailevina menetelminä, vaan menetelminä, jotka voivat täydentää toisiaan eri tavoilla. Yksi tapa on, että kvantitatiivinen vaihe edeltää kvalitatiivista vaihetta. (Hirsjärvi, Remes & Sajavaara 2009, 136-137.)

Kvantitatiivisen tutkimuksen voi ajatella käsittelevän numeroita ja kvalitatiivisen merkityksiä. Numerot ja merkitykset ovat kuitenkin riippuvaisia toisistaan vastavuoroisesti. Mittaaminen sisältää aina kvantitatiivisen puolen, jossa mitataan jotain ja kvalitatiivisen puolen, jossa mietitään mittaustulosten merkityksiä. (Hirsjärvi ym. 2009, 137.)

Kvantitatiivisessa tutkimuksessa on oleellista tehdä johtopäätöksiä aiemmista tutkimuksista, tutkia aiempia teorioita, määritellä käsitteitä ja tehdä hypoteeseja. Oleellista on myös suunnitella koejärjestely ja aineiston keruu, siten että havaintoaineisto soveltuu määrälliseen, numeeriseen mittaamiseen. Tämä aineisto tulee sitten esittää siten, että se on tilastollisesti käsiteltävässä muodossa. Tästä tehdään sitten päätelmiä tilastollisen analyysin perusteella. (Hirsjärvi ym. 2009, 140.)

Opinnäytetyön käytännön toteutusosassa voidaan ajatella käytettävän kvalitatiivista tutkimusta. Tässä tehdään tiedon hankintaa todellisissa tilanteissa, sekä luotetaan omiin havaintoihin, eikä mittausvälineillä hankittavaan tietoon. Tämän opinnäytetyön toteutusosassa todellinen tilanne tarkoittaa käytännössä tutkittavien järjestelmien ominaisuuksien testaamista käytännössä ja havaintojen tekemistä niistä dokumentointia varten. (Hirsjärvi ym. 2009, 164.)

3 Incident Response eli poikkeamanhallinta

3.1 Incident eli poikkeama ja Security Incident eli tietoturvapoikkeama

Jotta voidaan ymmärtää poikkeamanhallinta, tulee ymmärtää ensin mikä on poikkeama. Incident eli poikkeama on tietotekniikassa tapahtuma, joka ei ole osa normaalia toimintaa ja häiritsee yrityksen toimintaprosessia tai keskeyttää sen. Poikkeama voi sisältää esimerkiksi jonkin palvelun häiriön tai epäonnistumisen. Tietoturvapoikkeamat ovat tapahtumia, jotka ilmaisevat, että yrityksen järjestelmät tai data saattavat olla vaarantuneet. Poikkeamiin sisältyvät niin pienet häiriöt, kuten kiintolevytilan loppuminen, kuin suuret häiriöt, kuten tietomurrot joissa arkaluontoinen data on vaarantunut. (Rouse 2008.)

3.2 Tapahtuman ja poikkeaman ero

Tapahtuma eli Event on havainnoitava tapahtuma, joka tapahtui järjestelmässä johonkin aikaan. Tapahtuma voi esimerkiksi olla sähköposti, puhelu, järjestelmän kaatuminen tai pyyntö tehdä virusskannaus tiedostolle. (Pham 2001.)

Poikkeama eli Incident on haitallinen tapahtuma järjestelmässä tai tapahtuma, joka sisältää merkittävän uhan haitalliseen tapahtumaan. Eli käytännössä joku yrittää aiheuttaa tai aiheuttaa haittaa yritykselle. (Pham 2001.)

Poikkeamia voivat olla ainakin seuraavat tapahtumat:

1. Yrityksen tietoturvakäytänteiden loukkaus eli niiden vastainen käyttö
2. Yritys saada luvaton pääsy järjestelmään
3. Resurssit eivät ole saatavilla
4. Luvaton käyttö
5. Muutokset omistajan tietämättä, ilman hänen määräystään tai suostumustaan

Tapahtuma voi muuttua poikkeamaksi, mutta ei päinvastoin. Tässä kannattaa noudattaa yksinkertaista ohjetta: Jos jokin epäilyttää, raportoi siitä. Seuraavana on esitetty joitakin esimerkkejä tapahtuman ja poikkeaman erosta (Pham 2001):

- Hyökkäys haitallisella koodilla
    - Tapahtuma
        - Käyttäjä raportoi, että on saattanut joutua tietokoneviruksen kohteeksi
    - Mahdollinen Poikkeama
        - Käyttäjän järjestelmä osoittaa kyseiselle virukselle tyypillistä käytöstä
- Resurssi ei ole saatavilla
    - Tapahtuma
        - Käyttäjä ilmoittaa, ettei pääse palveluun
    - Mahdollinen Poikkeama
        - Useat käyttäjät ilmoittavat, etteivät pääse palveluun

- Tunkeutuminen
    - Tapahtuma
        - Järjestelmänvalvoja epäilee, että järjestelmään on tunkeuduttu
    - Mahdollinen Poikkeama
        - Järjestelmänvalvoja tarjoaa lokin siitä, että jotain epäilyttävää tapahtui
- Väärinkäyttö
    - Tapahtuma
        - Välityspalvelimen lokin perusteella käyttäjä on käynyt yrityksen käytänteissä kielletyllä nettisivulla (tietyn tyyppisillä nettisivuilla käynti on kielletty)
    - Mahdollinen Poikkeama
        - Useat käyttäjät ovat käyneet kielletyillä nettisivuilla
- Luvaton käyttö
    - Tapahtuma
        - käyttäjä löytää vahingossa dokumentoimattoman pelin kaupallisessa ohjelmistossa
    - Mahdollinen Poikkeama
        - Käyttäjä pelaa dokumentoimatonta peliä ja on olemassa käytänne, joka kieltää pelin pelaamisen
- Huijaukset
    - Tapahtuma
        - Käyttäjä lähettää valheellista tietoa sisältävän sähköpostin, joka liittyy massoille tarkoitettuun huijaussähköpostiketjuun
    - Mahdollinen poikkeama
        - Käyttäjä pyytää lisäksi muita tekemään samoin
- Tiedonkeräys
    - Esimerkiksi porttien skannaus
    - Voi olla joko tapahtuma, tai poikkeama
    - Riippuu, miten tärkeää tietoturva on
        - Miten arkaluontoista tieto on

Yllä olevan perusteella poikkeaman tunnistaminen ei ole helppoa. Ero normaalin tapahtuman ja poikkeaman välillä voi olla pieni. Juuri tätä varten tarvitaan hyviä työkaluja poikkeamanhallinnan tueksi, joita tässä opinnäytetyössä etsittiin.

3.3 Cyber kill chain eli kyberhyökkäyksen vaiheet

Yksi suurin harhakuvitelma perinteisessä tietoturvassa on oletus siitä, että tiedetään, mitä kautta hyökkääjä tulee verkkoon. Esimerkiksi hyökkääjä harvoin tulee sisään etuovesta (tässä tapauksessa verkon rajalla oleva palomuuri). Yleensä hyökkäykset noudattavat kuitenkin tietynlaisia kuvioita, joiden kuvaamiseen voidaan käyttää ”cyber kill chain” -mallia. (Insider’s guide to incident response n.d.)

Cyber kill chain kuvaa vaiheita, jotka hyökkääjän tulee suorittaa päästäkseen verkkoon sisään ja varastaakseen sieltä dataa. Jokainen vaihe kuvaa tiettyä tavoitetta hyökkääjän polulla. Tätä mallia voidaan käyttää apuna poikkeamanhallinnan suunnittelussa, koska se kuvaa sitä, miten oikeat hyökkäykset tapahtuvat. Cyber kill chain sisältää seuraavat vaiheet (Insider’s guide to incident response n.d.):

- Tiedustelu (Reconnaissance & Probing)
    - Etsitään kohde/kohteita hyökkäykselle
    - Tehdään hyökkäyssuunnitelma sen perusteella, mitä tietoturva-aukkoja tai haavoittuvuuksia voidaan hyväksikäyttää
- Toimitus ja hyökkäys (Delivery and Attack)
    - Asetetaan toimitusmekanismi käyttöön
    - Social engineering eli käyttäjän manipulointi
        - Esimerkiksi huijataan käyttäjä lataamaan haittaohjelma
- Hyväksikäyttö ja asennus (exploitation and installation)
    - Käytetään hyväksi kohdejärjestelmän haavoittuvuuksia, että päästään sisään
    - Korotetaan käyttöoikeuksia ja asennetaan haitallinen koodi
- Murtautuminen järjestelmään (system compromise)

- Kaapataan ja tallennetaan arvokasta dataa ja tehdään se mahdollisimman nopeasti ja huomaamattomasti
- Käytetään murrettua järjestelmää lisäkäyttöoikeuksien saamiseen, varastetaan resursseja ja hyökätään niiden avulla toista kohdetta vastaan

3.4 Incident Response

Incident Response (IR) eli poikkeamanhallinta on käsite, jolla kuvataan prosessia, jonka avulla yritys käsittelee tietomurtoa tai kyberhyökkäystä. Tässä yritetään myös rajoittaa poikkeaman seurauksia. Pääasiallinen tavoite on käsitellä poikkeama tehokkaasti rajoittaen vahinkoja. Tässä pyritään rajoittamaan palautumisaikaa ja kustannuksia sekä säilyttämään yrityksen maine. (Lord 2015.)

Yrityksellä tulisi olla selkeä poikkeamanhallintasuunnitelma (Incident Response Plan IRP). Suunnitelmassa pitäisi olla määritettynä, miten poikkeama määritetään yrityksessä. Suunnitelmassa tulisi olla myös selkeä ohjeistettu prosessi, miten toimitaan, kun poikkeama havaitaan. (Lord 2015.)

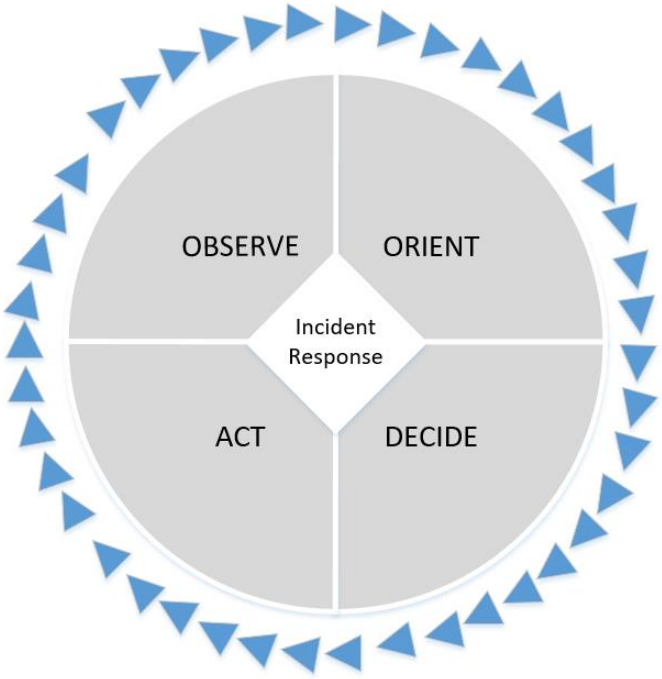
Yrityksessä olisi hyvä olla myös poikkeamanhallintatiimi (computer incident response team, CIRT). CIRT koostuu yleensä tietoturva- ja IT-henkilöstöstä ja sisältää jäseniä myös laki-, henkilöstö-, ja PR-osastoilta. CIRT on ryhmä, jonka vastuulla on reagoida tietomurtoihin, viruksiin ja muihin poikkeamiin. Teknisten osaajien lisäksi CIRTin tulisi sisältää asiantuntijoita, jotka voivat opastaa yrityksen johtoa asianmukaisessa kommunikoinnissa poikkeamatilanteissa. (Lord 2015.)

3.5 Poikkeamanhallintaprosessi ja sen vaiheet

3.5.1 OODA-loop

Poikkeamanhallintaprosessin kuvaamiseen on olemassa useita eri tapoja. Ehkä selkein ja yksinkertaisin tapa on käyttää niin sanottua OODA-luuppia. Tämä koostuu neljästä vaiheesta: Observe, Orient, Decide, Act, eli suomeksi Havainnoi, Arvioi tilanne,

Päätä, Toimi. Kuviossa 1 on esitetty OODA-loop (Insider's guide to incident response N.d.)



Kuvio 1 OODA-loop (Insider's guide to incident response n.d.)

OODA loopin on kehittänyt Yhdysvaltain ilmavoimien eversti John Boyd. Se on alun perin tarkoitettu sotilaalliseen päätöksentekoon ja strategiaan. Kyberturvallisuus on kuitenkin hyvin samanlaista, jos sitä verrataan sotaan. Molemmissa sekä hyökkääjä että puolustaja yrittävät saavuttaa tavoitteensa vihollista vastaan yrittäen minimoida vahingot itseään kohtaan. OODA-loop on hyvä esimerkki metodologiasta, joka on alun perin kehitetty sotatoimintaan ja jota nykyään käytetään kyberturvallisuudessa. OODA-loop keskittyy avaintaktiikoihin, joita käytetään reagoitaessa mihin vain kriisiin. Nämä avaintaktiikat ovat havainnointi, tilanteen arviointi, päättäminen ja toiminta. (Klinghofer 2014; Malik 2017.)

Havainnointivaiheessa (Observe) käytetään tietoturvamonitorointia epänormaalin, mahdollisesti toimenpiteitä vaativan toiminnan tunnistamiseen. Päätökset perustuvat havaintoihin muuttuvista tilanteista. Nämä havainnot ovat pohjatieto johon päätökset ja toiminta perustuvat. (Insider's guide to incident response n.d. ; Klinghofer 2014.)

Hyökkääjät tiedustelevat useita järjestelmiä ja tietoturvatasoja. Tämän takia on tärkeää, että jatkuvasti tarkkaillaan toimintaa kaikkien laitteiden laajuisesti. Havainnointivaiheessa on oleellista tapahtumien priorisointi uhkatietoisuuden (threat intelligence) perusteella sekä monitorointityökalujen jatkuva hienosäätö. Että tiedetään, milloin hyökkäys on tapahtumassa, tulee tietää mitä pitää tarkkailla. Tässä käytetään apuna tietoisuutta tietoturvauhista (threat intelligence). Kun verkosta havaitsee enemmän kuvioita/kaavoja verkkoliikenteessä, käyttäjien toiminnassa ja palveluiden saatavuustilastoissa halutaan todennäköisesti hienosäätää monitorointityökaluja. Näin varmistetaan, että kaikki poikkeamien tutkimisessa tarvittava informaatio tallennetaan. Kun uusia uhkia ilmaantuu, halutaan keskittyä avainindikaattoreihin. (Malik 2017.)

Tilanteen arviointivaiheessa (Orient) arvioidaan mitä tapahtuu tällä hetkellä kyberturvallisuusuhkien alueella ja yrityksen sisällä. Tämän perusteella tehdään loogisia ja reaaliaikaisia asiayhteyksiä, joiden avulla voidaan keskittyä tärkeisiin tapahtumiin. Tässä vaiheessa erotetaan oleelliset havainnot epäoleellisista. Näitä käytetään apuna päättämisessä ja toiminnassa. (Insider’s guide to incident response N.d. ;Klinghofer 2014.)

Tilanteen arviointivaiheessa on oleellista kaikki informaatio, joka on kerätty havainnointivaiheessa. Tätä tietoa tarvitaan sellaisten tapahtumien tunnistamiseen, jotka vaativat jatkotutkimusta. Tässä vaiheessa oleellista ei ole pelkästään tieto vaan myös asiayhteydet. Kaikki tieto maailmassa on käyttökelvotonta, jos ei tiedetä asiayhteyksiä, jotka tarvitaan tiedon ymmärtämiseen. Jos esimerkiksi johonkin järjestelmään ei saada yhteyttä se voi johtua esimerkiksi sähkökatkosta tai palvelunestohyökkäyksestä (DDoS, Dynamic Denial of Service). Ilman tarpeellisia asiayhteyksiä, ei pysty toteuttamaan tehokkaita vastatoimia. (Malik 2017.)

Tilanteen arviointivaihe voidaan jakaa vielä kolmeen alivaiheeseen. Ensin pitää päätellä ongelman laajuus ja vaikutukset uhkatietoisuuteen perustuen. Uhkatietoisuutta käytetään viimeisimpien hyökkäystyökalujen ja taktiikoiden monitorointiin ja analysointiin ja tämän tiedon perusteella voidaan määrittää automatisoituja toimintoja.

Näitä automatisoituja toiminnot voivat olla esimerkiksi sääntöjä, hälytyksiä ja tiketejä (tiketöintijärjestelmän). (Malik 2017.)

Seuraavaksi tutkitaan tapahtumaa asiayhteydessä verkon tapahtumiin ja luodaan aikajana tapahtumasta. Tämä voidaan tehdä manuaalisesti tai työkalun avulla. Tässä yhdistetään erilaiset, mutta toisiinsa liittyvät tapahtumat, muodostaen aikajana kaikille tapahtumille. (Malik 2017.)

Tilanteen arviointivaiheen kolmannessa vaiheessa tutkitaan hyökkäyksen lähdettä, että voidaan nimetä syyllinen, jos se on mahdollista. Vahva nimeäminen voi johtaa pelotteeseen. Pelotteella tarkoitetaan sitä, että pelotetaan vastapuolta, tässä tapauksessa hyökkääjää, tulevien hyökkäysten välttämiseksi. Syyllisen nimeäminen tarjoaa myös tarpeellista asiayhteyttä tulevien hyökkäysten tunnistamiseen ja välttämiseen samasta lähteestä sekä muilta hyökkääjiltä, joilla on samat tavoitteet, työkalut ja taktiikat. (Malik 2017.)

Kun kaikkien toimenpiteiden mahdollisuudet ja seuraukset on selvitetty, Päätöksentekijän tulee valita paras mahdollinen toimenpide. Päättämisvaiheessa (Decide) valitaan paras taktiikka vahinkojen ja palautumisajan minimointiin havaintoihin ja asiayhteyksiin perustuen. (Insider’s guide to incident response N.d. ;Klinghofer 2014.)

OODA-loopin kaksi ensimmäistä vaihetta (Havainnointi, Tilanteen arviointi) hyötyvät automatisoitujen työkalujen käytöstä tiedon keräämiseen ja analysointiin. Päätöksentekoa tietoon/tietämykseen perustuen ei voida ainakaan vielä automatisoida. Erityisesti tässä vaiheessa tarvitaan osaavia ihmisiä. (Malik 2017.)

Myös Päättämisvaihe voidaan jakaa kolmeen alivaiheeseen. Ensin päätetään välittömät toimenpiteet poikkeamaan reagoinnissa. Yksi suurimmista päätöksistä tässä vaiheessa on miten tasapainoillaan nopean palautumisen ja todisteiden säilyttämisen välillä. Pitää päättää tilanteesta riippuen, kumpi on tärkeämpää. Tämä päätös on hyvä tehdä etukäteen ennen kuin poikkeama on jo käynnissä. Vakiomenettely poikkeamien käsittelystä tulisi tulla yrityksen johdolta, lakiosaston avustuksella. Päätös siitä yritetäänkö palautua mahdollisimman nopeasti vai säilyttää todisteet, ei ole

helppo, mutta se tulee päättää mahdollisimman varhaisessa vaiheessa. Tähän päätökseen vaikuttavat seuraavat asiat: Yrityksen toimiala, lait, millaisesta tiedosta on kyse ja miten se saavutettiin sekä oliko hyökkääjä yrityksen sisäinen vai ulkoinen. Tätä päätöstä ei tulisi tehdä kevyesti ja ohjeistusta kannattaa pyytää asiantuntevilta tahoilta. (Malik 2017.)

Päätösvaiheen toisessa vaiheessa tutkitaan suojattavien kohteiden (asset) omistajuustietoja sekä suojattaviin kohteisiin liittyviä muita tietoja. Mitä paremmin verkossa olevat suojattavat kohteet (erityisesti palvelimet) tunnetaan, sitä parempia ollaan niihin liittyvien poikkeamien hallinnassa. Ei ole aina selkeää, kuka omistaa suojattavan kohteen, miten se on konfiguroitu ja mitä ohjelmistoa on asennettu. Tämän takia nämä asiat tulisi olla hyvin dokumentoitu jo etukäteen. (Malik 2017.)

Päätösvaiheen kolmannessa vaiheessa dokumentoidaan suunnitellut taktiikat poikkeamasta palautumiseen. Kun poikkeaman vaikutukset ja laajuus on selvitetty, tulee palautua mahdollisimman nopeasti lisävahinkojen välttämiseksi. Kaikki poikkeamasta palautumisen vaiheet kannattaa dokumentoida sisältäen tiedot suojattavista kohteista, joihin vaikutukset kohdistuivat ja tiedot siitä mitä tehtiin, kuka sen teki ja milloin. Tällaisesta jäljitysketjusta on hyötyä esimerkiksi johdolle raportoinnissa. (Malik 2017.)

Toimimisvaiheessa (Act) sananmukaisesti toimitaan, eli korjataan ongelma ja palaudutaan siitä. Opitun perusteella tehdään tarvittavia muutoksia Incident Response -prosessiin, jotta voidaan tulevaisuudessa välttää vastaava ongelma tai toimia paremmin vastaavassa tilanteessa. (Insider’s guide to incident response N.d. ;Klinghofer 2014.)

Toimisvaihe voidaan myös jakaa kahteen alivaiheeseen. Ensimmäisessä vaiheessa toteutetaan tarvittavat toimenpiteet vaikutuksenalaisille suojattaville kohteille palautumiseen ja varmistetaan, että ongelmasta palautuminen toteutettiin oikein. Tässä vaiheessa tehtävät toimenpiteet riippuvat uhasta, vaikutuksista, vaikutusten laajuudesta ja kohteena olleista suojattavista kohteista. Näihin toimenpiteisiin sisältyy esimerkiksi (Malik 2017):

- Tietoturvapäivitykset
- Tarpeettomien ja kiellettyjen ohjelmistojen poistaminen
- Uudelleenkonfigurointi
- Palomuurien konfigurointi
- Käyttöoikeuksien poistaminen
- salasanojen resetointi
- Poistetaan käyttämättömät ja tarpeettomat käyttäjätilit

Toimimisvaiheen toisessa vaiheessa tarkastetaan ja päivitetään tietoturvatietoisuuden koulutusohjelmat ja tietoturvakäytänteet asianmukaisesti. Jokainen poikkeama tarjoaa mahdollisuuden arvioida, kuinka hyvin tietoturvasuunnitelma ja koulutus, toimivat tietoturvatietoisuuden, käytänteiden ja toimenpiteiden osalta. Mitä valppaampia käyttäjät ovat kyberturvallisuudesta, sitä todennäköisemmin poikkeamien riski laskee niin määrän kuin vaikutustenkin osalta. (Malik 2017.)

OODA loop on jatkuva sykli, jossa pyritään jatkuvaan edistykseen. Jotta saavutetaan nopein ja tehokkain toiminta tulee ottaa huomioon kaikki aktiiviset muuttujat, mahdolliset skenaariot ja kaikkien toimenpiteiden seuraukset. (Klinghofer 2014.)

3.5.2 Poikkeamanhallinnan vaiheet

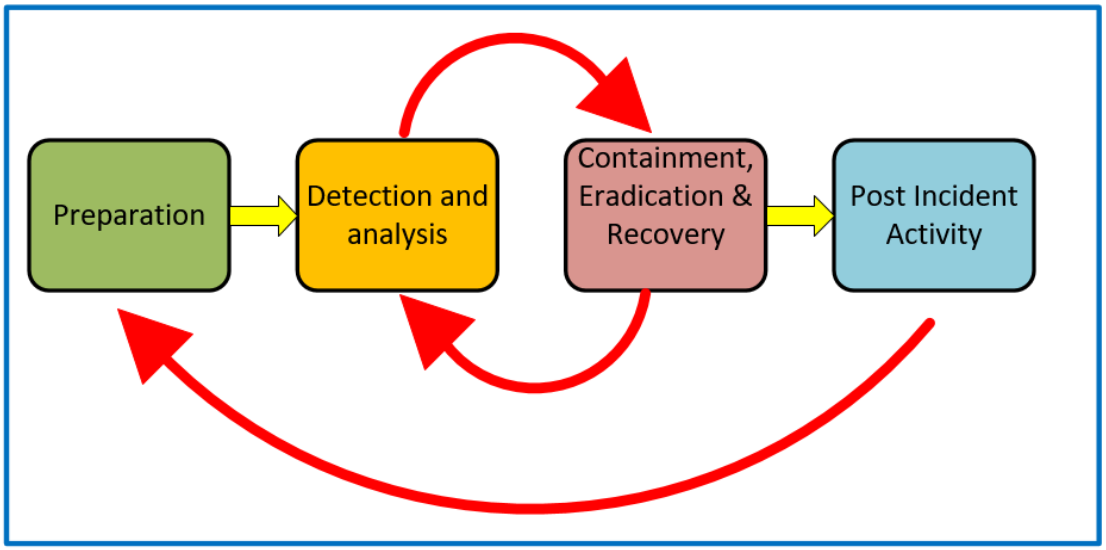
Toinen tapa kuvata on SANSin käyttämä kuusivaiheinen prosessi. Sen kuusi vaihetta ovat seuraavat (Kral 2011):

1. Preparation (Valmistautuminen)
2. Identification (Tunnistaminen)
3. Containment (Rajaaminen)
4. Eradication (Hävittäminen)
5. Recovery (Palautuminen)
6. Lessons Learned (Mitä opittiin?)

Myös NIST (National Institute of Standards and Technology) on kuvaillut vastavanlaisen prosessin, mutta siinä on 4 vaihetta, koska joitakin vaiheita on yhdistetty ja nimetty eri nimillä (Cichonski, Millar, Grance, Scarfone 2012):

1. Preparation (valmistautuminen)
2. Detection and Analysis (tunnistaminen ja analysointi)
3. Containment, Eradication and Recovery (rajaaminen, hävittäminen, palautuminen)
4. Post Incident Activity (Poikkeaman jälkeiset toimenpiteet)

Kuviossa 2 on esitelty poikkeamanhallinnan vaiheet. Ensimmäisenä on valmistautumisvaihe (preparation), tämän jälkeen mennään tunnistamis- ja analysointivaiheeseen (detection and analysis). Seuraavana on rajaaminen, hävittäminen ja palautuminen (containment, eradication & recovery). Näistä vaiheista palataan välillä tunnistamis- ja analysointivaiheeseen tarkistamaan, onko poikkeama laajentunut. Sitten taas palataan rajaamiseen, hävittämiseen ja palautumiseen. Tätä toistetaan niin kauan, että poikkeama on saatu korjattua. Kun ongelma on korjattu, voidaan mennä poikkeaman jälkeisiin toimenpiteisiin (post incident activity). Poikkeamanjälkeisiin toimenpiteisiin kuuluu kaiken dokumentointi sekä sen miettiminen mitä opittiin ja miten voidaan toimia paremmin seuraavalla kerralla tai ehkäistä poikkeama kokonaan. Tämän jälkeen palataan prosessin alkuun valmistautumisvaiheeseen ja tehdään tarvittavat korjaukset tietoturvakontrolleihin ja poikkeamanhallintaprosessiin. (Cichonski ym. 2012.)



Kuvio 2 Poikkeamanhallinnan vaiheet (Cichonski ym. 2012)

Valmistautumisvaiheessa (Preparation) Koulutetaan käyttäjät ja IT-henkilökunta toimimaan oikein Poikkeaman tapahtuessa. Tässä vaiheessa myös perustetaan ja koulutetaan poikkeamanhallintatiimi ja hankitaan tarvittavat työkalut ja resurssit. Tässä vaiheessa myös arvioidaan riskejä sekä valitaan ja toteutetaan tietoturvakontrollit sen perusteella. (Insider's guide to incident response N.d.; Cichonski ym. 2012.)

Tietoturvakontrollien toteuttamisen jälkeenkin kuitenkin jää riskejä jäljelle. Tämän takia tarvitaan tietoturvaloukkausten tunnistamista yrityksen varoittamiseen poikkeamista. Tunnistamis- ja analysointivaiheessa (Identification / Detection and Analysis) Tunnistetaan poikkeamat IR-suunnitelman määritysten mukaisesti, analysoidaan niitä ja päätetään, mitkä niistä pitää tutkia välittömästi ja mitkä eivät ole niin kiireellisiä. (Insider's guide to incident response N.d.; Cichonski ym. 2012.)

Kolmessa seuraavassa vaiheessa Yritys voi lieventää poikkeaman vaikutuksia rajaamalla sen ja loppujen lopuksi palautumalla siitä. Rajaamisvaiheessa (Containment) rajataan ongelma ja eristetään saastuneet järjestelmät lisävahinkojen estämiseksi. Hävittämisvaiheessa (Eradication) löydetään ja eliminoidaan ongelman alkulähde ja poistetaan saastuneet järjestelmät tuotantoympäristöstä. Palautumisvaiheessa (Recovery) palautetaan eristetyt järjestelmät takaisin tuotantoympäristöön ja tarkkaillaan huolellisesti, onko enää ongelmia. Näistä vaiheista usein palataan takaisin tun-

nistamis- ja analysointivaiheeseen, että voidaan esimerkiksi tarkistaa, onko saastuneita järjestelmiä ilmaantunut lisää. (Insider's guide to incident response N.d.; Cichonski ym. 2012.)

Poikkeaman jälkeisissä toimenpiteissä (Post-Incident Activity ja Lessons Learned) kirjataan kaikki ylös ja mietitään mitä opittiin sekä miten tulevaisuudessa voisi toimia paremmin vastaavassa tilanteessa tai ehkäistä vastaavan lainen ongelma kokonaan. Poikkeaman jälkeen tehdään yksityiskohtainen raportti poikkeaman syistä ja vahingoista/kustannuksista sekä kirjataan ylös vaiheet, joita seuraamalla voidaan välttää tulevia samanlaisia poikkeamia. (Insider's guide to incident response N.d.; Cichonski ym. 2012.)

Preparation eli valmistautuminen

Incident Response metodologiat usein painottavat valmistautumisvaiheen tärkeyttä. Tässä vaiheessa varmistetaan, että ollaan valmiita reagoimaan poikkeamiin, mutta myös pyritään ehkäisemään poikkeamia, varmistamalla, että järjestelmät, verkot ovat riittävän tietoturvallisia. Vaikka poikkeamanhallintatiimi ei tyypillisesti ole vastuussa poikkeamien ehkäisystä, on se oleellista poikkeamanhallintaohjelman menestykselle. Poikkeamien välttämisessä ovat oleellisia seuraavat asiat (Cichonski ym. 2012.):

- Riskien arviointi
- Päätelaitteiden tietoturva
- Verkon tietoturva
- Haittaohjelmien välttäminen
- Käyttäjien tietoisuus ja koulutus

Valmistautumisvaihe sisältää kaikki toimenpiteet, jotka tehdään, jotta ollaan valmiina käsittelemään poikkeamia, kun ne ilmaantuvat. Tämä on kaikkein tärkein vaihe verrattuna muihin, koska tämä vaihe määrittää kuinka hyvin tiimi voi reagoida kriisitilanteessa. On olemassa useita asioita, jotka tehdään tässä vaiheessa, jotta saadaan vähennettyä mahdollisia ongelmia, jotka voivat haitata työntekijän kykyä käsitellä poikkeamia. (Kral 2011.)

Seuraavaksi käydään läpi joitakin tärkeitä asioita, työkaluja ja resursseja, jotka ovat tärkeitä valmistautumisvaiheessa.

Policy eli käytänne

Käytänne tarjoaa kirjalliset periaatteet, säännöt ja käytännöt yrityksessä. Tämän on keskeinen elementti, joka tarjoaa opastusta siitä, onko yrityksessä tapahtunut poikkeama. Selkeät käytänteet ovat tärkeitä oikeustapausten välttämiseksi, kun esimerkiksi työntekijä irtisanotaan sopimattoman toiminnan vuoksi ja sitä ei ole ollut virallisesti kielletty yrityksen käytänteissä. (Kral 2011.)

Incident response plan (IRP) eli poikkeamanhallintasuunnitelma

Seuraavaksi pitää luoda suunnitelma ja strategia siitä, kuinka poikkeamia käsitellään. Poikkeamat tulisi priorisoida sen perusteella minkälaiset ja miten suuret vaikutukset sillä on yrityksen toimintaan. Kun poikkeamat priorisoidaan tällä tavalla se helpottaa saamaan yrityksen johdon tuen ja osallistumisen poikkeamanhallintaan. Ilman yrityksen johdon tukea poikkeamanhallintatiimille ei välttämättä anneta tarpeellisia resursseja oikeanlaiseen toimintaan kriisitilanteessa. (Kral 2011.)

Kommunikointi

Kommunikointisuunnitelma on tärkeä, koska poikkeaman ilmaantuessa voi olla tärkeää ottaa yhteyttä tiettyyn asiantuntijaan. Jos suunnitelmaa ei ole, on todennäköistä, että aikaa menee hukkaan, kun otetaan yhteyttää vääriin ihmisiin, ennen kuin löydetään oikea ihminen johon pitää ottaa yhteyttä. Koko IR-tiimin tulisi tietää keneen otetaan yhteyttä sekä milloin ja miksi. Yhteystietoihin voi sisältyä esimerkiksi poliisin ja muiden poikkeamanhallintatiimien yhteystiedot. Yhteystietoina on hyvä olla puhelin ja sähköposti ja oleellista on myös olla pääasiallinen ja varayhteystieto. Poikkeamien raportointimekanismit joiden kautta käyttäjät voivat raportoida epäilyistä poikkeamista ovat oleellisia kommunikoinnissa. Näihin mekanismeihin voi sisältyä puhelin, sähköposti, verkkolomake ja turvallinen pikaviestijärjestelmä. Kommunikoinnissa on oleellinen myös järjestelmä poikkeamien kirjaamiseen ja seurantaan. Järjestelmästä tulisi nähdä esimerkiksi poikkeaman tiedot ja tila. Oleellista on myös kommunikoinnin turvallisuus ja luottamuksellisuus. Tässä on tärkeää salaus/kryptausohjelmistot sekä todisteiden ja arkaluontoisten materiaalien turvallinen säilytys.

Poikkeamanhallintatiimillä on myös hyvä olla ns. ”sotahuone” keskitettyyn kommunikointiin ja ohjaukseen. (Kral 2011; Cichonski ym. 2012.)

Dokumentointi

Poikkeaman dokumentointi on tärkeää ainakin kahdesta syystä. Kun poikkeamaan epäillään liittyvän rikos, voidaan dokumentaatiota käyttää todisteena. Toinen syy dokumentointiin on se, että siitä voidaan oppia, miten toimitaan tulevaisuudessa paremmin. On tärkeää, että dokumentoidaan kaikki mitä poikkeamanhallintatiimi tekee. Dokumentaation tulisi vastata seuraaviin kysymyksiin: kuka, mitä, milloin, missä, miksi ja miten epäselvyyksien välttämiseksi. Valmistautumisvaiheessa on oleellista myös dokumentoida omaa ympäristöä. Tämä auttaa myöhemmissä vaiheissa. Oman ympäristön dokumentaatioon sisältyy seuraavat tärkeät asiat (Kral 2011; Cichonski ym. 2012.):

- Verkkotopologiat ja porttilistat
- Lista tärkeistä palvelimista, laitteista ja järjestelmistä
- Baseline tiedot
    - verkon, järjestelmien ja sovellusten käyttö normaalitilanteessa
- Järjestelmien dokumentaatio
    - käyttöjärjestelmät, sovellukset, protokollat, tunkeutumisen havaitseminen, virustorjunta

Tiimi (CIRT – Computer Incident Response Team)

CIRTin tulisi koostua eri alojen ihmisistä, jotka voivat käsitellä erilaisia ongelmia, jotka ilmaantuvat poikkeaman aikana. Tiimin jäsenet voivat olla Lakiosastolta, henkilöstöosastolta, PR-osastolta sekä tietenkin IT-osastolta eri asioihin erikoistuneita asiantuntijoita. (Kral 2011.)

Pääsynhallinta eli Access Control

On tärkeää varmistaa, että CIRTillä on asianmukaiset oikeudet työnsä tekemiseen. Järjestelmänvalvoja voi antaa oikeuksia CIRTin käyttäjätileille poikkeaman aikana ja sallia heidän korjata ongelma ja poistaa kyseiset oikeudet, kun niitä ei enää tarvita. (Kral 2011.)

Työkalut

Ilman työkaluja on vaikea saada mitään tehtyä. Kaikki tärkeät ohjelmistot ja tarvittavat työkalut tulisi olla helposti otettavissa käyttöön tarvittaessa. Esimerkiksi seuraavat työkalut voivat olla oleellisia (Kral 2011; Cichonski ym. 2012):

- Järjestelmä poikkeamien kirjaamiseen ja seuraamiseen
- Forensiikka- ja varmuuskopiointityöasemat sekä forensiikkaohjelmisto
    - levykuvien, lokien ja muun oleellisen incident response datan tallentamiseen ja varmuuskopiointiin
- kannettavat tietokoneet analysointiin, pakettien kaappaukseen, raportointiin
- Varatyöasemat, varapalvelimet, varaverkkolaitteet
- Siirrettävät tallennusvälineet
    - esimerkiksi muistitikut
        - tyhjät muistitikut tärkeiden tietojen tallennukseen
        - muistitikut, jotka sisältävät tärkeitä ohjelmia
- Työkalut verkkoliikenteen kaappaamiseen
- Todisteidentallennusvälineet: muistikirja, kamera, nauhuri

Koulutus

Koulutus on oleellista, koska ilman sitä tiimi ei ole valmistautunut poikkeamiin ja tämä voi johtaa täydelliseen epäonnistumiseen, hyvästä suunnitelmasta huolimatta. On suositeltavaa pitää harjoituksia säännöllisin väliajoin, että jokainen CIRTin jäsen tietää ja osaa tehdä tehtävänsä Poikkeaman aikana. (Kral 2011.)

Identification/Detection and analysis eli tunnistaminen ja analysointi

Tässä vaiheessa tunnistetaan ja päätellään, onko normaalista toiminnasta poikkeava tapahtuma poikkeama, ja mikä on poikkeaman laajuus. Tässä vaiheessa kerätään tietoa verkon tapahtumista. Tietolähteinä ovat lokitiedostot, virheviestit, IDS-järjestelmät ja palomuurit, jotka voivat tuottaa todisteita siitä, onko jokin tapahtuma poikkeama. Jos jokin tapahtuma tunnistetaan poikkeamaksi, se tulisi raportoida mahdollisimman pian, että siitä ehditään keräämään mahdollisimman paljon tietoa seuraavia vaiheita varten. Tässä vaiheessa kommunikointi CIRTin, Hallinnon ja järjestelmänvalvojien välillä on tärkeää, varsinkin, jos poikkeamalla on merkittävä vaikutus

liiketoimintaan. Yhdellä poikkeamalla olisi hyvä olla kaksi käsittelijää; yksi ensisijainen käsittelijä, joka tunnistaa ja arvioi poikkeamaa ja toinen auttamaan todisteiden keräämisessä. Kaikki tässä vaiheessa tehty tulisi dokumentoida vastaten seuraaviin kysymyksiin: kuka, mitä, missä, milloin, miksi. Kun poikkeaman laajuus on määritetty ja todisteet dokumentoitu voidaan siirtyä seuraaviin vaiheisiin. (Kral 2011.)

Tunnistamis- ja analysointivaiheessa oleellisia asioita ovat hyökkäysvektorit, poikkeaman tunnusmerkit, poikkeamien analysointi ja priorisointi. Seuraavaksi käsitellään näitä hieman lisää. (Cichonski ym. 2012.)

Hyökkäysvektorit

Hyökkäysvektorit ovat erilaisia toteutustapoja, joiden kautta poikkeamat voivat ilmentyä. Periaatteessa yrityksellä olisi tärkeää olla vaiheittaiset ohjeet kaikkien mahdollisien poikkeamien hallintaan. Käytännössä kannattaa kuitenkin keskittyä poikkeamiin jotka käyttävät yleisiä hyökkäysvektoreita. Hyökkäysvektoreita ovat esimerkiksi seuraavat (Cichonski ym. 2012.):

- Siirrettävä tallennusmedia
    - voi sisältää esimerkiksi haitallista koodia joka levitetään verkkoon
- Uuvutus
    - tähän sisältyy brute force menetelmät, esimerkiksi DDos (palvelunestohyökkäys)
- Web
    - Hyökkäys toteutetaan verkkosivun tai verkkopohjaisen sovelluksen kautta
- Sähköposti
    - Esimerkiksi sähköpostissa oleva haitallinen liitetiedosto tai linkki haitalliselle internet sivulle
- imitointi
    - korvataan jotain hyvälaatuista haitallisella
    - esimerkiksi man-in-the-middle hyökkäys
- sopimaton käyttö
    - valtuutettu käyttäjä toimii yrityksen käytänteiden vastaisesti

- laitteiston varastaminen
    - varastetaan esimerkiksi yrityksen käyttämä kannettava tietokone tai älypuhelin

Poikkeaman tunnusmerkit

Poikkeamanhallinnan vaikein osa poikkeamien tarkka tunnistaminen ja arviointi; onko poikkeama tapahtunut, minkä tyyppinen se on ja miten laajat ovat sen vaikutukset. Kolme tekijää tekee tästä haastavaa (Cichonski ym. 2012.):

1. Poikkeamia voidaan tunnistaa eri tavoilla ja niistä on tietoa vaihteleva määrä. Poikkeamia voidaan tunnistaa automaattisesti tunkeutumisen tunnistamisjärjestelmällä, virustorjuntaohjelmalla tai lokianalysaattorilla. Poikkeamia voidaan tunnistaa myös manuaalisesti käyttäjän raportilla. Käyttäjien raportit ovat usein virheellisiä. Myös tunkeutumisen tunnistamisjärjestelmä voi antaa virheellisiä ilmoituksia.
2. Tunnusmerkkejä mahdollisista poikkeamista tulee todella paljon: tuhansia tai jopa miljoonia päivässä
3. Tekninen tietämys ja kokemus ovat tärkeitä tiedon analysoinnissa

Poikkeamanhallinnan tunnusmerkit voidaan jakaa kahteen ryhmään; Ennakkomerkkeihin (precursor) ja tunnusmerkkeihin (indicator) jo tapahtuneesta poikkeamasta. Ennakkomerkki voi olla esimerkiksi web-palvelimen lokimerkintä haavoittuvuusskannerin käytöstä. Tunnusmerkki jo tapahtuneesta poikkeamasta voi olla esimerkiksi virustorjuntaohjelman ilmoitus siitä, että laite on saastunut haittaohjelmalla. Ennakkomerkit ovat oleellisia poikkeamien ehkäisemisessä, mutta kaikilla poikkeamilla ei ole ennakkomerkkejä. Merkit jo tapahtuneista poikkeamista ovat yleisiä. (Cichonski ym. 2012.)

Poikkeaman analysointi

Jos poikkeaman tunnusmerkit olisivat varmasti tarkkoja olisi tunnistaminen ja analysointi helppoa, näin ei kuitenkaan ole. Oikeiden poikkeamien löytäminen on pelottava tehtävä. Se vaatii hyvää harkintakykyä ja yhteistyötä muiden asiantuntijoiden kanssa. Jokainen havainnon todenmukaisuus pitää arvioida ja niitä on tuhansia tai

miljoonia päivässä. Kaikilla poikkeamilla ei ole selkeitä tunnusmerkkejä. Kaiken lisäksi, vaikka havainto olisi todenmukainen se ei välttämättä tarkoita, että on tapahtunut tietoturvapoikkeama. Esimerkiksi palvelimen kaatuminen voi johtua poikkeamasta tai inhimillisestä virheestä. Aina kuitenkin kannattaa epäillä, että poikkeama on tapahtunut. Kaikki tämä pitäisi tehdä myös mahdollisimman nopeasti. (Cichonski ym. 2012.)

Aluksi pitää tehdä pika-analyysi, jossa selvitetään mihin järjestelmiin poikkeama vaikuttaa, miten se ilmenee, mistä lähteestä se tulee ja mitä haavoittuvuuksia siinä hyödynnetään. Tämän analyysin pitäisi tarjota tarpeeksi tietoa seuraavien toimenpiteiden priorisointiin. Analysointia helpottaa seuraavat asiat (Cichonski ym. 2012.):

- Verkon ja järjestelmien profilointi, eli normaalin toiminnan mittaaminen
    - esimerkiksi: normaalit tapahtumat, kaistan keskimääräinen ja huippukäyttö
- Sen ymmärtäminen mikä on normaalia
    - On opiskeltava tämän saavuttamiseksi verkkoja, järjestelmiä ja sovelluksia
- Lokien säilytyskäytänteet
    - lokeja on monia: palomuuri, IDS, sovellusten lokit
    - päätettävä kuinka kauan lokeja säilytetään, koska joskus poikkeamat huomataan vasta pitkän ajan päästä
    - tieto aikaisemmista vastaavista poikkeamista arvokasta
- Eri tietolähteiden riippuvuussuhteiden ymmärtäminen
    - pitää pystyä yhdistämään tietoa eri tietolähteistä kokonaiskuvan muodostamiseksi
- Laitteiden kellonaikojen synkronointi
    - NTP (network time protocol) avulla
    - oleellista eri tietolähteiden riippuvuussuhteiden määrittämisessä
- Pidetään yllä tietokantaa kaikesta aiemmin opitusta tietämyksestä
    - saadaan sieltä apua uusien poikkeamien tutkimisessa
- Tiedon hakeminen internetistä

- Verkkoliikenteen kaappaus
    - saadaan lisätietoa meneillään olevasta poikkeamasta
- Datan suodattaminen
    - pyritään suodattamaan yleensä ei merkittävät havainnot pois
- Etsitään apua muualta
    - Poikkeamaa ei aina saa selvitettyä itse
    - Apua voi pyytää esimerkiksi toisilta poikkeamanhallintatiimeiltä

Poikkeaman priorisointi

Poikkeamien käsittelyn priorisointi on ehkä kriittisin päätös poikkeamanhallintaprosessissa. Priorisointiin vaikuttavat poikkeaman toiminnalliset vaikutukset, eli miten paljon se haittaa liiketoimintaa, sen vaikutukset arkaluontoiseen dataan eli vaarantuuko luottamuksellisuus, eheys tai saatavuus, ja miten se vaikuttaa liiketoimintaan. Kolmas asia joka vaikuttaa priorisointiin, on se, kuinka helposti poikkeamasta voidaan palautua. Joskus poikkeamasta ei voi sillä hetkellä palautua ja siihen ei kannata tuhlata resursseja. (Cichonski ym. 2012.)

Containment eli rajaaminen

Tässä vaiheessa tarkoituksena on rajoittaa vahinkoja ja ehkäistä lisävahinkoja. Ongelman rajaaminen on tärkeää, ennen kuin poikkeama ylikuormittaa resurssit ja vahingot lisääntyvät. Suurin osa poikkeamista vaatii rajaamista, ja se on tärkeä päätös käsittelyn alkuvaiheessa. Rajaaminen tarjoaa aikaa hyvän toimintasuunnitelman tekemiseen. Rajaamisessa on oleellista päättää, sammutetaanko järjestelmä kokonaan, eristetäänkö se verkosta vai laitetaanko vain joitakin ominaisuuksia käytöstä. Poikkeamanhallinnassa on hyvä olla hyväksyttävien riskien perusteella tehty strategia ja toimintasuunnitelma rajaamiseen. Tämä vaihe voidaan jakaa vielä pienempiin vaiheisiin, jotka kaikki ovat tärkeitä, että poikkeama saadaan täysin hoidettua ja ettei todisteita katoa. (Kral 2011; Cichonski ym. 2012.)

Ensimmäinen vaihe on lyhytaikainen rajaaminen. Tässä vaiheessa tavoitteena on rajoittaa vahinkoja mahdollisimman pian. Lyhytaikainen rajaaminen voi tarkoittaa esimerkiksi verkon segmentin eristämistä muusta verkosta tai tuotantopalvelimien ottamista pois käytöstä ja ottamalla varapalvelimet käyttöön. Lyhytaikainen rajaaminen

ei ole ratkaisu ongelmaan, tavoitteena on vain rajata poikkeamaa ennen kuin se menee pahemmaksi. (Kral 2011.)

Toinen vaihe on varmuuskopiointi, jossa otetaan rikostekninen levykuva (forensic image) järjestelmistä joihin poikkeama on vaikuttanut. Tähän käytetään Forensic Tool Kit (FTK) -sovelluksia. Tässä tallennetaan tilannekuva järjestelmästä sellaisena kuin se oli poikkeaman aikana. Näin säilytetään todisteet, siltä varalta, että poikkeamaan liittyy rikos. Todisteita myös hyödynnetään ”mitä opittiin” -vaiheessa, kun tutkitaan miksi ja miten poikkeama tapahtui. (Kral 2011.)

Kolmas vaihe on pitkäaikainen rajaaminen. Tässä vaiheessa voidaan järjestelmät korjata väliaikaisesti, että liiketoiminta voi jatkua, kun järjestelmiä puhdistetaan seuraavassa vaiheessa. Pääasiallinen tavoite on poistaa hyökkääjien jättämät takaovet, asentaa tietoturvapäivitykset sekä tehdä muita toimenpiteitä ehkäisemään poikkeaman laajenemista. (Kral 2011.)

Poikkeamanrajaamisstrategiat vaihtelevat poikkeaman tyypin mukaan, jokaiselle poikkeamatyypille pitäisi olla oma hyvin dokumentoitu rajaamisstrategia. Strategian valinta riippuu muutamista asioista (Cichonski ym. 2012.):

- Potentiaaliset vahingot
- Todisteiden säilyttämisen tarve
- Onko palvelun saatavuus tärkeää
- Käytettävissä oleva aika ja resurssit
- Strategian tehokkuus
    - rajataanko ongelma osittain vai täydellisesti
- Ratkaisun kesto
    - onko kyseessä pikakorjaus, väliaikainen korjaus vai lopullinen korjaus

Hyvä esimerkki rajaamisesta on eristää vaikutuksen alaiset järjestelmät verkosta irrottamalla verkkokaapelit tai sammuttamalla kokonaisen verkkosegmentin kytkimet ja reitittimet. (Kral 2011.)

Eradication eli hävittäminen

Tässä vaiheessa tehdään vaikutuksen alaisten järjestelmien varsinainen poistaminen ja palauttaminen. Kun poikkeama on saatu rajattua verkosta, hävittäminen on tarpeellista poikkeaman komponenttien eliminointiin. Aiemmissa vaiheissa tehtyä dokumentaatiota kaikesta mitä on tehty, hyödynnetään poikkeaman kokonaisvaikutusten arviointiin. Tässä vaiheessa varmistetaan myös, että oikeat toimenpiteet tehtiin haitallisen sisällön poistamiseksi vaikutuksen alaisista järjestelmistä ja varmistetaan, että järjestelmät ovat täysin puhtaita. Tämä tarkoittaa käytännössä kiintolevyjen tyhjentämistä ja levykuvan palauttamista. Hävittämisellä voidaan tarkoittaa myös haittaohjelmien poistamista, murrettujen käyttäjätilien poistamista tai haavoittuvuuksien tunnistamista ja korjaamista. Hävittämisen aikana on oleellista tunnistaa kaikki vaikutuksen alaiset laitteet, että ne voidaan korjata. Tässä vaiheessa myös puolustusmekanismeja parannetaan, kun opittiin, mikä aiheutti poikkeaman ja varmistetaan, että tämä ei tapahdu uudestaan. Tämä voidaan tehdä esimerkiksi asentamalla päivitykset, jotka korjaavat hyökkääjien hyväksikäyttämät haavoittuvuudet. (Kral 2011; Cichonski ym. 2012.)

Recovery eli palautuminen

Tässä vaiheessa palautetaan vaikutuksen alaiset järjestelmät takaisin tuotantoympäristöön varovasti, varmistaen, ettei tämä johda uuteen poikkeamaan. On tärkeää testata, monitoroida ja varmistaa, että palautettavat järjestelmät eivät saastu tai vaarannu jollakin muulla tavalla uudestaan. Tässä vaiheessa tulisi tehdä seuraavat tärkeät päätökset (Kral 2011.):

- Omistajat päättävät palautustoimenpiteiden toteutusajankohdan, CIRTin neuvojen perusteella.
- Päätetään, kuinka testataan ja varmistetaan, että palautetut järjestelmät ovat puhtaita ja täysin toimintakunnossa.
- Päätetään, kuinka kauan palautettuja järjestelmiä monitoroidaan tarkemmin epänormaalin toiminnan varalta.
- Päätetään testaamiseen ja monitorointiin käytettävät työkalut.

Palautumiseen sisältyy esimerkiksi seuraavia asioita (Cichonski ym. 2012):

- Järjestelmien palautus varmuuskopioista
- Järjestelmien rakentaminen uudelleen tyhjästä
- saastuneiden tiedostojen vaihtaminen puhtaisiin
- tietoturvapäivitysten asennus
- salasanojen vaihto
- verkon rajan tietoturvan parantaminen (eli palomuurisääntöjen parantaminen)
- haavoittuvuuksien korjaaminen

Hävittämisen ja palautumisen tulisi olla vaiheittainen prosessi, jonka vaiheet tulee priorisoida. Suurista poikkeamista palautuminen voi kestää kuukausia. Aikaisten vaiheiden tarkoitus on parantaa yleistä tietoturvaa suhteellisen nopeasti, tulevien poikkeamien välttämiseksi. Myöhemmissä vaiheissa keskitytään pitkän ajan muutoksiin ja jatkuvaan työhön yrityksen pitämiseen mahdollisimman tietoturvallisena. (Cichonski ym. 2012.)

Poikkeaman jälkeiset toimenpiteet

Lessons learned eli mitä opittiin

Tässä vaiheessa suoritetaan loppuun dokumentointi, jos sitä ei poikkeaman aikana saatu vielä valmiiksi, ja tehdään lisädokumentaatiota tarvittaessa uusien poikkeamien varalta. Dokumentointi kirjoitetaan raportin muotoon, joka vastaa kysymyksiin kuka, mitä, missä, miksi ja miten. Pääasiallinen tavoite on oppia poikkeamasta, että voidaan parantaa tiimin toimintaa ja tarjota lähdemateriaalia mahdollisessa vastaavanlaisessa poikkeamassa. Dokumentaatiota voidaan käyttää myös uusien työntekijöiden kouluttamiseen. (Kral 2011.)

Varsinkin suurien poikkeamien jälkeen on hyvä pitää mitä opittiin -palaveri. Tämä on hyödyllistä tietoturvan ja poikkeamanhallintaprosessin parantamiseen. Mitä opittiin -palaveri on hyvä pitää mahdollisimman pian. Siellä tehdään yhteenveto poikkeamasta, ja se tulisi pitää lyhyenä, että yleisön huomio ei harhaudu. Lopussa olisi

hyvä olla aikaa ehdotuksille ja keskusteluille, siitä kuinka tiimin toimintaa voisi parantaa tulevaisuudessa. Hyvässä yhteenvedossa käydään läpi seuraavat asiat (Kral 2011; Cichonski ym. 2012):

- Mitä tapahtui?
- Milloin ongelma huomattiin ensin ja kuka teki havainnon?
- Poikkeaman laajuus
- Kuinka poikkeama rajattiin ja hävitettiin?
- Palautusvaiheessa tehty työ
- Miten henkilökunta ja johto suoriutuivat poikkeaman käsittelyssä
- Noudatettiinko toimintasuunnitelmia ja olivatko ne asianmukaisia
- Millä alueilla CIRT-tiimi oli tehokas
- Millä alueilla olisi parannettavaa
    - Mitä tietoa olisi tarvittu aikaisemmin
    - Tehtiinkö jotain, mikä haittasi palautumista
    - Mitä tehdään eri tavalla seuraavalla kerralla
    - Miten tiedonjakoa muiden organisaatioiden kanssa voisi parantaa
    - Mitä korjaustoimenpiteitä tehdään vastaavien poikkeamien ehkäisemiseksi tulevaisuudessa
- Mitä ennakkomerkkejä ja tunnusmerkkejä tulisi seurata tulevaisuudessa vastaavan poikkeaman tunnistamiseen
- Mitä uusia työkaluja tarvitaan tulevaisuudessa

Kerätyn datan käyttö

Poikkeamasta kerätty data voi osoittautua hyödylliseksi poikkeamanhallinnan kehittämisessä. Poikkeaman kokonaiskeston perusteella voi esimerkiksi saada lisää rahoitusta, jos poikkeamia ei saada ratkaistua tarpeeksi tehokkaasti. Dataa voidaan käyttää tiimin suorituskyvyn mittaamiseen. Voidaan tarkkailla vaikka, miten kapasiteetin lisäys vaikuttaa tiimin suorituskykyyn. Poikkeamien tunnusmerkkejä voi tutkia kerätystä datasta ja hyödyntää tätä riskien arvioinnissa ja parempien tietoturvakontrollien toteuttamisessa. (Cichonski ym. 2012.)

Kerättävään dataan voi sisältyä esimerkiksi seuraavat asiat: Käsiteltyjen poikkeamien määrä, käsittelyyn käytetty aika, objektiivinen arviointi ja subjektiivinen arviointi. Käsittelyyn käytetyssä ajassa voidaan seurata kokonaiskestoa ja sitä, kuinka kauan kesti, että poikkeamasta tehtiin ensimmäinen havainto. Objektiivisessa arvioinnissa tarkastellaan dokumentaatiota, arvioidaan vahinkoja ja mietitään miten poikkeaman olisi voinut ehkäistä. Subjektiivisessa arvioinnissa arvioidaan omaa, tiimin muiden jäsenien ja koko tiimin suoriutumisesta. Siinä arvioidaan myös, oliko asiakas tyytyväinen poikkeaman käsittelyyn. Lisäksi voidaan arvioida menettelytapoja, työkaluja, resursseja, koulutusta ja dokumentointia. (Cichonski ym. 2012.)

Todisteiden säilyttäminen

Todisteiden säilyttämisestä tulee päättää, eli käytännössä päätetään miten ja kuinka kauan todisteita säilytetään. Tässä tulee ottaa huomioon mahdolliset oikeudenkäynnit, se miten kauan tietyntyyppistä dataa, esimerkiksi sähköposteja säilytetään sekä tallennuslaitteiden kustannukset. Mitä pidemmältä ajalta säilytetään dataa, sitä kalliimpaa se on. (Cichonski ym. 2012.)

3.6 Poikkeamanhallinnan työkaluja

Incident Responsessa tarvitaan työkaluja poikkeamien tehokkaaseen tunnistamiseen, luokitteluun, eristämiseen ja niihin reagointiin. Että yrityksen verkkoa voidaan puolustaa tehokkaasti, tarvitaan oikeat ”ammukset” (ammunition), pyritään nimeämään syyllinen (attribution) ja keskitytään parantamaan tietoisuutta (awareness). Näiden 3 A:n avulla pyritään vähentämään poikkeamia ja pienentämään niiden vaikutuksia yrityksessä. (Insider’s guide to incident response n.d.)

Ammunition (Ammukset)

Ammukset tarkoittavat käytännössä työkaluja. Tässä vaiheessa hankitaan incident response työkaluja ja muokataan niitä omiin käyttötarkoituksiin sopiviksi. Suuri osa incident response työntekijöistä käyttää suurimman osan ajasta tämän tekemiseen. (Insider’s guide to incident response n.d.)

Attribution (Syyllisen nimeäminen)

Tässä vaiheessa pyritään löytämään, mistä hyökkäys on tulossa. Tämä auttaa ymmärtämään hyökkääjän aikeet ja tekniikan, varsinkin, jos käytetään apuna myös reaaliaikaista tietoa tietoturvauhista.
(Insider's guide to incident response n.d.)

Awareness (tietoisuus)

Keskeisin tietoturvamekanismi on koulutettu ja tietoinen käyttäjä. Jokainen poikkeama tulisi käsitellä siten, että parannetaan tietoturvaa kokonaisvaltaisesti. Tietoisuus on keskeinen osa tätä.
(Insider's guide to incident response n.d.)

Seuraavaksi käydään läpi, mitä työkaluja tarvitaan ja käytetään incident responsen eri vaiheissa. Tässä käytetään apuna OODA-loopin vaiheita.

Observe (havainnointi)

Tässä vaiheessa käytetään tietoturvamonitorointia epänormaalin tutkimista vaativan toiminnan tunnistamiseen. Tässä vaiheessa käytetään seuraavia työkaluja: lokien hallinta ja analysointi, tunkeutumisen havaitsemisjärjestelmät, verkkoliikenneanalysaattorit, haavoittuvuusskannerit, saatavuusmonitorointi ja välityspalvelimet. (Insider's guide to incident response n.d.)

Lokien hallinta ja analysointi (log management and analysis)

Lokit ovat paras lähde ymmärtää, mitä verkossa tapahtuu. Näitä varten tarvitaan IR-työkalu, jonka avulla ymmärretään paremmin lokitiedostoja. Tämä on käytännössä lokien analysoinnin tarkoitus. (Insider's guide to incident response n.d.)

Tunkeutumisen havaitsemisjärjestelmät (Intrusion Detection System, IDS)

Monitoroidaan palvelimia ja verkkoja reaaliajassa käyttäen apuna merkkejä hyökkäyksestä ja vertaamalla tätä baseline-tietoon ja annetaan hälytyksiä, kun tunnettuja hyökkäyksiä tai epäilyttävää toimintaa havaitaan. (Insider's guide to incident response n.d.)

Verkkoliikenneanalysaattorit (Netflow analyzers)

Verkkoliikenneanalysaattorit analysoivat aitoa verkkoliikennettä verkon sisällä ja sen reunalla. Tämän avulla saadaan käsitys siitä mitä protokollia käytetään verkossa ja mitkä suojattavat kohteet kommunikoivat keskenään. Voidaan seurata yleisesti verkossa tapahtuvaa toimintaa trendien muodostamiseksi tai seurata tarkemmin tietyntyyppistä liikennettä. (Insider's guide to incident response n.d.)

Haavoittuvuusskannerit (vulnerability scanners)

Haavoittuvuusskannerien avulla voi verkosta ja järjestelmistä etsiä potentiaalisia riskejä. Tämä helpottaa arvioimaan yrityksen hyökkäyspinta-alaa ja tämän tiedon avulla voidaan päättää mahdollisista parannustoimenpiteistä tietoturvaan. (Insider's guide to incident response n.d.)

Saatavuusmonitorointi (Availability monitoring)

Incident Responsen tavoite on välttää järjestelmien alhaalla oloaikaa mahdollisimman paljon. Tämän takia tarvitaan työkaluja palveluiden tai järjestelmien saatavuusmonitorointiin. Jonkin palvelun alhaalla olo voi olla ensimmäinen merkki poikkeamasta. (Insider's guide to incident response n.d.)

Välityspalvelimet (Web proxies)

Välityspalvelimien usein ajatellaan olevan tarkoitettu vain verkkosivujen pääsynhallintaan. Niillä voidaan myös lokittaa, ketkä ovat yhteydessä verkkoon ja tämä on elintärkeää, koska monet modernit hyökkäykset tapahtuvat HTTP:n kautta. Välityspalvelimella voidaan lokittaa IP-osoitteiden lisäksi HTTP-yhteyttä tarkemmin ja tämä on tärkeää, kun tutkitaan ja paikannetaan ongelmaa. (Insider's guide to incident response n.d.)

Orient (tilanteen arviointi)

Tässä vaiheessa arvioidaan mitä tapahtuu kyberuhkien maisemassa ja yrityksen sisällä. Tämän perusteella tehdään loogisia reaaliaikaisia asiayhteyksiä, jotta voidaan keskittyä tärkeisiin tapahtumiin. Tässä käytettäviä työkaluja ovat suojattavien kohteiden listaus sekä järjestelmät tietoisuuden jakamiseen tietoturvauhista ja tietoturvatutkimukseen. (Insider's guide to incident response n.d.)

Suojattavien kohteiden listaus (Asset inventory)

Että verkon tapahtumia voidaan priorisoida, pitää olla lista kriittisistä järjestelmistä verkossa ja siitä mitä ohjelmistoja niihin on asennettu. Oma ympäristö pitää tuntea, että voidaan arvioida poikkeamien vakavuutta. Suojattavien kohteiden listaamiseen on hyvä olla automatisoitu järjestelmä, jossa voi päivittää tietoja tarvittaessa. (Insider's guide to incident response n.d.)

Tietoisuus uhista ja tietoturvatutkimus (Threat intelligence, Security research)

On olemassa järjestelmiä, jossa jaetaan globaalia tietoa riskeistä reaalimaailmassa. Niissä voidaan jakaa esimerkiksi merkkejä tietomurroista ja pahamaineisia IP-osoitteita. Niistä saatavaa tietoa voi verrata omaan verkkoon ja tehdä tämän perusteella asiayhteyksiä. (Insider's guide to incident response n.d.)

Decide (päätös)

Tässä vaiheessa havaintoihin ja tehtyihin asiayhteyksiin perustuen valitaan paras taktiikka poikkeaman käsittelyyn, joka minimoi vaikutukset ja palautumisajan. Tässä vaiheessa käytettävät työkalut ovat yrityksen tietoturvakäytänteet ja Fyysinen dokumetaatio (muistikirja, kynä, kello). (Insider's guide to incident response n.d.)

ACT (toiminta)

Tässä vaiheessa korjataan ongelma ja palaudutaan siitä. Opitun perusteella parannetaan IR-prosessia. Tässä käytetään seuraavia työkaluja: tiedonkaappaustyökalut, IR-tutkimustyökalut, varmuuskopiointi ja palautustyökalut, päivitystenhallinta sekä työkalut ja ohjelmat tietoturvatietoisuuskoulutukseen. (Insider's guide to incident response n.d.)

Tiedonkaappaus- ja IR-tutkimustyökalut (Data capture, IR forensic tools)

Tiedonkaappaustyökaluja käytetään muistin, tietokantojen ja verkon tutkimukseen. IR-tutkimustyökaluilla tutkitaan digitaalista mediaa ja tavoitteena on faktojen ja mielipiteiden tunnistus, säilytys, palautus, analysointi, esittäminen ja jäljitysketjun (audit trail) muodostaminen. (Insider's guide to incident response n.d.)

Varmuuskopionti/ palautus, päivitysten hallinta (backup/recovery, patch management)

Varmuuskopiointi- ja palautusjärjestelmät ja päivitystenhallintajärjestelmät ovat erityisen tärkeitä poikkeamasta palautumisessa. (Insider's guide to incident response n.d.)

Työkalut tietoturvatietoisuuden koulutukseen (security awareness training tools)

Tietoturvatietoisuuden koulutus on oleellista yleisen tietoturvan parantamisessa ja poikkeamien todennäköisuuden madaltamisessa. (Insider's guide to incident response n.d.)

Tässä opinnäytetyössä tutkittavat aikajana-työkalut ovat käytännössä IR-tutkimustyökaluja (incident response forensic tools).

3.7 Incident response team, poikkeamanhallintatiimi (CIRT, CSIRT)

Poikkeamanhallintatiimi, CSIRT (Computer Security Incident Response Team) tai CIRT (Computer Incident Response Team) on organisaatio, joka on vastuussa tietoturvapoikkeamien ja niistä tehtyjen raporttien vastaanottamisesta, tutkimisesta ja niihin reagoinnista. CSIRT yleensä tarjoaa palveluitaan tietylle määritetylle organisaatiolle, esimerkiksi yritykselle, hallitukselle, valtiolle tai maksavalle asiakkaalle. (CSIRT Frequently Asked Questions n.d.)

CSIRT on ryhmä tietoturvaeksperttejä, jonka pääasiallinen tehtävä on reagoida tietoturvapoikkeamiin. Se tarjoaa tarvittavat palvelut poikkeamien käsittelyyn ja tukee asiakkaitaan tietomurroista palautumisessa. (Bronk, Thorbuegge, Hakkaja 2006)

Riskien rajoittamiseksi ja reagointia vaativien poikkeamien minimoimiseksi, useimmat CSIRTit tarjoavat myös poikkeamia ehkäiseviä palveluita sekä koulutuspalveluita asiakkailleen. CSIRT voi jakaa ohjeita haavoittuvuuksista sekä ohjeistaa käyttäjiä hait-

taohjelmista tai viruksista, jotka hyödyntävät kyseisiä haavoittuvuuksia. Näiden ohjeiden avulla asiakas voi nopeasti päivittää järjestelmiään turvallisemmaksi. (Bronk ym. 2006)

CSIRT voi olla virallinen tiimi, jonka pääasiallinen tehtävä on harjoittaa poikkeamanhallintaa tai ad-hoc tyyppinen tarvittaessa yhteen kutsuttava ryhmä ihmisiä. CSIRT voidaan jakaa ainakin seuraaviin kategorioihin (CSIRT Frequently Asked Questions n.d.):

- Sisäinen CSIRT, joka tarjoaa poikkeamanhallintapalveluita emoyritykselleen
- Kansallinen CSIRT, joka tarjoaa poikkeamanhallintapalveluita valtiolle
- Koordinaatiokeskukset, jotka koordinoivat ja johtavat useita CSIRTejä
    - Esimerkiksi CERT Coordination Center
- Analysointikeskukset, jotka yhdistävät dataa eri lähteistä löytääkseen trendejä ja kuvioita poikkeamissa
    - Tätä tietoa voi hyödyntää ennakoinnissa
- Laite- ja ohjelmistotoimittajien tiimit, jotka käsittelevät raportteja haavoittuvuuksista omissa ohjelmisto- ja laitteistotuotteissaan.
- Poikkeamanhallintapalvelujen tarjoajat, tarjoavat palveluita muille yrityksille

Parhainkaan tietoturvainfrastruktuuri ei takaa, että tietomurtoja tai muita haitallisia tapahtumia ilmene. Kun tietoturvapoikkeama ilmenee, on kriittistä, että yrityksellä on tehokas tapa reagoida siihen. Tätä varten yrityksellä tulisi olla CSIRT. (CSIRT Frequently Asked Questions n.d.)

CSIRTistä on yritykselle paljon etuja. Se auttaa ehkäisemään ja lieventämään suuria poikkeamia ja auttaa suojaamaan yrityksen arvokkaita suojattavia kohteita. CSIRT tarjoaa keskitetyn ohjauksen ja yhteyspisteen (point of contact) tietoturvaongelmien ilmaantuessa. CSIRT tarjoaa myös keskitetyn ja erikoistuneen ryhmän poikkeamien käsittelyyn ja niihin reagointiin. CSIRTillä on lisäksi asiantuntemusta käyttäjien tukemiseen ja opastamiseen mahdollistaen nopean palautumiseen poikkeamista. CSIRTistä hyötyä oikeudellisissa kysymyksissä ja todisteiden säilyttämisessä mahdollisissa

käräjäasioissa. Lisäksi CSIRT seuraa mitä tapahtuu tietoturvan alalla ja tekee yhteistyötä asiakkaan kanssa tietoturvassa ja rakentaa tietoisuutta siitä. (Bronk ym. 2006)

Mitä nopeammin yritys pystyy havaitsemaan, analysoimaan ja reagoimaan poikkeamaan, sitä vähemmän aiheutuu kustannuksia ja sitä nopeampaa on palautuminen. CSIRT voi olla paikalla ja tehdä nopean suunnitelman, miten poikkeama rajataan ja miten palaudutaan siitä. (CSIRT Frequently Asked Questions n.d.)

CSIRT voi myös tuntea paremmin vaarantuneet järjestelmät ja tämän ansiosta sillä on paremmat valmiudet ohjata palautumista poikkeamasta ja tarjota strategioita vastatoimiin. Myös CSIRTin suhteista toisiin CSIRTeihin on hyötyä, kun he voivat jakaa keskenään strategioita ja varoituksia. (CSIRT Frequently Asked Questions n.d.)

CSIRTissä tulisi olla jäseniä, joilla on erilaisia teknisiä taitoja ja luonteenpiirteitä, kuten kommunikointitaitoja. CSIRT henkilöstön pitäisi olla omistautunutta, innovatiivista, yksityiskohtiin suuntautunutta, joustavaa ja analyyttistä. CSIRT työntekijät ovat hyviä ratkaisemaan ongelmia, hyviä kommunikoimaan ja pystyvät käsittelemään stressaavia tilanteita. CSIRT henkilöstöllä voi olla seuraavia rooleja (CSIRT Frequently Asked Questions n.d.):

- Tiimin johtaja
- apujohtajat, valvojat, ryhmän johtajat
- puhelinpalvelu, help desk ja luokittelu
- Poikkeaman käsittelijät
- Haavoittuvuuksien käsittelijät
- Analysoijat
- eri alustojen asiantuntijat
- kouluttajat
- teknologian tarkkailijat

Näiden lisäksi CSIRTin kanssa työskentelee seuraavissa rooleissa olevia työntekijöitä (CSIRT Frequently Asked Questions n.d.):

- tukihenkilöstö
- tekniset kirjoittajat
- verkon ja järjestelmänvalvojat
- ohjelmoijat ja ohjelmistokehittäjät
- web-kehittäjät ja ylläpitäjät
- mediasuhteet
- lakiosasto
- tarkastajat ja laadunvalvonta
- markkinointiosasto

3.8 CSIRTin tekninen infrastruktuuri

Ennen kuin aletaan listaamaan teknisen infrastruktuurin elementtejä pitää päättää mitä palveluja CSIRT tarjoaa. Tarjottavat palvelut määrittävät sen, mitä resursseja tarvitaan onnistuneeseen toimintaan. CSIRTin palvelujen tulisi mahdollistaa ja tukea asiakasyrityksen tavoitteiden saavuttamista. Palveluiden tulisi olla myös sellaiset, että ne voidaan realistisesti ja rehellisesti tarjota, riippuen tiimin koosta, kokemuksesta ja taidoista. (Penedo 2006.)

Puhelinlaitteistot

Puhelinta tarvitaan kommunikointiin asiakasyritysten sekä muiden CSIRTien, laitevalmistajien ja ulkoisten kontaktien kanssa. Puhelimen kautta saadaan henkilökohtainen ote kommunikointiin asiakasyrityksen kanssa. Myös puhelun reaaliaikaisuus on etu. CSIRTiin tulisi pystyä ottamaan yhteyttä 24/7. (Penedo 2006.)

Verkon infrastruktuuri

Domain

CSIRTillä tulisi olla oma domain-nimi: Helposti muistettava rajapinta sähköpostille ja CSIRTin web-palveluille. (Penedo 2006.)

Sähköposti

Sähköposti on CSIRTin eniten käytetty kommunikointimuoto. Tästä syystä sähköpostijärjestelmän tulisi olla hyvä ja luotettava. Sähköpostijärjestelmälle oleellista on

myös hyvät suodatus ja hakuominaisuudet sekä tietoturva/salaus. Automaattisten incident response työkalujen integrointimahdollisuus on myös eduksi. (Penedo 2006.)

Verkkosivu

Verkkosivu on tehokas tapa jakaa tietoa asiakkaiden kanssa. Myös tässä tietoturva oleellista. Verkkosivun tulisi sisältää hyödyllistä tietoa, esimerkiksi:

- Tiimin tehtävät
- havainnot
- hälytykset
- yhteystiedot
- raportointilomake
- parhaat käytännöt (Penedo 2006.)

Tietoturva ja palomuuri

Kun CSIRTin palvelut ovat tarjolla verkossa, se on myös itse kohde hyökkääjille. CSIRTin tulee olla varautunut tähän. (Penedo 2006.)

Kaiken liikenteen tulisi mennä palomuurin läpi. Palomuurilla kontrolloidaan ja tarkkaillaan pääsyä palveluihin. Siinä lokitus on tärkeää ja lokia tulisi tarkkailla säännöllisesti. (Penedo 2006.)

Palvelimet ja tietokoneet

Tarvittavan laitteiston määrä riippuu tiimin koosta ja sen tarjoamista palveluista. Laitteet valitan siten, että ne tukevat asiakasyritystä. CSIRTillä voi olla esimerkiksi seuraavat palvelimet: DNS, Sähköposti, verkkosivut ja CSIRT-työkalut. Jokaiselle työntekijälle tulisi olla oma tietokone. Lisäksi on hyvä olla muista verkoista erillään olevat järjestelmät ja verkot testaamiseen ja koulutukseen. (Penedo 2006.)

Käyttöjärjestelmät

Linux on useimpien hakkereiden ja tietoturvakonsulttien valinta. Linuxissa on tuki uusille protokollille ja teknologioille ja se on hyvin mukautuva. (Penedo 2006.)

Windowsista on nykyään tullut myös mukautuvampi ja joustavampi. Windowsille on tarjolla yhä enemmän verkonvalvontatyökaluja sekä muita työkaluja. (Penedo 2006.)

MacOS X on myös hyvä vaihtoehto. Siinä on unix käyttäjille tutut standardit komentotulkit. MacOS X tarjoaa myös kääntäjän (compiler) ja kirjastoja, helppoon työkalujen toteutukseen. (Penedo 2006.)

Nykyään on paljon erilaisia käyttöjärjestelmiä käytössä, ja CSIRTillä tulee olla valmiudet käsitellä kaikkia. Virtuaalikoneiden avulla voi olla monta eri käyttöjärjestelmää kätevästi samalla laitteella. (Penedo 2006.)

Incident Response-työkalut

CSIRTin tulee kirjata ylös kaikki tieto: Poikkeamien historiatiedot, kaikki kommunikaatio, lokitiedostot, todisteet ja tehdyt toimenpiteet. Tähän ei ole täydellistä ratkaisua ja ratkaisu riippuu asiakasyrityksen koosta ja tapahtumien määrästä verkossa. Isossa yrityksessä tulee olla tietokanta, johon kaikki tieto kirjataan ylös. Tiketöintijärjestelmää käytetään raporttien seuraamiseen. (Penedo 2006.)

Salaus

CSIRTissä salaus on tärkeää, koska siellä käsitellään luottamuksellista ja arkaluontoista tietoa. Vahvoja salausjärjestelmiä edistyneillä algoritmeilla on saatavana ilmaisina ja kaupallisina. Salausta on käytettävä, jos tieto on vähänkin arkaluontoista. Incident datan siirtäminen toimipisteiden välillä tulisi tehdä VPN:n avulla. (Penedo 2006.)

Tietoturvatyökalut

Tarvittavat tietoturvatyökalut määrittyvät sen perusteella, mitä palveluja CSIRT tarjoaa asiakkailleen. Tietoturvatyökaluihin sisältyvät niin pienemmät työkalut kuin isommat järjestelmät. (Penedo 2006.)

Varmuuskopiointi

Varmuuskopiot ovat ainoa toivo palauttaa alkuperäistä dataa. Varmuuskopiointi luokitellaan tietoturvamekanismiksi, koska se on viimeinen puolustuskeino. Useita var-

muuskopioita on pidettävä yllä ja säilytettävä fyysisesti eri paikoissa. Varmuuskopiointijärjestelmiä tulee myös testata eheyden varmistamiseksi ajoittain. (Penedo 2006.)

Forensiikkatyökalut

Forensiikkatyökaluja tarvitaan, jos CSIRT tekee forensiikkaa. Forensiseen analyysiin on olemassa järjestelmiä tallennuslaitteilla olevan datan tutkimiseen. Niiden avulla voidaan palauttaa dataa, kerätä todisteita ja tehdä raportteja. (Penedo 2006.)

4 Tilannetietoisuus

4.1 Situational awareness eli tilannetietoisuus

Tilannetietoisuuden parantaminen on tärkeä tavoite yrityksille, jotka toimivat useilla sektoreilla ja joilla on useita toimialueita. Tilannetietoisuus tukee päätöksentekijää auttamalla ymmärtämään tilanteen. Lisääntynyt tilannetietoisuus ympäristöstä mahdollistaa paremman päätöksenteon. (Hall, Hansen & Jones 2015.)

Päätöksiä tehdään yhä enemmän tilanteissa, jotka vaativat yhteistyötä yrityksen sisällä. Keskinäiset riippuvuudet yrityksen sisällä monimutkaistavat parhaiden mahdollisten tietoisten päätösten tekoa. Päätöksentekijän tilannetietoisuuden parantaminen johtaa parempiin päätöksiin ja parempiin toimenpiteisiin. (Hall ym. 2015.)

Tilannetietoisuus tarkoittaa tietämystä siitä, mitä on tapahtumassa ympäristössä riippuen sen tämänhetkisestä tilasta. Luonnollisesti tässä on tärkeää tietää mikä on tärkeintä milläkin hetkellä. Vaikka tilannetietoisuuden elementit vaihtelevat toimialoittain, tilannetietoisuuden luonto ja mekanismit voidaan kuvailla yleisesti. Tilannetietousuus on paljon tutkittu aihe, jota tarvitaan useilla aloilla kuten esimerkiksi avaruustekniikka, asevoimat ja tietenkin kyberturvallisuus. (Hall ym. 2015.)

Tilannetietoisuutta tarvitaan arkaluontoisen datan suojaamiseen sekä yritykselle olennaisten toimenpiteiden ylläpitämiseen. Tilannetietoisuus tarkoittaa sitä, että

tunnetaan oma ympäristö ja potentiaalisiin riskit sekä osataan reagoida niihin. Järjestelmissä ja verkoissa on haavoittuvuuksia, jotka aiheuttavat riskejä yrityksille. Kun osataan ennakoida mitä järjestelmille saattaa tapahtua, johtajat voivat tehdä tehokkaita vastatoimia kriittisten liiketoimintaoperaatioiden suojaamiseksi. (Situation Awareness n.d.)

Kokonaisvaltainen tilannetietoisuus kyberturvallisuudessa sisältää kolme osa-aluetta: tietoisuus omasta verkosta ja järjestelmistä (network awareness), tietoisuus mahdollisista uhista (threat awareness) ja tietoisuus kriittisistä tehtävistä (mission awareness). Tämän tason tilannetietoisuuden saavuttaminen vaatii investointeja tiedon keräykseen, hallintaan ja analysointiin, että saadaan ylläpidettyä kuvaa siitä, miten tietojärjestelmät, verkot ja käyttäjät toimivat verkossa. Kyberturvallisuus on tuonut uuden ulottuvuuden tietoisuuteen, jota tarvitaan liiketoimintaoperaatioissa. Tämän tietoisuuden avulla negatiiviset tilanteet voidaan tunnistaa ja hallita, kun ne ilmaantuvat. (Situation Awareness n.d.)

Network awareness (tietoisuus omasta verkosta)

Tähän osa-alueeseen sisältyy suojattavien kohteiden ja konfiguraatioiden hallinta. Eli käytännössä verkossa olevien laitteiden ja niissä olevien konfiguraatioiden dokumentointi. Tähän sisältyy myös verkossa olevien haavoittuvuuksien seuranta, tietoturvapäivitysten hallinta ja niistä raportointi. Lisäksi tunnistetaan ja jaetaan tietoisuutta poikkeamista yrityksessä. (Situation Awareness n.d.)

Threat awareness (tietoisuus uhista)

Tähän sisältyy tietoisuus sekä sisäisistä, että ulkoisista uhista. Tunnistetaan ja seurataan sisäisiä poikkeamia ja epäilyttävää toimintaa sekä sisällytetään tähän tieto ulkoisista uhista. Ulkoisista uhista saa tietoa esimerkiksi osallistumalla alanlaajuisiin yhteisöihin, joissa jaetaan tietoa mahdollisten uhkien tunnusmerkeistä. (Situation Awareness n.d.)

Mission awareness (tietoisuus kriittisistä tehtävistä)

Tässä muodostetaan kokonaisvaltainen kuva kriittisistä riippuvuuksista, joita tarvitaan toimintaan verkossa. Tätä hyödynnetään poikkeaman analysoinnissa poikkeaman jälkeen, poikkeaman luokittelussa ja reaaliaikaisessa reagoinnissa poikkeaman aikana, riskien ja valmiuksien arvioinnissa ennen tehtävän suorittamista (ennakoidaan ja vältetään tilanteita) sekä puolustussuunnittelussa, jossa valmistaudutaan lieventämään vaikutuksia vastaavanlaisen tilanteen mahdollisesti toistuessa. (Situation Awareness n.d.)

Edellä mainitut osa-alueet keskittyivät tilannetietoisuuden taktiseen tasoon. Tämä on tärkeää, mutta korkeamman tason tilannetietoisuutta tarvitaan, jotta voidaan ymmärtää tilanteen vaikutukset yrityksen kykyyn suorittaa tehtävänsä. (Situation Awareness n.d.)

Että voidaan saavuttaa tilannetietoisuus operaatiotasolla, alemman tason yksityiskohdat pitää suhteuttaa yrityksen tehtävään ja liiketoimintaan. Tässä ei ole kyse vain kaiken taktisen tason tiedon kokoamisesta yhteen. Sen sijaan tilannetieto pitää korreloida yrityksen tehtävään ja liiketoimintaan. Esimerkiksi miten se, että puolet palvelimista ovat alhaalla vaikuttaa jonkin palvelun käyttöön. (Situation Awareness n.d.)

Korkein tilannetietoisuuden taso on strateginen taso. Tässä pitää pystyä katsomaan selvästi korkeammalle kuin yksinkertaiseen tietoon uhkatekijöistä, trendeistä niiden toiminnassa, haitallisen toiminnan tunnistamiseen. Tämän tason tietoisuus on olennaista taistelussa hienostuneita vihollisia vastaan kyberturvallisuudessa ja tehokkaiden puolustussuunnitelmien muodostamiseen. (Situation Awareness n.d.)

4.2 Tilannetietoisuuden määritelmiä

Tilannetietoisuudelle on olemassa monia määritelmiä; toiset yleisiä, toiset tiukasti tiettyyn käyttötarkoituksen sidottuja (esimerkiksi avaruustekniikka, kyberturvallisuus). Endsleyn malli tilannetietoisuudesta on parantanut tietämystä päätöksiä tukevien järjestelmien suunnittelusta. Endsleyn antama määritelmä tilannetietoisuudesta on laajasti hyväksytty. Endsley määrittelee tilannetietoisuuden seuraavasti:

”the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future”
"Tietoisuus elementeistä ympäristössä tietyssä tilassa ja ajassa, ymmärrys niiden merkityksestä ja ennuste niiden tilasta lähitulevaisuudessa" (Hall ym. 2015; Endsley 1995)

Seuraavaksi vielä 3 muuta määritelmää tilannetietoisuudesta:

"SA is an abstraction that exists within our minds, describing phenomena that we observe in humans performing work in a rich and usually dynamic environment."
"Tilannetietoisuus on abstraktio ihmisen mielessä, joka kuvailee ilmiötä, jota tarkkailemme ihmisissä, jotka tekevät töitä rikkaassa ja dynaamisessa työympäristössä." (Hall ym. 2015)

"SA provides the primary basis for subsequent decision making and performance in the operation of complex, dynamic systems..."
"Tilannetietoisuus tarjoaa ensisijaisen perustan peräkkäisten päätösten tekemiseen ja toimintaan monimutkaisissa ja dynaamisissa järjestelmissä." (Hall ym. 2015)

"Situational awareness is probably the pre-requisite state of knowledge for making adaptive decisions in situations involving uncertainty. Situational awareness is the knowledge, cognition and anticipation of events, factors and variables affecting the safe, expedient and effective conduct of the mission.”
"Tilannetietoisuus on edeltävä tila tietämykselle joustavaan päätöksentekoon epävarmoissa tilanteissa. Tilannetietoisuus on tieto, tajunta ja ennakointi tapahtumista, tekijöistä ja muuttujista, jotka vaikuttavat turvalliseen, sopivaan ja tehokkaaseen tehtävän suorittamiseen." (Hall ym. 2015)

4.3 Tilannetietoisuus: Endsleyn malli

Endsleyn malli määrittää tilannetietoisuudelle kolme tasoa:

1.Havainto (Perception)
-Ensimmäinen vaihe tilannetietoisuuden saavuttamiseen on havaita tärkeiden elementtien tila, piirteet ja dynamiikka ympäristössä. (Hall ym. 2015)

2. Ymmärrys (Comprehension)
-Ymmärrys tilanteesta perustuu synteesiin irrallisista vaiheen 1 elementeistä. Kun ensimmäisessä vaiheessa on saavutettu tietoisuus elementeistä, tässä vaiheessa pyritään saavuttamaan ymmärrys niiden tärkeydestä suhteessa tavoitteeseen. (Hall ym. 2015)

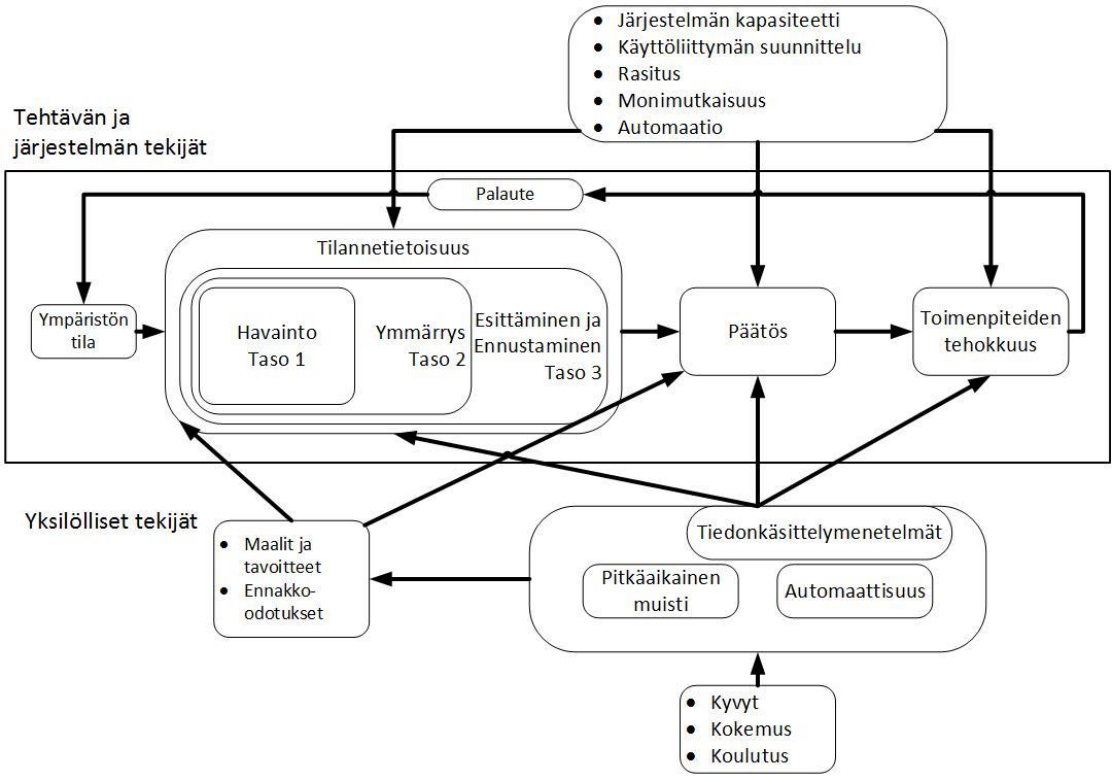
3. Esittäminen ja ennustaminen (Projection)
Kolmas vaihe tilannetietoisuuden saavuttamisessa on tulevan tilan ennustaminen. Tämä saavutetaan tiedolla elementtien tilasta ja dynamiikasta sekä ymmärryksellä tilanteesta (Hall ym. 2015)

Tilannetietoisuuden pitää muuttua koko ajan tilanteiden dynaamisesta luonteesta johtuen. Tilannetietoisuus on pääedellytys päätöksentekoon. Voi olla hyvä tilannetietoisuus ja tehdä huono päätöksen tai olla huono tilannetietoisuus ja tehdä hyvän päätöksen (vaikkakin vain tuurilla). Tilannetietoisuus ei ole päätöksentekoa, eikä päätöksenteko tilannetietoisuutta. Kuitenkin päätökset muodostetaan tilannetietoisuuden avulla ja tilannetietoisuus muodostuu päätösten avulla. (Hall ym. 2015.)

Päätökset ovat keskeisiä tilannetietoisuuden muodostamisessa. Ihmisten tiedon prosessointia monimutkaisissa järjestelmissä toiminnassa voi kuvata kahdella tavalla. Nämä tavat ovat tietoon perustuva prosessointi (alhaalta ylöspäin) ja tavoitteisiin perustuva prosessointi (ylhäältä alaspäin). Tietoon perustuvassa prosessoinnissa havaintojen perusteella etsitään uusia tavoitteita. Tavoitteisiin perustuvassa proses-

soinnissa tavoitteisiin perustuen tulkitaan olemassa olevaa tietoa. Dynaaminen vaihtelu näiden välillä on tärkeää menestyneeseen toimintaan monissa ympäristöissä. (Hall ym. 2015.)

Endsleyn tilannetietoisuusmalli esitellään alla (kuvio 3). Kuviossa keskellä on tilannetietoisuuden prosessi, jossa ensin on ympäristön tila, josta muodostetaan tilannetietoisuus, jonka perusteella tehdään päätöksiä, jotka johtavat toimenpiteisiin, joista annetaan palautetta ja lopulta mennään taas prosessin alkuun. Kuviossa ylhäällä on tehtävän ja järjestelmän tekijöitä jotka vaikuttavat tilannetietoisuuteen, päätöksiin ja toimenpiteiden tehokkuuteen. Näitä tekijöitä ovat järjestelmän kapasiteetti, käyttöliittymän suunnittelu, järjestelmän rasitus, monimutkaisuus ja automaatio. Kuviossa alhaalla on yksilöllisiä tekijöitä, jotka vaikuttavat tilannetietoisuuden prosessiin. Yksilöllisiin tekijöihin kuuluu tiedonkäsittelymenetelmät, pitkäaikainen muisti ja toiminnan automaattisuus, joita voi kehittää kykyjen, kokemuksen ja koulutuksen avulla. Kaikki edeltävät yksilölliset tekijät vaikuttavat tavoitteisiin ja ennakko-odotuksiin. (Endsley 1995.)



Kuvio 3 Endsleyn malli tilannetoisuudesta (Endsley 1995)

4.4 Tilannetietoisuus kyberturvallisuudessa

Täyden tilannetietoisuuden saavuttaminen kyberturvallisuudessa vaatii ainakin seuraavien seitsemän osa-alueen integroinnin (Hall ym. 2015):

1. Ole tietoinen tämänhetkisestä tilanteesta (Verkon tietoturva ja laajempi kybervaikutus)
2. Ole tietoinen hyökkäyksen vaikutuksista
3. Ole tietoinen tilanteiden kehittymisestä
4. Ole tietoinen vastustajan käyttäytymisestä
5. Ole tietoinen, miten ja miksi kyseinen tilanne aiheutui
6. Ole tietoinen tilannetietoisuusinformaation laadusta ja luotettavuudesta
7. Arvioi tilanteen todennäköisiä piirteitä

Kyberturvallisuus on osa kokonaisvaltaista tilannetietoisuutta. Sitä ei saa eristää muusta tilannetietoisuudesta. Vaikka Kyber SA (Situational Awareness) liittyy tietoisuuteen kyberturvallisuusongelmista, tämä tieto tulee liittää muuhun tietoon täyden ymmärryksen saavuttamiseen tilanteesta. Kybertapahtumat tarjoavat lisää käsitystä kokonaistilanteesta eikä erillisestä kybertilanteesta.
(Hall ym. 2015.)

4.5 Yhteistyön tärkeys kyberturvallisuudessa ja tilannetietoisuudessa

Työpaikat ovat nykyään yhä monimuotoisempia; on erilaisia väestöryhmiä, maantieteellisesti hajautuneita työpaikkoja ja tämä muuttaa myös hierarkkista ketjua. Tämän takia pitää miettiä uudelleen, miten tiimejä organisoidaan. (Hall ym. 2015.)

Yhteistyötä kyberturvallisuudessa voidaan kuvata käsitteellä CFT (Cross Functional Team). CFT on ryhmä ihmisiä, joilla on erilaisia taitoja, joiden avulla ryhmä tekee työtä yhteistä tavoitetta kohti. CFT:t ovat tehokkaita, kun ne on toteutettu oikein. CFT mahdollistaa ihmisten erilaisilta osaamisalueilta vaihtaa tietoa, koordinoida kohti senhetkistä tavoitetta ja ratkaista monimutkaisia ongelmia. CFT tiimi voi sijaita samassa paikassa tai olla maantieteellisesti hajaantunut. (Hall ym. 2015.)

Kyberturvallisuudessa CFT:t ovat kriittisiä, koska verkkoinsinööritaito ei ole ainoa tarvittava taito kyberturvallisuudessa. Kyberturvallisuus ei ole rajoittunut vain siihen mitä verkossa tapahtuu vaan siihen sisältyy myös hyökkäyksen mahdolliset seuraukset, käytänteiden muuttaminen uusiin tarpeisiin ja liiketoimintaprosessit, että voidaan päätellä hyökkäyksien mahdollisia seurauksia. Tämän takia kyberturvallisuudessa tarvitaan myös operatiivista, logistista, maantieteellistä, kulttuurillista, psykologista ja markkinoinnillista osaamista. (Hall ym. 2015.)

Onnistuneen toiminnan saavuttamiseksi tiimin pitää saavuttaa synergia. Tämä mahdollistaa parempien tuloksien saavuttamisen nopeammin. Synergian saavuttamiseksi pitää seuraavat kolmea avainasiaa olla kunnossa: kommunikointi, yhteistoiminta ja koordinointi. Tehokkaat CFT:t sisältävät jäseniä, jotka kommunikoivat hyvin toistensa kanssa, tekevät hyvin yhteistyötä ja ovat hyvin koordinoituneita. Tämä on tärkeää kaikilla tiimin tasoilla, alhaalta ylöspäin hierarkkisessa ketjussa. (Hall ym. 2015.)

4.6 Timeline analysis

Poikkeamanhallinnan tutkimuksessa pyritään löytämään vastaukset seuraaviin kysymyksiin: kuka, mitä, missä, milloin, miten, miksi. Yksi välttämätön tässä tarvittava työkalu on aikajana. Se voi olla mitä tahansa muistikirjassa olevan luonnoksen ja interaktiivisen tietokoneella tehdyn grafiikan väliltä. Aluksi aikajana on lajittelematon ja jäsentämätön kokoelma dataa, mutta jos se hallitaan oikein, siitä tulee työkalu, joka voi yhtenäistää tutkimusta, auttaa täyttämään aukkoja tiedossa ja mahdollistaa selkeän kommunikoinnin johdon kanssa. (Liston 2012.)

Aikajanan ydinelementti on tapahtuma, jota voidaan kuvailla seuraavien määreiden avulla (Liston 2012):

- Aika
    - Tarkka tai epävarma
- Paikka
    - Fyysinen sijainti, IP-osoite, tiedostosijainti
- Tekijä

    - tunnettu tai tuntematon
- Toiminta
    - mitä tapahtui
- kohde
    - joku tai jokin johon tapahtuma kohdistui
- Lisäksi merkintää voidaan rikastaa
    - Tageilla
        - lisätään tageja (merkintöjä) tärkeiden tapahtumien tunnistamiseen ja tutkimuksen myöhemmän dokumentoinnin avustamiseen
    - Todisteilla
        - Kirjataan todisteita siitä, että jotain tapahtui. Ei vain anneta vaikeasti tulkittavaa lokimerkintää, vaan myös tulkinta siitä.

Aikajanojen käytöstä on useita etuja (Inglot, Liu & Antonopoulos 2013):

- Nähdään mitä on tapahtunut lähiaikoina
- Auttaa sulkemaan teorioita pois
- Löydetään todisteita, jotka vaativat jatkokäsittelyä, ja joita ei ehkä ilman aikajanaa huomattaisi
- Aikajanasta on apua myös tapahtumien lajittelussa

Edistynyt aikajana-analyysi

Avain ketterään analyysiin ja nopeaan reagointiin on analysoitavan datan määrän rajoittaminen. Tämä voi tarkoittaa käytännössä myös datan keräämistä helpommin hallittavaan muotoon tai kitkemällä normaalit, ei haitalliset tapahtumat tutkittavasta datasta. On olemassa joitakin toimintatapoja, jotka helpottavat aikajana-analyysiä (Inglot ym. 2013):

- Samanlaisten tapahtumien ryhmittely
- Olennaisten tapahtumien korostus
- epäolennaisten tapahtumien piilottaminen
- muistiinpanojen lisääminen
- tehokkaat suodatus ja hakutoiminnot

- visualisointi
- raportointi

Tapahtumien ryhmittely

Tapahtumien ryhmittelyssä pyritään luomaan ylemmän tason tapahtuma, jonka alle tulee alitapahtumia. Esimerkiksi kun käyttäjä Seppo kirjautuu sisään laitteeseen, tapahtuu siinä useita asioita. Tässä voidaan luoda ylemmän tason tapahtuma ”Käyttäjä Seppo kirjautui sisään”. Tämän merkinnän alle saattaa tulla 20 eri tapahtumaa. Näin voidaan rajata tutkittavan tiedon määrää. Tutkija voi tutkia alitapahtumia kuitenkin tarkemmin halutessaan. Tapahtumien ryhmittelyä voidaan tehdä manuaalisesti, mutta tämä vie aikaa ja perustuu täysin tutkijan tietämykseen. Automaattinen ryhmittely on paljon käytöllisempää ja tehokkaampaa. (Inglot ym. 2013.)

Kohinanpoisto

Suurin osa verkon tapahtumista on epäoleellisia eli ”kohinaa”. Kohinan poistaminen tarkoittaa epäoleellisten tapahtumien suodattamista. (Inglot ym. 2013.)

Edistynyt esittäminen

Visualisointi on ongelmallista, koska se on tehokasta vain, jos valitaan oikea tapa. Visualisoinnin tulisi olla myös interaktiivista ja joustavaa. Tähän mennessä aikajanaanalyysin alalla ei ole paljon onnistunutta työtä. Tästä käy ilmi aukko aikajanadatan ja sen käsittelyyn saatavilla olevien ohjelmien välillä. Tällä alalla on tarve tehdä lisää tutkimusta. Hyvä ratkaisu olisi alustasta riippumaton runko/viitekehys. (Inglot ym. 2013.)

5 TheHive ja FIR vertailu

5.1 Johdanto

Tässä osioissa vertailtiin kahta poikkeamien kirjaamiseen tarkoitettua järjestelmää, jotka ovat TheHive ja FIR. Näistä tutkittiin ja vertailtiin ominaisuudet sekä testattiin rajapinnat tiedon tuontiin järjestelmään ja tiedon vientiin järjestelmästä. Tavoitteena oli saada selville, kumpi järjestelmä on parempi ja saako näihin järjestelmiin tuotua

järkevästi esimerkiksi monitorointidataa ja saako tietoa vietyä järkevästi esimerkiksi aikajanatyökaluun.

5.2 TheHive esittely

TheHive on skaalautuva avoimen lähdekoodin 3-in-1 poikkeamanhallinta-alusta. Sen on tarkoitus helpottaa CSIRTien elämää. TheHiven avulla voi tehdä yhteistyötä (collaborate), käsitellä poikkeamia yksityiskohtaisesti (elaborate) ja analysoida havaintoja (analyze). (CERT-BDF/TheHive n.d.)

Collaborate eli yhteistyö

TheHive on suunniteltu yhteistyötä varten. Useat poikkeamankäsittelijät voivat työskennellä järjestelmässä samanaikaisesti. Twitter-tyylisen ”virtauksen” (flow) ansiosta kuka vain voi seurata mitä järjestelmässä tapahtuu reaaliajassa. (CERT-BDF/TheHive n.d.)

Elaborate eli yksityiskohtainen tutkiminen

TheHivessa jokaiselle tutkimuksella luodaan oma tapaus (case). Tapauksia voi luoda tyhjästä ja tehtäviä voidaan lisätä niihin matkan varrella. Tapauksia voi luoda myös mallipohjien (template) avulla. (CERT-BDF/TheHive n.d.)

Tapaukseen liittyvät tehtävät voidaan määrittää tietylle käyttäjälle tehtäväksi. Käyttäjä voi myös itse merkitä, että ottaa tehtävän hoitaakseen, ilman, että kenenkään tarvitsee määrittää tehtävää hänelle. Jokainen tehtävä voi sisältää useita työlokeja, joihin työntekijät voivat kuvailla mitä ovat tehneet, mikä oli lopputulos ja liittää todisteita ja muita oleellisia tiedostoja. Lokien kirjoittamiseen on TheHivessa tekstieditori. (CERT-BDF/TheHive n.d.)

Analyze eli analysointi

Jokaiseen tapaukseen voi lisätä satoja tai tuhansia havaintoja. TheHive osaa automaattisesti tunnistaa havainnot, jotka ovat olleet jo aiemmissa tapauksissa. Havaintoihin voi myös yhdistää tageilla tiedot niiden tekijästä. Havaintoja voi rajata helposti hakutoiminnon avulla. (CERT-BDF/TheHive n.d.)

Työntekijät voivat analysoida satoja havaintoja muutamalla klikkauksella käyttämällä apuna Cortexin analysaattoreja. Cortex on havaintojen analysoimisalusta, jonka voi yhdistää TheHiveen. Versio 1.0.1 sisältää 13 analysaattoria, ja niitä voi luoda myös itse skriptaamalla. Analysaattoreiden avulla voi esimerkiksi tarkistaa domain-nimiä ja IP-osoitteita, tutkia sijaintia (geolocation) ja tarkistaa epäilyttäviä tiedostoja ja linkkejä viruksien tai muiden haittaohjelmien varalta. (CERT-BDF/TheHive n.d.)

TheHive on kirjoitettu Scalalla ja se käyttää taustalla tallennukseen ElasticSearch versiota 2.x. Käyttöliittymä on toteutettu Bootstrapilla ja AngularJS:llä. TheHiven Rest API on tilaton ja mahdollistaa horisontaalisen skaalautuvuuden lisäämällä palvelimia. (CERT-BDF/TheHive n.d.)

TheHiven suositellut järjestelmävaatimukset ovat 8-ydin prosessori, 8 gigatavua muistia ja 60 gigatavun kiintolevy. Ohjeet TheHiven asentamiseen löytyy sen github-sivuilta. (CERT-BDF/TheHive n.d.)

5.3 FIR (Fast Incident Response) esittely

FIR on kyberturvallisuuden poikkeamanhallinta-alusta. Se on suunniteltu ketteryys ja nopeus mielessä. FIR mahdollistaa helpon poikkeamien luonnin, seuraamisen ja raportoinnin. FIR on tarkoitettu niille, joiden tarvitsee seurata kyberturvallisuus-poikkeamia, eli käytännössä CSIRTeille. FIR on CERT Societe Generale nimisen yrityksen suunnittelema ja on alun perin suunniteltu kyseisen yrityksen tarpeisiin ja käyttöön. Siitä on kuitenkin yritetty tehdä mahdollisimman geneerinen, että muutkin tiimit ympäri maailman voivat muokata ja käyttää sitä kuten parhaaksi näkevät. (certsocietegenerale/FIR n.d.)

FIR on kirjoitettu Pythonilla käyttäen Django 1.9 Frameworkia. Käyttöliittymän tekemiseen visuaalisesti miellyttäväksi on käytetty Bootstrap 3, Ajax ja d3js. Taustalla käytetään tiedon tallentamiseen oletuksena MySQL-tietokantaa. Tässä voidaan käyttää myös muuta tietokantaa, kunhan se on yhteensopiva Djangon kanssa. (certsocietegenerale/FIR n.d.)

FIR:n järjestelmävaatimuksien luvataan olevan matalat. Sen kerrotaan toimivan hyvin ubuntu 14.04 virtuaalikoneessa yhdellä prosessorin ytimellä, 40 gigatavun kiintolevyllä ja 1 gigatavun RAM-muistilla. FIR on avoimen lähdekoodin järjestelmä ja ohjeet sen asentamiseen löytyvät FIR:n github-sivulta. (certsocietegenerale/FIR n.d.)

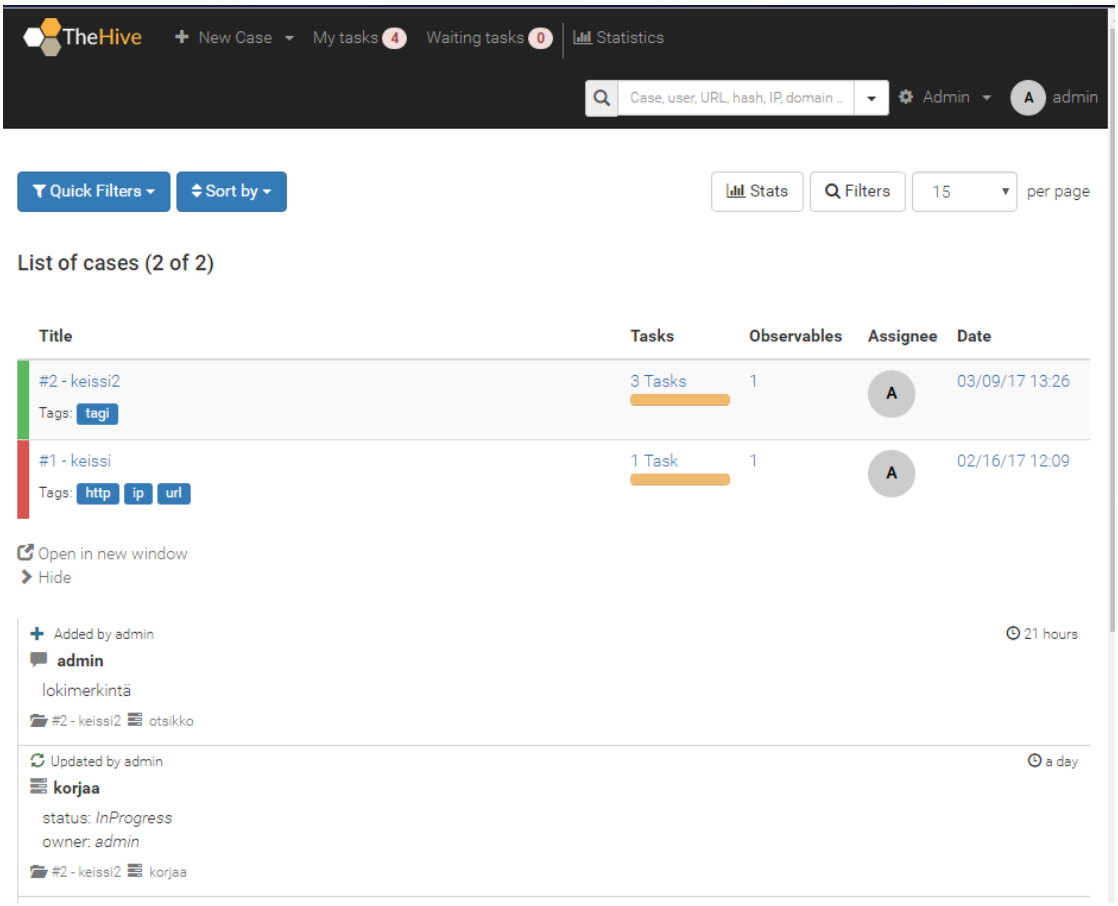
5.4 Asennus

Vertailua ja testaamista varten molemmat järjestelmät asennettiin omalle tietokoneelle virtualboxiin ubuntu-virtuaalikoneille. Molempiin oli selkeät ohjeet, eikä asentamisessa ollut isompia ongelmia, ja asentamiseen molemmissa meni aikaa 1-2 tuntia.

5.5 TheHive ominaisuudet

TheHive etusivu

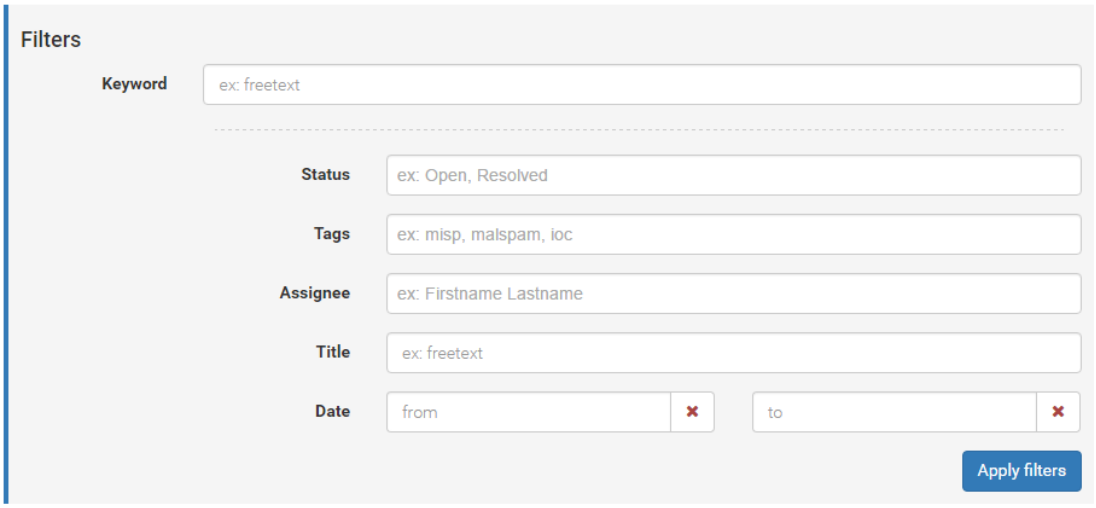
TheHiven etusivulla on lista tapauksista (cases). Listan järjestyksen voi helposti vaihtaa vanhimmasta uusimpaan ja uusimmasta vanhimpaan. Listan voi myös helposti rajata näyttämään avoimet (open) tapaukset, suljetut (closed) tapaukset ja omat tapaukset. Lisäksi etusivulla näkyy reaaliaikainen lista järjestelmän tapahtumista, esimerkiksi tapauksen lisääminen, tehtävien (task) luonti ja havaintojen (observable) lisäys. Etusivun yläreunassa on hakutoiminto, jolla voi hakea tapauksia, havaintoja ja tehtäviä esimerkiksi tagien, tilan (status) ja käyttäjän perusteella. Alla kuviossa 4 on esitetty TheHiven etusivu.



Kuvio 4 TheHive etusivu

TheHiven etusivulta on myös linkit tilastonäkymään (stats) ja suodatustoimintoon (filters). Tilastonäkymässä näkyy tapauksien määrät tilan mukaan eli avoimet ja suljetut tapaukset, tapaukset niiden ratkaisun perusteella (esimerkiksi TruePositive, FalsePositive) sekä 5 eniten käytettyä tagia. TheHivessä on myös tarkempi tilastonäkymä, joka esitellään myöhemmin. Suodatustoiminnolla voi suodattaa tapauksia avainsanojen, tilan, tagien, käyttäjän (assignee), otsikon ja päivämäärän (alkaen, päättyen perusteella). Alla kuviossa 5 on esitetty tilastonäkymä ja kuviossa 6 suodatustoiminto.

Kuvio 5 TheHive tilastonäkymä (stats)

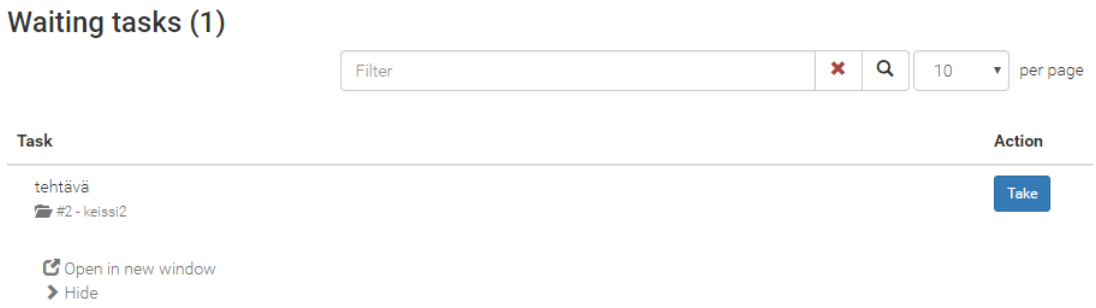


Kuvio 6 TheHive suodatustoiminto (filters)

My tasks ja Waiting tasks näkymät

TheHivessa on my tasks ja waiting tasks näkymät. Waiting tasks näkymässä on lista tehtävistä, joita ei ole määritetty kenellekään tehtäväksi vielä. Siellä voi take-painiketta painamalla ottaa itselleen tehtävän hoitaakseen. My tasks näkymässä on kaikki

omat tehtävät listattuna. Molemmissa näkymissä on suodatus- ja hakutoiminto tehtäville. Alla kuviossa 7 on esitetty waiting tasks -näkymä ja kuviossa 8 my tasks -näkymä.



Kuvio 7 TheHive waiting tasks näkymä

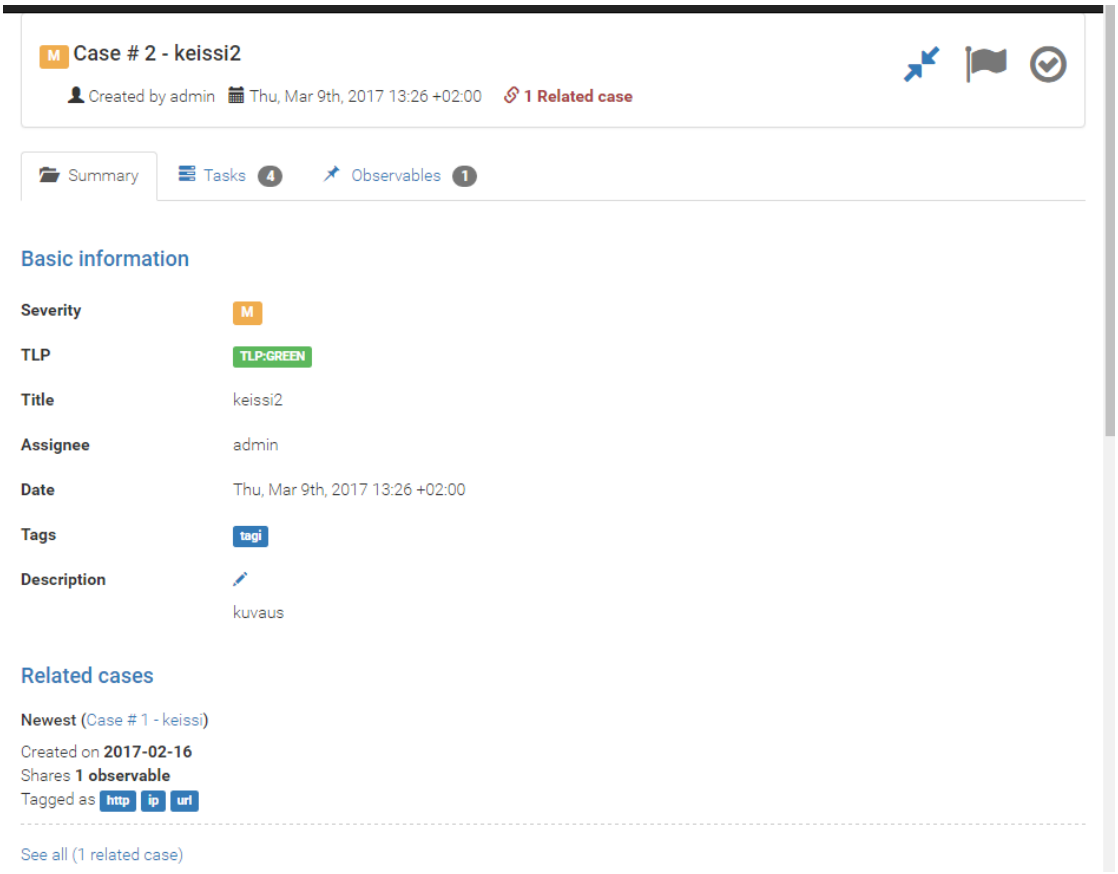
Kuvio 8 TheHive my tasks näkymä

Tietyn tapauksen (case) tarkempi näkymä

TheHivessa voi avata tietyn tapauksen tarkempaan näkymään. Tämän näkymä on vielä jaettu kolmeen eri välilehteen: Summary (yhteenveto), Tasks (tehtävät) ja Observables (havainnot).

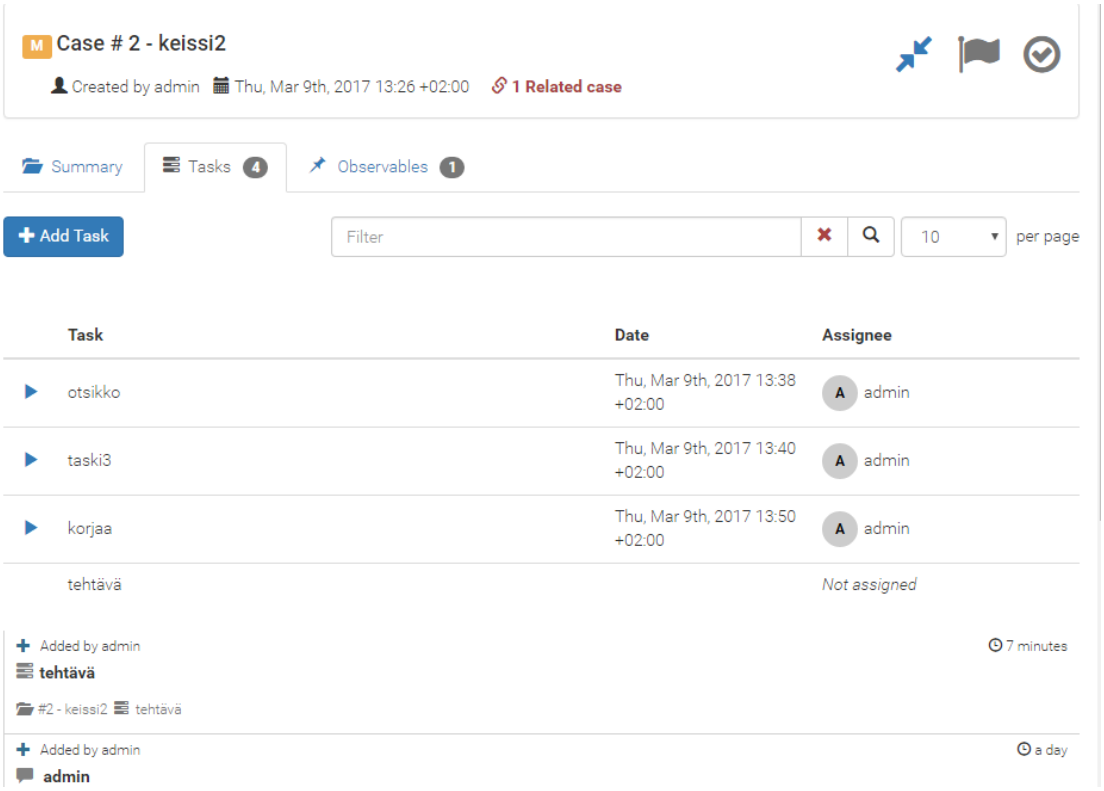
Summary-välilehdellä näkyy tapauksen tiedot: vakavuus, TLP (traffic light protocol), otsikko, käyttäjä (assignee), päivämäärä/aika (luomisaika), tagit ja kuvaus (description), jota voi muokata. Summary-näkymän alareunassa näkyy tapaukseen liittyvät tapahtumat järjestelmässä. Alhaalla näkyy myös tapaukseen liittyvät muut tapaukset,

eli tapaukset joiden kanssa on yhteinen havainto (observable). Lisäksi summary-näkymässä on painikkeet, joilla voi yhdistää tapauksen (merge case), merkitä tapauksen lipulla (flag case) ja sulkea tapauksen (close case). Alla kuvassa 9 on esitetty tapauksen summary-näkymä.

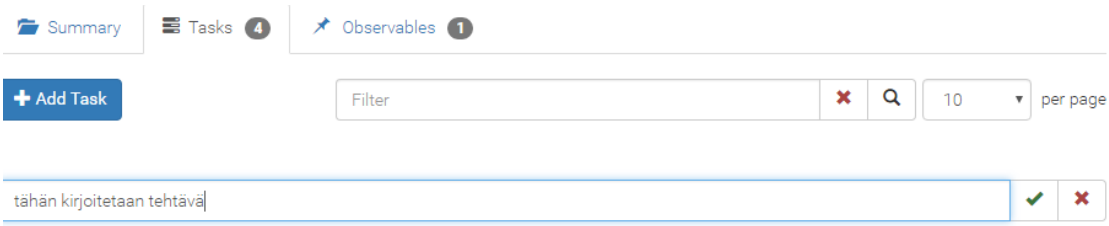


Kuvio 9 TheHive tapauksen yhteenvetonäkymä (case summary)

Tapauksen tarkemmassa näkymässä voidaan myös mennä Tasks-välilehdelle. Tällä välilehdellä näkyy lista kaikista tapaukseen liittyvistä tehtävistä. Siellä voi myös lisätä uusia tehtäviä. Uuden tehtävän lisäämiseen on painike, josta aukeaa yhden rivin kokoinen tekstikenttä, johon tehtävä kirjoitetaan. Tehtävien löytämiseen on haku- ja suodatustoiminnot. Alla on kuviossa 10 on esitetty Tasks-välilehti ja kuviossa 11 tehtävän lisääminen.

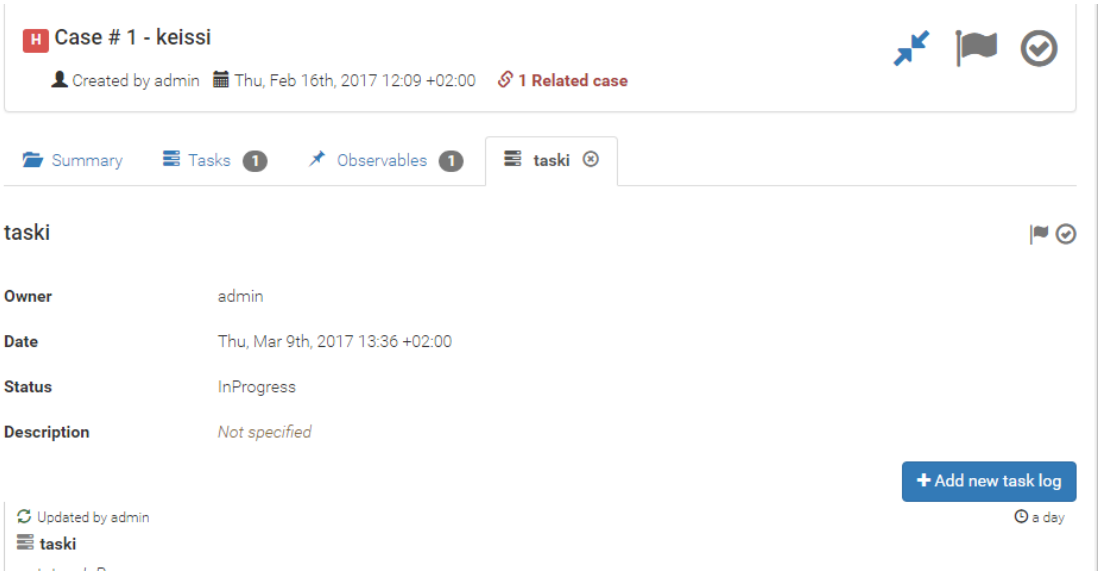


Kuvio 10 TheHive Tasks-välilehti



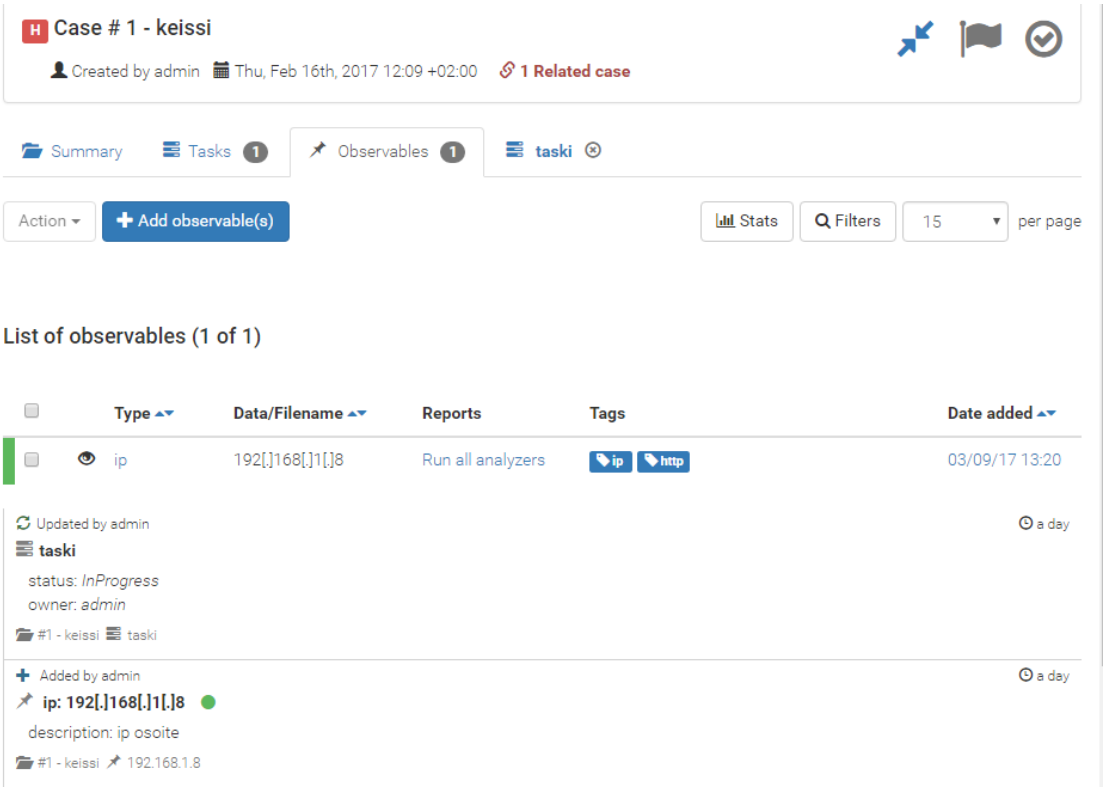
Kuvio 11 TheHive tehtävän lisääminen

Tietyn tehtävän voi myös avata tarkempaan näkymään. Tässä näkymässä näkyy tehtävän omistaja, päivämäärä ja aika, tehtävän tila ja tehtävän kuvaus, jota voi muokata. Tehtävään voi täällä myös lisätä päiväkirjamerkinnän (task log) tehtävälle. Näillä merkinnöillä voi lisätä tietoa tehtävän suorittamisesta. Merkintöjen tekemistä varten on tekstieditori, jolla voi lisätä kuvan ja linkin sekä luoda listoja, taulukoita ja laatikon johon voi laittaa löydettyä koodia. Merkintään voi liittää myös liitetiedoston. Alla on kuviossa 12 esitetty tehtävän tarkempi näkymä.



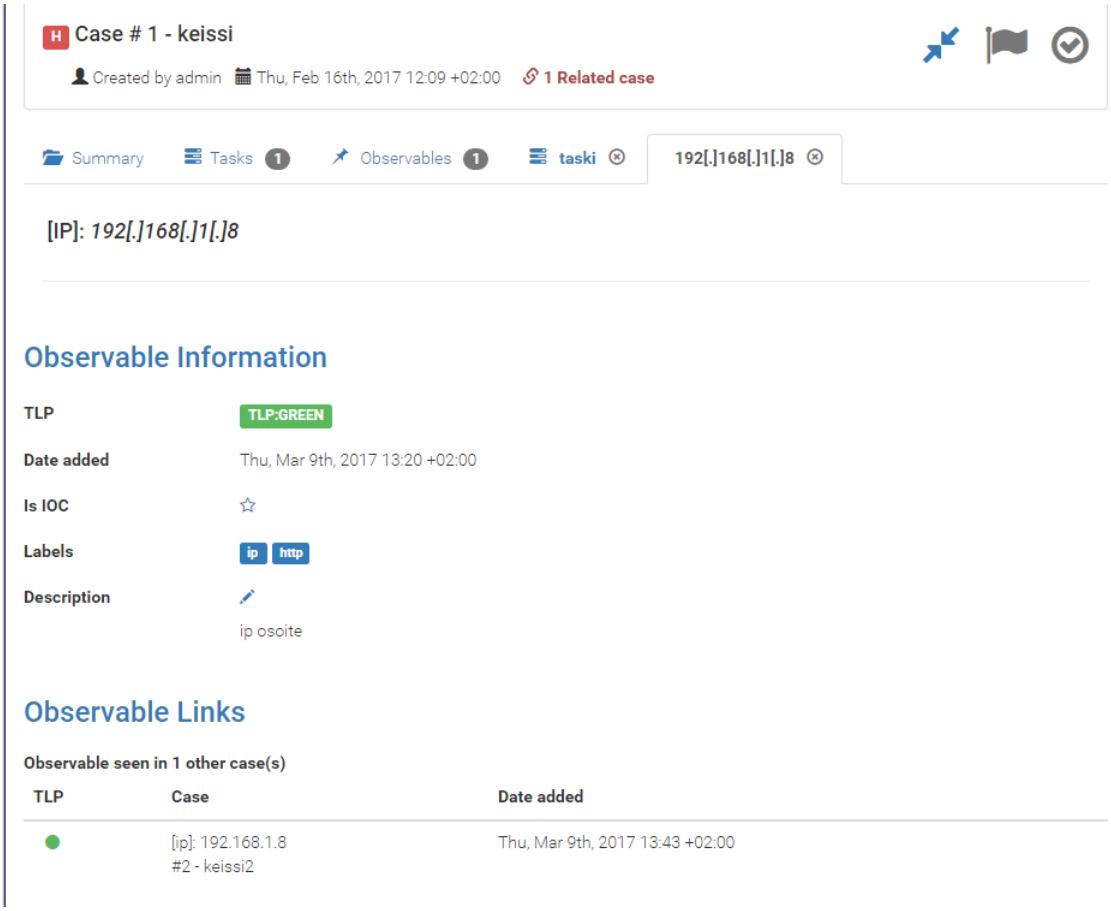
Kuvio 12 TheHive tehtävän tarkempi näkymä

Tapauksen tarkemman näkymän kolmas näkymä on observables-välilehti. Tällä välilehdellä näkyy listattuna kaikki havainnot, jotka on lisätty tapaukselle. Listassa näkyy havainnosta seuraavat tiedot: tyyppi, sisältö/nimi, lisätyt tagit ja päivämäärä/aika. Tällä välilehdellä voi myös lisätä havaintoja. Havainnolle voi antaa seuraavia tietoja: tyyppi (esimerkiksi IP, toimialue, tiedosto, sähköposti), data (sisältö), tagi, TLP (traffic light protocol) sekä havainnon kuvaus. Myös havainnoille on suodatustoiminto, jossa havaintoja voi suodattaa tyypin, tagien, datan, päivämäärän (alkaen > päättyen), kuvauksen ja sen perusteella onko havainto merkki poikkeamasta (IOC, indicator of compromise). Lisäksi observables-välilehdellä voi avata tilastoja havainnoista. Tilastoista voi katsoa havaintojen määriä tyypin perusteella, IOC-havaintojen määrää ja yleisimpiä tageja. Alla on kuviossa 13 esitetty observables-välilehti.



Kuvio 13 TheHive Observables-välilehti

Tietyn havainnon voi avata tarkempaan näkymään. Tässä näkymässä näkyy seuraavia tietoja: TLP, havainnon päivämäärä ja aika, IOC, tagit ja kuvaus, jota voi muokata. Lisäksi tässä näkymässä voi katsoa, onko sama havainto linkitetty myös muihin tapauksiin. Alla on kuviossa 14 on esitetty havainnon tarkempi näkymä.



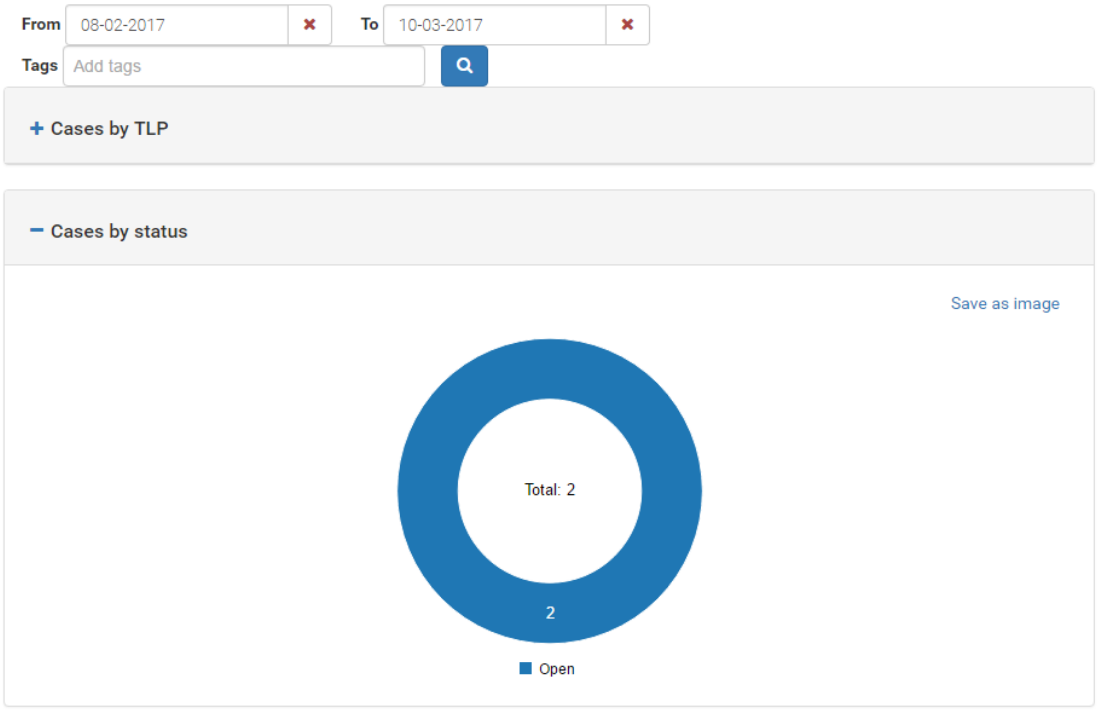
Kuvio 14 TheHive havainnon tarkempi näkymä

Statistics-näkymä

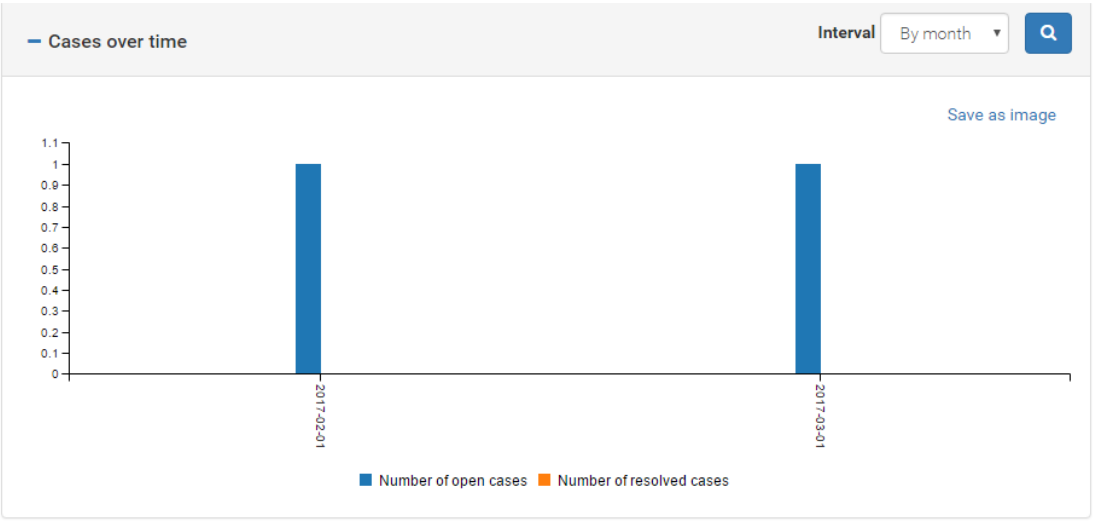
TheHivesta löytyy tilastonäkymä, josta löytyy tapauksiin liittyviä ympyrä- ja pylväskaavioita. Tilastonäkymää voi rajoittaa ajanjakson ja tagien perusteella. Kuvaajia voi myös tallentaa kuvatiedostoiksi. Tilastonäkymästä löytyy seuraavat kuvaajat:

- Tapaukset TLP:n perusteella
- Tapaukset tilan perusteella
- Tapaukset ratkaisun perusteella
- Tapaukset vakavuuden perusteella
- Tapaukset ajanjaksolla
- Tapauksien käsittely ajanjaksolla
- Tapauksien mittaukset (metrics) ajanjaksolla
- havainnot tyypin perusteella
- havainnot IOC perusteella
- havainnot ajanjaksolla

Alla on esimerkkeinä kuviossa 15 esitetty kuvaaja tapauksien tilan perusteella sekä kuviossa 16 kuvaaja tapahtumista ajanjaksolla.



Kuvio 15 TheHive kuvaaja tapaukset tilan perusteella

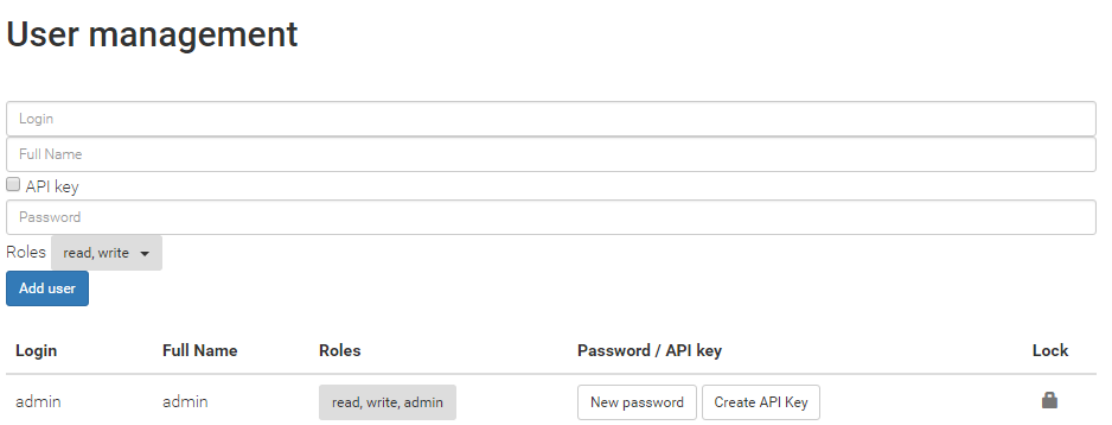


Kuvio 16 TheHive kuvaaja tapaukset ajanjaksolla

TheHive admin-sivu

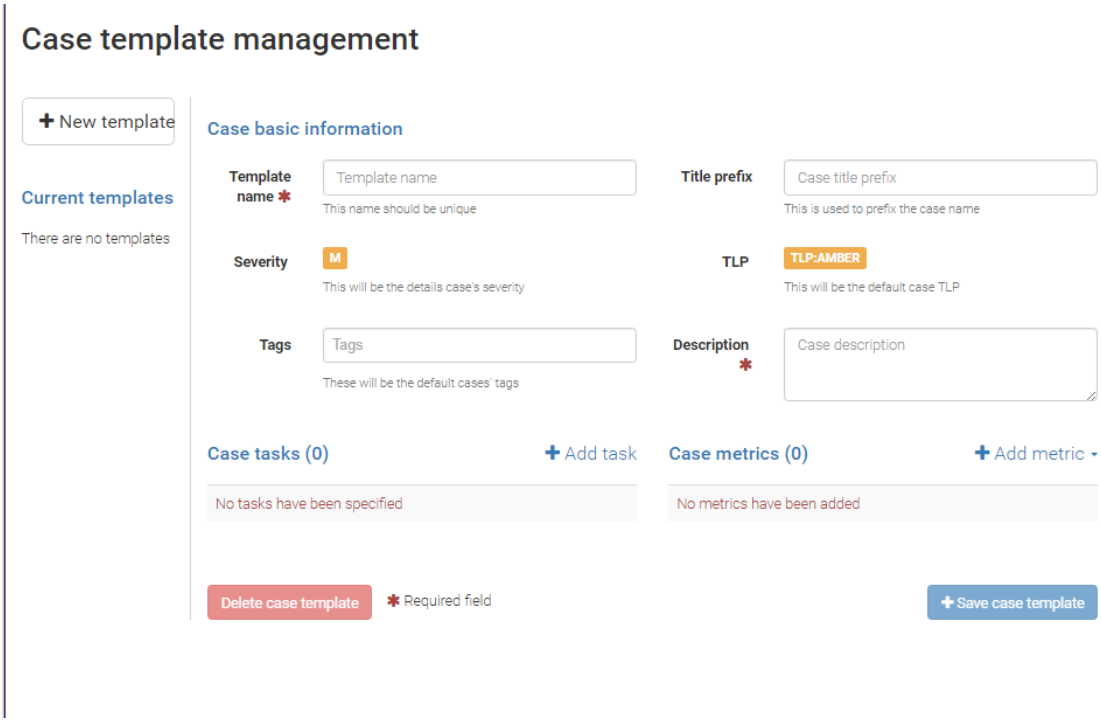
TheHivesta löytyy admin-sivu, jolla voi tehdä hallinnollisia toimenpiteitä. Tällä sivulla voi hallita ja lisätä käyttäjiä, lisätä tapauksen mallipohjia, lisätä raporttien mallipohjia, lisätä tapauksien mittareita (metrics) ja lisätä havaintojen tyyppejä.

Käyttäjien hallinnassa voidaan luoda käyttäjiä ja määrittää käyttäjille oikeudet. Käyttäjille voi määrittää luku, kirjoitus ja järjestelmänvalvojan (admin) oikeudet. Alla on kuviossa 17 esitetty käyttäjien hallintasivu.



Kuvio 17 TheHive käyttäjien hallinta

Tapauksien mallipohjien luonnissa voi määrittää mallipohjalle seuraavia asioita: mallipohjan nimi, otsikon etuliite, vakavuus ja TLP, mallipohjan oletustagit, tapauksen oletuskuvaus, tapauksen tehtävät ja tapauksen mittarit (metrics). Alla on kuviossa 18 esitetty tapauksen mallipohjan luontisivu.



Kuvio 18 TheHive tapauksen mallipohjan luominen

5.6 FIR ominaisuudet

Uuden tapauksen (new event) lisäys

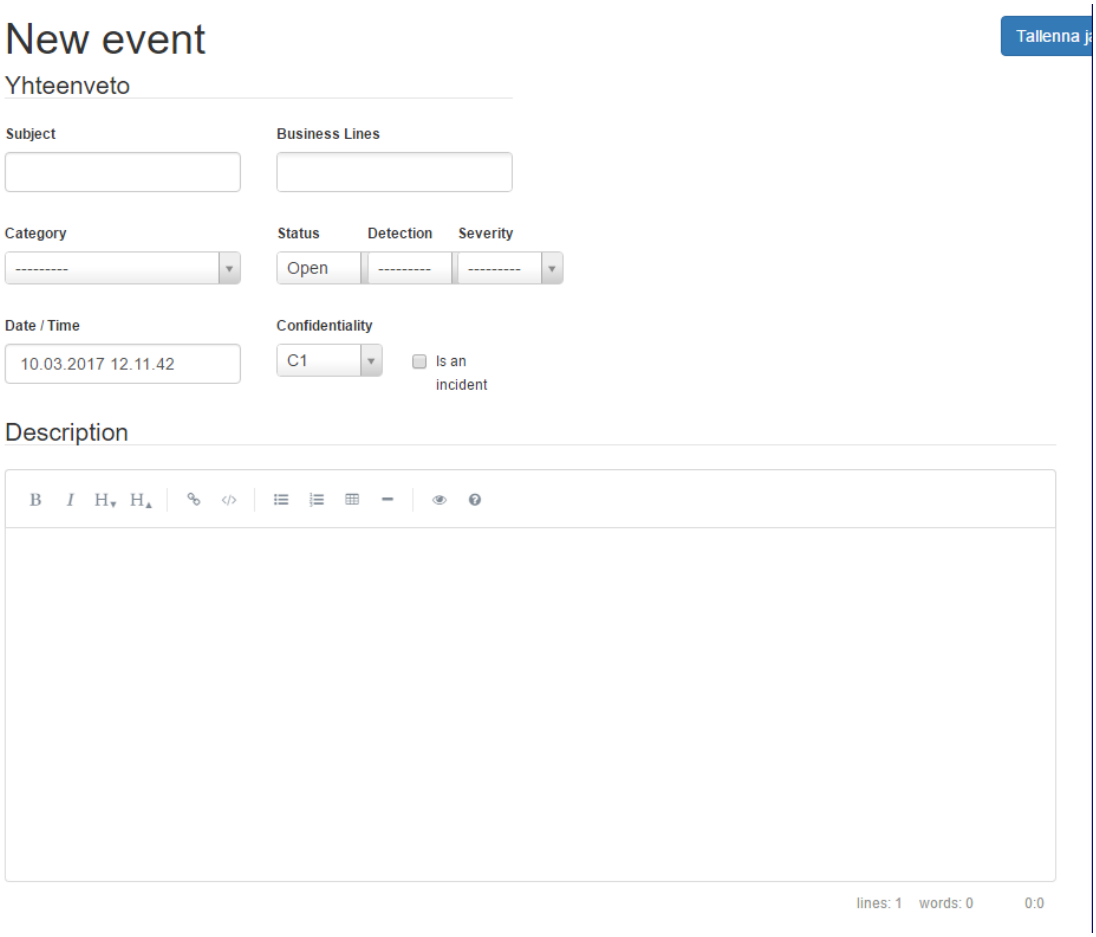
FIR:ssä voi luoda uusi tapauksia, joille voi antaa seuraavat tiedot: aihe, business line, kategoria, tila, kuka teki havainnon, vakavuus, päivämäärä ja aika, luottamuksellisuus, onko tapahtuma poikkeama sekä tapauksen kuvaus.

Business line tarkoittaa sitä, mitä yrityksen osaa tämä tapaus koskee. Tapaukselle voi määrittää seuraavia kategorioita:

- Phishing, Scam (web), Malware, Dataleak, Cybersquatting
- Stolen data, Scam msg, Unavailability, IS integrity, Fraud
- Social Eng., Consulting, Threatintel, Insider
- Blackmail, Dos, Scam (tel), Scam (social), Security Assess

Tapauksen tilaksi voi määrittää avoin, suljettu ja estetty (open, closed, blocked). Tapauksen vakavuuden voi määrittää numeroilla (1,2,3,4). Päivämääräksi ja ajaksi tulee oletuksena kellonaika, jolloin havainto merkittiin FIR-järjestelmään. Päivämäärän ja ajan voi määrittää myös manuaalisesti. Valitettavasti molempia kellonaikoja ei voi

olla samaan aikaan, jos haluaisi esimerkiksi määrittää merkintäajankohdan lisäksi alkuperäisen havainnon ajankohdan. Kuvauskenttään voi lisätä taulukoita, listoja, linkin ja laatikon koodille (jos havaitaan esimerkiksi jotain haitallista koodia). Alla on kuviossa 19 esitetty uuden tapahtuman lisäämisnäkymä.

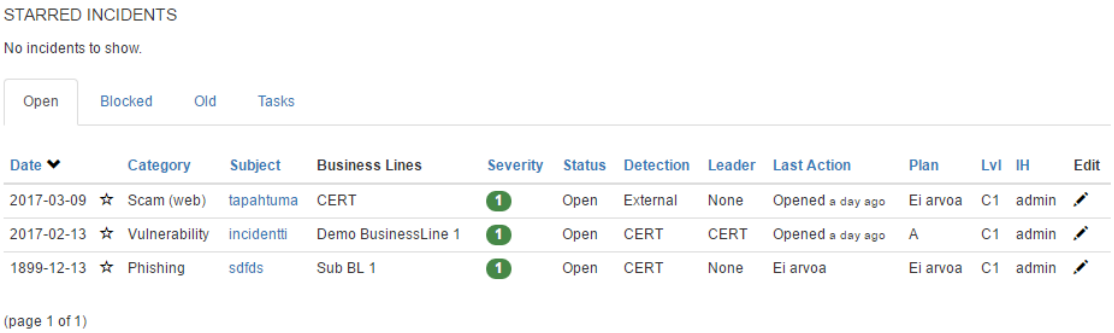


Kuvio 19 FIR uuden tapauksen lisääminen

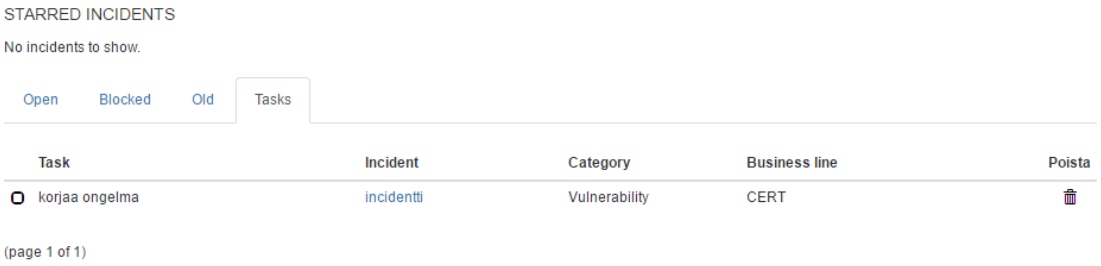
Dashboard-näkymä

FIR:ssä on dashboard-näkymä, jossa voi nähdä listattuna poikkeamia sekä niihin liittyviä tehtäviä. Dashboard näkymässä on neljä välilehteä: Open (näytetään avoimet poikkeamat, Blocked (näytetään estetyt poikkeamat), Old (näytetään vanhat poikkeamat) sekä Tasks, jossa näkyy lista tehtävistä. Tehtävän voi merkitä tehdyksi täällä, kun se on tehty.

Open, Blocked ja Old välilehdillä näkyy poikkeamista seuraavat tiedot: Päivämäärä, kategoria, aihe, vakavuustaso, tila, mikä taho havaitsi poikkeaman, kuka johtaa poikkeaman käsittelyä, viimeisimmän toimenpiteen ajankohta ja kuka teki merkinnän poikkeamasta alun perin. Alla on kuviossa 20 esitetty dashboardin poikkeamalista ja kuviossa 21 tehtävälista.



Kuvio 20 FIR Dashboard lista poikkeamista

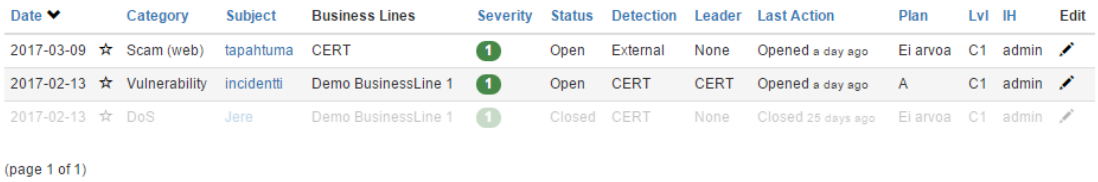


Kuvio 21 FIR lista tehtävistä

Events ja Incidents näkymät

FIRissä on Events-näkymä, jossa näkyy tapahtumat, joita ei ole merkitty poikkeamaksi ja Incidents-näkymä tapahtumista, jotka on merkitty poikkeamiksi. Incidents näkymässä on vastaavanlainen lista poikkeamista kuin dashboard-näkymässä ja samat tiedot. Listan voi laittaa järjestykseen eri sarakkeiden perusteella. Events-näkymässä on tapahtumat myös vastaavanlaisessa listassa, mutta siinä näkyy vähemmän tietoja. Alla on kuviossa 22 esitetty events-näkymä ja kuviossa 23 incidents-näkymä.

Kuvio 22 FIR events-näkymä



Kuvio 23 FIR incidents-näkymä

Hakutoiminto

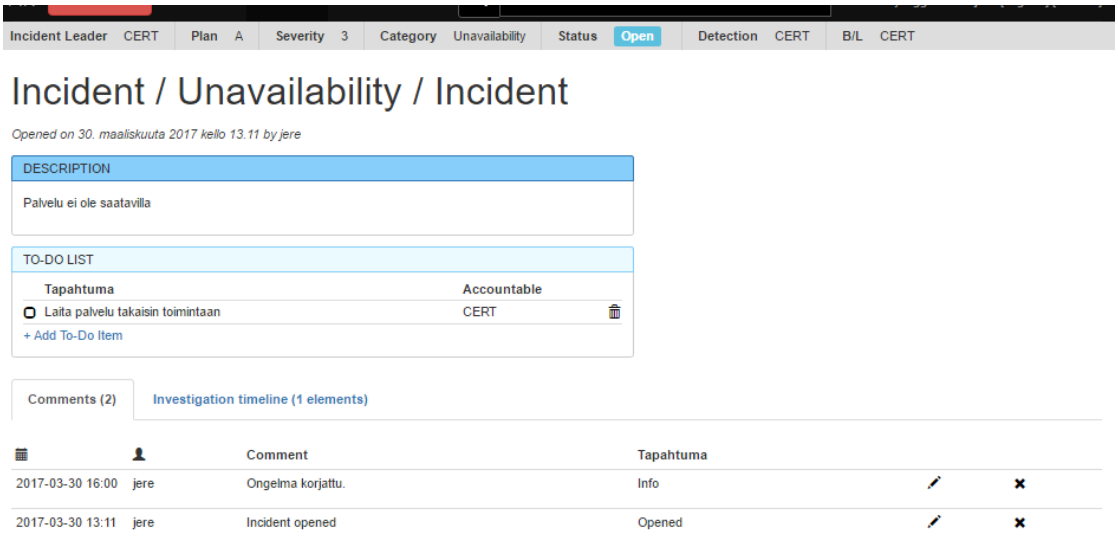
FIRissä on hakutoiminto poikkeamien hakemiseen. Oletuksena sillä voi hakea aiheen, kommentin ja kuvauksen perusteella. Jos haluaa hakea muiden tietojen perusteella, tulee käyttää avainsanoja. Esimerkiksi kategorian perusteella voi hakea seuraavasti category:hakusana (esimerkiksi category:Phishing) ja tilan perusteella status:hakusana (esimerkiksi suljetut poikkeamat status:C). Alla on kuviossa 24 esitetty hakutoiminto FIR:n käyttöliittymän yläreunassa.

Kuvio 24 FIR hakutoiminto

Poikkeaman tarkempi näkymä

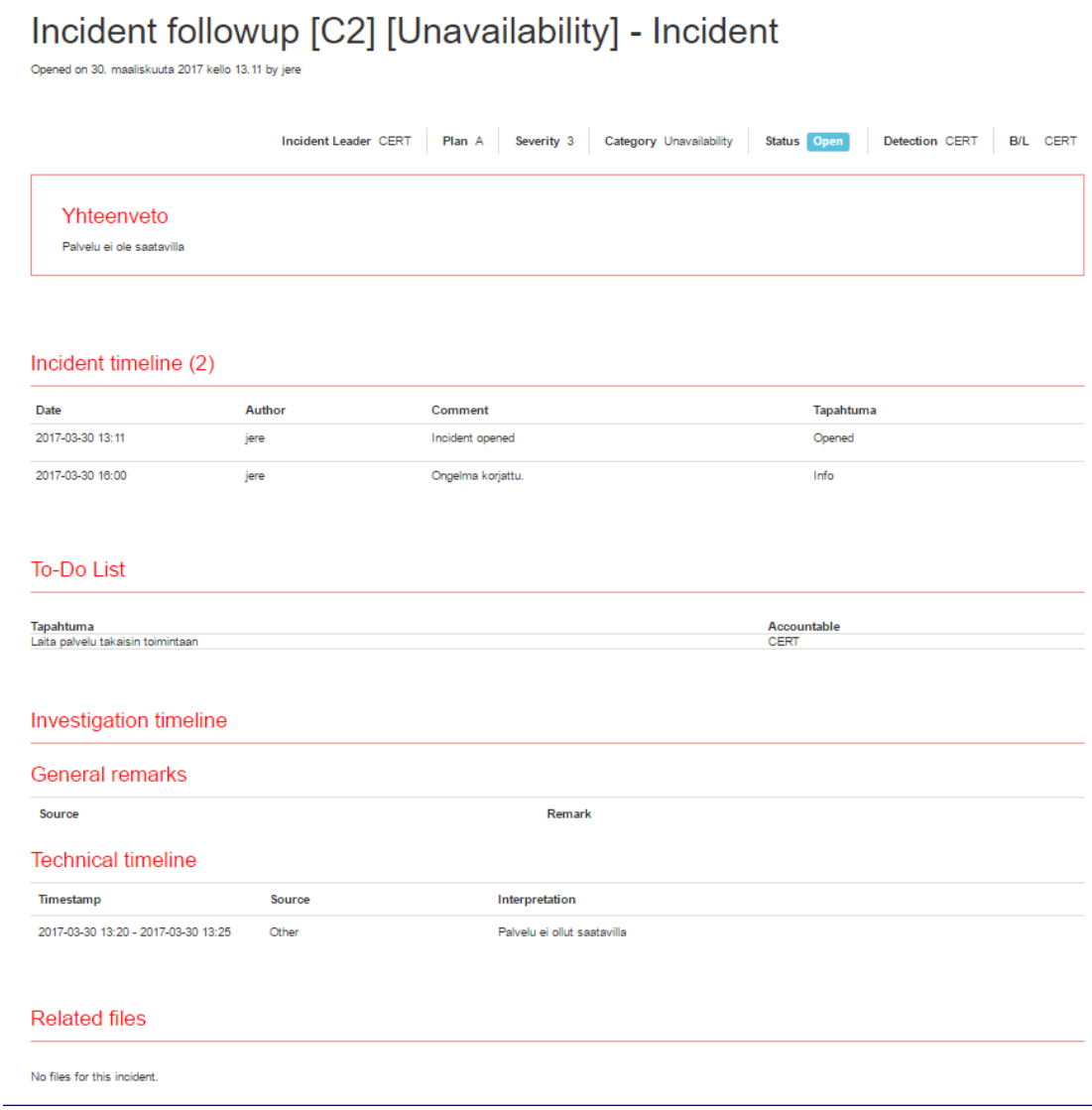
FIR:ssä on mahdollista avata tietty poikkeama tarkempaan näkymään. Tässä näkymässä voi nähdä poikkeamaan liittyvät kommentit ja nähdä poikkeamaan tutkimiseen liittyvät merkinnät (nugget) listassa (investigation timeline). Tällä sivulla voi alareunassa olevilla painikkeilla käyttää lisää-toimintoa poikkeamaan liittyvien asioiden lisäämiseen, lisätä kommentin, muokata alkuperäistä merkintää, sulkea (close) ja es-

tää (block) tapauksen, avata incident followup -sivun ja lähettää hälytyksen sähköpostirajapinnan avulla. Lisää-toiminnolla voi lisätä tiedoston, tehtävän tehtävälistaan (todo) ja poikkeamaan liittyvän havainnon/tiedon (nugget). Alla kuviossa 25 on esitetty poikkeaman tarkempi näkymä.



Kuvio 25 Poikkeaman tarkempi näkymä FIR

Poikkeaman tarkemmassa näkymässä on myös “Incident followup” -sivu, jolla näkyy yhteenvetoa kaikesta poikkeamaan liittyvästä. Siellä näkyviä oleellisia tietoja ovat yhteenveto (tapauksen kuvaus), Incident timeline, To-Do List sekä Technical timeline. Kohdassa Incident timeline näkyy tapaukseen liittyvät kommentit. Poikkeamaan liittyvät tehtävät näkyvät To-Do List -kohdassa. Technical timeline kohdassa näkyvät tapaukseen liitetyt havainnot (nugget). Alla kuviossa 26 on esitetty Incident followup -sivu.



Incident followup [C2] [Unavailability] - Incident

Opened on 30. maaliskuuta 2017 kello 13.11 by jere

Incident Leader CERT | Plan A | Severity 3 | Category Unavailability | Status Open | Detection CERT | B/L CERT

Yhteenveto

Palvelu ei ole saatavilla

Incident timeline (2)

Date	Author	Comment	Tapahtuma
2017-03-30 13:11	jere	Incident opened	Opened
2017-03-30 16:00	jere	Ongelma korjattu.	Info

To-Do List

Tapahtuma	Accountable
Laita palvelu takaisin toimintaan	CERT

Investigation timeline

General remarks

Source	Remark

Technical timeline

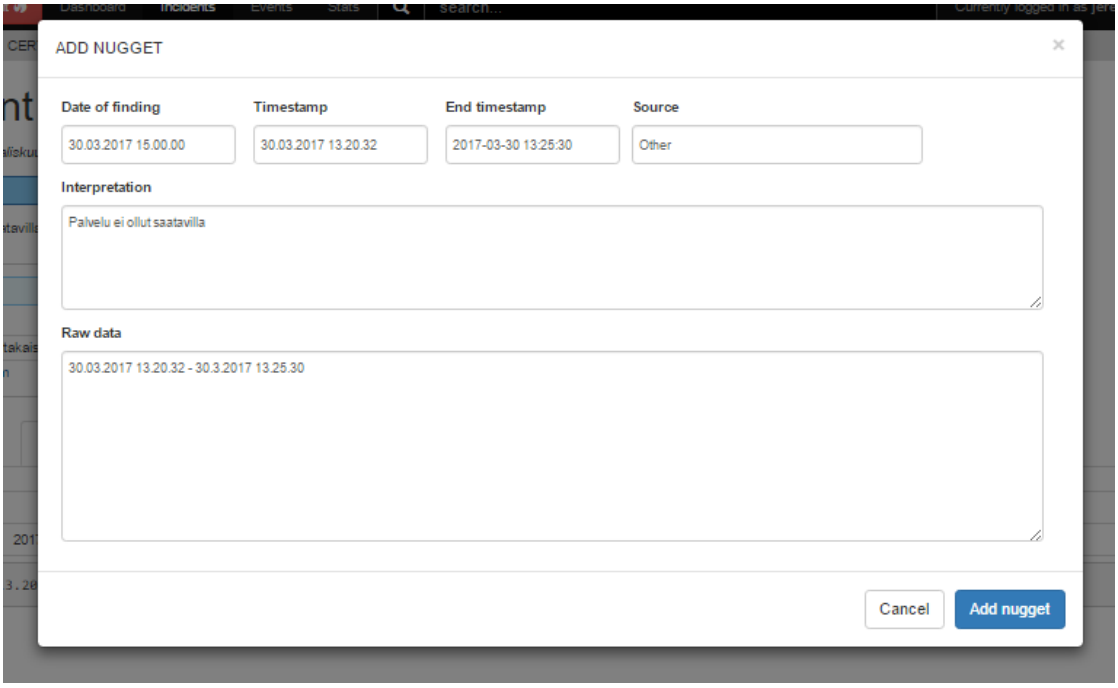
Timestamp	Source	Interpretation
2017-03-30 13:20 - 2017-03-30 13:25	Other	Palvelu ei ollut saatavilla

Related files

No files for this incident.

Kuvio 26 FIR Incident followup sivu

Poikkeaman tarkemman näkymän lisää -toiminnon kautta voi lisätä poikkeamaan liittyvän havainnon (nugget). Havaintoon voi antaa seuraavia tietoja: Löytymisajankohdan aikaleima, alkuperäinen aikaleima/alkamisajankohdan aikaleima, päättymisajankohdan aikaleima, havainnon selitys ja havainnon raakadata. Havainnonlisäämistoiminto on esitetty alla kuviossa 27.



Kuvio 27 Havainnon lisääminen poikkeamaan FIR

5.7 Ominaisuuksien vertailu

5.7.1 Käyttöliittymä yleisesti

TheHive -käyttöliittymä vaikuttaa viimeistellymmältä ja havainnollisemmalta, koska siinä esimerkiksi käytetään värejä havainnollistamiseen, esimerkiksi tapausten vakavuuden ja luottamuksellisuuden määrityksessä. TheHive ylipäätään vaikuttaa selkeämmän ja hienomman näköiseltä, niin sitä on mukavampi käyttää. TheHive:n käyttöliittymässä on hyvää myös se, että siinä näkyy kaikilla sivuilla sivun alareunassa reaaliajassa järjestelmän tapahtumat (esimerkiksi tapauksien, havaintojen ym. lisääminen).

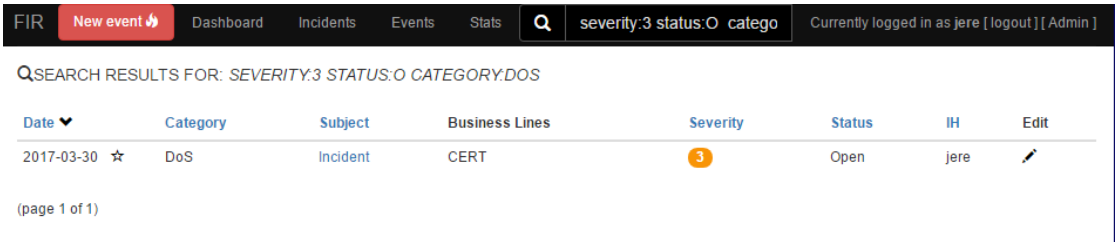
FIR käyttöliittymä ei ole niin viimeistellyn näköinen, eikä niin havainnollinen ja vaikuttaa vähän ankealta. Lähes kaikki on esitetty järjestelmässä tekstillä ja numeroilla, eikä käytetä värejä havainnollistamiseen ainakaan niin paljon kuin TheHive -käyttöliittymässä.

TheHive vaikuttaa käyttöliittymän perusteella viimeistellymmälle järjestelmälle. Molemmissa järjestelmissä on kuitenkin sellaisia hyviä ominaisuuksia joita ei toisessa järjestelmässä ole. Seuraavaksi vertaillaan tarkemmin järjestelmien tärkeimpiä ominaisuuksia.

5.7.2 Haku ja suodatustoiminnot

FIR-järjestelmässä on pelkkä hakutoiminto, eikä erillistä suodatustoimintoa. Hakukenttään voi kuitenkin laittaa avainsanoja, joiden avulla voi suodattaa hakutuloksia. FIR-hakutoiminto hakee vain poikkeamia/tapahtumia eikä havaintoja.

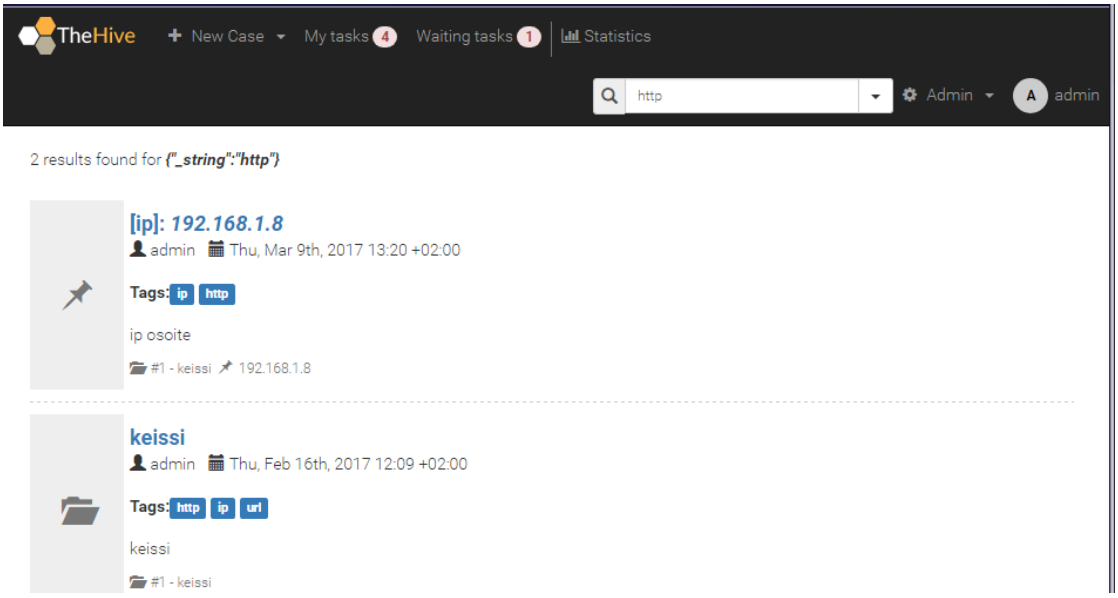
Alla on esimerkki FIR-hakutoiminnon käytöstä:
Esimerkiksi, jos, haluaa hakea poikkeamat, joiden vakavuustaso on 3, tila on avoin ja kategoria on DoS, se tehdään kirjoittamalla seuraavaa hakukenttään:
severity:3 status:O category:DoS
Kuviossa 28 on esitetty haun tulos.



Kuvio 28 FIR-hakutoiminto esimerkki

TheHive-järjestelmässä on hakutoiminto, jolla voi hakea tapauksia, havaintoja ja tehtävälistan tehtäviä. Lisäksi TheHive sisältää erillisen suodatustoiminnon. Erillisen suodatustoiminnon avulla on selkeämpi suodattaa hakutuloksia. Näiden perusteella TheHive:n haku ja suodatustoiminnot ovat paremmat ja selkeämmät.

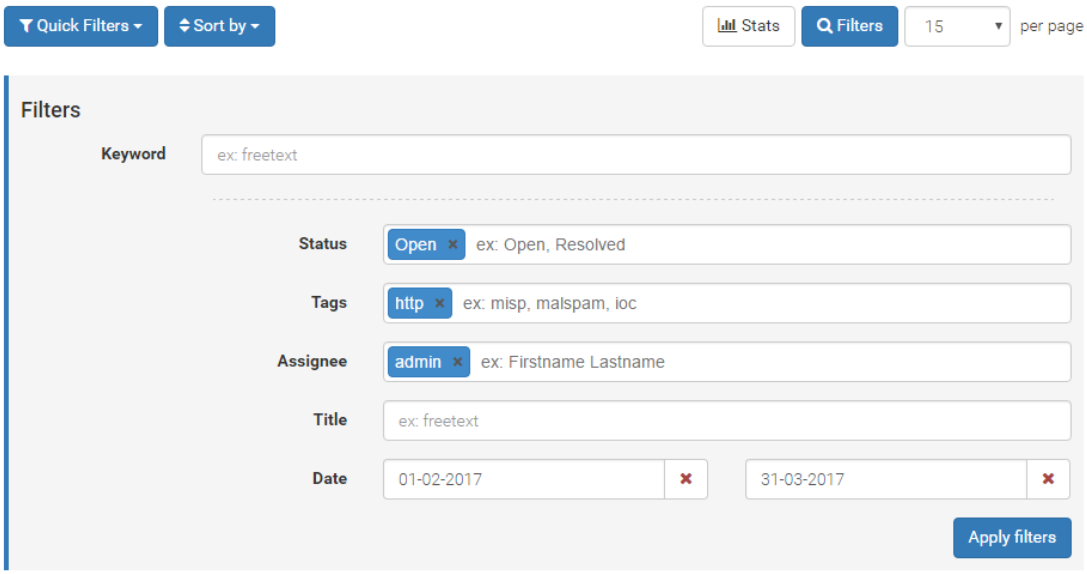
Alla esimerkki TheHiven hakutoiminnon käytöstä:
Kun TheHive:n hakukenttään kirjoittaa hakusanaksi tagin ”http”, löytää hakutoiminto sekä tapauksen, että havainnon. Haun tulos on esitetty kuviossa 29.



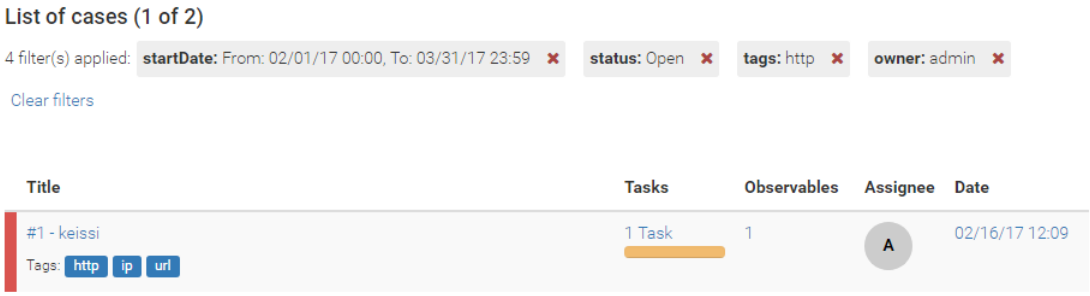
Kuvio 29 TheHive hakutoiminnon esimerkki

Alla on esimerkki TheHive-suodatustoiminnon käytöstä:

Suodatetaan tapauksista avoimet tapaukset, joissa on käytetty tagia ”http” ja jotka on merkinnyt käyttäjä admin ja jotka on merkitty aikavälillä 1.2.2017 – 31.3.2017. Kuviossa 30 on esitetty TheHive:n suodatustoiminto. Kuviossa 31 on esitetty suodatuksen tulokset.



Kuvio 30 TheHive suodatustoiminto esimerkki



Kuvio 31 TheHive suodatustoiminto hakutulos

5.7.3 Tapauksien kirjaaminen

Kummassakin järjestelmässä voi uutta tapausta kirjattaessa merkitä:

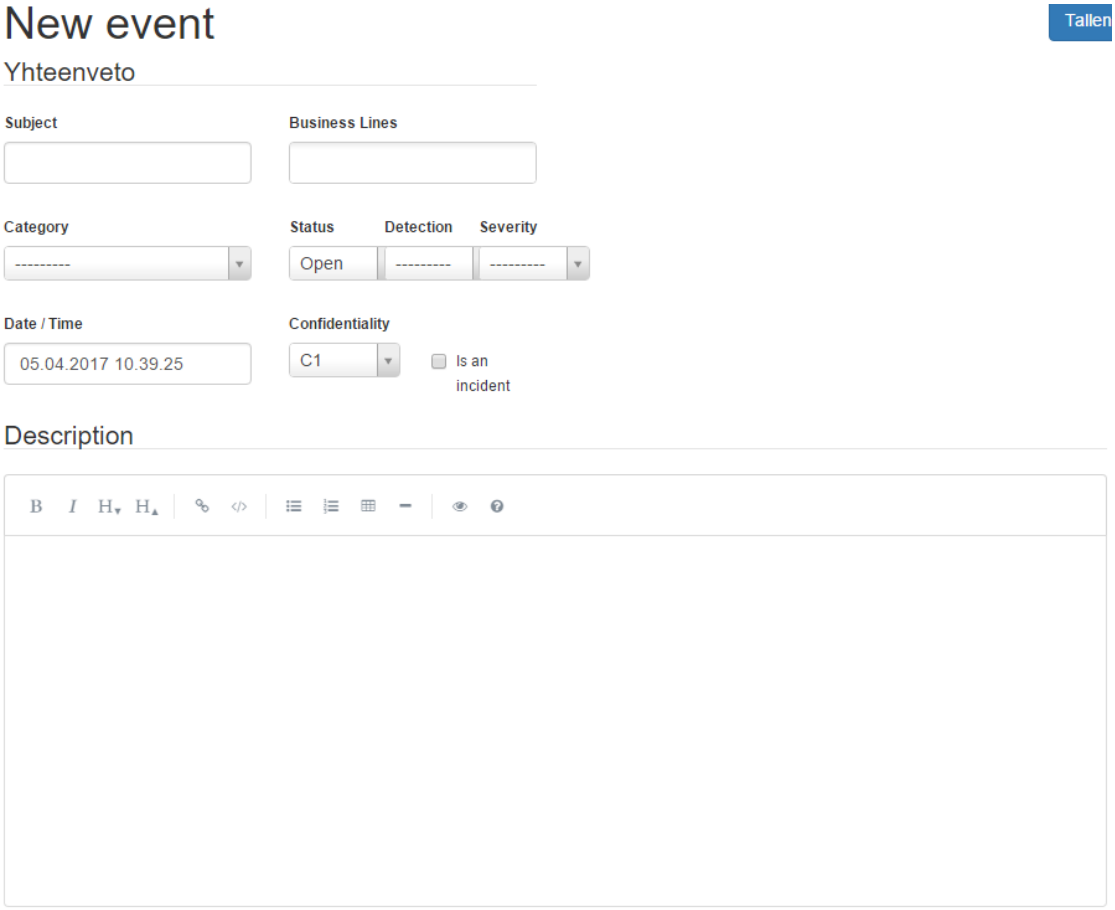
- Otsikko/aihe
- aika (oletuksena kirjaamisaika)
- tapauksen kuvaus
- Vakavuus
    - FIR-järjestelmässä esitetty numeroilla (1,2,3,4)
    - TheHive-järjestelmässä esitetty kirjaimilla ja väreillä (low: L sininen, medium: M keltainen, high: H punainen)
- Luottamuksellisuus
    - FIR:ssä merkitään (C0, C1, C2, C3)
    - TheHive käyttää TLP (Traffic light protocol) väreillä (valkoinen, vihreä, keltainen, punainen)

FIR-järjestelmässä tapauksen kirjaaminen sisältää seuraavat ominaisuudet, joita ei ole TheHive:ssa:

- Business Lines, voi merkitä mitä yrityksen osaa tapaus koskee (admin-sivun kautta näitä voi lisätä ja muokata)
- Detection, voi merkitä mikä taho tunnisti poikkeaman (myös näitä voi lisätä admin-sivun kautta)
- Is an incident, tämän avulla voi määrittää näkyykö tapaus tapahtumissa vai poikkeamissa järjestelmässä

- Kategoria, eli tapauksen tyyppi (esimerkiksi phishing, unavailability, DoS) (näitä voidaan luoda lisää admin sivun kautta)
- Status, voi merkitä jo tapauksen kirjaamisvaiheessa onko poikkeama avoin vai suljettu

Kuviossa 32 on esitetty FIR tapauksen kirjaamissivu.

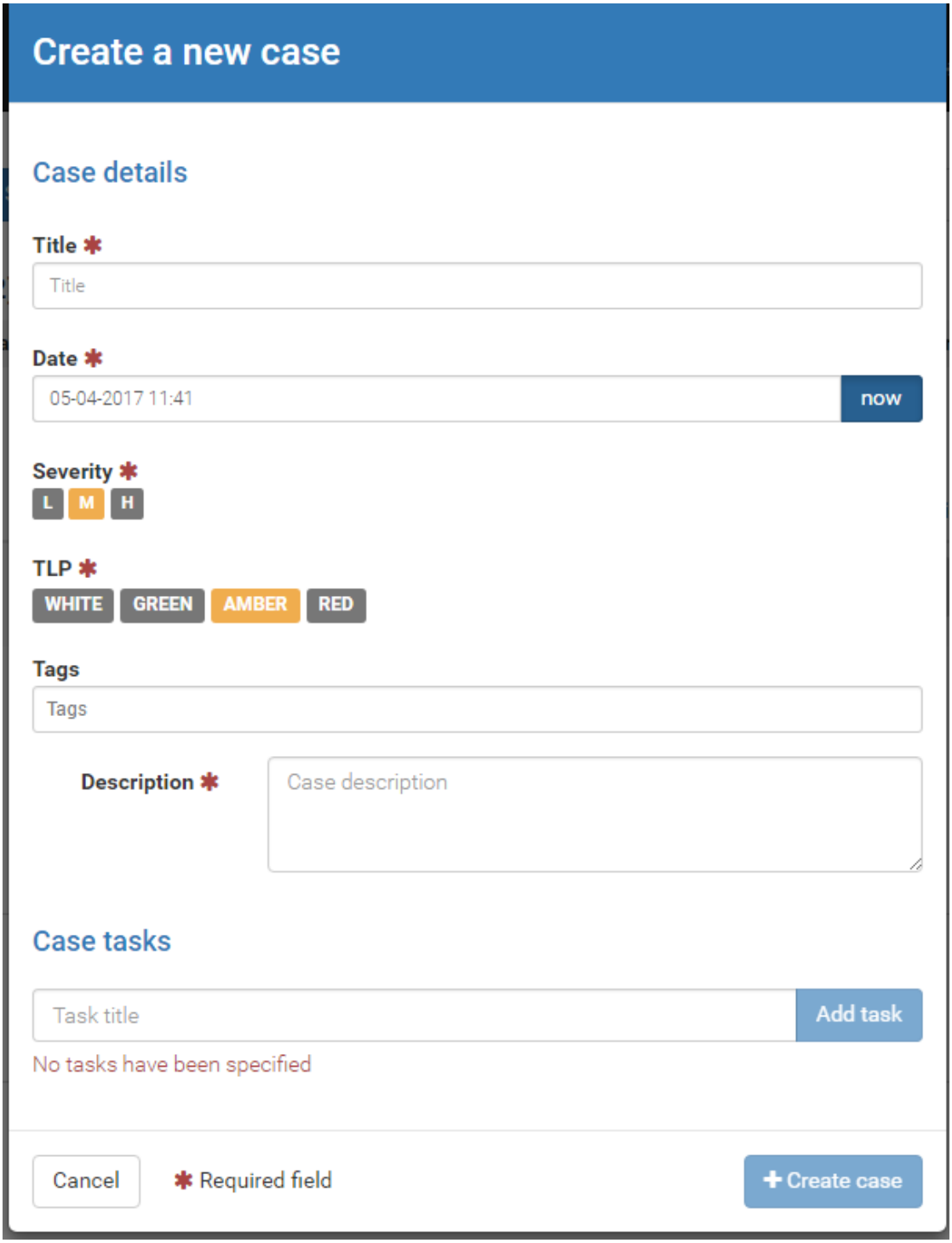


Kuvio 32 FIR tapauksen kirjaaminen

TheHive-järjestelmässä tapauksen kirjaaminen sisältää seuraavat ominaisuudet, joita ei ole FIR:ssä:

- Tagit, voi merkitä erilaisia avainsanoja, joiden avulla tapauksia on helpompi hakea järjestelmästä
- Voidaan lisätä tehtäviä (task) jo tapausta luodessa

Kuviossa 33 on esitetty TheHive:n tapauksen kirjaamissivu.



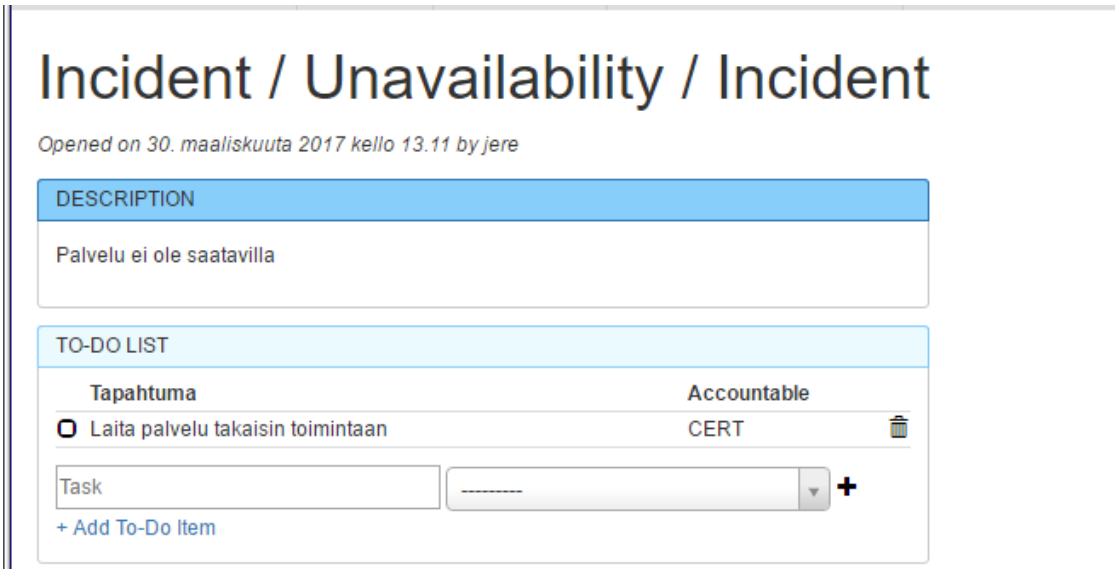
Kuvio 33 TheHive tapauksen kirjaaminen

TheHive:ssa on hyvää, että voi lisätä tageja, vakavuus ja luottamuksellisuus merkitään väreillä (on havainnollisempaa) sekä tehtävien lisääminen tapausta luodessa. TheHiven tapauksen kirjaamisessa ei mielestäni ole merkittäviä puutteita. Joitakin asioita, joita voi merkitä FIR:ssä omaan kenttäänsä, voi TheHive:ssa hyvin merkitä tageihin.

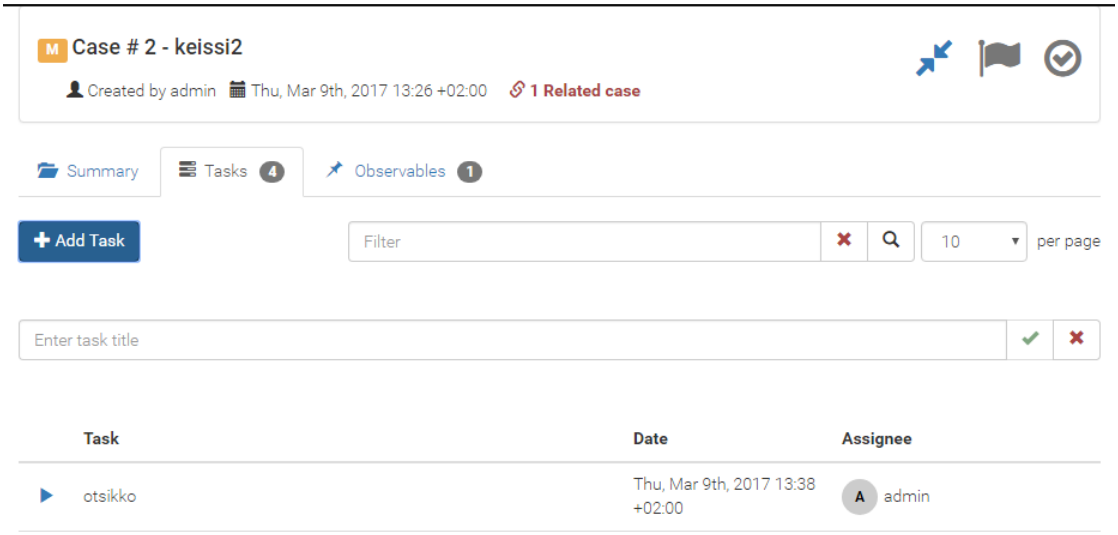
FIR-järjestelmässä on hyvää, että voi merkitä mikä taho havaitsi poikkeaman, voi merkitä mihin yrityksen osastoon se vaikuttaa, voi merkitä tapauksen tilan sitä luotaessa (open, closed) sekä ”is incident” -merkintä jonka avulla voi rajata tapaukset poikkeamiin ja tapahtumiin. FIR:n puutteena on, että siinä ei voi lisätä tageja. Tämä olisi aika oleellinen ominaisuus. TheHive on parempi tapauksien kirjaamisessa, koska siinä voi merkitä tageja.

5.7.4 Tehtävien lisääminen

Molemmissa järjestelmissä onnistuu tehtävien lisääminen, kun avaa tietyn poikkeaman tarkasteltavaksi. Molemmissa on tekstikenttä, johon kirjoitetaan tehtävä. FIR:ssä voi lisäksi merkitä mitä yrityksen osastoa (business line) tehtävä koskee. Kuviossa 34 on esitetty tehtävän lisääminen FIR:ssä ja kuviossa 35 on esitetty tehtävän lisääminen TheHive:ssa.



Kuvio 34 Tehtävän lisääminen FIR



Kuvio 35 Tehtävän lisääminen TheHive

Tehtävän lisäämisominaisuuksissa ei siis ole merkittäviä eroja. Molemmat toimivat ihan hyvin.

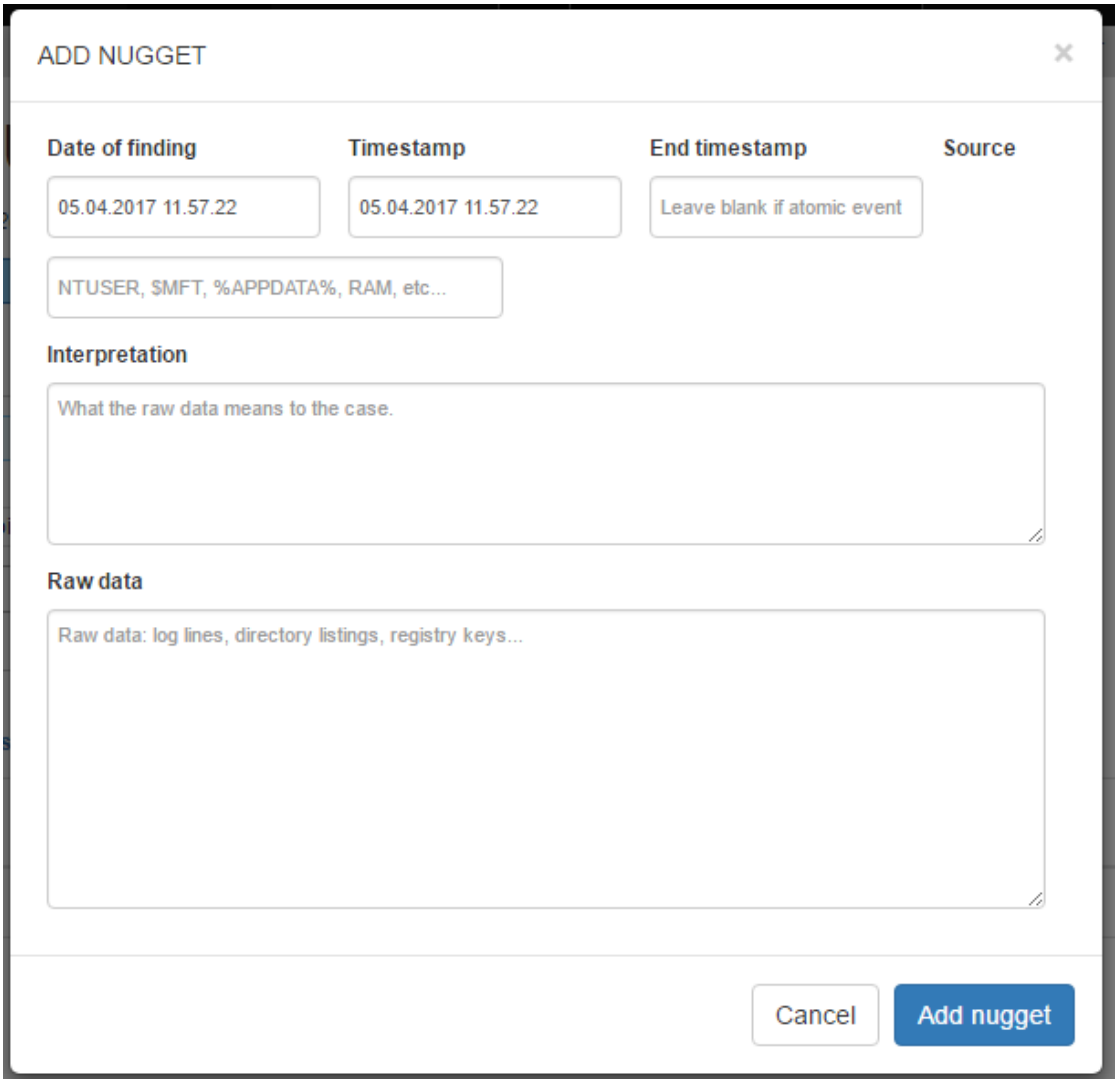
5.7.5 Havaintojen lisääminen

Molemmissa järjestelmissä on mahdollisuus lisähavaintojen merkitsemiseen tapaukseen. FIR:ssä niitä kutsutaan nimellä nugget ja TheHive:ssa nimellä observable.

FIR:ssä havaintoon voi merkitä:

- Kolme päivämäärää/aikaa
    - havainnon tekemisajankohta
    - alkuperäisen tapahtuman ajankohta tai alkamisajankohta
    - päättymisajankohta
- Havainnon lähde
- havainnon selitys
- havainnon sisältö/raakadata

Alla kuviossa 36 on esitetty havainnon merkitsemissivu FIR:ssä.

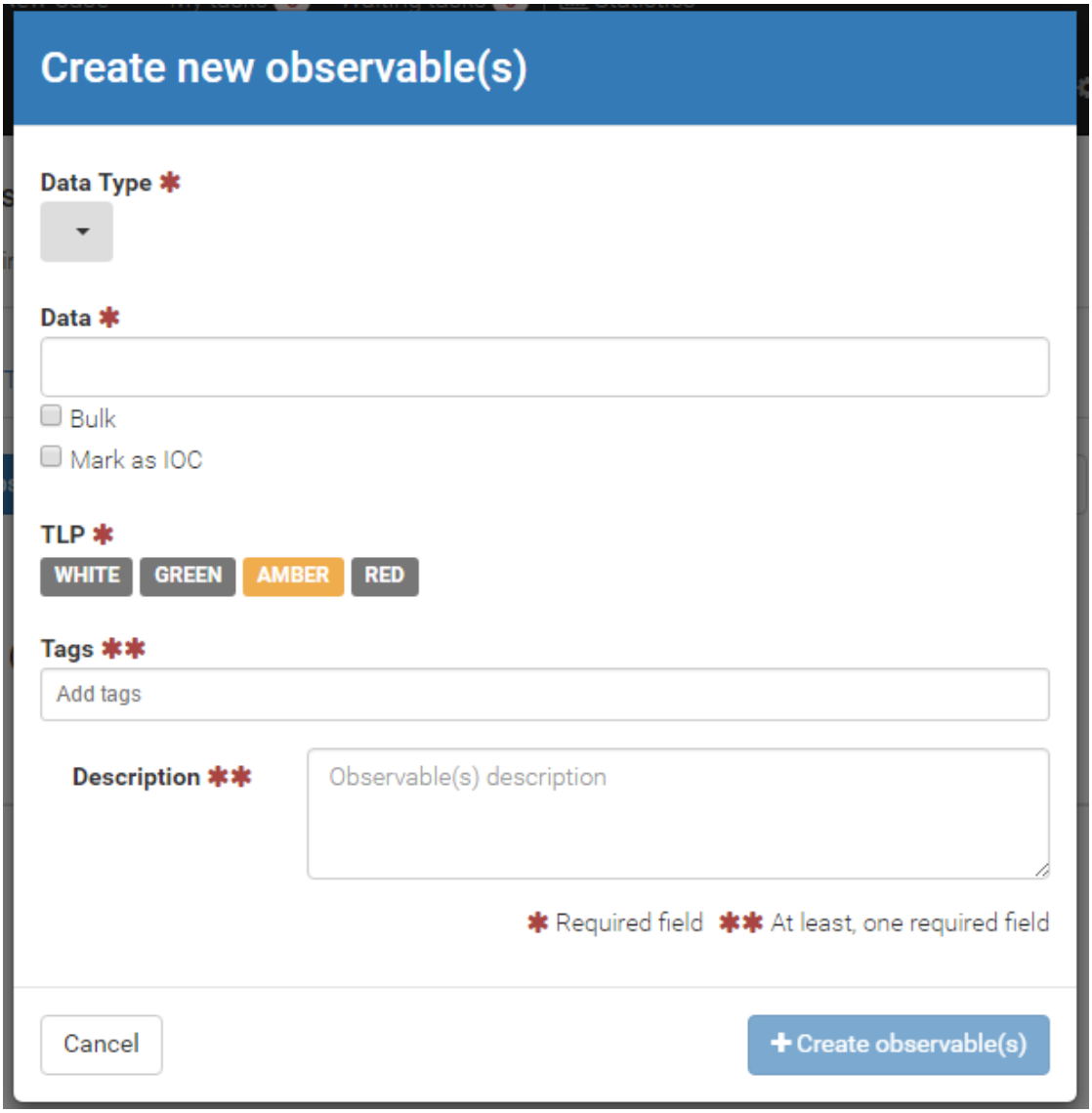


Kuvio 36 FIR havainnon merkitseminen

TheHive:ssa voi merkitä havaintoon seuraavat asiat:

- Havaintodatan tyyppi (esimerkiksi IP, URL, File)
- Varsinainen havainnon data
- Voidaan merkitä IOC:ksi (indicator of compromise, tunnusmerkki poikkeamasta)
- Voidaan käyttää TLP:tä myös tässä luottamuksellisuuden merkitsemiseen
- Voidaan lisätä tageja myös havaintoihin
- Voidaan lisätä havainnon kuvaus

Alla kuviossa 37 on esitetty havainnon merkitseminen TheHive:ssa.



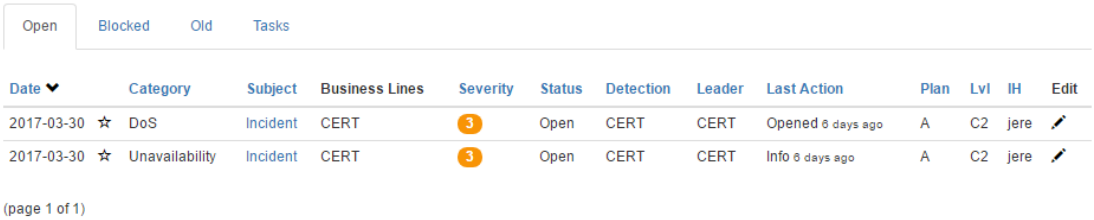
Kuvio 37 Havainnon lisääminen TheHive

Havainnon lisäämisessä FIR:ssä on hyvää, että voi merkitä useita päivämääriä/aikoja. TheHivessa on hyvä, että voi lisätä tageja, merkitä IOC:ksi ja merkitä luottamuksellisuuden TLP:llä. Niin useat päivämäärämerkinnät kuin tagit ovat oleellisia ominaisuuksia, joten tässä on hankala valita kummassa järjestelmässä havainnon merkitsemisominaisuudet ovat paremmat. Tässä pitää valita tilanteen ja tarpeiden mukaan parempi järjestelmä käytettäväksi.

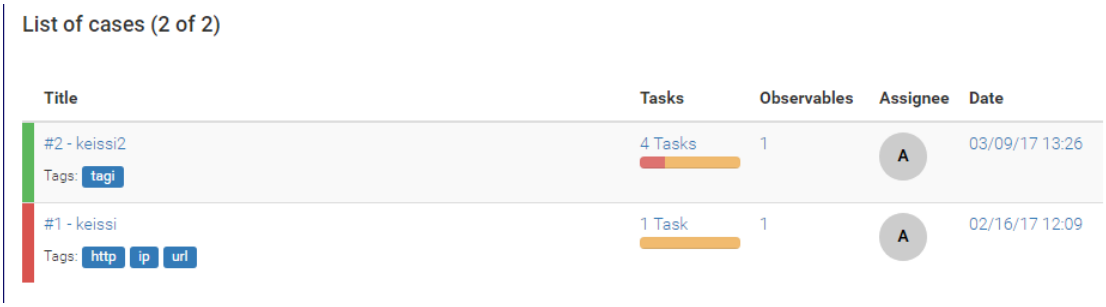
5.7.6 Listat kaikista tapauksista etusivuilla

Molemmissa järjestelmissä on tapauslistat etusivulla. Näistä FIR näyttää enemmän tietoa, mutta TheHive näyttää selkeämmältä. FIR:ssä on vain tekstiä sarakkeissa,

mutta TheHive käyttää myös värejä ja kuvioita. Tapauslista on parempi ja selkeämpi TheHive:ssa edellä mainittujen asioiden perusteella. Kuviossa 38 on esitetty etusivun tapauslista FIR:ssä ja kuviossa 39 on esitetty tapauslista TheHive:n etusivulla.



Kuvio 38 FIR lista tapauksista

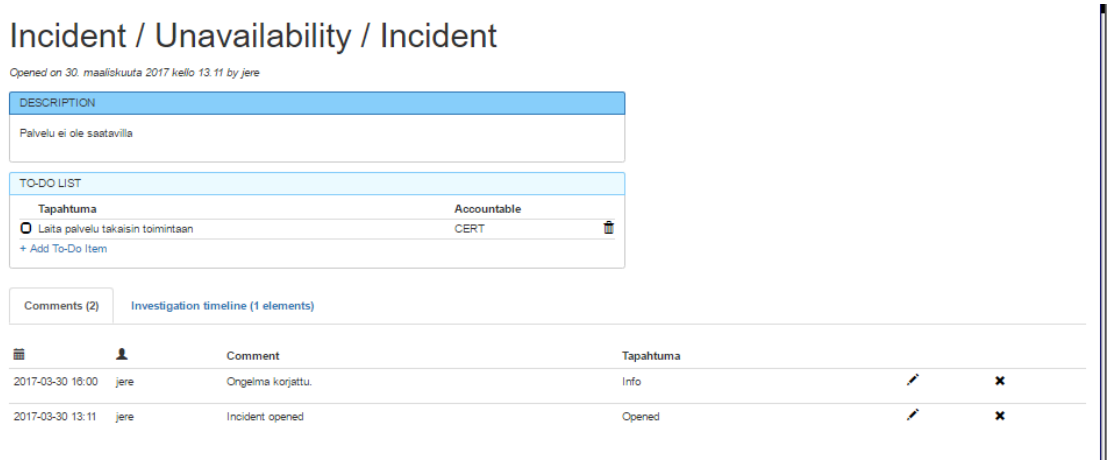


Kuvio 39 TheHive lista tapauksista

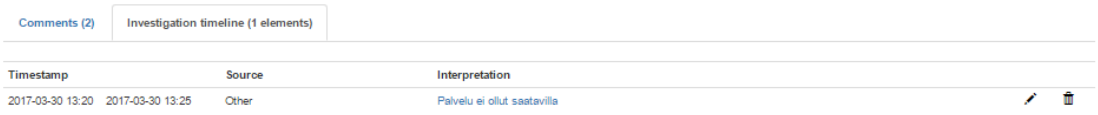
5.7.7 Tietyn tapauksen yhteenvetonäkymä

Molemmissa järjestelmissä voi avata tietyn tapauksen tarkempaan näkymään, jossa näkyy kyseisen tapauksen tiedoista yhteenveto. FIR:ssä on tätä varten kaksi erilaista näkymää: Normaali näkymä, joka avautuu, kun avaa poikkeaman tiedot poikkeamalistan kautta ja ”incident followup” -sivu johon pääsee edellisen sivun kautta.

Normaalissa näkymässä näkyy sivun yläreunassa samat tiedot kuin poikkeamalistassa etusivulla. Alempana sivulla on seuraavia tietoja: tapauksen tyyppi (category), kuvaus, tehtävät, kommentit sekä investigation timeline, jossa näkyy tapaukseen merkityt havainnot (nugget) aikajärjestyksessä. Alla kuvioissa 40 ja 41 on esitetty poikkeaman yhteenvetonäkymä.

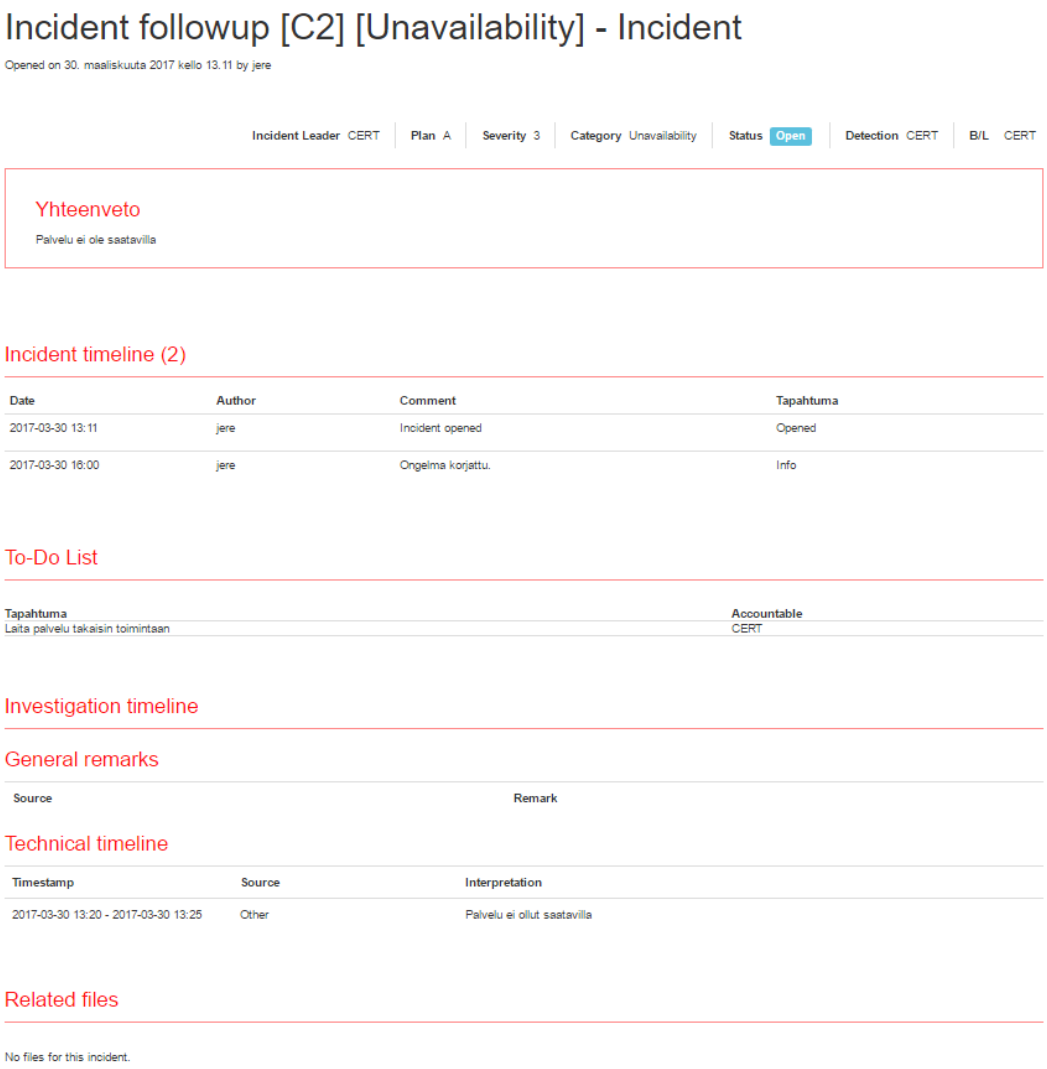


Kuvio 40 Tapauksen yhteenvetonäkymä FIR osa 1



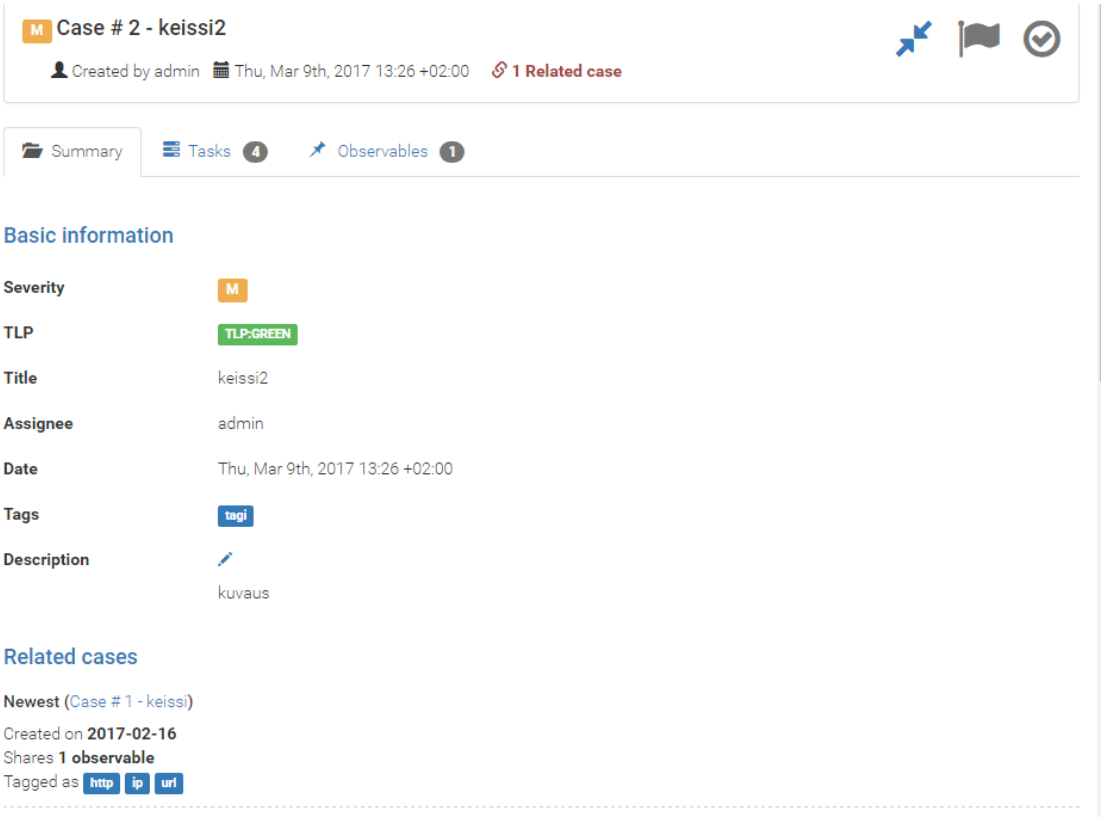
Kuvio 41 Tapauksen yhteenvetonäkymä FIR osa 2

Incident followup -sivulla yhteenveto kaikesta tapaukseen liittyvästä tiedosta. Siellä on tapauksen kuvaus, kommentit, tehtävät ja havainnot ja lista liittyvistä tiedostoista. Tämän sivun kautta saa kokonaiskuvan tapauksesta. Incident followup-sivu vaikuttaisi olevan tulostettavassa muodossa. Tästä voi olla apua raportoinnissa. Incident followup -sivu on esitetty kuviossa 42.



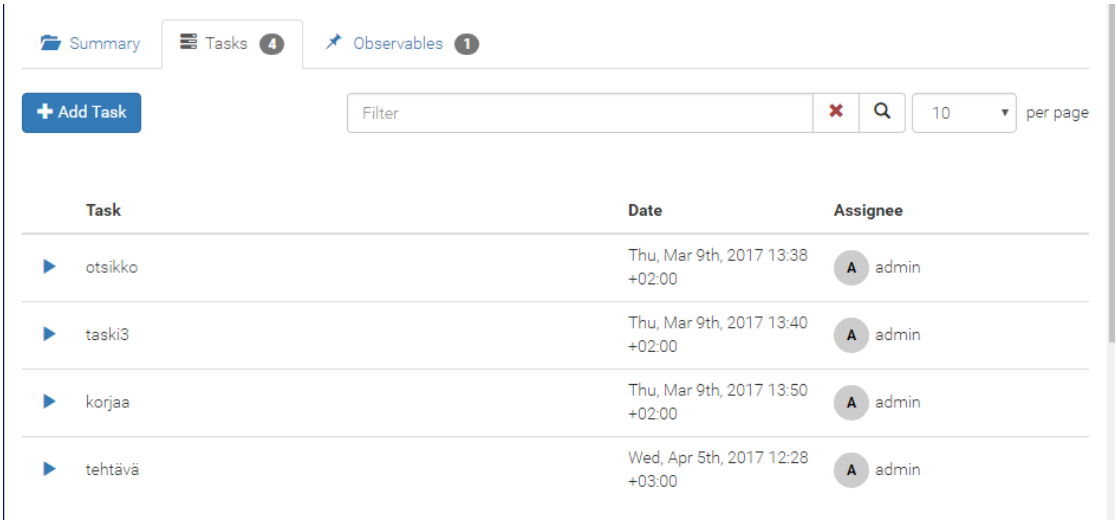
Kuvio 42 FIR Incident followup -sivu

TheHive:ssa on tietyn poikkeaman yhteenvetonäkymässä kolme eri välilehteä: Summary, tasks ja observables. Summary-välilehdellä on yhteenveto tapauksen tiedoista: vakavuus, TLP, otsikko, kuka merkitsi, päivämäärä, tagit, kuvaus ja tapaukset joiden kanssa on jaettuja havaintoja. Kuviossa 43 on esitetty summary-välilehti.



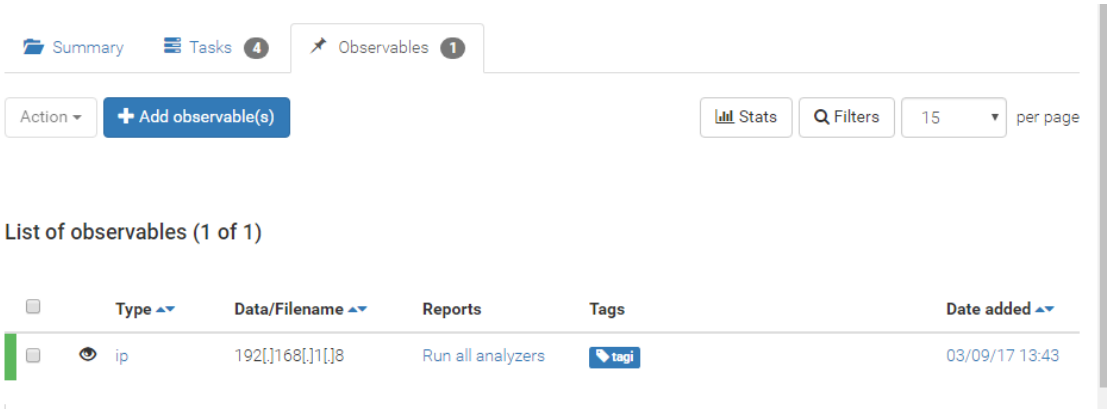
Kuvio 43 Tapauksen yhteenveto: Summary-välilehti TheHive

Tasks-välilehdellä on lista tehtävistä ja siellä voi lisätä tehtäviä. Tehtäville on myös suodatus/hakutoiminto tällä sivulla. Tasks-välilehti on esitetty kuviossa 44.



Kuvio 44 Tapauksen yhteenveto: Tasks-välilehti TheHive

Observables-välilehdellä on lista havainnoista, jossa näkyy tiedot: havainnon tyyppi, tagit, sisältö ja päivämäärä. Tätä kautta voi lisätä tapaukseen havaintoja ja täältä löytyy myös haku- ja suodatustoiminnot havainnoille. Kuviossa 45 on esitetty Observables-välilehti.



Kuvio 45 Tapauksen yhteenveto Observables-välilehti TheHive

TheHivessa tapauksen yhteenvetonäkymässä on hyvää haku- ja suodatustoiminnot ja tietojen selkeämpi ja havainnollisempi esittäminen. FIR:ssä on hyvää Incident followup -sivu, josta saa hyvin kokonaiskuvan tapauksesta ja siitä on hyötyä myös raportoinnissa. Edellä mainittujen perusteella TheHive:ssa on parempi yhteenvetonäkymä tapauksille.

5.7.8 Ominaisuuksien vertailun yhteenveto

TheHive-järjestelmässä käyttöliittymä vaikuttaa selkeämmälle ja viimeistellymmälle. Myös haku- ja suodatustoiminnot ovat monipuolisemmat ja helpommat käyttää. Tapauksien lisäämisessä järjestelmään FIR:ssä on oma kenttänsä useille tiedoille, kun taas TheHivessa voi samoja tietoja merkitä tagimerkinnöillä. Tagimerkinnät ovat kätevämpiä kuin se, että kaikille mahdollisille tiedoille olisi oma kenttänsä. Tagimerkinnöistä näkee yhdellä vilkaisulla paljon tietoa. Tehtävienlisäämistoiminnot ovat molemmissa järjestelmissä yhtä hyvät.

Havaintojen lisäämisessä FIR:ssä on etuna, että voi merkitä useita päivämääriä ja aikoja (esimerkiksi havainnon tekemisajankohta ja alkuperäisen tapahtuman ajankohta

oleellisia joissakin tapauksissa). TheHive:n tapauksien lisäämisessä on hyvää tagimerkinnät, joilla voi selkeästi esittää havaintoihin liittyviä tietoja. Tapauksesta riippuen saattaa olla toinen tai molemmat näistä ominaisuuksista oleellisia. Tapauksesta riippuen siis pitää päättää kumpi järjestelmä on parempi.

Erilaiset näkymät tapauksille ovat selkeämpiä TheHive-järjestelmässä. FIR:ssä on hyvä yhteenvetonäkymä (incident followup) tapaukselle, joka esittää yhdellä sivulla kaikista tapauksen tiedoista yhteenvedon. Tämä yhteenveto on kätevä, jos haluaa tulostaa tietoja tai raportoida tapausta.

Yleisesti ottaen TheHive vaikuttaa paremmalle ja viimeistellymmälle järjestelmälle. Kuitenkin myös FIR-järjestelmässä on hyödyllisiä ominaisuuksia (esimerkiksi incident followup -sivu ja useiden päivämäärien merkitseminen havainnolle), joita ei ole TheHive:ssa.

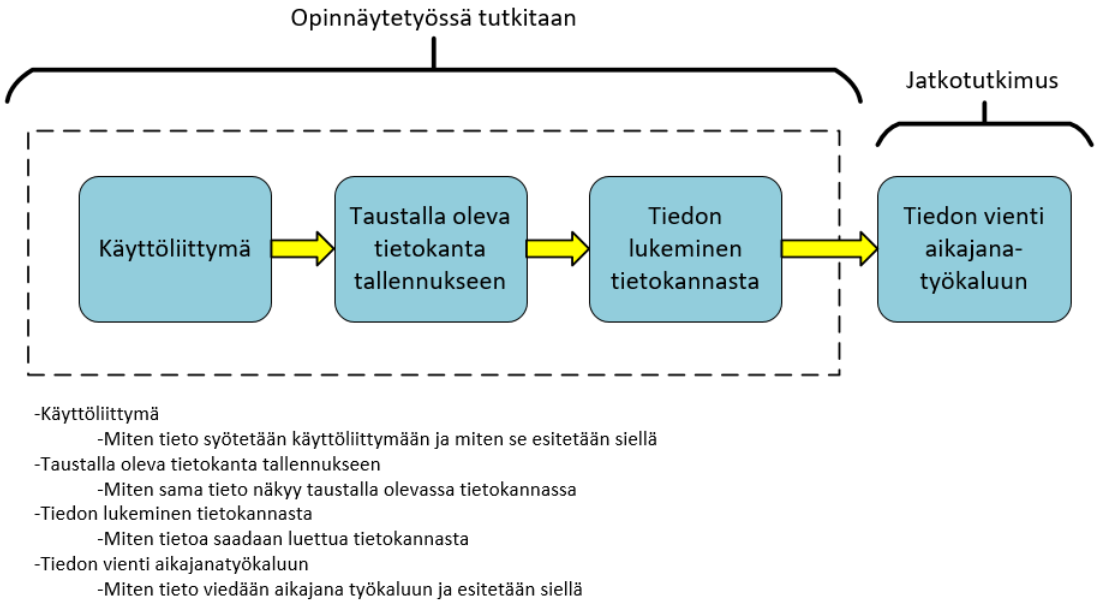
5.8 Järjestelmien rajapinnat tiedon tuontiin ja vientiin

5.8.1 Johdanto

Kummassakaan järjestelmässä ei ole käyttöliittymässä tiedon tuontiin ja vientiin sellaisia toimintoja, joita tässä haettiin. Eli ei saa järkevässä muodossa tietoa ulos, että sen voi viedä toiseen järjestelmään. Tietoa ei myöskään saa tuotua automatisoidusti järjestelmään. Tiedot syötetään järjestelmiin käyttöliittymän kautta manuaalisesti tapaus ja havainto kerrallaan. Tässä keskityn tutkimaan järjestelmien API-rajapintoja sekä järjestelmien taustalla olevia tietokantoja.

Tässä vaiheessa tutkitaan prosessi sille, miten tieto saadaan tutkittavista järjestelmistä toiseen järjestelmään, tässä tapauksessa käytännössä aikajanatyökaluun. Tässä opinnäytetyössä tutkitaan prosessia siihen asti, kun tieto on saatu luettua järjestelmien taustalla olevista tietokannoista sopivaan muotoon. Se miten tieto viedään käytännössä aikajanatyökaluun ja miten se esitetään siellä, menee jatkotutkimukseen.

Alla kuviossa 46 näkyy tutkittava prosessi. Tässä ensin tutkitaan, miten tieto syötetään poikkeamanhallinnan kirjaamistyökaluun ja miten se esitetään sen käyttöliittymässä. Seuraavaksi tutkitaan, miten tämä sama tieto esitetään taustalla olevassa tietokannassa. Sitten tutkitaan, miten tieto saadaan luettua tietokannasta sellaiseen muotoon, että sitä voi hyödyntää. Viimeinen vaihe, joka jää tämän opinnäytetyön ulkopuolelle, on aikajanatyökalun toteuttaminen ja tiedon vienti tähän työkaluun. Aikajanatyökalu jää opinnäytetyön ulkopuolelle, koska valmiita aikajanatyökaluja ei ollut saatavilla poikkeamanhallintaan.



Kuvio 46 Rajapintojen tutkiminen

5.8.2 Tiedon tuonti

Tiedon tuonti onnistuu kumpaankin järjestelmään vain manuaalisesti syöttämällä tapaus ja havainto kerrallaan. Molemmissa järjestelmissä tämän voi tehdä käyttöliittymän kautta. TheHive -järjestelmässä voi tietoa syöttää myös API:n kautta, mutta se ei tuo mitään uusia ominaisuuksia tiedon syöttämiseen. Tietoa voi syöttää edelleen vain tapaus tai havainto kerrallaan. Tiedon tuonti taustalla olevien tietokantojen kautta manuaalisesti ei ole järkevää, koska järjestelmät luovat automaattisesti id- ja indeksi-arvoja, kun tapauksia syötetään käyttöliittymän kautta. Jos tietoja syöttää tietokantaan taustalle manuaalisesti, järjestelmä ei ymmärrä tätä tietoa.

5.8.3 Tiedon vienti

Tiedon vienti onnistuu järjestelmistä API:n tai taustalla olevan tietokannan kautta. TheHive -järjestelmästä testasin tiedon vientiä kutsumalla järjestelmän API:ta curl-komennolla HTTP:n kautta. FIR -järjestelmää varten tehtiin PHP-skripti, joka hakee tietoa taustalla olevasta MySQL -tietokannasta.

5.8.4 Tiedon viennin testaaminen TheHive

Tiedon viennin testaamiseen TheHive -järjestelmästä käytettiin Curl -komentoa. Curl on Linuxissa oleva työkalu tiedon hakemiseen järjestelmistä eri protokollien kautta, joista tässä käytettiin HTTP -protokollaa. Curl -työkalun tuloste on JSON -muodossa. Lisäksi käytettiin json_reformat -työkalua JSON-tulosteen muuttamiseen ihmiselle helpommin luettavaan muotoon.

Tiedon viennissä testattiin seuraavat asiat: kaikkien järjestelmässä olevien tapauksien hakeminen, yksittäisen tapauksen hakeminen järjestelmästä, kun tapauksen ID tiedetään sekä kaikkien havaintojen (observable) hakeminen järjestelmästä.

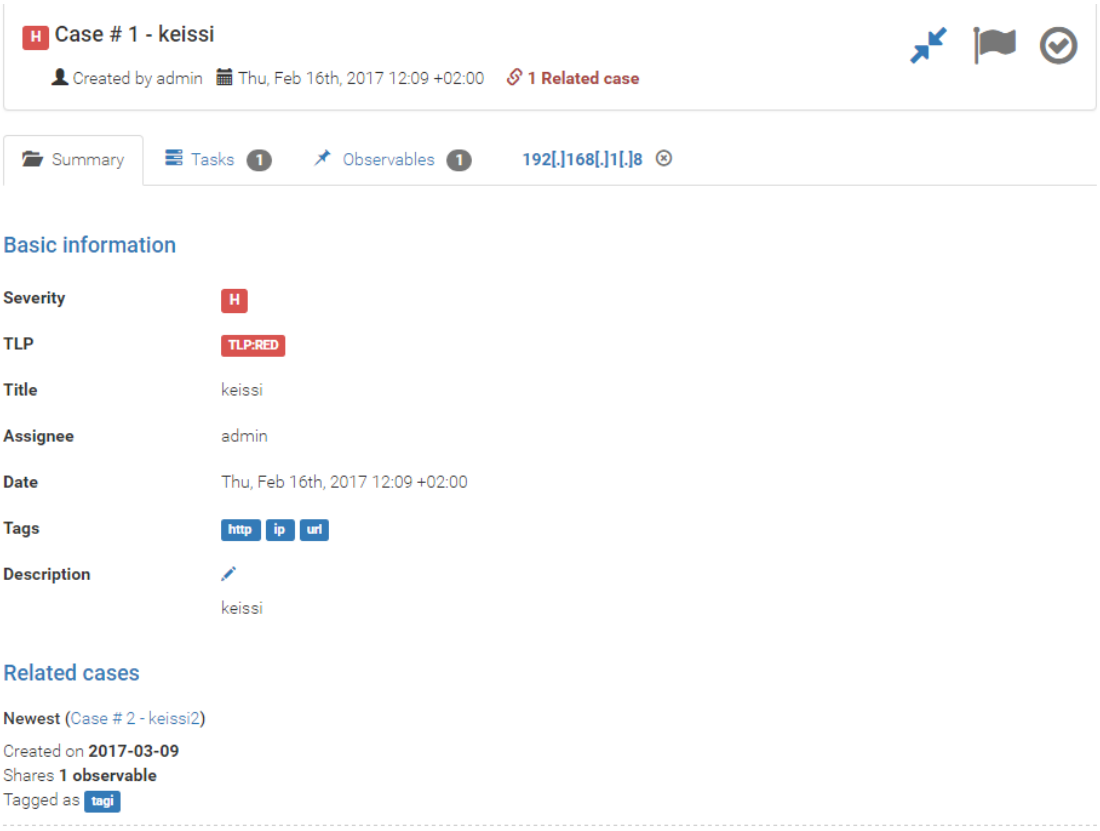
Kaikkien tapauksien hakeminen TheHive -järjestelmästä
Kaikkien tapauksien hakeminen onnistuu seuraavan laisella komennolla:
curl -u käyttäjätunnus:salasana http://IP-OSOITE:9000/api/case | json_reformat
Tässä tapauksessa komento oli seuraavan lainen:
curl -u admin:admin http://192.168.1.8:9000/api/case | json_reformat

Komennossa ensin annetaan käyttäjätunnukset ja sen jälkeen kutsutaan TheHive -järjestelmän API:a oikealla osoitteella ja lopuksi muokataan json_reformat komennolla tuloste helpommin luettavaan muotoon. Alla kuviossa 47 on esitetty komennon tulosteesta yksi tapaus, tulosteessa on kaikki tapaukset peräkkäin.

```
{
    "tlp": 3,
    "severity": 3,
    "createdBy": "admin",
    "tags": [
        "http",
        "ip",
        "url"
    ],
    "caseId": 1,
    "startDate": 1487239740000,
    "owner": "admin",
    "createdAt": 1487239828731,
    "status": "Open",
    "description": "keissi",
    "user": "admin",
    "title": "keissi",
    "flag": false,
    "resolutionStatus": "TruePositive",
    "summary": "1234",
    "impactStatus": "NoImpact",
    "updatedBy": "admin",
    "updatedAt": 1489058283758,
    "id": "AVpGZtWDc7-N1ct_QfF2",
    "type": "case"
}
```


Kuvio 47 Tapauksien tietojen hakeminen TheHive

Alla kuviossa 48 on esitetty, miten sama tapaus näkyy TheHive -käyttöliittymässä.



Kuvio 48 Tapaus TheHive -käyttöliittymässä

Tietyn tapauksen hakeminen TheHive -järjestelmästä

Tapahtuman, jonka ID tiedetään, voi hakea järjestelmästä lisäämällä aiemmin käytettyyn komentoon ID-arvon /api/case/ kohdan jälkeen. Alla on esitetty käytetty komento:

curl -u admin:admin http://192.168.1.8:9000/api/case/AVpGZtWDc7-N1ct_QfFZ | json_reformat

Alla kuviossa 49 on esitetty komennon tuloste. Tuloste on samassa muodossa kuin kaikkia tapauksia haettaessa ja esimerkkinä tässä käytetään samaa tapausta.


```
{
    "tlp": 3,
    "severity": 3,
    "createdBy": "admin",
    "tags": [
        "http",
        "ip",
        "url"
    ],
    "caseId": 1,
    "startDate": 1487239740000,
    "owner": "admin",
    "createdAt": 1487239828731,
    "status": "Open",
    "description": "keissi",
    "user": "admin",
    "title": "keissi",
    "flag": false,
    "resolutionStatus": "TruePositive",
    "summary": "1234",
    "impactStatus": "NoImpact",
    "updatedBy": "admin",
    "updatedAt": 1489058283758,
    "id": "AVpGZtWDc7-N1ct_QfFZ",
    "type": "case"
}
```


Kuvio 49 Tietyn tapauksen hakeminen TheHive

Kaikkien havaintojen hakeminen TheHive -järjestelmästä

Myös kaikkien havaintojen hakemiseen TheHive -järjestelmästä voidaan käyttää curl -komentoa. Nyt komento kuitenkin syötetään eri muodossa. Käytetty komento on seuraavan lainen:

curl --data -u http://käyttäjätunnus:salasana@IP-OSOITE:9000

/api/case/artifact/_search | json_reformat

Tässä tapauksessa komentoa käytettiin seuraavasti:

curl --data -u http://admin:admin@192.168.1.8:9000/api/case/artifact/_search | json_reformat

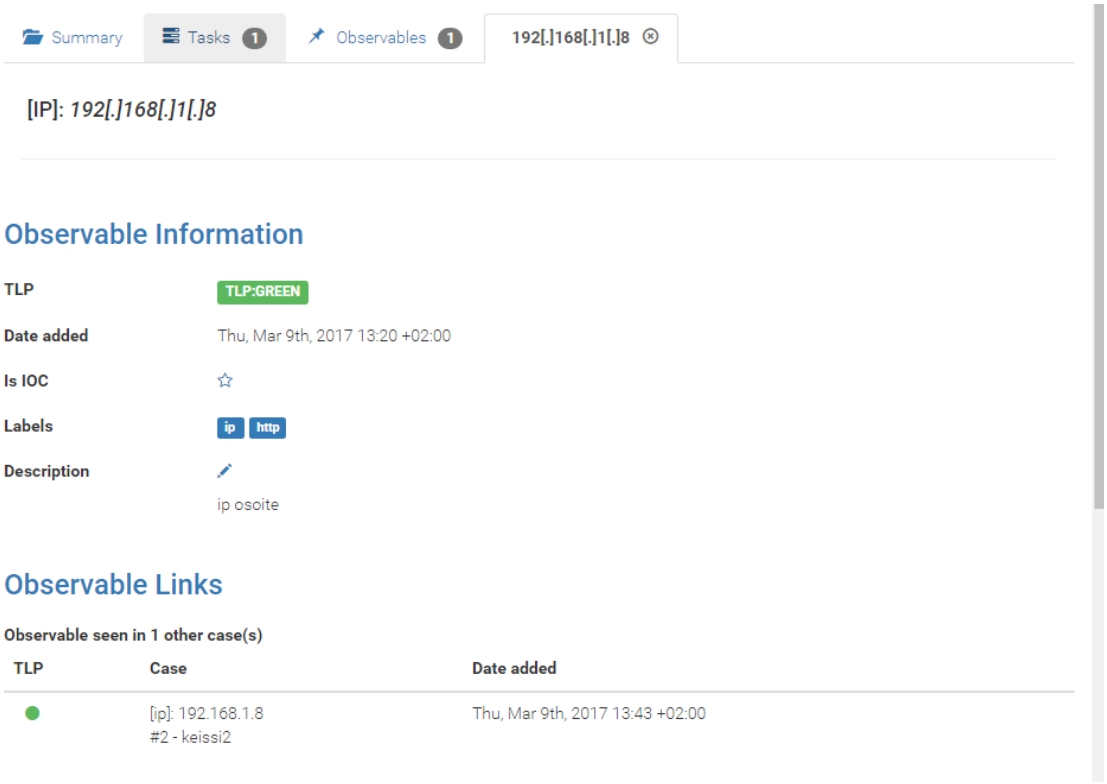
Komennon tulosteessa on peräkkäin kaikkien havaintojen tiedot. Alla kuviossa 50 on esitetty yksi havainto komennon tulosteesta.


```
{
    "ioc": false,
    "createdAt": 1489058422209,
    "createdBy": "admin",
    "tlp": 1,
    "startDate": 1489058422251,
    "dataType": "ip",
    "message": "ip osoite",
    "user": "admin",
    "tags": [
        "ip",
        "http"
    ],
    "reports": {

    },
    "data": "192.168.1.8",
    "status": "Ok",
    "_id": "8619b9f261c7f526e7162d58c38b0b4b",
    "id": "8619b9f261c7f526e7162d58c38b0b4b",
    "type": "case_artifact"
},
```


Kuvio 50 havaintojen tietojen hakeminen TheHive

Alla kuviossa 51 on esitetty sama havainto TheHive käyttöliittymässä.



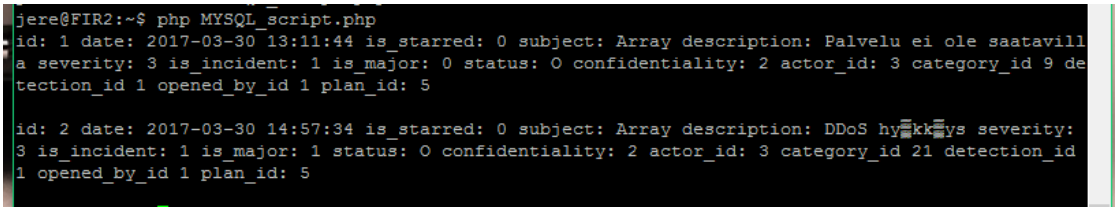
Kuvio 51 Havainto TheHive -käyttöliittymässä

5.8.5 Tiedon viennin testaaminen FIR

Tiedon viennin testaamiseen FIR -järjestelmästä tein PHP-skriptin, joka hakee tietoa FIR:n taustalla olevasta MySQL -tietokannasta. Skriptin toiminta on seuraavan lainen:

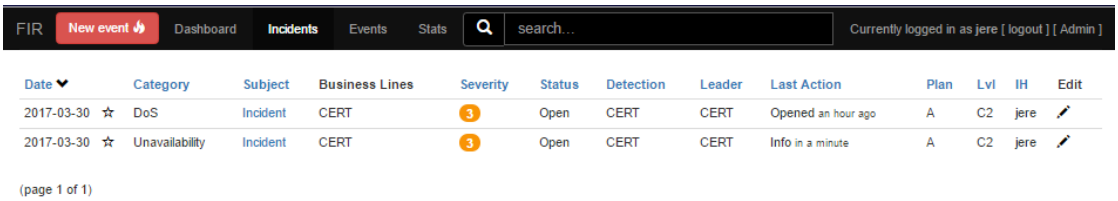
1. Skripti muodostaa yhteyden tietokantaan
2. Skripti valitsee tietokannan taulusta tietoa (poikkeamien tiedot taulussa ”incidents_incident”)
3. Skripti muokkaa tulosteen luettavampaan muotoon

Ensin testasin kaikkien poikkeamien tietojen hakemisen tietokannasta. Tähän käytetty skripti on liitteessä 1. Skriptin antama tuloste poikkeamien tiedoista on esitetty alla kuviossa 52.



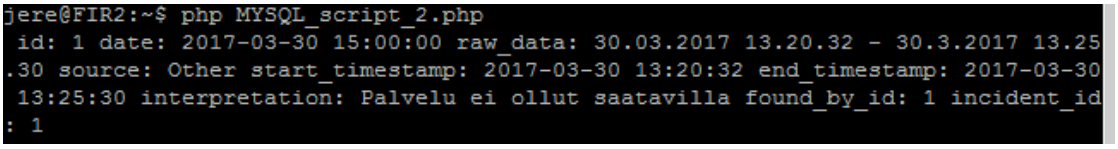
Kuvio 52 FIR PHP -skriptin tuloste poikkeamien tiedoista

Alla kuviossa 53 on esitetty, miten samat poikkeamat näkyvät FIR -käyttöliittymässä.



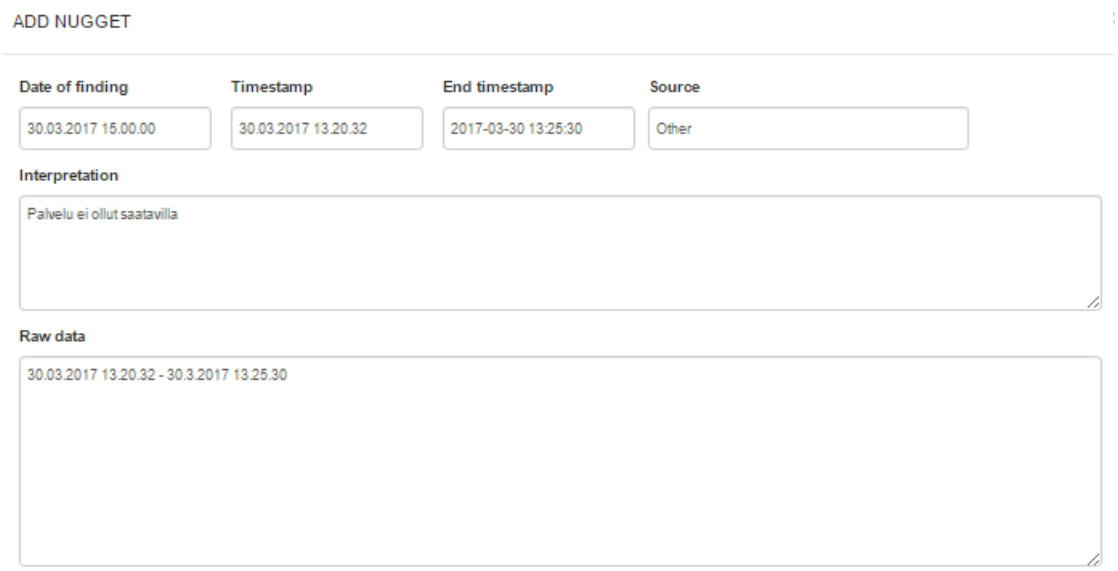
Kuvio 53 FIR poikkeamat käyttöliittymässä

Seuraavaksi testattiin havaintojen hakemista muokkaamalla aiemmin käytettyä skriptiä. Skriptissä piti vaihtaa taulu mistä haetaan tietoa (havaintojen tiedot taulussa ”fir_nuggets_nugget” ja muokata osioita, joka muotoilee tulosteen helpommin luettavaan muotoon. Tämä skripti on liitteessä 2. Alla kuviossa 54 on esitetty skriptin antamasta tulosteesta yhden havainnon tiedot.



Kuvio 54 Havaintojen tietojen hakeminen FIR

Alla kuviossa 55 on esitetty, miten sama havainto näkyy FIR -käyttöliittymässä.



Kuvio 55 Havainto FIR-käyttöliittymässä

FIR -järjestelmän tietokannassa on myös muita tauluja, mutta tietojen hakemista niistä ei lähdetä tässä esittämään, koska ne menevät ihan samalla periaatteella. Esimerkiksi taulussa ”fir_todos_todoitem” säilytetään tehtävälistat ja taulussa ”incidents_comments” näkyvät poikkeamaan liitetyt kommentit. Alla kuviossa 56 on esitetty lista FIR -järjestelmän MySQL-tietokannassa olevista tauluista.

```
+-----------------------------------------------------+
| Tables_in_fir                                       |
+-----------------------------------------------------+
| auth_group                                          |
| auth_group_permissions                              |
| auth_permission                                     |
| auth_user                                           |
| auth_user_groups                                    |
| auth_user_user_permissions                          |
| authtoken_token                                     |
| django_admin_log                                    |
| django_content_type                                 |
| django_migrations                                   |
| django_session                                      |
| django_site                                         |
| fir_artifacts_artifact                              |
| fir_artifacts_artifactblacklistitem                 |
| fir_artifacts_file                                  |
| fir_artifacts_file_hashes                           |
| fir_nuggets_nugget                                  |
| fir_todos_todoitem                                  |
| fir_todos_todolisttemplate                          |
| fir_todos_todolisttemplate_concerned_business_lines |
| fir_todos_todolisttemplate_todolist                 |
| incidents_accesscontrolentry                        |
| incidents_attribute                                 |
| incidents_balecategory                              |
| incidents_businessline                              |
| incidents_categorytemplate                          |
| incidents_comments                                  |
| incidents_incident                                  |
| incidents_incident_artifacts                        |
| incidents_incident_concerned_business_lines         |
| incidents_incident_main_business_lines              |
| incidents_incidentcategory                          |
| incidents_incidenttemplate                          |
| incidents_incidenttemplate_concerned_business_lines |
| incidents_label                                     |
| incidents_labelgroup                                |
| incidents_log                                       |
| incidents_profile                                   |
| incidents_recipienttemplate                         |
| incidents_validattribute                            |
| incidents_validattribute_categories                 |
+-----------------------------------------------------+
```

Kuvio 56 FIR MySQL -tietokannan taulut

5.9 Tiedot käyttöliittymässä ja tietokannassa

5.9.1 Johdanto

Osana tutkimusta verrattiin myös käyttöliittymässä näkyvää tietoa tietokannasta haettuun tietoon. Tämä toteutettiin käytännössä siten, että tehtiin taulukko tietokannasta löytyvistä tiedoista ja numeroitiin ne ja merkittiin numeroilla käyttöliittymän kuvaan mistä vastaava tieto löytyy käyttöliittymästä. Molemmille järjestelmille tehtiin vastaavanlainen vertailu sekä tapauksien että havaintojen tiedoille.

5.9.2 TheHive tiedot käyttöliittymässä ja tietokannassa

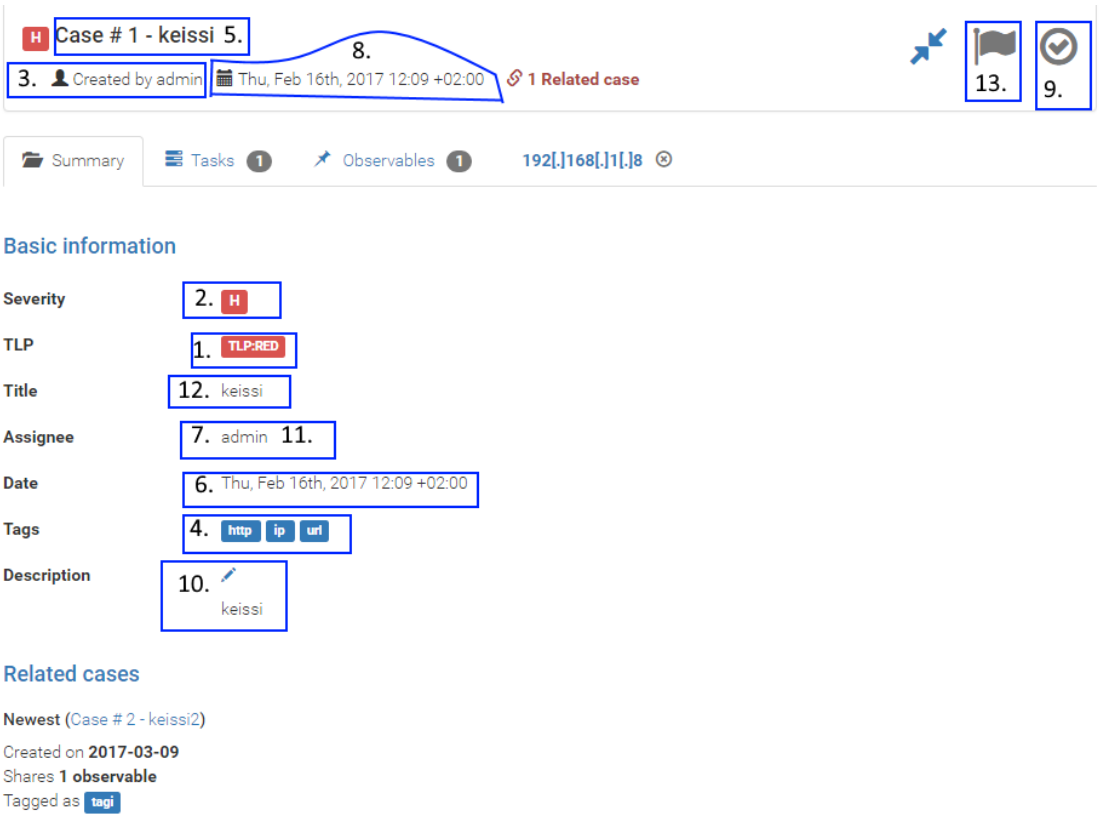
Ensin verrattiin tapauksien tietoja käyttöliittymässä ja taustalla olevassa tietokannassa. Tietokannassa olevista tiedoista tehtiin taulukko. Taulukossa on tiedot numeroituna, kerrotaan, näkyykö tieto käyttöliittymässä ja lyhyt selitys tiedoista. Vastaavilla numeroilla on merkitty käyttöliittymästä otettuun kuvakaappaukseen mistä tieto löytyy käyttöliittymästä. Tapauksien tiedot tietokannassa on esitetty taulukossa 1 ja vastaavat tiedot käyttöliittymässä on esitetty kuviossa 57.

Taulukko 1 Tapauksien tiedot tietokannassa TheHive

TheHive tapauksen tiedot		näkyykö UI	selitys
tlp	1.	näkyy	Luottamuksellisuus
severity	2.	näkyy	Vakavuus
createdby	3.	näkyy	Kuka loi tapauksen
tags	4.	näkyy	Tapauksen tagit
caseId	5.	näkyy	Tapauksen numero käyttöliittymässä
startDate	6.	näkyy	Itse merkitty kellonaika
owner	7.	näkyy	tapauksen omistaja
createdAt	8.	näkyy	Tapauksen merkitsemiskellonaika
status	9.	näkyy	tapauksen tila (open, closed)
description	10.	näkyy	tapauksen kuvaus
user	11.	näkyy	käyttäjä
title	12.	näkyy	otsikko
flag	13.	näkyy	lippu

resolutionStatus	14.	ei näy	oliko false- vai true-positive
summary	15.	ei näy	yhteenveto
impactStatus	16.	ei näy	vaikutukset
updatedBy	17.	ei näy	kuka on viimeksi päivittänyt
updatedAt	18.	ei näy	milloin viimeksi päivitetty
id	19.	ei näy	tapauksen varsinainen ID-arvo
type	20.	ei näy	tiedon tyyppi (tässä aina "case")

Pääasiassa näistä kaikki oleelliset ominaisuudet näkyvät käyttöliittymässä. Hyödyllistä olisi, jos käyttöliittymässä näkyisi ID-arvo (19) sekä se että näkyisi milloin tapausta on päivitetty ja kenen toimesta (17 updatedBy ja 18 updatedAt).



Kuvio 57 Tapauksen tiedot käyttöliittymässä TheHive

Vastaavanlainen vertailu tehtiin myös havaintojen tiedoille tietokannassa. Taulukossa 2 on esitetty havaintojen tiedot tietokannassa ja kuviossa 58 on esitetty vastaavat tiedot numeroituna käyttöliittymän kuvakaappauksessa.

Taulukko 2 Havaintojen tiedot tietokannassa TheHive

TheHive havainto-jen tiedot		nä-kyykö UI	selitys
ioc	1.	näkyy	onko tunnusmerkki poikkeamasta (indicator of compromise)
createdAt	2.	näkyy	merkitsemisaika
createdBy	3.	ei näy	Kuka havainnon merkitsi
tlp	4.	näkyy	vakavuus
startDate	5.	näkyy	merkitty päivämäärä
dataType	6.	ei näy	datatyyppi (IP, HTTP, File
message	7.	näkyy	havainnon kuvaus
user	8.	ei näy	käyttäjä
tags	9.	näkyy	tagit
reports	10.	ei näy	ei dokumentoitu ei tietoa mitä käytännössä tarkoittaa
data	11.	näkyy	havainnon data
status	12.	ei näy	havainnon tila "ok"
_id	13.	ei näy	havaintokohtainen ID-arvo
id	14.	ei näy	sama ID-arvo toiseen kertaan jostain syystä
type	15.	ei näy	tiedon tyyppihavainnon yhteydessä aina "case_artifact"

Joitakin hyödyllisiä tietoja näistä ei näy käyttöliittymässä. Käyttöliittymässä olisi hyvä näkyä kuka merkitsi havainnon (createdBy), merkityn datan tyyppi (dataType) sekä havainnon ID-arvo (_id).

Kuvio 58 TheHive havainnon tiedot käyttöliittymässä

5.9.3 FIR tiedot käyttöliittymässä ja tietokannassa

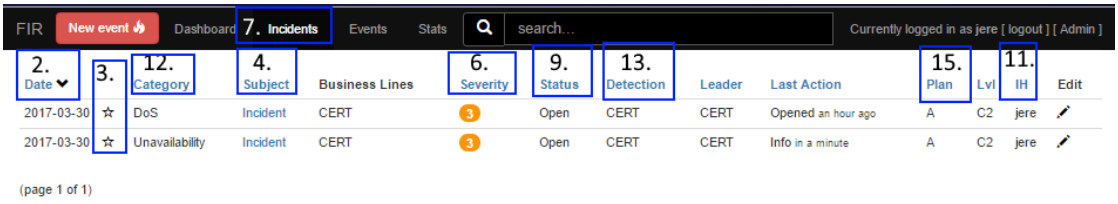
FIR-järjestelmän tiedot tapauksille käyttöliittymässä ja tietokannassa vertailtiin samalla tavalla kuin TheHive-järjestelmässä. Taulukossa 3 on esitetty numeroituna tietokannan tiedot ja kuviossa 59 on esitetty numeroituna vastaavat tiedot FIR:n käyttöliittymästä otetussa kuvakaappauksessa.

Taulukko 3 FIR tapauksen tiedot tietokannassa

tieto		näkyykö UI	selitys
id	1.	ei näy	poikkeaman ID-arvo
date	2.	näkyy	päivämäärä
is starred	3.	näkyy	onko merkitty tähdellä
subject	4.	näkyy	aihe
description	5.	ei näy (näkyy kun avaa tarkempaan näkymään)	kuvaus
severity	6.	näkyy	vakavuus
is incident	7.	näkee siitä, että merkintä näkyy incidents näkymässä	onko poikkeama/ vai normaali tapahtuma
is major	8.	ei näy	onko suuri poikkeama

status	9.	näkyy	tila (open/closed
confi-den-tiality	10.	ei näy	luottamuksellisuus
actor_id	11.	näkyy	käyttäjä
cate-gory_id	12.	näkyy	poikkeaman luokka (Dos, Unavailability)
detec-tion_id	13.	näkyy	taho joka tunnisti poik-keaman
opened by id	14.	ei näy	kuka avasi poikkeaman
plan id	15.	näkyy	mikä toimintasuunnitelmaa käytetään

Näistä tiedoista käyttöliittymässä olisi hyvä näkyä ID-arvo (id), onko poikkeama suuri (is major), luottamuksellisuus (confidentiality) sekä kuka avasi poikkeaman (opened by id).



Kuvio 59 FIR tapauksen tiedot käyttöliittymässä

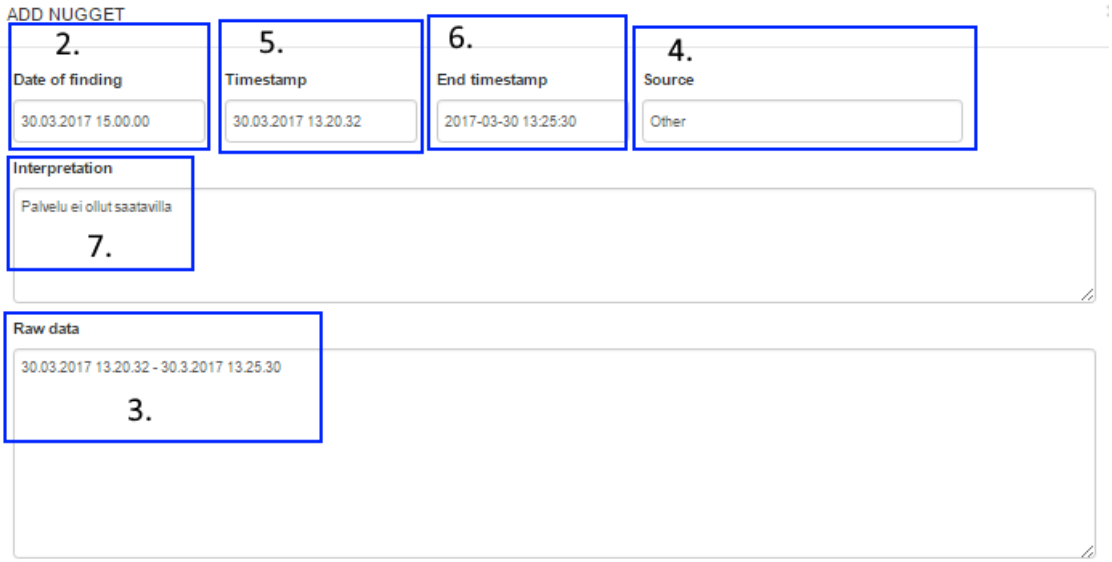
Vastaavanlaisen vertailu tehtiin havainnon tiedoille tietokannassa. Taulukossa 4 on esitetty numeroituna tietokannan tiedot ja kuviossa 60 on esitetty vastaavat tiedot numeroituna käyttöliittymässä.

Taulukko 4 FIR havainnon tiedot tietokannassa

tieto		näkyykö UI	selitys
id	1.	ei näy	havainnon numero
date	2.	näkyy	havainnon merkitsemisajankohta
raw_data	3.	näkyy	havainnon sisältö
source	4.	näkyy	havainnon lähde
start_times-tamp	5.	näkyy	alkamisajankohdan aikaleima

end_times-tamp	6.	näkyy	päättymisajankohdan aikaleima
interpretation	7.	näkyy	havainnon selitys
found_by_id	8.	ei näy	kuka löysi
incident_id	9.	ei näy	poikkeaman ID (johon havainto liit-tyy)

Havainnon tiedoista olisi hyvä näkyä käyttöliittymässä lisäksi kuka löysi poikkeaman (found_by_id) sekä poikkeaman ID-arvo (incident_id).



Kuvio 60 FIR havainnon tiedot käyttöliittymässä

6 Tulokset

Työn tuloksina ovat järjestelmien esittely ja niiden ominaisuuksien dokumentointi, Järjestelmien rajapintojen tutkiminen tiedon tuontiin ja vientiin sekä järjestelmien oleellisten ominaisuuksien vertailu.

Ominaisuuksien dokumentoinnissa käydään läpi järjestelmien kaikki ominaisuudet, joista oleellisimpia vertaillaan vertailuosioissa. Oleellisia ominaisuuksia ovat tapauksien, havaintojen ja tehtävien kirjaaminen sekä se, miten ne esitetään järjestelmässä. Ominaisuuksien vertailun perusteella molemmissa järjestelmissä on sellaisia hyödyllisiä tai oleellisia ominaisuuksia, joita ei toisessa ole. Järjestelmistä kuitenkin TheHive vaikuttaa havainnollisemmalta ja selkeämmältä. Saattaa olla, että käytännössä on

hyvä ottaa molemmat järjestelmät käyttöön ja valita tilanteen mukaan, kumpaa on parempi käyttää. Myös toimeksiantaja mainitsi, että järjestelmistä saatetaan ottaa molemmat käyttöön tarvittaessa.

Rajapintojen tutkimisessa tiedon tuontiin ja vientiin tulokset ovat seuraavat. Järjestelmissä ei ole käyttöliittymässä toimintoja tiedon tuontiin ja vientiin järkevässä muodossa. Tämän takia tutkitaan tiedon tuontiin ja vientiin järjestelmien API-rajapintoja ja taustalla olevia tietokantoja. Kummassakaan järjestelmässä tiedon tuonti järjestelmään ei onnistu automatisoidusti vaan manuaalisesti tapaus ja havainto kerrallaan, joko käyttöliittymän kautta ja TheHive:ssa myös sen API:n kautta komentoriviltä. Tämän takia tutkimuksessa keskitytään tutkimaan tiedon vientiä järjestelmistä. Tiedon vienti olikin toimeksiantajan mukaan oleellisempaa, koska tavoitteena olisi, että tietoa saataisiin vietyä aikajanatyökaluun.

TheHive-järjestelmästä onnistuu tiedon vienti sen API:n kautta. API:in voi ottaa yhteyden HTTP:n kautta. Testaamisessa käytetään Curl-työkalua. Curl on työkalu, jolla voi hakea tietoa järjestelmistä eri protokollien avulla. Tässä tapauksessa käytetään HTTP-protokollaa. Curl-komentoon laitetaan oikea linkki API:in ja käyttäjätunnukset ja tällä tavalla saadaan tietoa JSON-muodossa ulos. JSON-tulosteen muokkaamiseen luettavampaan muotoon käytetään json_reformat -työkalua. Tietoa ei tarvitse muokata ihmiselle luettavampaan muotoon, kun tietoa viedään aikajanatyökaluun. Tässä tapauksessa käytetään json_reformat-työkalua, että tulosteen saisi esitettyä selkeämmin opinnäytetyössä.

FIR-järjestelmässä käytetään taustalla MySQL-tietokantaa, eikä järjestelmässä ole vielä API:a tiedon vientiin. FIR:stä vietiin siis tietoa MySQL-tietokannan kautta. Tiedon vientiä varten tehtiin PHP-skripti, joka yhdistää MySQL-tietokantaan, valitsee sieltä oikeasta taulusta tiedot ja muokkaa skriptin tulosteen helpommin luettavaan muotoon. Skriptistä tehtiin kaksi eri versioita: toinen hakee tapauksien tiedot järjestelmästä ja toinen hakee havaintojen tiedot järjestelmästä.

Yleisesti ottaen TheHive vaikuttaa paremmalle ja viimeistellymmälle järjestelmälle. Käyttöliittymä ja erilaiset näkymät vaikuttavat selkeämmiltä. TheHive:n hyödyllisimpiä ominaisuuksia on tagimerkinnät, joiden avulla on helppo saada käsitys tapauksesta ja merkitä oleellisia tietoja. Niistä on hyötyä myös käytettäessä hakutoimintoa. FIR ei vaikuta niin viimeistellyltä järjestelmältä eikä käyttöliittymä ole niin selkeä. FIR:ssä on kuitenkin hyödyllisinä ominaisuuksia useiden päivämäärien ja aikojen merkitseminen havainnoille (usein on oleellista tietää havainnon tekemisajankohdan lisäksi alkuperäisen tapahtuman ajankohta) sekä kätevä yhteenvetosivu (incident followup), josta on hyötyä esimerkiksi tapauksen raportoinnissa. Loppujen lopuksi käytettävä järjestelmä kannattaa valita tapauksen mukaan. Joskus saattaa olla hyödyllistä, että käytetään molempia järjestelmiä rinnakkain.

7 Yhteenveto ja pohdinta

Alun perin opinnäytetyössä oli tavoitteena löytää aikajanatyökaluja poikkeamanhallintaan. Tarkoituksena tällaisissa työkaluissa on tilannetietoisuuden muodostaminen. Alustavan tutkimuksen perusteella kuitenkaan sopivia aikajanatyökaluja ei ollut saatavilla poikkeamanhallintaan, jonkin verran löytyi forensiikkaan tarkoitettuja työkaluja, mutta sellaisia tässä ei haettu. Oli hieman yllättävää, vaikka kyberturvallisuus ja poikkeamanhallinta on nykyään paljon käsitelty asia, ei sitä varten löytynyt aikajanatyökaluja. Kyberturvallisuusala kuitenkin kehittyy koko ajan ja myöhemmin tällaisia työkaluja saattaa olla saatavilla.

Työssä vertailtiin kahta työkalua poikkeamien kirjaamiseen: TheHive ja FIR. Työn päätuloksina olivat työkalujen oleellisten ominaisuuksien vertailu ja tiedon tuonnin ja viennin testaaminen. Ominaisuuksien vertailussa oli tuloksena, että kummassakin järjestelmässä on hyödyllisiä ominaisuuksia, joita ei toisessa ole ja käytännössä kannattaa valita käytettävä järjestelmä tilanteen mukaan. Tiedon tuonnin ja viennin testaamisessa oli tuloksena, että tiedon tuonti ei onnistu kummassakaan järjestelmässä kuin manuaalisesti yksi merkintä kerrallaan. Tiedon vienti onnistui TheHive-järjestelmästä ottamalla TheHiven API:n yhteyden HTTP:n kautta ja tietoa saatiin ulos JSON-muodossa. Tiedon vienti FIR-järjestelmästä onnistui tekemällä PHP-skripti, joka ottaa

yhteyden MySQL-tietokantaan ja hakee tietokannan taulusta haluttua tietoa ja muokkaa tulosteen luettavaan muotoon.

Ominaisuuksien vertailussa ja tiedon viennin testaamisessa kvalitatiivinen tutkimus toimi käytännössä hyvin. Tässä tehtiin omia laadullisia havaintoja, joihin tutkimus perustui. Tutkimus oli myös käytännössä kokeellista, varsinkin tietojen viennin testaamisessa. Siinä kokeiltiin jotain ja jos se ei toiminut kokeiltiin jotain muuta, kunnes saatiin tiedon vienti toimimaan.

Jatkotutkimuksena on mahdollisesti aikajanatyökalun toteuttaminen, koska JYVSECTEC:llä on tarvetta sellaiselle ja sellaisia ei löytynyt. Tässä työssä oli tavoitteena tutkia saako tietoa vietyä ulos järjestelmistä, että se saataisiin vietyä aikajanatyökaluun. Aikajanatyökalun toteuttaminen jäi tämän opinnäytetyön ulkopuolelle. Jatkotutkimusta voi tehdä myös seuraamalla julkaistaanko poikkeamanhallintaan uusia työkaluja ja tutkimalla niitä.

Poikkeamanhallinta ei ollut minulle tuttu aihe, mutta teoriaosuutta kirjoittaessa sain peruskäsityksen siitä, mitä poikkeamanhallinta on ja miten sen prosessi toimii. Poikkeamanhallinta ja kyberturvallisuus ovat hyvin ajankohtaisia ja mielenkiintoisia aiheita. Poikkeamanhallinta ja kyberturvallisuus ovat tärkeitä nykymaailmassa ja varsinkin tulevaisuudessa.

Oli yllättävää, että aikajanatyökaluja tilannetietoisuutta varten poikkeamanhallintaan ei oikein ollut saatavilla ja miten vähän löytyy tutkimusta tilannetietoisuudesta poikkeamanhallinnassa. Tilannetietoisuutta on paljon tutkittu, mutta ei niin paljon poikkeamanhallintaa varten. Tutkimusta on tehty lähinnä sen verran, että on tiedossa, että sitä pitäisi tutkia lisää.

Opinnäytetyössä olisi voinut teoriaosassa yrittää tutkia asioita useammasta lähteestä. Kaikkeen tähän opinnäytetyöhön liittyvään vain ei tuntunut löytyvän sopivia lähteitä. Opinnäytetyössä olisi voinut etsiä vaihtoehtoisia järjestelmiä TheHive:lle ja FIR:lle, jos olisi ollut aikaa. Se ei kuitenkaan ollut opinnäytetyön toimeksiannossa vaatimuksena.

Opinnäytetyössä sain teoriaosuudessa käsiteltyä poikkeamanhallintaan liittyvät oleelliset asiat aika hyvin, mutta joitakin olisi voinut tutkia tarkemmin. Vertailuosuudessa sain vertailtua järjestelmien oleelliset ominaisuudet. Toteutusosuudessa sain ihan hyvin testattua tiedon viennin järjestelmistä. PHP-skriptinkin sain tehtyä, vaikka ei ollut sellaisesta oikeastaan yhtään kokemusta. Loppujen lopuksi sain hyvin käsiteltyä poikkeamanhallintaprosessin, dokumentoitua ja vertailtua ominaisuudet ja testattua tiedon viennin järjestelmistä, joka oli tässä opinnäytetyössä oleellisin asia, koska tavoitteena on saada tietoa vietyä aikajanatyökaluun.

Opinnäytetyötä tehdessä opin jonkin verran poikkeamanhallinnasta ja siinä tarvittavasta tilannetietoisuudesta. Opin myös poikkeamanhallinnassa tarvittavista työkaluista. Opinnäytetyön toteutuksessa opin tietoturvapoikkeamien kirjaamiseen tarkoitetuissa työkaluissa tarvittavista ominaisuuksista. Opin lisäksi jonkin verran MySQL-tietokannan käyttöä, PHP-skriptin tekoa ja ongelmanratkaisua.

Sain opinnäytetyössä mielestäni käsiteltyä hyvin poikkeamanhallinnan oleellisia asioita. Joitakin asioita olisi ehkä voinut käsitellä tarkemmin useammasta lähteestä. Järjestelmien ominaisuuksien dokumentoinnissa sain esiteltyä järjestelmien oleelliset ominaisuudet ja vertailuosiossa sain vertailtua oleelliset ominaisuudet. Tiedon viennin testaamisessa onnistuin mielestäni melko hyvin. PHP-skriptinkin sain tehtyä sitä varten, vaikka ei skriptaamisesta ollut kokemusta aiemmin. Siihen löytyi hyvin ohjeita internetistä.

Ylipäätään opinnäytetyö onnistui mielestäni hyvin siihen nähden, että ei ollut aiheesta aikaisempaa kokemusta tai tietoa itsellä. Täytin mielestäni opinnäytetyön toimeksiannossa annetut tavoitteet hyvin.

Lähteet

Bronk, H., Thorbuegge, M. & Hakkaja, M. 2006. enisa A step-by-step approach on how to set up a csirt. Viitattu 23.3.2017. https://www.enisa.europa.eu/publications/csirt-setting-up-guide

CERT-BDF/TheHive. N.d. TheHive-järjestelmän github-sivu. Viitattu 10.4.2017. https://github.com/CERT-BDF/TheHive

certsocietegenerale/FIR. N.d. FIR-järjestelmän github-sivu. Viitattu 10.4.2017 https://github.com/certsocietegenerale/FIR.

Cichonski, P., Millar, T., Grance, T. & Scarfone, K. 2012. NIST Computer Security Incident Handling Guide. Viitattu 14.3.2017. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

CSIRT Frequently Asked Questions. N.d. CERTin web-sivu. Viitattu 2.2.2017. http://www.cert.org/incident-management/csirt-development/csirt-faq.cfm

Endsley M. 1995. Toward a Theory of Situation Awareness in Dynamic Systems. Viitattu 23.3.2017. http://uwf.edu/skass/documents/HF.37.1995-Endsley-Theory_000.pdf

Hall, M., Hansen, D. & Jones K. 2015. Cross-domain situational awareness and collaborative working for cyber security. Viitattu 27.2.2017. http://ieeexplore.ieee.org/document/7166110/

Hirsjärvi, S., Remes, P., Sajavaara, P. 2009. Tutki ja kirjoita. 15. p. Helsinki: Kustannusosakeyhtiö Tammi.

Inglot, B., Liu, L. & Antonopoulos, N. 2013. A FrameWork for Enhanced Timeline Analysis in Digital Forensics. Viitattu 28.2.2017. http://ieeexplore.ieee.org/document/6468322/

Insider's guide to incident response. N.d. Alienvaultin web-sivulla. Viitattu 25.1.2017. https://www.alienvault.com/resource-center/ebook/insider-guide-to-incident-response

JYVSECTEC N.d. Viitattu 6.3.2017. http://jyvsectec.fi/fi/
KlinGhofer, B. 2014. Applying OODA Loop in Incident Response Programs. Viitattu 7.3.2017. https://www.hexadite.com/blog/applying-ooda-loop-incident-response-programs/

Kral, P. 2011. The Incident Handlers Handbook. Viitattu 30.1.2017 https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901

Liston, K. 2012. Investigator's Tool-kit: Timeline. Viitattu 6.2.2017. https://isc.sans.edu/forums/diary/Investigators+Toolkit+Timeline/13537/

Lord, N. 2015. What is Incident Response. Viitattu 27.1.2017 https://digitalguardian.com/blog/what-incident-response

Malik, J. 2017. Incident response with the OODA loop method. Viitattu 7.3.2017 https://community.spiceworks.com/how_to/137901-incident-response-with-the-ooda-loop-method

Penedo, D. 2006. Technical Infrastructure of a CSIRT. Viitattu 28.2.2017. http://ieeexplore.ieee.org/document/1690411/

Pham, C. 2001. From Events to Incidents. Viitattu 26.1.2017 https://www.sans.org/reading-room/whitepapers/incident/events-incidents-646

Rouse, M. 2008. incident definition. Viitattu 26.1.2017. http://whatis.techtarget.com/definition/incident

Situation Awareness. N.d. MITREn web-sivulla. Viitattu 1.2.2017. https://www.mitre.org/capabilities/cybersecurity/situation-awareness

Liitteet

Liite 1. PHP-skripti tapauksien tietojen vientiin FIR:n MySQL-tietokannasta

```
<?php
$servername = "192.168.1.9";
$username = "fir";
$password = "1234";
$dbname = "fir";

error_reporting(0);

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM incidents_incident";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
   // output data of each row
   while($row = $result->fetch_assoc()) {

      echo "id: " . $row["id"]. " date: " . $row["date"]. " is_starred: " .
$row["is_starred"].
" subject: " . ["subject"]. " description: " . $row["description"]. " severity: " .
$row["severity
"]. " is_incident: " . $row["is_incident"]. " is_major: " . $row["is_major"]. " status: " .
$row["
status"]. " confidentiality: " . $row["confidentiality"]. " actor_id: " . $row["actor_id"].
" cate
gory_id " . $row["category_id"]. " detection_id ". $row["detection_id"]. "
opened_by_id " . $row["
opened_by_id"]. " plan_id: " . $row["plan_id"]. "\r\n" . "\r\n" ;
   }
} else {
   echo "0 results";
}
$conn->close();
?>
```

Liite 2. PHP-skripti havaintojen tietojen vientiin FIR:n MySQL-tietokannasta

```
<?php
$servername = "192.168.1.9";
$username = "fir";
$password = "1234";
$dbname = "fir";

error_reporting(0);



// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}




$sql = "SELECT * FROM fir_nuggets_nugget";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
   // output data of each row
   while($row = $result->fetch_assoc()) {

      echo " id: " . $row["id"]. " date: " . $row["date"]. " raw_data: " .
$row["raw_data"]. " source: " . $row["source"]. " start_timestamp: " .
$row["start_timestamp"]. " end_timestamp: " . $row["end_timestamp"]. " interpre-
tation: " . $row["interpretation"]. " found_by_id: " . $row["found_by_id"]. " inci-
dent_id: " . $row["incident_id"].  "\r\n" . "\r\n" ;
```

```
    }
} else {
    echo "0 results";
}
$conn->close();
?>
```