Tampereen yliopisto

Teemu Lång

AVOIMEN LÄHDEKOODIN ISÄNTÄPOHJAISTEN TUNKEUTUMISENHAVAITSEMISJÄRJESTELMIEN VERTAILU

Kandidaatintutkielma
Informaatioteknologian ja viestinnän tiedekunta
Tarkastaja: yliopisto-opettaja Matti Monnonen
Toukokuu 2020

TIIVISTELMÄ

Teemu Lång: Avoimen lähdekoodin isäntäpohjaisten tunkeutumisenhavaitsemisjärjestelmien vertailu
Kandidaatintutkielma
Tampereen yliopisto
Tieto- ja sähkötekniikka, TkK
Toukokuu 2020

Tunkeutumisenhavaitsemis- ja estojärjestelmän (engl. intrusion detection and prevention system) tehtävänä on havaita ja estää tietojärjestelmään kohdistuvat tapahtumat, jotka ovat tietoturvapolitiikkojen, hyväksyttävien käyttötapojen tai tietoturvakäytäntöjen vastaisia. Tällaiset luvattomat tapahtumat ovat yleensä seurausta haittaohjelmista, kyberhyökkäyksistä tai organisaation oman henkilöstön toimista. Tunkeutumisenhavaitsemisjärjestelmät jaetaan yleisesti isäntäpohjaisiin ja verkkopohjaisiin järjestelmiin. Isäntäpohjaiset järjestelmät valvovat sen isäntäkoneen tapahtumia, jolle ne on asennettu. Ne soveltuvatkin hyvin esimerkiksi kaupallisilta hostingpalveluntarjoajilta vuokratuille virtuaali- ja pilvipalvelimille. Verkkopohjaiset järjestelmät valvovat kokonaisen verkkosegmentin verkkoliikennettä käyttäen ohjelmisto- tai laitteistopohjaisia sensoreita.

Tässä työssä vertaillaan joitakin avoimeen lähdekoodiin pohjautuvia isäntäpohjaisia tunkeutumisenhavaitsemisjärjestelmiä, jotka toimivat Linux-ympäristössä. Työn tavoite on selvittää, mitä tällaisia ohjelmistoja on saatavilla ja mitä ominaisuuksia niissä on. Vertailua varten ohjelmistot asennettiin virtuaalipalvelimille. Kaikkien vertailussa mukana olevien ohjelmistojen arkkitehtuuri koostuu valvonnan suorittavasta agenttiohjelmistosta ja erillisestä keskitetystä hallintapalvelinohjelmistosta.

Työ jakaantuu kolmeen osaan. Teoriaosassa perehdytään erilaisiin tunkeutumishavaitsemisjärjestelmiin, tunkeutumisenestossa käytettyihin vastatoimiin sekä menetelmiin, joilla tunkeutumisia voidaan havaita. Vertailuosassa tarkastellaan vertailtavien isäntäpohjaisten tunkeutumisenhavaitsemisjärjestelmien asennusta ja ominaisuuksia. Viimeisessä osassa esitetään yhteenveto vertailun tuloksista.

Vertailuun valikoitui kolme työssä asetetut ehdot täyttävää ohjelmistoa: Ossec, Wazuh ja Samhain. Jokaisen vertailussa mukana olevan ohjelmiston ominaisuuksiin sisältyvät lokitapahtumien valvonta, tiedostojen eheyden valvonta, haittaohjelmien havaitseminen ja aktiivisten vastatoimien suorittaminen tunnistettujen tunkeutumisten torjumiseksi. Wazuhin ominaisuuksiin kuuluvat lisäksi muun muassa avoimen lähdekoodin selainkäyttöliittymä ja ohjelmistohaavoittuvuuksien tunnistaminen haavoittuvuustietokantoja hyödyntäen. Wazuh onkin vertailluista järjestelmistä monipuolisin sekä ominaisuuksien määrän että laadun perusteella. Ossecissa on useita samanlaisia ominaisuuksia ja yksinkertaisempi arkkitehtuuri ja asennus kuin Wazuhissa. Samhain on ominaisuuksiltaan vertailun huonoin ohjelmisto.

Avainsanat: tunkeutumisenhavaitseminen, tunkeutumisenesto, Ossec, Wazuh, Samhain

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

SISÄLLYSLUETTELO

1. JOHDANTO 1
2. TUNKEUTUMISENHAVAITSEMISJÄRJESTELMÄT 2
2.1 Erilaiset havaitsemisjärjestelmät 2
2.1.1 Isäntäpohjaiset järjestelmät 3
2.1.2 Verkkopohjaiset järjestelmät 4
2.2 Vastatoimet 5
2.3 Havaitsemismenetelmät 6
2.3.1 Tunnistetietoon pohjautuva menetelmä 6
2.3.2 Anomaliaan pohjautuva menetelmä 7
2.3.3 Tilallinen protokolla-analyysi 8
3. ISÄNTÄPOHJAISTEN JÄRJESTELMIEN VERTAILU 9
3.1 Ossec 9
3.1.1 Asennus 11
3.1.2 Lokitapahtumien valvonta 11
3.1.3 Tiedostojen eheyden valvonta 13
3.1.4 Haittaohjelmien havaitseminen 13
3.1.5 Aktiiviset vastatoimet 14
3.2 Wazuh 14
3.2.1 Asennus 16
3.2.2 Lokitapahtumien valvonta 17
3.2.3 Tiedostojen eheyden valvonta 18
3.2.4 Ohjelmistohaavoittuvuuksien havaitseminen 19
3.3 Samhain 20
3.3.1 Asennus 21
3.3.2 Lokitapahtumien valvonta 22
3.3.3 Tiedostojen eheyden valvonta 23
3.3.4 Haittaohjelmien havaitseminen 23
4. YHTEENVETO 25
LÄHTEET 27
LIITE A: OSSEC ASENNUSKOMENNOT 29
LIITE B: WAZUH ASENNUSKOMENNOT 30
LIITE C: SAMHAIN ASENNUSKOMENNOT 32

KUVALUETTELO

Kuva 1. *Ossecin arkkitehtuuri [9]. 10*

Kuva 2. *Esimerkki lokitapahtuman analysoinnin vaiheista Ossecissa [9]. 12*

Kuva 3. *Wazuhin arkkitehtuuriin kuuluvat ohjelmat [18]. 15*

Kuva 4. *Wazuhin ohjelmien väliset yhteydet [18]. 16*

Kuva 5. *Wazuhin käyttöliittymän tietoturvatapahtumanäkymä. 18*

Kuva 6. *Wazuhin käyttöliittymän eheydenvalvontanäkymä. 19*

Kuva 7. *Samhainin arkkitehtuuri [27]. 21*

1. JOHDANTO

Day määrittelee tunkeutumisen luvattoman tahon teoksi, jolla on kielteinen vaikutus tietojärjestelmän sisältämän tai käsittelemän tiedon luottamuksellisuuteen, eheyteen tai saatavuuteen [1]. Tunkeutumisenhavaitseminen (engl. intrusion detection) pyrkii huomaamaan tällaiset luvattomat toimet valvomalla tietojärjestelmien tai tietoverkkojen tapahtumia. Tapahtumat analysoidaan, etsien niistä merkkejä tietoturvaloukkauksien uhkista ja jo tapahtuneista loukkauksista. Tunkeutumisenesto (engl. intrusion prevention) pyrkii havaitsemisen lisäksi estämään havaittujen uhkien toteutumisen ja pysäyttämään jo käynnissä olevat tietoturvaloukkaukset. [2]

Tunkeutumisenhavaitsemis- ja estojärjestelmiä (engl. intrusion detection and prevention system) on saatavilla sekä ohjelmistoina että erillisinä laitteina. Näitä järjestelmiä voidaan käyttää yhdessä perinteisten palomuurien, virustorjuntaohjelmistojen ja sähköpostisuodattimien kanssa lisäämään tietojärjestelmien tietoturvaa. Seuraavan sukupolven palomuurit (engl. next-generation firewall) sisältävät tunkeutumiseneston ominaisuuksia integroituna osaksi palomuuria [3]. Tunkeutumisenhavaitsemisjärjestelmiä kutsutaan joko isäntäpohjaisiksi tai verkkopohjaisiksi sen mukaan, havainnoivatko ne yhden isäntäkoneen vai kokonaisen verkkosegmentin tapahtumia.

Tässä työssä vertaillaan joitakin avoimeen lähdekoodiin pohjautuvia isäntäpohjaisia tunkeutumisenhavaitsemisjärjestelmiä. Tarkoituksena on selvittää, mitä tällaisia ohjelmistoja on saatavilla ja mitä ominaisuuksia niissä on. Vertailuun on sisällytetty vain Linux-pohjaisissa pilvi- ja virtuaalipalvelimissa toimivia ohjelmistoja. Erityisesti on huomioitu soveltuvuus kaupallisilta hosting-palveluntarjoajilta vuokrattuihin virtuaalipalvelimiin, joihin asiakas saa pääkäyttäjän käyttöoikeudet. Näihin ympäristöihin isäntäpohjainen yhtä isäntäkonetta valvova tunkeutumisenhavaitsemisjärjestelmä sopii hyvin. Verkkopohjaisen järjestelmän käyttö tällaisessa ympäristössä voi olla mahdotonta johtuen sen vaatimuksesta asettaa verkkosovitin valikoimattomaan tilaan (engl. promiscuous mode).

Työn toisessa luvussa perehdytään tarkemmin erilaisten tunkeutumisenhavaitsemisjärjestelmien eroihin ja järjestelmien käyttämiin havaitsemismenetelmiin. Kolmannessa luvussa tarkastellaan kolmen isäntäpohjaisen järjestelmän asennusta ja ominaisuuksia. Neljännessä luvussa kootaan yhteen vertailun olennaisimmat tulokset.

2. TUNKEUTUMISENHAVAITSEMISJÄRJESTELMÄT

Tunkeutumisenhavaitsemisjärjestelmät etsivät tapahtumia, jotka ovat tietoturvapolitiikkojen, hyväksyttävien käyttötapojen tai tietoturvakäytäntöjen vastaisia. Tällaiset tapahtumat voivat olla seurausta esimerkiksi haittaohjelmista, kyberhyökkäyksistä tai organisaation oman henkilöstön toimista. Järjestelmän luvalliset käyttäjät voivat väärinkäyttää oikeuksiaan tai tavoitella laajempia oikeuksia luvattomasti. Kaikki löydetyt tapahtumat eivät ole seurausta pahantahtoisista toimista, vaan osa niistä voi johtua esimerkiksi käyttäjien vahingossa tekemistä virheistä. [2]

Keskeinen haaste tunkeutumisenhavaitsemisjärjestelmissä on havaintojen tarkkuus. Järjestelmien tulisi minimoida väärät positiiviset löydökset, jotka eivät ole oikeita tietoturvaloukkauksia ja aiheuttavat aiheettomia hälytyksiä. Aiheettomien hälytysten tutkiminen kuluttaa turhaan ylläpitäjien aikaa. Toisaalta järjestelmien tulisi välttää myös vääriä negatiivisia havaintoja, jotka tarkoittavat löytämättä jääneitä oikeita tietoturvaloukkauksia. [4] Käytännössä vääriltä positiivisilta ja negatiivisilta havainnoilta ei voida välttyä. Tyypillisesti havaitsemisjärjestelmän muokkaaminen niin, että väärät negatiiviset havainnot vähenevät, lisää väärien positiivisten havaintojen määrää, ja päinvastoin. Monet organisaatiot minimoivat mieluummin väärien negatiivisten havaintojen määrän, jotta tietoturvaloukkauksia ei jäisi huomaamatta, ja hyväksyvät suuremman määrän aiheettomia positiivisia havaintoja. [2]

2.1 Erilaiset havaitsemisjärjestelmät

Tunkeutumisenhavaitsemisjärjestelmät jaetaan yleisesti isäntäpohjaisiin ja verkkopohjaisiin järjestelmiin. Isäntäpohjainen järjestelmä toimii yhdellä isäntäkoneella tarkkaillen sen tapahtumia ja etsien merkkejä epäilyttävästä toiminnasta [2]. Isäntäpohjaisia järjestelmiä käsitellään tarkemmin alaluvussa 2.1.1. Verkkopohjainen järjestelmä valvoo tiettyjen verkkosegmenttien tai laitteiden verkkoliikennettä etsien merkkejä epäilyttävästä toiminnasta [2]. Verkkopohjaisia järjestelmiä esitellään alaluvussa 2.1.2. Isäntäpohjaisia ja verkkopohjaisia havaitsemisjärjestelmiä käytetään usein yhdessä, jolloin saadaan molempien vahvuudet hyödynnettyä [4].

Hybridijärjestelmiksi kutsutaan tunkeutumisenhavaitsemisjärjestelmiä, jotka sisältävät sekä isäntäpohjaisten että verkkopohjaisten järjestelmien ominaisuuksia [5]. Jotkin järjestelmät esimerkiksi toimivat isäntäkoneen sijasta omalla laitteellaan, mutta valvovat

vain yhden isäntäkoneen tai sovelluksen verkkoliikennettä [2]. Hybridijärjestelmiksi voidaan kutsua myös sellaisia tunkeutumisenhavaitsemisjärjestelmiä, jotka hyödyntävät toiminnassaan useita erilaisia havaitsemismenetelmiä [6]. Havaitsemismenetelmiä käsitellään alaluvussa 2.3.

Muitakin tunkeutumisenhavaitsemisjärjestelmien jakotapoja käytetään, kuten langattomat tunkeutumisenhavaitsemisjärjestelmät ja verkkokäyttäytymisen analyysiin perustuvat järjestelmät. Langattomat järjestelmät tarkkailevat langattomien verkkojen liikennettä analysoiden langattomien verkkojen protokollia epäilyttävän toiminnan havaitsemiseksi. Niillä voidaan etsiä myös luvatonta langatonta verkkoliikennettä paikoissa, joissa langattomat verkot eivät ole sallittuja. Verkkokäyttäytymisen analyysiin perustuvia järjestelmiä käytetään yleensä organisaatioiden sisäverkoissa tunnistamaan epätavallisia liikennevirtoja verkkoliikenteessä. Poikkeamia liikennevirtoihin voivat aiheuttaa esimerkiksi hajautetut palvelunestohyökkäykset, jotkin haittaohjelmat ja verkon skannaus. [2]

2.1.1 Isäntäpohjaiset järjestelmät

Isäntäpohjainen tunkeutumisenhavaitsemisjärjestelmä (engl. host-based intrusion detection system) asennetaan suoraan valvottavalle isäntäkoneelle. Järjestelmä valvoo isäntäkoneen tapahtumia pyrkien havaitsemaan epäilyttävän käytöksen. [4] Isäntäkone voi olla palvelin tai työasema. Yleisimmin isäntäpohjainen havaitsemisjärjestelmä asennetaan palvelimelle, jonne on pääsy julkisesta verkosta tai joka sisältää arkaluonteista tietoa. [2]

Isäntäkoneelle asennettavaa ohjelmistoa kutsutaan agentiksi. Valvonnan lisäksi agentti voi pyrkiä torjumaan havaitun tunkeutumisen vastatoimillaan. Usein agentit lähettävät havaintonsa keskitetylle hallintapalvelimelle käyttäen isäntäkoneiden olemassa olevia verkkoyhteyksiä. Hallintapalvelimet tarjoavat tyypillisesti käyttöliittymän valvonnan seuraamiseen ja hallinnointiin. [2]

Monet isäntäpohjaiset järjestelmät muokkaavat isäntäkoneen sisäistä arkkitehtuuria voidakseen seurata ja tarvittaessa estää esimerkiksi järjestelmäkutsuja, tiedostojärjestelmien käyttöä ja verkkoliikennettä. Järjestelmät, jotka eivät muokkaa isäntänsä arkkitehtuuria, ovat yleensä tehottomampia havaitsemaan uhkia ja kyvyttömämpiä tekemään estotoimia. [2]

Isäntäpohjaiset järjestelmät voivat valvoa muun muassa isäntäkoneen

- käynnissä olevia prosesseja ja niiden järjestelmäkutsuja
- langallista ja langatonta verkkoliikennettä

- tiedostojen hakuja
- kriittisten tiedostojen sisällön ja pääsyoikeuksien muutoksia
- järjestelmän lokitiedostoja
- verkkoasetusten muutoksia
- järjestelmän ja sovellusten asetusmuutoksia [2].

Erityinen etu isäntäpohjaisissa järjestelmissä on niiden mahdollisuus analysoida salattu verkkoliikenne sen jälkeen, kun salaus on purettu isäntäkoneella. Näin myös salattuja yhteyksiä käyttävät hyökkäykset voidaan tunnistaa. [1]

Isäntäpohjaisen järjestelmän haittapuolia ovat seuraavat asiat:

- Havaintojen näkyvyys rajoittuu vain yksittäiseen valvottavaan isäntäkoneeseen [4].
- Havaitsemisjärjestelmä kuluttaa isäntäkoneen resursseja ja voi vaikuttaa haitallisesti isäntäkoneen suorituskykyyn [4].
- Hyökkäykset voidaan havaita vasta kun ne ovat jo saavuttaneet isäntäkoneen [4].
- Hälytykset ja keskitetty raportointi eivät aina ole reaaliaikaisia, jolloin hyökkäykset havaitaan viiveellä [2].
- Isäntäpohjainen järjestelmä voi olla epäyhteensopiva ohjelmistopohjaisten palomuurien, VPN-ohjelmistojen tai muiden tietoturvaohjelmistojen kanssa [2].
- Jos hyökkääjä saa isäntäkoneen hallintaansa edes osittain, hänen voi olla mahdollista vaikuttaa tunkeutumisenhavaitsemisjärjestelmän toimintaan [4].

2.1.2 Verkkopohjaiset järjestelmät

Verkkopohjainen tunkeutumisenhavaitsemisjärjestelmä (engl. network intrusion detection system) valvoo ja analysoi yhden tai useamman verkkosegmentin verkkoliikennettä käyttäen erityisiä sensoreita. Sensorit ovat joko ohjelmiston sisältäviä valmiita laitteita tai pelkästään ohjelmistoja, jotka voidaan asentaa mihin tahansa ohjelmiston vaatimukset täyttävään isäntäkoneeseen. [4] Sensorit välittävät havaintonsa hallintapalvelimelle, usein hyödyntäen erillistä hallintaverkkoa, joka on erotettu valvottavan liikenteen käyttämästä verkosta [2].

Sensorin mahdollisia asennustapoja on kaksi erilaista. Sensori voidaan kytkeä kuten palomuuri niin, että valvottava verkkoliikenne kulkee sen läpi, jolloin sensori voi myös tarvittaessa palomuurin tavoin helposti estää haitalliseksi havaitsemansa liikenteen läpipääsyn tai poistaa haitallisen sisällön. Vaihtoehtoisesti sensori voi olla passiivinen kuuntelija, joka valvoo verkkoliikenteen kopiota pystymättä kuitenkaan itse vaikuttamaan liikenteeseen. Passiivinen sensori voidaan erillisen laitteen lisäksi toteuttaa myös esimerkiksi verkkokytkinten peilaustoimintoa hyödyntäen. [2]

Verkkopohjaiset järjestelmät havaitsevat uhkia verkko-, siirto- ja sovellusprotokollien liikenteessä, tukien yleisesti esimerkiksi IP-, ICMP-, IGMP-, TCP- ja UDP-protokollia ja lukuisia yleisiä sovellusprotokollia. Lisäksi jotkin järjestelmät tunnistavat verkkopaketeista niiden lähettäjän käyttöjärjestelmän tai sovelluksen tyypin ja version, jolloin myös lähettäjän tunnetut haavoittuvuudet ovat tiedossa. [2]

Verkkopohjaisten järjestelmien haasteita ovat seuraavat asiat:

- Sensorit eivät pysty analysoimaan salatun verkkoliikenteen sisältöä [2].
- Verkon kuormituksen ollessa korkea, sensorien suorituskyky saattaa loppua kesken ja ne voivat kadottaa osan verkkopaketeista, minkä seurauksena jotkin hyökkäykset voivat jäädä havaitsematta. Jos verkkoliikenne kulkee sensorin läpi, kadotetut paketit voivat lisäksi aiheuttaa häiriöitä verkkopalvelujen saatavuuteen. [2]
- Sensorit ovat alttiita palvelunestohyökkäyksille, joissa hyökkääjä aiheuttaa epätavallisen suuren verkkoliikenteen määrän. Kuormituksen vuoksi havaitsematta jääneiden hyökkäysten lisäksi hyökkääjä voi pyrkiä naamioimaan todellisen hyökkäyksensä hämäystarkoituksessa aiheutetun verkkoliikenteen sekaan. [2]
- Jos sensori estää haitallisen liikenteen läpipääsyn, väärät positiiviset havainnot ovat tavanomaista vakavampia koska ne aiheuttavat häiriöitä palvelun saatavuuteen [1].

2.2 Vastatoimet

Löydettyään tietoturvaloukkauksen, tunkeutumisenhavaitsemisjärjestelmä lähettää ilmoituksen järjestelmän ylläpitäjille. Tätä kutsutaan passiiviseksi vastatoimeksi. Ilmoitus voidaan lähettää esimerkiksi sähköpostina tai SNMP-hälytyksenä. [5] Ilmoitusten muotoa on pyritty standardoimaan esimerkiksi dokumentissa RFC 4766 [7]. Tyypillisesti järjestelmät myös kirjoittavat lokiin havainnoistaan laajat tiedot, kuten päiväyksen ja kellon-

ajan, tapahtuman tyypin, tärkeysluokituksen, ja mahdolliset suoritetut estotoimet. Järjestelmässä paikallisesti säilytettävän lokin lisäksi useimmat järjestelmät mahdollistavat lokitapahtumien lähetyksen keskitetylle lokipalvelimelle. [2] Tapahtumat voidaan lähettää myös tunkeutumisenhavaitsemisjärjestelmää täydentävälle tietoturvatiedon ja -tapahtumien hallintajärjestelmälle (engl. security information and event management) [2].

Aktiiviksi vastatoimiksi kutsutaan tunkeutumisenestojärjestelmän yrityksiä estää tai pysäyttää tietoturvaloukkaus. Järjestelmä voi esimerkiksi katkaista hyökkääjän TCP-yhteyden tai lisätä palomuureihin sääntöjä hyökkääjän pääsyn estämiseksi. [5] Verkkopohjaiset järjestelmät voivat myös rajoittaa hyökkäyksen käyttämän verkkoprotokollan siirtonopeutta tai poistaa verkkopaketeista haitallista sisältöä [2].

2.3 Havaitsemismenetelmät

Monet tunkeutumisenhavaitsemisjärjestelmät hyödyntävät useita erilaisia havaitsemismenetelmiä tunnistaessaan tunkeutumisia. Eri menetelmiä voidaan käyttää toisistaan erillään tai integroidusti toisiaan tukien. [2]

Havaitsemismenetelmät voidaan jakaa kolmeen pääluokkaan: tunnistetietoon pohjautuviin menetelmiin, anomaliaan pohjautuviin menetelmiin ja tilalliseen protokolla-analyysiin [2]. Seuraavissa alaluvuissa esitellään näitä tarkemmin.

2.3.1 Tunnistetietoon pohjautuva menetelmä

Tunnistetietoon pohjautuva havaitseminen (engl. signature-based detection) perustuu yksittäiset hyökkäykset tunnistaviin malleihin [4]. Tapahtumaa, kuten verkkopakettia tai lokitiedoston riviä, verrataan ennalta tunnettuihin malleihin. Jos tapahtuma vastaa mallia, tunnistus on tapahtunut. Esimerkiksi sähköpostin liitteenä lähetetyt haittaohjelmat tai kirjautumisyritykset luvattomalla käyttäjätunnuksella on mahdollista havaita. Tunnistetietoon pohjautuva havaitseminen on havaitsemismenetelmistä yksinkertaisin. Sitä kutsutaan joskus myös nimellä väärinkäytösten havaitseminen. [2]

Tunnistetietoon pohjautuva havaitseminen toimii hyvin tunnettuja uhkia vastaan, mutta on pääosin tehoton ennalta tuntemattomia uhkia vastaan. Myös tunnettujen uhkien pienetkin variaatiot jäävät helposti havaitsematta. Tunnistetietoon pohjautuvat menetelmät ovat tilattomia, joten ne eivät osaa yhdistää esimerkiksi HTTP-pyyntöä ja -vastausta toisiinsa. Tästä syystä ne eivät tunnista uhkia, joiden havaitseminen vaatii usean eri tapahtuman yhdistämistä toisiinsa. On tosin syytä mainita, että joskus tunnistetietoon pohjautuvien menetelmien katsotaan sisältävän myös tilallisen protokolla-analyysin toimintaperiaatteet. [2] Tilallista protokolla-analyysiä käsitellään alaluvussa 2.3.3.

Mallit sisältävä tietokanta vaatii säännöllisiä päivityksiä, jotta uudetkin uhat voidaan tunnistaa [6]. Mallien tekeminen uusien hyökkäysten pohjalta on työlästä. Mallit eivät saisi olla liian tarkkoja, jotta pienet muutokset hyökkäyksissä eivät jäisi havaitsematta. Toisaalta mallit eivät myöskään saisi olla liian ylimalkaisia, jotta ne eivät aiheuttaisi vääriä positiivisia löydöksiä. [4]

2.3.2 Anomaliaan pohjautuva menetelmä

Anomaliaan pohjautuva havaitseminen (engl. anomaly-based detection) pyrkii tunnistamaan järjestelmän toiminnassa tai verkkoliikenteessä poikkeamia normaaliin toimintaan verrattuna. Vertailu perustuu profiileihin, jotka voivat sisältää muun muassa käyttäjien, isäntäkoneiden, verkkoyhteyksien ja sovellusten normaalin käyttäytymisen mallit. Profiilit luodaan seuraamalla tavanomaista toimintaa tietyn aikaa. Toimintaa verrataan aiemmin luotuihin profiileihin tyypillisesti käyttäen tilastollisia menetelmiä. [2] Vertailussa voidaan hyödyntää myös useita muita tekniikoita kuten koneoppimista [8].

Anomaliaan pohjautuvan havaitsemisen suurin etu on sen kyky tunnistaa ennalta tuntemattomia uhkia. Esimerkiksi uusi haittaohjelma voisi lisätä prosessoritehon käyttöä, lähettää poikkeuksellisen määrän sähköposteja, avata suuren määrän verkkoyhteyksiä tai tehdä muita järjestelmän normaalista toiminnasta selvästi poikkeavia toimia. [2]

Havaitsemisessa käytetyt profiilit voivat olla joko staattisia tai dynaamisia. Staattiset profiilit luodaan seuraamalla järjestelmän normaalia toimintaa päivien tai joskus viikkojenkin ajan. Tämän opetusjakson jälkeen niihin ei enää tehdä muutoksia, vaan normaalia toimintaa verrataan kerran luotuihin staattisiin profiileihin. Staattiset profiilit täytyy luoda uudestaan määrävälein järjestelmien ja verkkojen normaalin käyttäytymisen muuttuessa ajan kuluessa. Dynaamisia profiileja sen sijaan muokataan jatkuvasti automaattisesti järjestelmän päivittäistä toimintaa seuraamalla. [2]

Profiileja luotaessa on riskinä, että järjestelmässä tapahtuu opetusjakson aikana luvatonta toimintaa, jolloin tällainen toiminta tulkitaan profiilissa osaksi normaalia toimintaa. Dynaamisia profiileja käytettäessä hyökkääjä voi hiljalleen lisätä haitallisen toiminnan määrää, jolloin toimintaa ei tunnisteta haitalliseksi ja profiili oppii virheellisesti jatkuvasti lisääntyvän haitallisen toiminnan olevan osa normaalia toimintaa. [2]

Anomaliaan pohjautuvan havaitsemisen haasteena ovat myös harvemmin tapahtuvat poikkeavat mutta luvalliset toimet, jotka aiheuttavat helposti vääriä positiivista löydöksiä. Esimerkiksi kuukausittain tapahtuvat suuret varmuuskopioiden siirrot saattavat tapahtua opetusjakson jälkeen, jolloin ne voivat aiheuttaa aiheettoman hälytyksen. Anomalian aiheuttaman väärän hälytyksen syytä voi olla myös jälkikäteen vaikeaa selvittää. [2]

2.3.3 Tilallinen protokolla-analyysi

Tilallinen protokolla-analyysi (engl. stateful protocol analysis) pyrkii tunnistamaan järjestelmän toiminnassa tai verkkoliikenteessä poikkeamia yleisesti hyväksyttyyn toimintaan nähden. Se on siis verrattavissa anomaliaan pohjautuvaan menetelmään. Erona kuitenkin on, että tilallinen protokolla-analyysi käyttää tunkeutumisenhavaitsemisjärjestelmän toimittajan valmiita profiileja, jotka määrittävät, mikä on hyväksyttyä toimintaa. Profiilit pohjautuvat yleensä protokollastandardeihin, pyrkien kuitenkin sallimaan variaatiot protokollatoteutuksissa. [2]

Tilallinen protokolla-analyysi ymmärtää ja seuraa tukemiensa verkko-, siirto- ja sovellusprotokollien tilaa. Esimerkiksi FTP-protokollaa käytettäessä se tietää istunnon tilan, kuten onko käyttäjä todennettu hyväksytysti, seuraamalla FTP-protokollan vastauksia. Monet FTP-komennot voidaan tulkita epäilyttäviksi, jos ne annetaan ennen käyttäjän sisäänkirjautumista, mutta onnistuneen sisäänkirjautumisen jälkeen ne katsotaan hyväksytyksi toiminnaksi. Jos epäilyttävää toimintaa tapahtuu käyttäjän todennuksen jälkeen, voidaan hälytykseen sisällyttää tieto käytetystä käyttäjätunnuksesta. [2]

Tilallisen protokolla-analyysin haaste on sen aiheuttama kuormitus järjestelmälle johtuen useiden yhtäaikaisten istuntojen seurannasta ja tilallisen analyysin monimutkaisuudesta. Se ei myöskään tunnista esimerkiksi palvelunestohyökkäystä, joka suoritetaan antaen suuri määrä hyväksyttyjä komentoja lyhyen ajan sisällä. [2]

3. ISÄNTÄPOHJAISTEN JÄRJESTELMIEN VERTAILU

Tässä luvussa perehdytään kolmeen avoimen lähdekoodin isäntäpohjaiseen tunkeutumisenhavaitsemisjärjestelmään. Erityisesti tarkastellaan järjestelmien merkittävimpiä ominaisuuksia ja asennusta Linux-palvelimille.

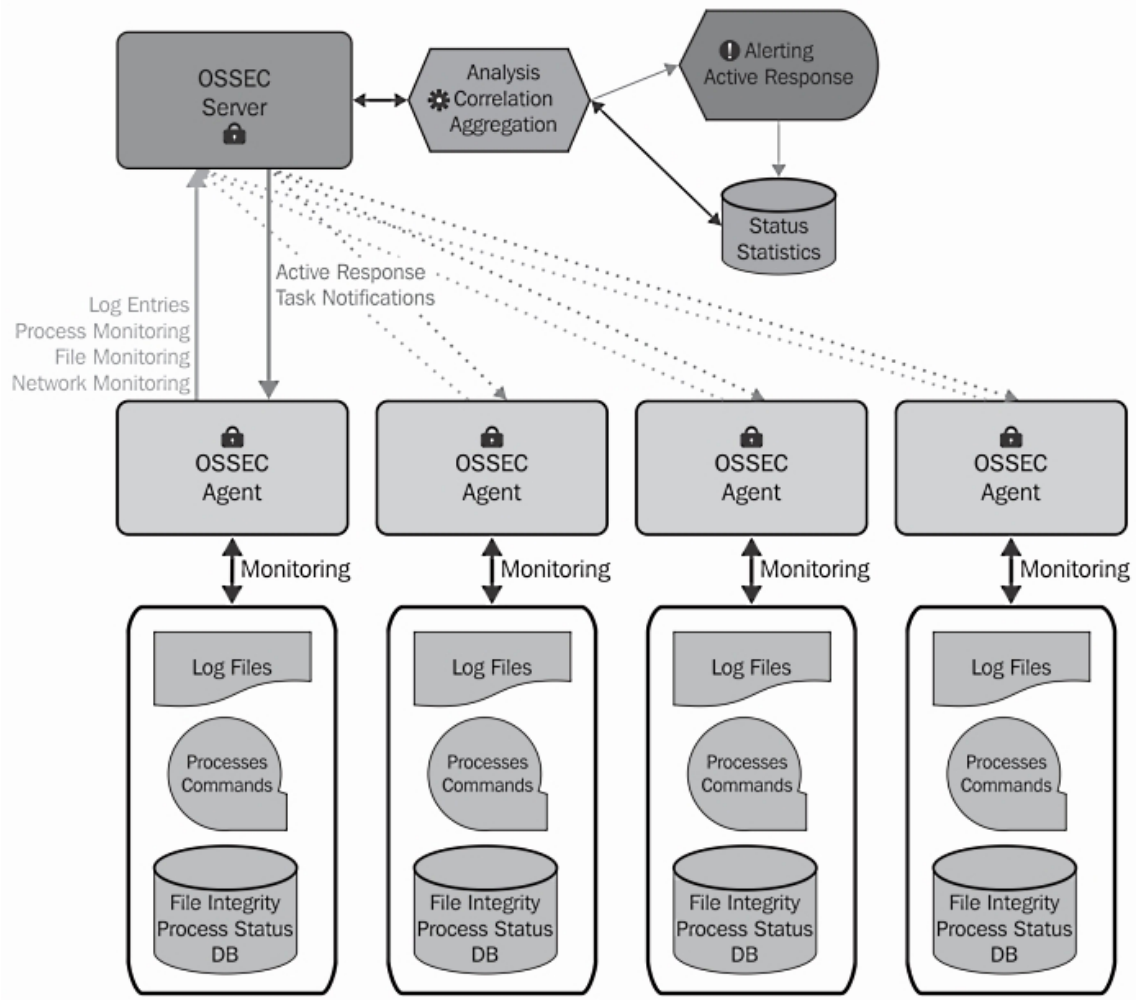
Järjestelmien asennusta ja käyttöä testataan asentamalla ne kahdelle Debian 10 Buster Linux-jakelua käyttävälle virtuaalipalvelimelle, agenttiohjelmisto ja hallintapalvelin erillisille palvelimille. Järjestelmistä asennetaan uusin saatavilla oleva versio, joko paketinhallintajärjestelmän kautta tai lähdekoodista kääntämällä. Kaikki asennukset suoritetaan pääkäyttäjätunnusta käyttäen.

3.1 Ossec

Ossec (Open Source Security Event Correlator [9]) on avoimeen lähdekoodiin pohjautuva isäntäpohjainen tunkeutumisenhavaitsemisjärjestelmä, josta on saatavissa myös yhdysvaltalaisen Atomicorpin tarjoama laajempi kaupallinen versio. Maksullinen versio sisältää esimerkiksi selainpohjaisen käyttöliittymän toisin kuin ilmainen versio, joka tukee vain komentorivipohjaista hallintaa. [10] Ossecin oman verkkosivuston mukaan Ossec on maailman käytetyin isäntäpohjainen tunkeutumisenhavaitsemisjärjestelmä [11]. Ensimmäinen versio on tehty vuonna 2003 [12].

Ossecin keskeisiä ominaisuuksia ovat reaaliaikainen **lokitapahtumien** valvonta ja analysointi, reaaliaikainen tiedostojen ja Windowsin rekisterin **eheyden** valvonta, **haittaohjelmien** havaitseminen analysoimalla prosesseja ja tiedostoja sekä **aktiiviset vastatoimet** tunkeutumisen estämiseksi [13]. Ossecin arkkitehtuuri koostuu keskitetystä hallintapalvelimesta ja erillisestä valvottaville isäntäkoneille asennettavasta agentista. Ossec pystyy tekemään eheystarkistuksia myös ilman asennettua agenttiohjelmistoa, mikä mahdollistaa esimerkiksi palomuurien ja reitittimien valvonnan. [14]

Kuva 1 havainnollistaa Ossecin arkkitehtuuria. Agentit keräävät valvottavat tiedot ja välittävät ne hallintapalvelimelle, joka analysoi ja tallentaa tiedot. Hallintapalvelin voi myös käskeä agentteja suorittamaan vastatoimia.



Kuva 1. *Ossecin arkkitehtuuri [9].*

Agenttikoneilla toimiva ohjelma *ossec-agentd* välittää kerätyt tiedot hallintapalvelimella ajettavalle ohjelmalle *ossec-remoted*. Muita hallintapalvelimella tavallisesti ajettavia ohjelmia ovat agenttiyhteyksiä valvova ja vanhat lokit pakkaava *ossec-monitord*, lokitapahtumat vastaanottava ja analysoiva *ossec-analysisd* sekä sähköpostihälytykset lähettävä *ossec-maild*. Sekä agenttikoneilla että hallintapalvelimella toimivat lokitapahtumat keräävä *ossec-logcollector*, tiedostojen eheyttä tarkkaileva *ossec-syscheckd* ja aktiivisia vastatoimia suorittava *ossec-execd*. [15]

Ossecin agenttiohjelmisto tukee useimpia käyttöjärjestelmiä, kuten Linux, OpenBSD, FreeBSD, MacOS, Solaris ja Windows [13]. Hallintapalvelin toimii vain Unix-pohjaisissa järjestelmissä [9]. Paketinhallintajärjestelmän kautta asennettava Ossec on saatavilla Amazon Linux, CentOS, Debian, Fedora, Kali Linux, OpenSUSE, Red Hat Enterprise Linux ja Ubuntu Linux-jakeluille Atomicorpin ylläpitämästä repositoriosta [16].

3.1.1 Asennus

Ossec on mahdollista asentaa Linux-ympäristöön asennuspaketista pakettihallintajärjestelmää käyttäen tai hyödyntäen komentotulkissa toimivaa asennuskomentosarjaa, joka kääntää ohjelmiston lähdekoodista [14]. Tätä työtä varten Ossecin hallintapalvelin asennettiin yhdelle virtuaalipalvelimelle ja agentti toiselle virtuaalipalvelimelle hyödyntäen Atomicorpin repositorion valmiita paketteja [16]. Sekä hallintapalvelimesta että agentista asennettiin uusin saatavilla ollut versio 3.6.0, joka on julkaistu 14.2.2020 [17]. Asennuksessa käytetyt komennot on esitetty liitteessä A.

Tiedostot asentuvat pääosin hakemistoon */var/ossec*. On huomattava, että agenttiohjelmiston sisältävää pakettia *ossec-hids-agent* ei ole tarkoitus asentaa samalle isäntäkoneelle, jolle hallintapalvelinohjelmiston sisältävä paketti *ossec-hids-server* asennetaan. Hallintapalvelimen paketti sisältää myös paikallisen agenttitoiminallisuuden.

Asennuksen jälkeen hallintapalvelimelle lisätään tieto uudesta agentista. Se tehdään käyttäen ohjelmaa */var/ossec/bin/manage_agents*, joka ajetaan ensin hallintapalvelimella. Ohjelmalle syötetään agentin isäntänimi, IP-osoite sekä ID-numero. Ohjelma luo agentille jaetun salaisen avaimen, joka voidaan tulostaa näytölle. [9]

Seuraavaksi sama ohjelma */var/ossec/bin/manage_agents* ajetaan agenttikoneella. Hallintapalvelimella luotu avain syötetään ohjelmalle, joka tallentaa avaimen agenttikoneelle. Avainten hallinta palvelimella ja agenteilla on myös mahdollista automatisoida käyttäen ohjelmia *ossec-authd* ja *agent-auth*. [9]

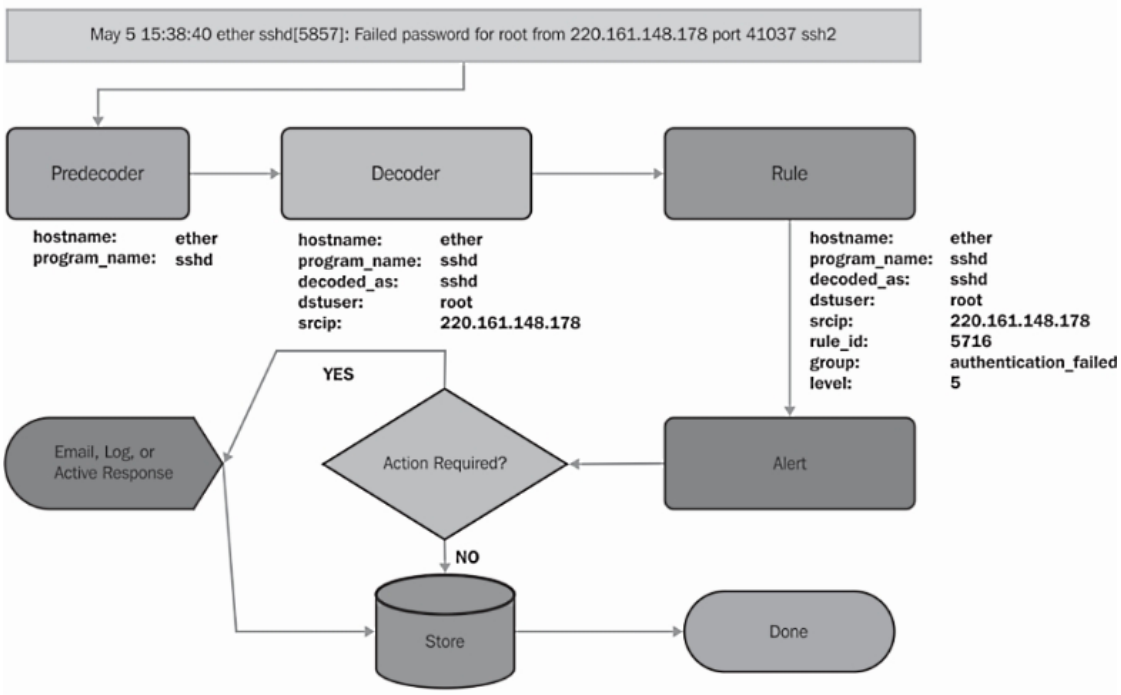
Agenttikoneelle tulee myös lisätä hallintapalvelimen IP-osoite ja hallintapalvelimelle syöttää sähköpostiasetukset hälytyksiä varten. Molemmat muutokset tehdään tiedostoon */var/ossec/etc/ossec.conf*. [9] Lopuksi sekä hallintapalvelimella että agenttikoneella ajetaan seuraava käynnistyskomento:

```
/var/ossec/bin/ossec-control start
```

3.1.2 Lokitapahtumien valvonta

Ossecin lokitapahtumienvalvonta pystyy seuraamaan reaaliajassa paikallisen isäntäkoneen lokitiedostoja, Windowsin tapahtumalokia sekä vastaanottamaan verkon yli syslog-tapahtumia. Kaikilla valvottavilla isäntäkoneilla ajettava ohjelma *ossec-logcollector* kerää lokitapahtumat. [14] Vain hallintapalvelimella ajettava ohjelma *ossec-analysisd* vastaanottaa lokitapahtumat, analysoi ne vertaamalla niitä Ossecin sääntötiedostoihin ja luo tarvittaessa hälytyksiä [15]. Analysointi koostuu dekoodaus-, suodatus- ja luokitteluvaiheista [14].

Kuva 2 esimerkissä esitetään, kuinka Ossec analysoi lokitapahtuman, joka on aiheutunut SSH-yhteydelle annetusta väärästä salasanasta. Eri vaiheissa lokitapahtumasta poimitaan sääntöjen käyttämiä tietoja, kuten IP-osoite ja ohjelman nimi, ja etsitään tapahtumaan täsmääviä sääntöjä. Jos sopiva sääntö löytyy, muodostetaan hälytys. Jos hälytyksen vakavuus vastaa ennalta asetettuja kriteerejä, voidaan esimerkiksi lähettää sähköpostihälytys tai suorittaa vastatoimi, kuten IP-osoitteen estäminen palomuurissa. Lopuksi hälytyksen tiedot tallennetaan hallintapalvelimelle tilastollista raportointia ja myöhempää analyysiä varten. [9]



Kuva 2. *Esimerkki lokitapahtuman analysoinnin vaiheista Ossecissa [9].*

Ossec voi myös ajaa käyttäjän määrittämiä komentoja ja analysoida komentojen tulosteita lokitiedostojen tavoin samanlaisia sääntöjä käyttäen. Esimerkiksi komennon *df* tulostama levytilan käyttöaste olisi mahdollista lokisääntöjen avulla analysoida ja luoda hälytys, jos levytila on täyttymässä. Komentojen kokonaisia tulosteita voidaan myös verrata aiempien suorituskertojen tulosteisiin ja muodostaa hälytys, jos tulosteet eroavat toisistaan. Vertaamalla esimerkiksi komennon

```
netstat -nltu
```

tulosteita keskenään on mahdollista huomata, jos järjestelmässä avoinna olevat verkkoportit muuttuvat. [9]

Ossecin lokitapahtumienanalysoinnissa käyttämät säännöt sijaitsevat XML-muotoisissa tiedostoissa hallintapalvelimen hakemistossa */var/ossec/rules*. Ossec sisältää suuren

määrän valmiita sääntöjä. Säännöt voivat periytyä toisistaan ja muodostaa siten hierarkioita. Ossec myös ylläpitää tilatietoja säännöille, mikä mahdollistaa säännöt, jotka vaativat tietyn määrän toistoja määrätyn ajan sisällä. [9]

3.1.3 Tiedostojen eheyden valvonta

Ossec valvoo tiedostojen eheyttä seuraamalla muutoksia tiedostojen sisällössä, koossa, aikaleimoissa, omistajassa ja oikeuksissa. Sisällön muutoksien havaitsemiseen käytetään MD5- tai SHA1-tiivisteitä. Valvonnasta kullakin isäntäkoneella vastaa ohjelma *ossec-syscheckd*. [14]

Tavallisesti agentit suorittavat eheydentarkistuksen ja lähettävät tulokset hallintapalvelimelle säännöllisesti, esimerkiksi kuuden tunnin välein. Ossec mahdollistaa lisäksi käyttäjän määrittämien tiedostojen reaaliaikaisen eheydenvalvonnan. Hälytyksiin voidaan sisällyttää tekstitiedostoihin tehtyjen muutosten sisältö. Valvottaviin hakemistoihin luotavista uusista tiedostoista on myös mahdollista saada välitön hälytys. [14]

Ossecin oletusasetuksilla yhden tiedoston muutoksista lähetetään enintään kolme hälytystä, jotta usein muuttuvat tiedostot eivät aiheuta jatkuvia turhia hälytyksiä. Tämän ominaisuuden voi kuitenkin kytkeä pois päältä, jolloin kaikki muutokset aiheuttavat aina hälytyksen. [9]

3.1.4 Haittaohjelmien havaitseminen

Ossec etsii isäntäkoneilta haittaohjelmia säännöllisin väliajoin. Haittaohjelmien etsinnän suorittaa sama ohjelma *ossec-syscheckd*, joka valvoo tiedostojen eheyttä. [9] Ossec sisältää valmiit tunnistetietokannat useille haittaohjelmille.

Haittaohjelmientunnistus etsii seuraavia asioita:

1. Rootkit-ohjelmia ja näiden muokkaamia tiedostoja tunnetuista sijainneista.
2. Hakemistosta */dev* sinne kuulumattomia tiedostoja.
3. Tiedostoja, joissa on epätavallisia oikeuksia, kuten SUID-bitti asetettuna.
4. Piilotettuja prosesseja, jotka eivät näy komennolla *ps*.
5. Piilotettuja avoimia verkkoportteja, jotka eivät näy komennolla *netstat*.
6. Verkkosovittimia, jotka on asetettu valikoimattomaan tilaan. [14]

Haittaohjelmientunnistusta voidaan laajentaa lisäsäännöillä, jolloin sitä voidaan käyttää myös laajempaan järjestelmän auditointiin. Sillä voidaan valvoa esimerkiksi kriittisten

palveluiden asetustiedostojen oikeellisuutta tai havainnoida organisaation tietoturvapolitiikan poikkeamia. [9]

3.1.5 Aktiiviset vastatoimet

Ossecin aktiiviset vastatoimet perustuvat agenttikoneilla tai hallintapalvelimella ajettaviin komentosarjoihin ja ohjelmiin. Sääntötiedostoissa määritetään eri tilanteissa suoritettavat vastatoimet. Vastatoimet suorittaville komentosarjoille ja ohjelmille välitetään parametreilla tieto siitä, onko kyseessä vastatoimen asennus vai purku. Parametreissa välitetään lisäksi esimerkiksi hälytyksen aiheuttanut IP-osoite ja mahdollinen käyttäjätunnus. Vastatoimien suorituksesta vastaa ohjelma *ossec-execd*. [9]

Vastatoimet voidaan asentaa määräajaksi, jonka jälkeen ne puretaan. Jos sama tunkeutumisyritys havaitaan uudestaan, voidaan vastatoimi asentaa seuraavalla kerralla pidemmäksi ajaksi. Esimerkiksi virheellisten SSH-kirjautumisyritysten jälkeen voidaan palomuurissa estää käytetty IP-osoite ensin lyhyeksi ajaksi. Jos virheelliset kirjautumisyritykset samasta IP-osoitteesta vielä jatkuvat, voidaan IP-osoite estää palomuurissa pidemmäksi ajaksi. Ossec sisältää valmiin komentosarjan muun muassa Linuxin iptables-palomuuria varten. Vastatoimikomentosarjat sijaitsevat hakemistossa */var/ossec/active-response/bin*. [9]

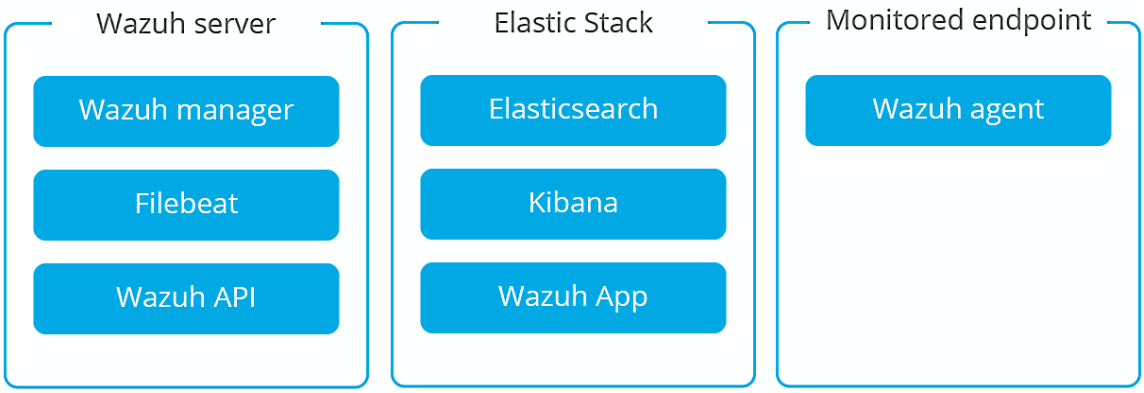
3.2 Wazuh

Wazuh on alun perin Ossecin lähdekoodiin pohjautuva isäntäpohjainen tunkeutumisenhavaitsemisjärjestelmä. Myöhemmin Wazuhiin on integroitu myös avoimeen lähdekoodiin pohjautuvat järjestelmät Elastic Stack ja OpenScap. Elastic Stackia käytetään lokitietojen käsittelyyn ja OpenScapia järjestelmäasetusten tarkistuksiin. [18] Wazuhin ensimmäinen versio on julkaistu vuonna 2015 [19].

Wazuh sisältää kaikki Ossecin merkittävät ominaisuudet, lukuisia parannuksia niihin ja joitakin uusia ominaisuuksia. Ossecin tavoin Wazuh kerää ja analysoi **lokitapahtumia**, valvoo tiedostojen ja Windowsin rekisterin **eheyttä**, tunnistaa **haittaohjelmia** ja suorittaa **aktiivisia vastatoimia**. Osseciin verrattuna uusia ominaisuuksia ovat muun muassa Wazuhin Elastic Stack -integraation mahdollistama selainpohjainen **käyttöliittymä**, **ohjelmistohaavoittuvuuksien** ja tietoturvapolitiikkojen tarkistukset, **inventaarion** muodostus agenttikoneiden laitteistosta ja ohjelmista, Amazon AWS ja Microsoft Azure -pilviympäristöjen valvonta sekä Docker-konttien valvonta. [18] Jotkin näistä Wazuhin ominaisuuksista ovat saatavissa myös Osseciin sen maksullisessa versiossa [10].

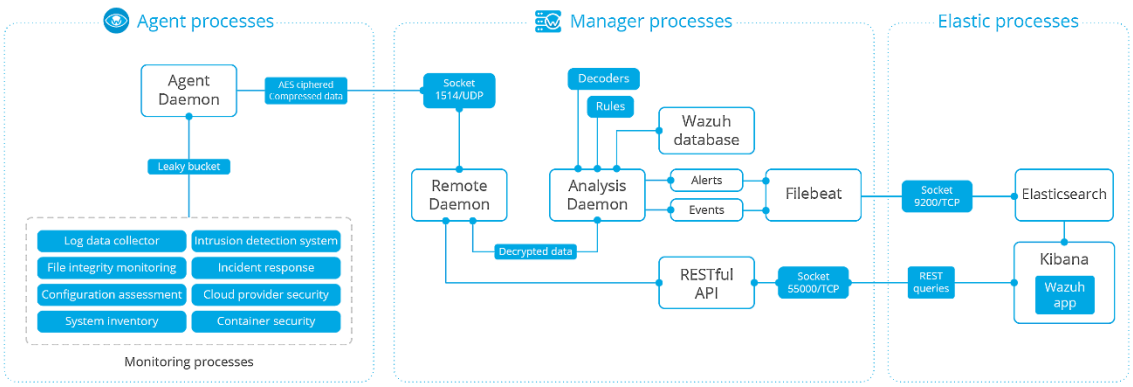
Wazuh tukee Ossecia monipuolisempia sääntöjä lokitapahtumia analysoitaessa ja tarkempia muutostietoja tiedostojen eheyttä valvottaessa [18]. Haittaohjelmien havaitseminen ja aktiiviset vastatoimet toimivat samoin kuin Ossecissa. Näitä käsiteltiin alaluvuissa 3.1.4 ja 3.1.5.

Wazuhin arkkitehtuuri koostuu keskitetystä hallintapalvelimesta, yhdestä tai useammasta Elastic Stack -palvelimesta ja valvottaville isäntäkoneille asennettavasta agentista. Kuva 3 esittää arkkitehtuuriin kuuluvat ohjelmat. Hallintapalvelin sisältää palvelinohjelman *Wazuh manager* lisäksi myös REST-rajapinnan tarjoavan ohjelman *Wazuh API* sekä Elastic Stackiin kuuluvan ohjelman *Filebeat*. Elastic Stack -palvelimilla toimivat ohjelmat *Elasticsearch*, *Kibana* sekä *Wazuh App*. Alle 50 agentista koostuvissa ympäristöissä hallintapalvelimen ja Elastic Stackin ohjelmat on mahdollista asentaa samalle palvelinkoneelle. Agenttikoneilla toimii ohjelma *Wazuh agent*. Ossecin tavoin Wazuh pystyy valvomaan myös palomuureja ja muita laitteita, joihin agenttia ei ole mahdollista asentaa. [18]



Kuva 3. *Wazuhin arkkitehtuuriin kuuluvat ohjelmat [18].*

Kuva 4 havainnollistaa eri ohjelmien välisiä yhteyksiä. REST-rajapinnan tarjoava *Wazuh API* on vaatimus useille selainpohjaisen käyttöliittymän toiminnoille. *Filebeat* siirtää hälytykset ja tapahtumat Elastic Stack -palvelimille. *Elasticsearch* indeksoi, tallentaa ja analysoi tapahtumat. *Kibana* tuottaa selainkäyttöliittymän. *Wazuh App* hakee REST-rajapinnan kautta asetus- ja tilatietoja ja toimii käyttöliittymän osana. [18]



Kuva 4. *Wazuhin ohjelmien väliset yhteydet [18].*

Ossecista poiketen Wazuhin ilmainen versio sisältää kaikki järjestelmän ominaisuudet, myös selainpohjaisen käyttöliittymän. Wazuhin kaupallinen tarjonta koostuu valmiiksi asennetusta pilvipohjaisesta hallintapalvelimesta [20]. Lisäksi saatavilla on tuki- ja koulutuspalveluja [21].

Ossecin tavoin Wazuhin agenttiohjelmisto tukee useita käyttöjärjestelmiä, kuten Linux, BSD-järjestelmät, MacOS, Solaris ja Windows. Hallintapalvelimen asennus vaatii Unix-pohjaisen käyttöjärjestelmän. Wazuhin omasta pakettirepositoriosta Wazuh on asennettavissa Amazon Linux, CentOS, Debian, Fedora, OpenSUSE, Oracle Linux, Red Hat Enterprise Linux, SUSE ja Ubuntu Linux-jakeluille. [18]

3.2.1 Asennus

Wazuhin hallintapalvelin ja agentti voidaan asentaa Linux-ympäristöön asennuspaketista pakettihallintajärjestelmää käyttäen tai hyödyntäen komentotulkissa toimivaa asennuskomentosarjaa, joka kääntää ohjelmiston lähdekoodista. Elastic Stackin ohjelmien asentamiseen Wazuhin asennusohje suosittelee asennuspakettiin pohjautuvaa asennusta. [18] Tätä työtä varten hallintapalvelin ja Elastic Stackin ohjelmat asennettiin yhdelle virtuaalipalvelimelle ja agentti toiselle virtuaalipalvelimelle. Pakettiasennuksissa hyödynnettiin Wazuhin virallista repositoriota. Asennuksessa käytettiin Wazuhin uusinta saatavissa ollutta versiota 3.12.2, joka on julkaistu 9.4.2020 [19]. Elastic Stackista asennettiin asennusohjeen suosittelema versio 7.6.2 [18]. Asennuksessa käytetyt komennot on esitetty liitteessä B.

Wazuhin tiedostot asentuvat pääosin hakemistoon */var/ossec*. On huomattava, että agenttiohjelmiston sisältävää pakettia *wazuh-agent* ei ole tarkoitus asentaa samalle isäntäkoneelle, jolle hallintapalvelimen paketti *wazuh-manager* asennetaan. Hallintapalvelimen paketti sisältää myös paikallisen agenttitoiminnallisuuden.

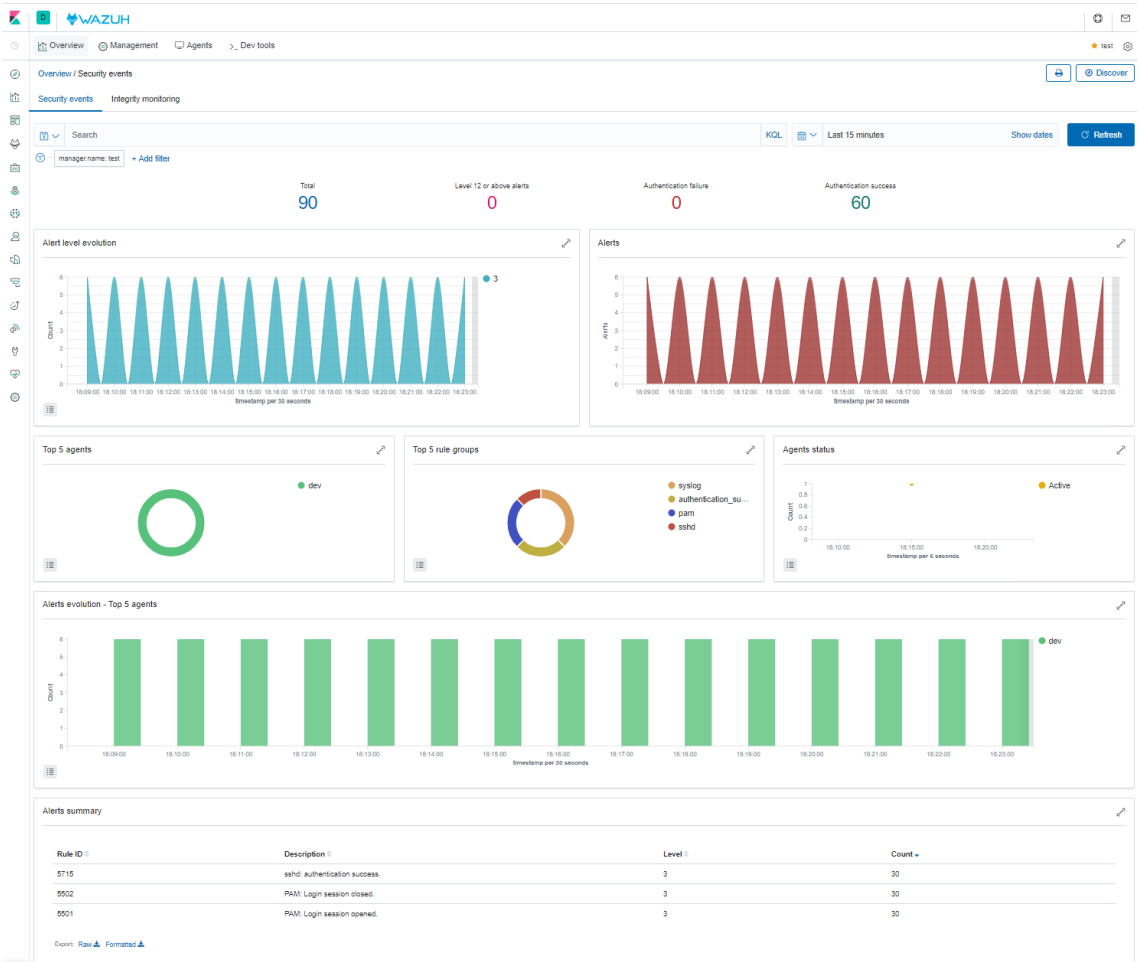
Toisin kuin Ossecissa, Wazuhissa agentteja ei oletusasetuksilla tarvitse erikseen lisätä hallintapalvelimelle. Agentit rekisteröityvät hallintapalvelimelle automaattisesti, kunhan niihin on asetettu palvelimen IP-osoite esimerkiksi asennuksen yhteydessä. Hallintapalvelimelle tulee kuitenkin syöttää sähköpostiasetukset tiedostoon */var/ossec/etc/ossec.conf* tai selainkäyttöliittymän kautta. Hallintapalvelimen käyttöliittymä vastaa HTTP-pyyntöihin portissa 5601.

3.2.2 Lokitapahtumien valvonta

Wazuh valvoo reaaliajassa isäntäkoneen lokitiedostoja, Windowsin tapahtumalokia ja verkon yli vastaanottamiaan syslog-tapahtumia. Se sisältää samat lokivalvontaominaisuudet ja -toimintaperiaatteen kuin alaluvussa 3.1.2 käsitelty Ossecin lokivalvonta. Wazuh mahdollistaa kuitenkin Ossecia monipuolisemmat säännöt lokitapahtumia käsiteltäessä. Wazuh sallii lokitapahtuman purun rajattomaan määrään käyttäjän nimeämiä muuttujia, joita voidaan hyödyntää esimerkiksi muotoiltaessa hälytysviestien sisältöä. Wazuh sisältää myös dekooderin JSON-muotoisten lokien käsittelyyn. [18]

Elastic Stack -integraation mahdollistama lokitapahtumien etsintä ja visualisointi selainkäyttöliittymässä on merkittävä parannus Osseciin verrattuna. Tapahtumia voidaan etsiä monipuolisin hakukriteerein ja tallentaa hakukriteerit myöhempää käyttöä varten. Hakutuloksista voidaan muodostaa esimerkiksi pylväsdiagrammeja ja piirakkakuvioita.

Kuva 5 esittää testiympäristön tietoturvatapahtumia visualisoivan näkymän. Wazuhiin liittymätön erillinen järjestelmä avaa testiympäristöön SSH-yhteyksiä minuutin välein, mikä näkyy kuvan diagrammien säännöllisyytenä.

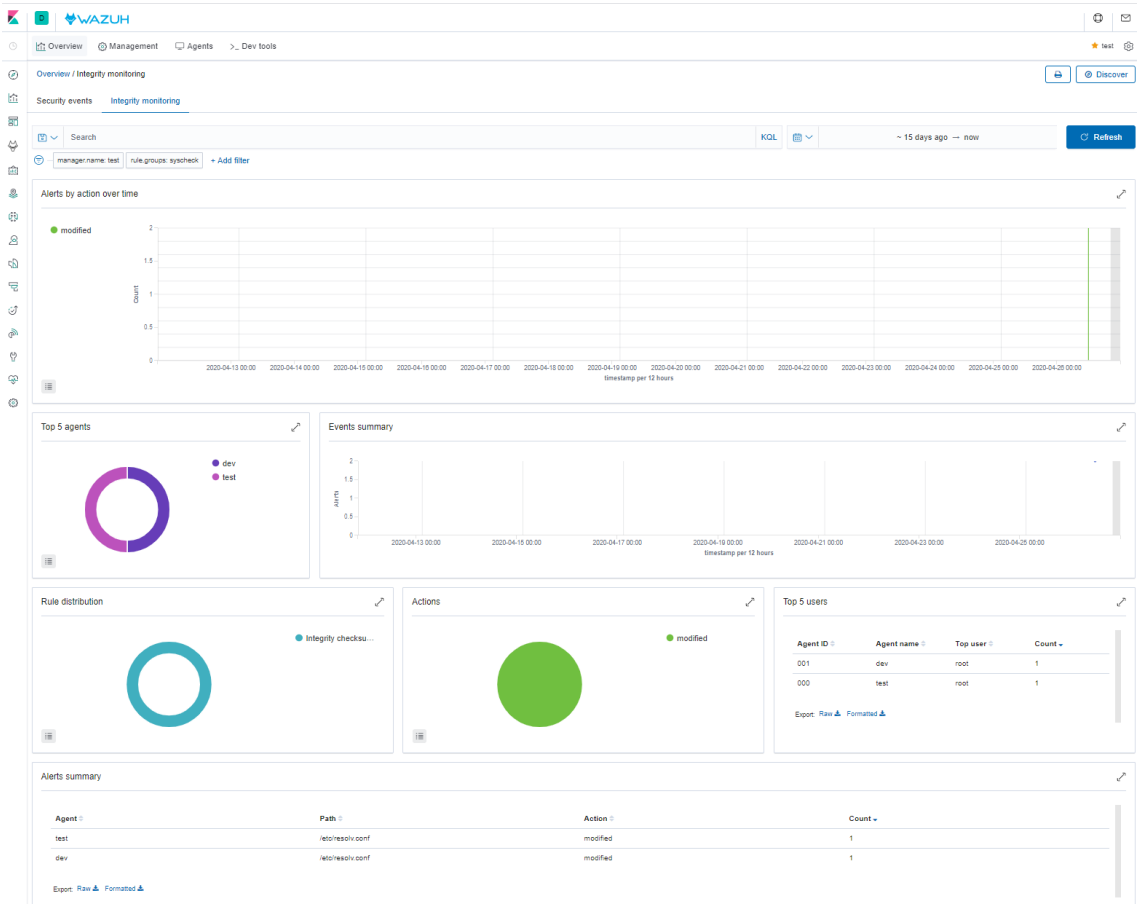


Kuva 5. *Wazuhin käyttöliittymän tietoturvatapahtumanäkymä.*

3.2.3 Tiedostojen eheyden valvonta

Wazuh seuraa muutoksia tiedostojen sisällössä, koossa, aikaleimoissa, omistajassa ja oikeuksissa samalla tavoin kuin alaluvussa 3.1.3 käsitelty Ossecin tiedostovalvonta. Sisällön muutoksien havaitsemiseen voidaan käyttää MD5- ja SHA1-tiivisteiden lisäksi SHA256-tiivisteitä. Toisin kuin Ossec, Wazuh seuraa ja raportoi myös, mikä käyttäjätunnus ja ohjelma tai prosessi aiheutti muutoksen tiedostossa. Tämän ominaisuuden käyttö Linux-ympäristössä vaatii ohjelman *auditd* asentamista agenttikoneille. [18]

Selainkäyttöliittymä sisältää yhteen näkymään koostettuna kaikkien agenttien viimeisimmät tiedostomuutokset. Kuva 6 esittää testiympäristön eheydenvalvonnan koostenäkymän. Valitsemalla hiirellä koostenäkymän taulukoista esimerkiksi tietyn tiedoston polun, saadaan näkyviin kooste vain kyseistä tiedostoa koskevista muutoksista kaikilla agenttikoneilla.



Kuva 6. *Wazuhin käyttöliittymän eheydenvalvontanäkymä.*

3.2.4 Ohjelmistohaavoittuvuuksien havaitseminen

Wazuh pystyy tunnistamaan agenttikoneille asennettujen ohjelmistojen haavoittuvuuksia vertaamalla niitä ajantasaisiin haavoittuvuustietokantoihin. Agentit lähettävät säännöllisesti hallintapalvelimelle listan agenttikoneille asennetuista ohjelmista. Hallintapalvelin vertaa näitä listoja tunnettuihin haavoittuvuuksiin ja muodostaa hälytyksen, jos haavoittuvuus löytyy. Hallintapalvelin on mahdollista asettaa hakemaan uusimmat haavoittuvuustietokannat automaattisesti määrävälein. [18]

Wazuhin käyttämät haavoittuvuustietokannat ovat

- Red Hat Security Data -kanta Amazon Linux, Red Hat Enterprise Linux ja CentOS Linux-jakeluille
- Debian Security Bug Tracker -kanta Debian Linux-jakelulle
- Canonical Ubuntu CVE Tracker -kanta Ubuntu Linux-jakelulle
- National Vulnerability Database -kanta Windowsille [18].

Samoja haavoittuvuustietokantoja on mahdollisuus käyttää myös muiden yhteensopivien Linux-jakelujen kanssa. Esimerkiksi Red Hat Security Data -kanta on mahdollista ottaa käyttöön Red Hat Enterprise Linux -jakeluun perustuvan Oracle Linux -jakelun kanssa. [18]

Agenttikoneille asennettujen ohjelmien listauksien lähetys hallintapalvelimelle on osa laajempaa inventaariotoiminnallisuutta. Ohjelmien lisäksi agenttikoneet raportoivat tietoja niiden käyttämästä laitteistosta, käyttöjärjestelmästä, verkkoliitynnöistä, avoimista verkkoporteista ja käynnissä olevista prosesseista. Windows-agentit raportoivat lisäksi niihin asennetut Windows-päivitykset. [18]

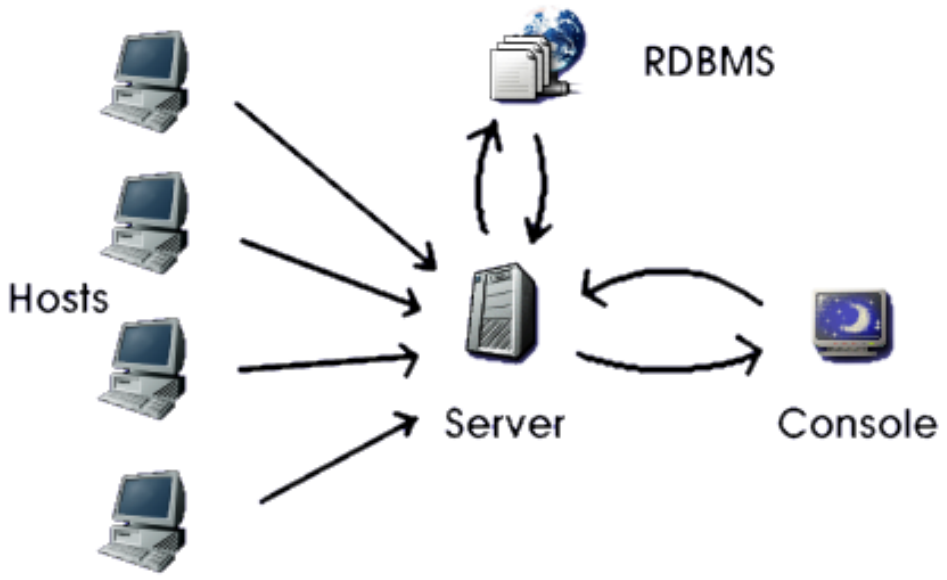
3.3 Samhain

Samhain on avoimeen lähdekoodiin pohjautuva isäntäpohjainen tunkeutumisenhavaitsemisjärjestelmä [22]. Sen selainpohjainen käyttöliittymä, Beltane II, on kuitenkin maksullinen. Ilmaisversiona on saatavilla käyttöliittymän vanha versio, joka on hitaampi ja sisältää vähemmän ominaisuuksia. [23] Samhainin ensimmäinen saatavilla oleva versio on julkaistu vuonna 2001 [24].

Samhainin keskeisiä ominaisuuksia ovat **lokitapahtumien** valvonta ja analysointi sekä reaaliaikainen tiedostojen **eheyden** valvonta. Se pystyy myös havaitsemaan **haittaohjelmia** yleisellä tasolla käynnissä olevia prosesseja ja avoimia verkkoportteja valvomalla, mutta ei sisällä haittaohjelmien tunnistetietoja. [25]

Samhain voi suorittaa **aktiivisia vastatoimia** komentosarjoja tai ohjelmia ajamalla. Ajettavalle ohjelmalle annetaan syötteenä Samhainin muotoilema lokiviesti [26]. Toisin kuin Ossecissa ja Wazuhissa, Samhainin mukana ei toimiteta valmiita vastatoimikomentosarjoja. Vastatoimien toteutus jääkin kokonaan käyttäjän vastuulle.

Valvottaville agenttikoneille asennettava ohjelma *samhain* lähettää keräämänsä tiedot keskitetylle hallintapalvelimelle, mutta toimii tarvittaessa myös itsenäisenä ohjelmana ilman erillistä hallintapalvelinta. Hallintapalvelimelle asennettava ohjelma *yule* käsittelee agenttien lähettämät tapahtumat. Hallintapalvelin voi tallentaa tapahtumat SQL-tietokantaan. Tuettuja tietokantaohjelmistoja ovat Oracle, MySQL ja PostgreSQL. [25] Selainpohjainen käyttöliittymä Beltane vaatii PHP-tuella varustetun WWW-palvelinohjelmiston asennuksen [23]. Käyttöliittymä myös edellyttää SQL-tietokannalla varustettua hallintapalvelinta [25]. Kuva 7 esittää Samhainin arkkitehtuurin.



Kuva 7. *Samhainin arkkitehtuuri [27].*

Samhainin ohjelmat toimivat POSIX-yhteensopivissa käyttöjärjestelmissä kuten Linux, BSD-järjestelmät, MacOS ja Solaris. Agenttitoiminnallisuus on mahdollista asentaa Windows-koneille käyttäen esimerkiksi Cygwin-ympäristöä. [25] Paketinhallintajärjestelmän kautta asennettava Samhain on saatavissa muun muassa Debian ja Ubuntu Linux-jakeluille jakeluiden omista pakettirepositorioista [28]. Samhainin verkkosivustolla ei ole saatavissa valmiita paketteja, vain lähdekoodi [29].

3.3.1 Asennus

Samhainin käyttöohjeissa asennustapana käytetään lähdekoodista kääntämistä. Sama lähdekoodi sisältää sekä agentti- että palvelintoiminnallisuuden. Komentoriviparametreilla valitaan, käännetäänkö agentti- vai palvelinohjelma. [26] Beltane-käyttöliittymällä on oma lähdekoodipakettinsa, joka käännetään erikseen [23]. Vaikka Beltane on pääosin PHP-kielellä toteutettu, sisältää se myös kääntämistä vaativia C-kielisiä ohjelmia.

Tätä työtä varten käännettiin lähdekoodista palvelinohjelma *yule* ja agenttiohjelma *samhain*. Sekä *yule* että MySQL-yhteensopiva MariaDB-tietokantaohjelmisto asennettiin yhdelle virtuaalipalvelimelle ja agenttiohjelma *samhain* toiselle virtuaalipalvelimelle. Asennuksessa käytettiin Samhainin uusinta saatavilla ollut versiota 4.4.1, joka on julkaistu 27.2.2020 [29]. Asennuksessa käytetyt komennot on esitetty liitteessä C.

Hallintapalvelimelle lisätään tieto kustakin uudesta agenttikoneesta luomalla agentille salasana komennolla

```
yule --gen-password
```

ja tallentamalla agenttikoneen isäntänimi ja salasana tiedoston *yulerc* loppuun [26]. Agenttikone lisätään esimerkiksi seuraavalla komennolla:

```
yule -P salasana | sed 's/HOSTNAME/agenttikoneen isäntänimi/'
>> /opt/samhain/etc/yulerc
```

Agenttikoneella sama salasana tallennetaan ohjelman *samhain* sisälle käyttäen ohjelmaa *samhain_setpwd* [26]. Salasana tallennetaan esimerkiksi seuraavilla komennoilla:

```
cd /opt/samhain/sbin
samhain_setpwd samhain new salasana
mv -f samhain.new samhain
```

Agenttikoneella tulee myös alustaa valvottavien tiedostojen metatiedot sisältävä tietokanta seuraavalla komennolla:

```
samhain -t init
```

Tietokanta sijaitsee tiedostossa *samhain_file*. Eheydenvalvonta vertaa myöhemmin tiedostoja tämän tietokannan sisältöön. Lisäksi tiedostoa *samhainrc* muokkaamalla valitaan, mitkä tapahtumat lähetetään palvelimelle. [26]

Tätä työtä varten yritettiin asentaa myös Beltane-käyttöliittymän vuonna 2016 julkaistu viimeinen maksuton versio 1.0.21 kääntämällä se lähdekoodista. Useimmat käyttöliittymän toiminnot eivät kuitenkaan toimineet, minkä vuoksi Beltanen testaus rajattiin pois työstä. Samhainin verkkosivusto suosittelee käyttämään käyttöliittymän uudempaa maksullista versiota [23].

3.3.2 Lokitapahtumien valvonta

Samhain pystyy valvomaan syslog, Apache HTTP Server, Samba ja pacct -muotoisia lokitiedostoja. Erityisen hyödyllinen ominaisuus on mahdollisuus valvoa käytössä olevia lokitiedostoja niin, että uusien rivien lisäys tiedostojen loppuun sallitaan, mutta vanhojen rivien muokkaus aiheuttaa hälytyksen. Samhain myös tunnistaa esimerkiksi ohjelman *logrotate* suorittaman vanhojen lokitiedostojen kierrätyksen automaattisesti. Vaikka lokitiedosto nimetään uudelleen, turhaa hälytystä ei muodosteta. [26]

Samhainin lokitapahtumienanalysoinnissa käyttämät säännöt sijaitsevat asetustiedostossa *samhainrc* [26]. Toisin kuin Ossecissa ja Wazuhissa, Samhainin mukana ei toimiteta valmiita sääntöjä.

Samhain tukee tilallista lokitapahtumien analysointia. Lokitapahtumia voidaan tallentaa muistiin ja analysoida niitä yhdessä myöhempien lokitapahtumien kanssa. Siten hälytyksen muodostus voi pohjautua useampaan toisiinsa liittyvään lokitapahtumaan. Samhain osaa myös tunnistaa toistuvat samankaltaiset lokitapahtumat. [26]

Samhain voi seurata järjestelmän käyttäjien sisään- ja uloskirjautumisia. Hälytys voidaan muodostaa, jos käyttäjä kirjautuu sisään

- uudesta osoitteesta, josta hän ei ole aiemmin kirjautunut
- esiasetettujen kellonaikojen ulkopuolella, kuten yöaikaan
- kellonaikaan, joka poikkeaa tilastollisesti käyttäjän aiemmista kirjautumisajoista. [26]

3.3.3 Tiedostojen eheyden valvonta

Samhain valvoo tiedostojen eheyttä seuraamalla muutoksia muun muassa tiedostojen sisällössä, koossa, aikaleimoissa, omistajassa ja oikeuksissa. Se osaa seurata tiedostojen oikeuksia ja ominaisuuksia laajemmin kuin Ossec ja Wazuh. Sisällön muutoksien havaitsemiseen voidaan käyttää TIGER-, MD5-, SHA1- tai SHA256-tiivisteitä. Samhain voi tallentaa kopion pienien tiedostojen sisällöistä, jolloin sisällön muuttuessa on mahdollista verrata vanhaa ja uutta sisältöä keskenään. [26]

Agentit suorittavat eheydenvalvonnan määrävälein vertaamalla tiedostoja paikallisesti tallennettuun tietokantaan. Myös reaaliaikainen valvonta on mahdollista. Vertailun tulokset voidaan lähettää hallintapalvelimelle. Agenttien käyttämä tietokanta on myös mahdollista ladata hallintapalvelimelta, jolloin useat agentit voivat verrata tiedostojaan samaan tietokantaan. [26]

Samhainin suorittama eheydenvalvonta on monipuolisesti muokattavissa erilaisiin käyttötilanteisiin sopivaksi. Wazuhin tavoin Samhain osaa myös raportoida tiedostomuutoksen tehneen käyttäjätunnuksen. Tämä ominaisuus vaatii ohjelman *auditd* asentamisen. [26]

3.3.4 Haittaohjelmien havaitseminen

Samhain voi verrata komennon *ps* tulostamaa prosessilistausta järjestelmän todellisiin prosesseihin. Jos komennon *ps* listauksesta puuttuu jokin prosessi, voi se olla haittaohjelman piilottama. Myös listaukseen sisältyvät ylimääräiset prosessit, joita ei todellisuudessa ole olemassa, aiheuttavat hälytyksen. Lisäksi on mahdollista määritellä, mitkä prosessit on aina oltava suorituksessa. [26] Samhainin piilotettujen prosessien tunnistus on samankaltainen kuin Ossecissa ja Wazuhissa.

Avoimien verkkoporttien valvonta perustuu ennalta asetettuihin konekohtaisiin porttilistauksiin. Asetustiedostoon määritellään, minkä porttien tulee aina olla avoinna ja mitkä

portit voivat joskus olla avoinna. Avoin portti, jota ei ole syötetty asetustiedostoon, aiheuttaa hälytyksen. Samoin hälytys aiheutuu suljetusta portista, jonka pitäisi asetustiedoston mukaan olla aina avoinna. [26] Toisin kuin Ossec ja Wazuh, Samhain ei osaa etsiä piilotettuja portteja komennon *netstat* tulostetta hyödyntämällä.

Samhain osaa etsiä tiedostoja, joissa on SUID- tai SGID-bitti asetettuna. Tällaiset tiedostot voidaan myös automaattisesti poistaa tai siirtää erilliseen karanteenihakemistoon. [26]

4. YHTEENVETO

Työssä perehdyttiin kolmen isäntäpohjaisen tunkeutumisenhavaitsemisjärjestelmän asennukseen, ominaisuuksiin ja eroihin. Tarkastelussa olivat mukana Ossec, Wazuh ja Samhain. Kaikki vertaillut järjestelmät ovat asennettavissa Linux-ympäristöön ja sisältävät samankaltaisia ominaisuuksia. Järjestelmien merkittävimpiä ominaisuuksia ovat lokitapahtumien valvonta, tiedostojen eheyden valvonta, haittaohjelmien havaitseminen ja aktiiviset vastatoimet. Wazuhin ominaisuuksiin kuuluvat lisäksi muun muassa avoimen lähdekoodin selainkäyttöliittymä ja ohjelmistohaavoittuvuuksien tunnistaminen haavoittuvuustietokantoja käyttäen.

Vertailtujen järjestelmien toiminta perustuu valvottaville isäntäkoneille asennettavaan agenttiohjelmistoon. Wazuh ja Ossec vaativat lisäksi erillisen keskitetyn hallintapalvelinohjelmiston asentamisen. Samhainia voi käyttää joko hallintapalvelimen kanssa tai ilman. Wazuh ja Ossec voivat valvoa myös palomuureja ja muita laitteita, joihin ei ole mahdollista asentaa agenttiohjelmistoa.

Järjestelmien asennuksen helppoudessa oli selviä eroja. Ossecin agentin ja hallintapalvelimen asennus valmiista paketeista oli vaivatonta. Wazuhin asennus vei enemmän aikaa johtuen sen vaatimuksesta asentaa hallintapalvelimelle useita erillisiä ohjelmia ja Elastic Stack -järjestelmä. Toisaalta agenttien liittäminen Wazuhin hallintapalvelimelle oli helpompaa kuin Ossecissa. Työläin asennus oli Samhainissa, jonka uusin versio oli saatavissa vain lähdekoodin muodossa. Lähdekoodi piti kääntää erikseen agenttikoneella ja hallintapalvelimella.

Ominaisuuksiltaan monipuolisin järjestelmistä on Wazuh, joka pohjautuu alun perin Ossecin lähdekoodiin. Wazuh sisältää kaikki Ossecin merkittävät ominaisuudet ja monia uusia ominaisuuksia. Vähiten ominaisuuksia on Samhainissa.

Kaikki vertaillut järjestelmät osaavat valvoa lokitiedostoja ja verrata lokitapahtumia valmiisiin sääntöihin. Lisäksi ne voivat ylläpitää tilatietoja lokitapahtumille, mikä mahdollistaa esimerkiksi lokitapahtumien toistojen määrään perustuvat säännöt. Monipuolisin lokivalvonta on Wazuhissa ja Ossecissa, joiden mukana myös toimitetaan paljon valmiita sääntöjä. Wazuh mahdollistaa hieman Ossecia laajemmat säännöt. Samhainissa on suppeimmat lokivalvontaominaisuudet, eikä sen mukana toimiteta valmiita sääntöjä.

Jokaisessa vertaillussa järjestelmässä on kattava tiedostojen eheyden valvonta, joka voi lähettää esimerkiksi sähköpostihälytyksen epäilyttävästä tiedostomuutoksesta. Tiedostoista voidaan valvoa muun muassa sisältöä, aikaleimoja ja oikeuksia. Wazuh ja

Samhain voivat sisällyttää hälytykseen muokkauksen tehneen käyttäjätunnuksen. Kaikki järjestelmät kykenevät myös reaaliaikaiseen tiedostojen valvontaan.

Ossec ja Wazuh osaavat etsiä haittaohjelmia sisältämiensä tunnistetietokantojen avulla. Lisäksi ne tunnistavat järjestelmän anomalioita, kuten epätavallisia tiedosto-oikeuksia, piilotettuja prosesseja ja verkkoportteja tai valikoimattomassa tilassa olevia verkkosovittimia. Samhain ei sisällä haittaohjelmien tunnistetietoja, mutta osaa etsiä piilotettuja prosesseja ja verrata avoimia verkkoportteja käyttäjän määrittämiin porttilistauksiin.

Kaikki vertaillut järjestelmät voivat suorittaa aktiivisia vastatoimia komentosarjoja tai ohjelmia ajamalla. Wazuh ja Ossec sisältävät valmiin komentosarjan esimerkiksi Linuxin iptables-palomuuria varten. Ne myös mahdollistavat suoritetun vastatoimen automaattisen purun tietyn ajan jälkeen.

Wazuh soveltuu vertailluista järjestelmistä parhaiten monipuoliseen käyttöön. Ossec voi olla parempi vaihtoehto yksinkertaisemman asennuksen ja arkkitehtuurin vuoksi, jos sen ominaisuudet riittävät aiottuun käyttötarkoitukseen. Samhainin käyttö voi olla perusteltua, jos pääasiallisena tarpeena on tiedostojen eheyden valvonta.

Vertaillut järjestelmät käyttävät havaitsemismenetelmänä pääasiassa tunnistetietoon pohjautuvaa havaitsemista, johon on paikoin yhdistetty tilatietoja. Anomaliaan pohjautuvan havaitsemisen käyttö on vähäistä, eikä mikään järjestelmistä mahdollista automaattista profiilin luontia isäntäkoneen tavanomaista toimintaa seuraamalla.

LÄHTEET

[1] C. Day, Intrusion Prevention and Detection Systems, in: J.R. Vacca (ed.), Computer and Information Security Handbook, 3rd ed., Morgan Kaufmann Publishers, Cambridge Massachusetts, USA, 2017, pp. 1011–1025.

[2] K. Scarfone, P. Mell, Guide to Intrusion Detection and Prevention Systems (IDPS), National Institute of Standards and Technology (NIST), Special Publication 800-94, USA, 2007.

[3] J. Pescatore, G. Young, Defining the Next-Generation Firewall, Gartner, ID: G00171540, 2009. Saatavissa (viitattu 29.3.2020): http://img1.custompublish.com/getfile.php/1434855.1861.sqqycbrdwq/Defining+the+Next-Generation+Firewall.pdf

[4] T.M. Chen, Guarding Against Network Intrusions, in: J.R. Vacca (ed.), Computer and Information Security Handbook, 3rd ed., Morgan Kaufmann Publishers, Cambridge Massachusetts, USA, 2017, pp. 149–163.

[5] J.W.G. Stênico, L.L. Ling, Network Traffic Monitoring and Analysis, in: A.-S.K. Pathan (ed.), The State of the Art in Intrusion Prevention and Detection, CRC Press, Boca Raton Florida, USA, 2016, pp. 23–46.

[6] B. Pranggono, K. McLaughlin, Y. Yang, S. Sezer, Intrusion Detection Systems for Critical Infrastructure, in: A.-S.K. Pathan (ed.), The State of the Art in Intrusion Prevention and Detection, CRC Press, Boca Raton Florida, USA, 2016, pp. 115–138.

[7] M. Wood, M. Erlinger, Intrusion Detection Message Exchange Requirements, Internet Engineering Task Force (IETF), 2007. Saatavissa (viitattu 23.3.2020): https://tools.ietf.org/html/rfc4766

[8] V. Jyothsna, V.V. Rama Prasad, A Review of Anomaly based Intrusion Detection Systems, International Journal of Computer Applications, Vol. 28, No. 7, Sep 2011, pp. 26–35. Saatavissa (viitattu 23.3.2020): https://www.ijcaonline.org/volume28/number7/pxc3874730.pdf

[9] B. Lhotsky, Instant OSSEC Host-based Intrusion Detection, Packt Publishing, Birmingham, UK, 2013, 49 p.

[10] OSSEC Products, OSSEC Project Team, 2020. Saatavissa (viitattu 18.4.2020): https://www.ossec.net/products/

[11] OSSEC, OSSEC Project Team, 2020. Saatavissa (viitattu 18.4.2020): https://www.ossec.net/

[12] OSSEC Overview, Atomicorp Inc., 2020. Saatavissa (viitattu 2.5.2020): https://atomicorp.com/about-ossec/

[13] About OSSEC HIDS, OSSEC Project Team, 2020. Saatavissa (viitattu 18.4.2020): https://www.ossec.net/about/

[14] OSSEC 3.6.0 Manual, OSSEC Project Team, 2020. Saatavissa (viitattu 18.4.2020): https://www.ossec.net/docs/docs/manual/index.html

[15] OSSEC 3.6.0 Man pages, OSSEC Project Team, 2020. Saatavissa (viitattu 18.4.2020): https://www.ossec.net/docs/docs/programs/index.html

[16] Atomicorp Archive, Atomicorp Inc., 2019. Saatavissa (viitattu 27.4.2020): https://updates.atomicorp.com/channels/ossec/

[17] OSSEC Downloads, OSSEC Project Team, 2020. Saatavissa (viitattu 18.4.2020): https://www.ossec.net/downloads/

[18] Wazuh 3.12 Docs, Wazuh Inc., 2020. Saatavissa (viitattu 25.4.2020): https://documentation.wazuh.com/3.12/index.html

[19] Wazuh Releases, GitHub Inc., 2020. Saatavissa (viitattu 2.5.2020): https://github.com/wazuh/wazuh/releases

[20] Wazuh Cloud, Wazuh Inc., 2020. Saatavissa (viitattu 25.4.2020): https://wazuh.com/cloud/

[21] Wazuh Services, Wazuh Inc., 2020. Saatavissa (viitattu 25.4.2020): https://wazuh.com/professional-services/

[22] R. Wichmann, The SAMHAIN file integrity / host-based intrusion detection system, Samhain Design Labs. Saatavissa (viitattu 3.5.2020): https://la-samhna.de/samhain/index.html

[23] R. Wichmann, Beltane, Samhain Design Labs. Saatavissa (viitattu 2.5.2020): https://www.la-samhna.de/beltane/

[24] R. Wichmann, Archive of samhain releases, Samhain Design Labs. Saatavissa (viitattu 2.5.2020): https://la-samhna.de/samhain/archive.html

[25] R. Wichmann, The Samhain HIDS: Overview of available features, 2011. Saatavissa (viitattu 2.5.2020): https://la-samhna.de/samhain/samhain_leaf.pdf

[26] R. Wichmann, The Samhain Host Integrity Monitoring System manual, 2019. Saatavissa (viitattu 2.5.2020): https://www.la-samhna.de/samhain/MANUAL-2_4.pdf

[27] R. Wichmann, SAMHAIN: An open-source Host Intrusion Detection System (HIDS), 2011. Saatavissa (viitattu 11.5.2020): https://la-samhna.de/samhain/Samhain.pdf

[28] Search Results for samhain, Packages Search for Linux and Unix, 2020. Saatavissa (viitattu 3.5.2020): https://pkgs.org/search/?q=samhain

[29] R. Wichmann, The SAMHAIN file integrity / intrusion detection system download, Samhain Design Labs. Saatavissa (viitattu 2.5.2020): https://la-samhna.de/samhain/s_download.html

LIITE A: OSSEC ASENNUSKOMENNOT

Ossecin **hallintapalvelin** asennettiin ensimmäiselle virtuaalipalvelimelle käyttäen seuraavia komentoja:

```
  wget -qO- https://www.atomicorp.com/RPM-GPG-KEY.atomicorp.txt
| apt-key add -
  echo "deb https://updates.atomicorp.com/channels/ossec/debian buster main"
> /etc/apt/sources.list.d/ossec.list
  apt update
  apt install -y ossec-hids-server
```

Ossecin **agentti** asennettiin toiselle virtuaalipalvelimelle käyttäen seuraavia komentoja:

```
  wget -qO- https://www.atomicorp.com/RPM-GPG-KEY.atomicorp.txt
| apt-key add -
  echo "deb https://updates.atomicorp.com/channels/ossec/debian buster main"
> /etc/apt/sources.list.d/ossec.list
  apt update
  apt install -y ossec-hids-agent
```

Asennukset tehtiin Debian 10 Buster Linux-jakelua käyttäville palvelimille. Asennuksissa käytettiin pääkäyttäjätunnusta.

LIITE B: WAZUH ASENNUSKOMENNOT

Wazuhin **hallintapalvelin** asennettiin ensimmäiselle virtuaalipalvelimelle käyttäen seuraavia komentoja:

```
   # Wazuh Manager asennus
   wget -qO- https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -
   echo "deb https://packages.wazuh.com/3.x/apt/ stable main"
> /etc/apt/sources.list.d/wazuh.list
   apt update
   apt install -y wazuh-manager
   # Wazuh API asennus
   wget -qO- https://deb.nodesource.com/setup_10.x | bash -
   apt install -y nodejs
   apt install -y wazuh-api
   # Filebeat asennus
   wget -qO- https://artifacts.elastic.co/GPG-KEY-elasticsearch
| apt-key add -
   echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
> /etc/apt/sources.list.d/elastic-7.x.list
   apt update
   apt install -y filebeat=7.6.2
   # Filebeat asetukset
   wget     -qO     /etc/filebeat/filebeat.yml     https://raw.githubusercon-
tent.com/wazuh/wazuh/v3.12.2/extensions/filebeat/7.x/filebeat.yml
   # Elasticsearch hälytysmalli
   wget   -qO   /etc/filebeat/wazuh-template.json   https://raw.githubusercon-
tent.com/wazuh/wazuh/v3.12.2/extensions/elasticsearch/7.x/wazuh-template.json
   # Wazuh moduuli
   wget -qO- https://packages.wazuh.com/3.x/filebeat/wazuh-filebeat-0.1.tar.gz
| tar -xz -C /usr/share/filebeat/module
   sed -i 's/YOUR_ELASTIC_SERVER_IP/localhost/' /etc/filebeat/filebeat.yml
   systemctl daemon-reload
   systemctl enable filebeat.service
   systemctl start filebeat.service
```

Elastic Stack asennettiin samalle palvelimelle käyttäen seuraavia komentoja:

```
   # Elasticsearch asennus
   wget -qO- https://artifacts.elastic.co/GPG-KEY-elasticsearch
| apt-key add -
   echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
> /etc/apt/sources.list.d/elastic-7.x.list
   apt update
   apt install -y elasticsearch=7.6.2
   systemctl daemon-reload
   systemctl enable elasticsearch.service
   systemctl start elasticsearch.service
   filebeat setup --index-management -E setup.template.json.enabled=false
   # Kibana asennus
   apt install -y kibana=7.6.2 sudo
   cd /usr/share/kibana/
   sudo     -u     kibana     bin/kibana-plugin     install     https://pack-
ages.wazuh.com/wazuhapp/wazuhapp-3.12.2_7.6.2.zip
   sed     -i     's/#server.host:     "localhost"/server.host:     "0.0.0.0"/'
/etc/kibana/kibana.yml
```

```
echo NODE_OPTIONS=\"--max_old_space_size=2048\" >> /etc/default/kibana
systemctl daemon-reload
systemctl enable kibana.service
systemctl start kibana.service
```

Wazuhin **agentti** asennettiin toiselle virtuaalipalvelimelle käyttäen seuraavia komentoja:

```
  # Wazuh Agent asennus
  wget -qO- https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -
  echo "deb https://packages.wazuh.com/3.x/apt/ stable main"
> /etc/apt/sources.list.d/wazuh.list
  apt update
  WAZUH_MANAGER="hallintapalvelimen IP-osoite" apt install -y wazuh-agent
```

Asennukset tehtiin Debian 10 Buster Linux-jakelua käyttäville palvelimille. Asennuksissa käytettiin pääkäyttäjätunnusta.

LIITE C: SAMHAIN ASENNUSKOMENNOT

Samhainin **hallintapalvelin** ja MariaDB-tietokanta asennettiin ensimmäiselle virtuaalipalvelimelle käyttäen seuraavia komentoja:

```
   wget -qO- http://la-samhna.de/samhain/samhain-current.tar.gz | tar -xz
   tar -xf samhain-4.4.1.tar.gz
   cd samhain-4.4.1
   # käytetään MySQL:n sijasta MariaDB:tä, joka on saatavilla repositoriosta
   apt install -y make gcc libpcre3-dev mariadb-server libmariadb-dev
   ln -s -r /usr/include/mariadb /usr/include/mysql
   sed -i 's/mysqlclient/mariadbclient/' configure
   ./configure MYSQL_LIB_DIR=/usr/lib/x86_64-linux-gnu --prefix=/opt/samhain
--enable-network=server --enable-xml-log --with-database=mysql
--enable-login-watch --enable-mounts-check --enable-logfile-monitor
--enable-process-check --enable-port-check --enable-userfiles
--enable-suidcheck
   make
   # make install -komento keskeytyy virheeseen jos samhainadmin.pl puuttuu
   touch scripts/samhainadmin.pl
   make install
   make install-boot
   # luodaan SQL-tietokanta
   mariadb <sql_init/samhain.mysql.init
   mariadb -e "CREATE USER 'samhain'@'localhost';
GRANT SELECT,INSERT ON samhain.log TO 'samhain'@'localhost';"
```

Samhainin **agentti** asennettiin toiselle virtuaalipalvelimelle käyttäen seuraavia komentoja:

```
   wget -qO- http://la-samhna.de/samhain/samhain-current.tar.gz | tar -xz
   tar -xf samhain-4.4.1.tar.gz
   cd samhain-4.4.1
   apt install -y make gcc libpcre3-dev
   ./configure --prefix=/opt/samhain --enable-network=client
--with-logserver="hallintapalvelimen IP-osoite" --enable-login-watch
--enable-mounts-check --enable-logfile-monitor --enable-process-check
--enable-port-check --enable-userfiles --enable-suidcheck
   make
   make install
   make install-boot
```

Asennukset tehtiin Debian 10 Buster Linux-jakelua käyttäville palvelimille. Asennuksissa käytettiin pääkäyttäjätunnusta.