Linux-pohjaisen järjestelmän analysointi tietoturvapoikkeamatilanteessa

Jarno Virtanen

2024 Laurea

Laurea-ammattikorkeakoulu

Linux-pohjaisen järjestelmän analysointi tietoturvapoikkeamatilanteessa

Jarno Virtanen

Tietojenkäsittelyn koulutus

Opinnäytetyö

12, 2024

Laurea-ammattikorkeakoulu **Tiivistelmä**
Tietojenkäsittelyn koulutus, kyberturvallisuus
Liiketalouden ammattikorkeakoulututkinto (AMK)

Jarno Virtanen

Linux-pohjaisen järjestelmän analysointi tietoturvapoikkeamatilanteessa

Vuosi 2024 Sivumäärä 61

Opinnäytetyö käsittelee Linux-järjestelmän analysointia tietoturvapoikkeamatilanteessa, jossa järjestelmään tai sen osaan on mahdollisesti tunkeuduttu hyökkääjän toimesta.

Opinnäytetyössä esitellään oppaan kaltaisesti tietoturvapoikkeamatilanteessa ensivaiheessa analysoitavat kohteet. Tietojen analysoitiin käytetään Bash -komentotulkkia, jonka avulla selvitetään mistä ja miten tietoturvapoikkeamatilanteen tunnistamiseksi tärkeät tiedot ovat käyttäjän itsensä löydettävissä.

Tiedot on pyritty esittämään mahdollisimman yksinkertaisesti, jotta myös vähemmän Linuxia käyttävä pystyisi toteuttamaan analysointia. Lukijalta oletetaan kuitenkin perusosaamista Linuxista.

Opinnäytetyötä voivat hyödyntää kaikki tahot, joilla on tarvetta analysoida tapahtumia Linux -järjestelmässä.

Asiasanat: linux, kyberturvallisuus, tietoturva

Laurea University of Applied Sciences **Abstract**
Degree Programme in Business Information Technology, Cyber Security
Business Administration

Jarno Virtanen

Analysis of a Linux-based system in the event of a data security breach

Year 2024 Pages 61

The thesis examines the analysis of a Linux-based system in the event of a information security incident, where the system or a part of it may have been compromised by an attacker.

Thesis follows a guidebook like approach, presenting the items to be analyzed in the first stage of an information security incident event. A Bash shell program is used to analyze the data, to find out where and how relevant information for identifying a security incident can be found by the user.

The information has been presented as simply as possible, so that even a less skilled Linux user can perform the analysis. However, the reader is expected to have a basic knowledge of Linux.

The thesis can be used by anyone who needs to analyze events on a linux-based system.

Keywords: linux, cyber security, information security

Sisällys

1 Johdanto

Opinnäytetyössä esitellään oppaan kaltaisesti tärkeimmät tietoturvapoikkeamatilanteen ensivaiheessa analysoitavat kohteet. Tarkoituksena on selvittää mistä ja miten tietoturvapoikkeamatilanteen tunnistamiseksi tärkeät tiedot ovat käyttäjän itsensä löydettävissä.

Linux -tietoturvaosaamisen tarve ei ole koskaan ollut suurempi. Linux-järjestelmät ovat yhä useammin kehittyneiden hyökkäysten kohteena. Hyökkääjät vaihtelevat kansallisista/valtiollisista toimijoista aina järjestäytyneeseen rikollisuuteen. (Wake, T. 2023b.)

Trend Micro:n -raportissa vuodelta 2023, korostetaan Linuxiin kohdistuvien hyökkäysten ja haittaohjelmien merkittävää lisääntymistä. Linuxista on tullut valtavan suosittu käyttöjärjestelmä, niin henkilökohtaisessa käytössä, kuin yritysten ja organisaatioiden palvelinten ja pilvi-infrastruktuurin osana. Raportin tietojen mukaan Linux oli käyttöjärjestelmänä yli 81 prosentissa verkkosivujen taustalla ja on arvioitu, että jopa 90 prosenttia julkisista pilvipalveluista pyörii Linuxilla. Laajentunut käyttöönotto on tehnyt siitä myös merkittävän kohteen erilaisille kyberuhkille ja hyökkäyksille. (Kinger, P., Bharti, S., Oliveira, M. 2023.)

Windows-käyttöjärjestelmän osalta tietoturvaosaamisen ja tietoturvapoikkeamien käsittelyyn liittyen on saatavilla runsaasti tietoa, mutta Linuxin kohdalla vastaavaa tietoa on tarjolla huomattavasti vähemmän. Haittaohjelmien aiheuttamien uhkien lisääntyminen, kehittyneet hyökkäysmenetelmät ja haavoittuvuuksien hyödyntäminen viittaavat siihen, että tarvitaan koulutusta ja taitojen kehittämistä Linux -tietoturvaosaamiseen (Wake, T. 2023b).

Useimpien poikkeamatilanteisiin vastaavien henkilöiden on jossain uransa vaiheessa suoritettava tietojen analysointia manuaalisesti komentoriviä käyttäen (Anson, S. 2020, 194). Poikkeamatilanteen selvittämisen kannalta on tärkeää, että poikkeamatilanteeseen vastaavalla on etukäteen hankittua osaamista ja tietoa työkaluista, joista voi olla hyötyä poikkeamatilanteen selvittämisessä (Cichonski, P., Millar, T., Grance, T. & Scarfone, K. 2012, 43).

Linux-järjestelmään kohdistuvien uhkien lisääntyminen edellyttää tietoon perustuvaa ja ennakoivaa toimintaa. On tärkeää ymmärtää hyökkääjien käyttäytymistä ja pystyä toteuttamaan oikeita toimenpiteitä poikkeamatilanteiden havaitsemiseksi ja niihin reagoimiseksi. (Wake, T. 2023b.)

Linuxissa on useita sisäänrakennettuja työkaluja ja komentoja, joilla saadaan esille tietoja käyttäjän toiminnasta. Lisäksi on tarjolla erilaisia kaupallisia ja avoimen lähdekoodin työkaluja esimerkiksi lokitietojen kokoamiseksi yhteen käyttöliittymään ja näiden tietojen monitorointiin. Tämän kaltaiset ratkaisut hakevat automaatiolla samoja tietoja, jotka käyttäjä voi

hakea manuaalisesti esille. Tietoturva-alan ammattilaisen tulee tuntea järjestelmä ja osata hyödyntää saatavilla olevia työkaluja, pelkkään automaatioon luottamisen sijaan.

2 Menetelmät

Opinnäytetyö toteutettiin toiminnallisena opinnäytetyönä, jonka tarkoituksena oli koota tietoa siitä, miten tunnistaa ja löytää Linux-järjestelmästä merkkejä mahdollisesta tietoturvapoikkeamatilanteesta.

Linuxilla tarkoitetaan opinnäytetyössä Unix -käyttöjärjestelmään pohjautuvaa avoimen lähdekoodin käyttöjärjestelmää.

Tavoitteena oli kehittää vaihe vaiheelta etenevä kirjallinen prosessi tietojen analyysista, jonka tuloksia voivat hyödyntää kaikki tahot, joilla on tarvetta analysoida tapahtumia Linux -käyttöjärjestelmässä.

Tutkimuskysymykset, joihin työssä on pyritty vastaamaan ovat:

1. Miten löytää Linux-järjestelmästä merkkejä tietoturvapoikkeamatilanteesta ja mitä nämä merkit ovat?
2. Mitä työkaluja ja komentoja on käytettävissä näiden tietojen esille saamiseksi?

Tiedonhankinta suoritettiin dokumenttianalyysina keräämällä monipuolisesti aineistoa, ja yhdistelemällä informaatio mahdollisimman selkeäksi ja yksinkertaiseksi kokonaisuudeksi siten, että tiedon avulla on helppo päästä aiheeseen sisälle.

Opinnäytetyö on rajattu tietojen esille saamiseen, alkuvaiheen analyysiin ja tilanteen tunnistamiseen, eikä tarkoituksena ole esitellä tietojen taltiointia tai raportointia.

Opinnäytetyön toiminnallista osuutta varten loin Oracle VM VirtualBox -ohjelmistolla Ubuntu Desktop 24.04 LTS -virtuaalikoneen, jolla testasin, että lähteissä mainitut seikat ovat paikkaansa pitäviä.

Internetistä on ladattavissa lukuisia erilaisia jakeluja, joista useat on optimoitu tiettyihin käyttötarkoituksiin. Ubuntu valikoitui siksi, että se on yleisesti käytössä oleva ja perustuu Debian Linux-jakelupakettiin, joka on yksi kehitetyimmistä ja tunnetuimmista jakeluversioista. Linux-jakelu on täydellinen itsenäinen Linux -käyttöjärjestelmä. Lisäksi Ubuntu mainostaa itseään käytetyimpänä käyttöjärjestelmänä pilviympäristöissä (Ubuntu 2024a).

Tietojen esille saamiseen käytetään Bash -komentotulkkia ja siinä käytettäviä komentoja ja komentoriviohjelmia. Ideana on, että lukija pystyy hyödyntämään komentoja sellaisenaan,

joutumatta asentamaan uusia ohjelmia. Kun työssä asennetaan työkaluja, joita ei toimiteta esiasennettuna, niin myös asennukseen tarvittavat komennot on esitetty.

Työssä pyrittiin käyttämään jakeluriippumattomia komentoja, mutta osa komennoista ja tietojen sijainnista vaihtelee jakelupakettien mukaan. Työn rajaamiseksi keskitytään kuitenkin Ubuntuun.

Tiedot on pyritty esittämään mahdollisimman yksinkertaisesti, jotta myös vähemmän Linuxia käyttävä pystyisi toteuttamaan analysointia. Lukijalta oletetaan kuitenkin perusosaamista Linuxista.

Jos jo ennen tarkempia selvityksiä on varmaa, että asiaan liittyen on tapahtunut rikos, tulee lukijan huomioida, että useimmat työkalut tekevät muutoksia tiedostoihin ja esimerkiksi lokien osalta niihin kirjoittuu koko ajan uuttaa data, jolloin mahdollinen todistusaineisto saattaa tuhoutua pysyvästi.

3 Tietoturvapoikkeama

Tietoturvapoikkeamalla tarkoitetaan tässä opinnäytetyössä tahallista tai tahatonta tapahtumaa tai järjestelmän tilaa, jonka seurauksena tietoturvallisuus saattaa olla vaarantunut, tai on vaarantumassa.

Tietoturvallisuus on järjestely, jolla pyritään varmistamaan tiedon saatavuus, eheys ja luottamuksellisuus (Kokonaisturvallisuuden sanasto 2017, 35). Tietoturvaloukkauksella tarkoitetaan tietoturvallisuuden loukkaamista, tai välitöntä uhkaa sen tapahtumisesta (Cichonski ym. 2012, 6).

Hyökkääjä voi yrittää murtautua järjestelmään tai on voinut päästä sisälle jo aiemmin kenenkään huomaamatta. Olennaista vahinkojen minimoimiseksi on, että potentiaalinen tietoturvapoikkeama pitää ensin tunnistaa, jotta voi alkaa tekemään korjaavia toimenpiteitä. Ammattikielessä puhutaan termistä Incident Response, jonka voi kääntää vahinko- ja vaaratilanteisiin reagoimiseksi tai poikkeamiin vastaamiseksi.

Jokainen poikkeamatilanne alkaa siitä, kun organisaatio saa tietää tapahtumasta tai tapahtumasarjasta, joka viittaa haitalliseen toimintaan. Havaitseminen voi tapahtua automaattisena hälytyksenä tai esimerkiksi ulkopuolisen tahon ilmoittaessa mahdollisesta tietoturvaongelmasta. Tämän jälkeen tietoturvapoikkeamaan reagoidaan yleensä ennalta määrättyjen prosessien mukaisesti. (Johansen, G. 2020, 8.)

Johansenin (2020, 8–11.) mukaan reagointiprosessi voidaan jakaa kuuteen eri vaiheeseen:

1. Valmistelu (Engl. Preparation)
2. Havaitseminen (Engl. Detection)
3. Analyysi (Engl. Analysis)
4. Eristäminen (Engl. Containment)
5. Hävittäminen ja palauttaminen (Engl. Eradication and Recovery)
6. Tapahtuman jälkeiset toimet (Engl. Post-Incident Activity)

Voidaan myös käyttää Incident Response elinkaarta (Engl. Life Cycle), jossa vaiheita on neljä (Cichonski ym. 2012, 21):

1. Valmistelu
2. Havaitseminen ja analyysi
3. Eristäminen, hävittäminen ja palauttaminen
4. Tapahtuman jälkeiset toimet

Molemmissa prosesseissa toimenpiteet ovat kuitenkin samoja.

Opinnäytetyössä ei käydä läpi koko prosessia, vaan keskitytään tilanteeseen, jossa on havaittu mahdollisia merkkejä poikkeamatilanteesta. Tässä vaiheessa aletaan analysoida tapahtumia, kuten lokitietoja, verkkoyhteyksiä ja tietoa käynnissä olevista ohjelmistoprosesseista. Analyysin tavoitteena on määrittää tapahtuman perimmäinen syy ja selvittää mahdollisen hyökkääjän toimet järjestelmässä (Johansen, G. 2020, 10).

Havaitsemis- ja analysointivaihe on yleensä reagointiprosessin haasteellisin ja vaativin vaihe, johon kuuluu sen varmistaminen, että poikkeama on tapahtunut, missä laajuudessa se on tapahtunut ja kuinka isosta poikkeamasta on kyse (Cichonski ym. 2012, 26).

Havaitsemisvaiheessa organisaatio saa ensimmäistä kertaa tietoonsa tapahtumia, jotka viittaavat mahdolliseen haitalliseen toimintaan. Tapahtuma voi olla esimerkiksi järjestelmänvalvojan tilin aktivoituminen työajan ulkopuolella tai ulkopuoliselta toimijalta tuleva ilmoitus organisaation verkosta peräisin olevasta oudosta liikenteestä. Voi olla myös, että työntekijä ilmoittaa saaneensa oudon sähköpostin ja avanneensa sen liitteenä olevan epäilyttävän tiedoston. Tästä alkaa reaktiivinen prosessi tietoturvapoikkeaman tutkimiseksi. (Johansen, G. 2020, 8–9.)

Havaitsemisen tukena voi olla esimerkiksi erilaisia SIEM-ohjelmistoja (Security Incident and Event Management), jotka tuottavat valtavasta datamassasta oleellisia tietoja ja automaattisia hälytyksiä, jotta poikkeamatilanteisiin voidaan reagoida mahdollisimman nopeasti. Myös muut automaattiset työkalut, kuten virustorjuntaohjelmistot voivat toimia apuna (Cichonski ym. 2012, 26).

Joskus merkit vaarantuneesta järjestelmästä ovat itsestäänselviä. Web -sivuston tilalle on voinut tulla hyökkääjän jättämä viesti tai koko järjestelmä on lukkiutunut ja näytöllä näkyy vain kiristyshaittaohjelman tiedot. Aina merkit eivät kuitenkaan ole näin selviä. Järjestelmä saattaa toimia hitaasti, järjestelmän prosesseissa voi olla epäilyttävä prosessi, palvelin on tavallista kuormittuneempi tai esimerkiksi verkossa on tunnistamatonta liikennettä. Järjestelmä saatetaan asentaa uudelleen, jolloin ongelmat katoavat, mutta alkuperäinen syy ei selviä. (Boelen, M. 2024a.)

Poikkeamatilanteen merkit voidaan jakaa kahteen ryhmään: esiasteisiin/edeltäviin (Engl. precursors) ja indikaattoreihin (Engl. Indicators). Esiasteen merkit ovat esimerkiksi suorat uhkaukset hyökkääjältä hyökkäyksen toteuttamisesta tai uusien käytössä olevaan järjestelmään tai palveluun liittyvien haavoittuvuuksien ilmitulo. Indikaattori on merkki siitä, että poikkeamatilanne on saattanut tapahtua tai tapahtuu juuri sillä hetkellä. (Cichonski ym. 2012, 26.)

Indikaattorien ymmärtäminen on analyysin kannalta tärkeää, koska niiden avulla on mahdollista selvittää miten hyökkäys on toteutettu ja mitä sillä on saatu aikaan.

Esimerkkejä indikaattoreista (Cichonski ym. 2012, 26):

- Laitteelta löytyy tuntematon tiedosto, joka on nimetty epätavallisesti.
- Useat epäonnistuneet kirjautumisyritykset sovellukseen tai järjestelmään.
- Epätavallinen verkkoliikenne.
- Palvelimen kaatuminen.

Vaikka indikaattorin perusteella vaikuttaisikin siltä, että tietoturvapoikkeama on tapahtunut, niin jotkut indikaattorit, kuten palvelimen kaatuminen voi johtua useista muista syistä.

On tärkeää analysoida miten järjestelmä on vaarantunut ja miten hyökkääjä on päässyt järjestelmään sisälle. Järjestelmän altistuminen hyökkääjän kohteeksi voi johtua esimerkiksi paikkaamattomasta haavoittuvuudesta, tai helposti murrettavista salasanoista käyttäjätileillä. Mikäli näitä tietoja ei saada selville, niin järjestelmän uudelleen asennuksesta ei ole mitään hyötyä, sillä hyökkääjä tulee pääsemään uudelleen sisälle, mikäli alkuperäistä haavoittuvuutta ei korjata. Sisäänpääsytavan ymmärtäminen voi myös auttaa määrittämään, ovatko organisaation muut laitteet vaarantuneet, vai koskeeko poikkeama vain yhtä laitetta. (Baker & Green 2005, 3)

Esimerkkejä hyökkäyksistä (Cichonski ym. 2012, 6):

- Hyökkääjän komentama bottiverkko (Engl. botnet) lähettää suuria määriä yhteyspyyntöjä verkkopalvelimelle aiheuttaen sen kaatumisen.

- Käyttäjälle lähetetään sähköposti, jonka liitteenä on haittaohjelma. Haittaohjelman avulla hyökkääjä saa luotua yhteyden laitteeseen.
- Hyökkääjä suorittaa tietomurron, jonka seurauksena saa haltuunsa arkaluonteisia tietoja ja kiristää niiden julkaisemisella organisaatiota.

Hyökkääjät voivat käyttää järjestelmää esimerkiksi ladatakseen työkaluja ja haittaohjelmia, joilla voidaan tartuttaa muita järjestelmiä samassa ympäristössä. Järjestelmää voidaan myös käyttää esimerkiksi kryptovaluutan louhintaan tai osana Command and Control (C2) serveriä. (Kinger ym. 2023.)

Yleisimmät Linux -järjestelmiin kohdistuneet haittaohjelmatyypit vuonna 2022 olivat (Kinger ym. 2023):

- Webshell
- Troijalaiset (Engl. Trojan)
- Takaportit (Engl. Backdoor)
- Kryptovaluutan louhinta (Engl. Cryptocurrency miner)
- Mainosohjelmat (Engl. Adware)
- Tietojenkalastelu (Engl. Phishing)
- Hakkerointiin käytettävät ohjelmat (Engl. Hacktool)

Yleisimmät hyökkäyksissä hyväksikäytetyt haavoittuvuudet voidaan yleensä jakaa neljään kategoriaan (Kinger ym. 2023):

1. Päivittämätön ohjelmisto (Engl. Unpatched software). Monet hyökkäykset hyödyntävät vanhentuneiden ohjelmistojen tunnettuja haavoittuvuuksia.
2. Virheelliset asetukset (Engl. Misconfigurations). Virheet asetuksissa ja konfiguroinnissa, kuten avoimeksi jätetyt suojaamattomat portit ovat ensisijaisia kohteita, joilla voidaan päästä sisälle järjestelmään.
3. Turvaton koodi (Unsecure code). Erityisesti Web-pohjaiset hyökkäystekniikat käyttävät hyväkseen huonoja koodauskäytäntöjä ja virheitä koodissa.
4. Tietojen kalastelu ja sosiaalinen manipulointi (Engl. Phishing and social engineering). Käyttäjä saadaan esimerkiksi asentamaan haittaohjelmia tai paljastamaan tunnistetietoja.

Pahimmassa tapauksessa järjestelmää on jo hyödynnetty laajempaan tietomurtoon, kuten esimerkiksi Vastaamon tapauksessa, jonka uutisoinnin perusteella tietoturvapoikkeamatilanteen laajuuden tunnistamiseen liittyi ongelmia, silloin kun ensimmäiset merkit järjestelmän vaarantumisesta ovat ilmenneet. Mikäli indikaattoreita esiintyy, niin on järkevää lähteä siitä ajatuksesta liikkeelle, että tietoturvapoikkeama on tapahtunut tai käynnissä ja toimia sen mukaisesti (Cichonski ym. 2012, 28).

Hyökkääjät voivat järjestelmään sisälle päästyään esimerkiksi (Hussain, S. 2020):

- vaihtaa olemassa olevien käyttäjien salasanoja
- avata vanhentuneita ja lukittuja käyttäjätilejä
- luoda uusia käyttäjiä
- poistaa käyttäjätilejä
- tuhota lokitiedostoja
- muuttaa lokitiedostojen sisältöä
- poistaa tai muuttaa järjestelmän konfigurointitiedostoja

Kokonaisuuden hahmottamiseksi opinnäytetyön konteksti voidaan sitoa kuvitteelliseen skenaarioon, jossa yrityksessä X työskentelevä työntekijä Y on valittanut käyttämänsä tietokoneen hitaudesta ja toistuvasta palveluiden kaatumisesta laitetta käytettäessä. Yrityksen IT-työntekijän on selvitettävä, onko kyseessä tietoturvapoikkeama ja onko yritykseen kohdistunut tietoturvaloukkaus tai mahdollisesti tietomurto.

Mistä tällaisessa tilanteessa voidaan lähteä etsimään tietoa ja miten?

4 Analyysin suorittaminen

Sandfly Securityn (2021) mukaan viisi tärkeintä selvitettävää asiaa analyysin kannalta ovat:

1. Prosessit – Epäilyttävät prosessit ja verkkotoiminta
2. Hakemistot - Epäilyttävät hakemistot, jotka sisältävät haittaohjelmien käyttämiä hyötykuormia (Engl. payloads), dataa, tai työkaluja, jotka mahdollistavat sivuttaisen liikkumisen verkossa (Engl. lateral movement).
3. Tiedostot – Haitalliset tiedostot sekä tiedostot, joita on muokattu, luotu tai siirretty hyökkääjän toimesta.
4. Käyttäjät – Epäilyttävän käyttäjätoiminnan selvittäminen.
5. Lokit – Muokattujen lokitiedostojen havaitseminen ja yleisimpien lokitiedostojen tarkastelu hyökkääjän toimenpiteiden, kuten jälkien peittelyn varalta.

Analyysin kohteet haetaan esille Bourne-again shell (Bash) -komentotulkin kautta. Komentotulkkeja on erilaisia, mutta Bash on lukuisten Linux-jakeluiden oletus komentotulkki ja siksi käytössä myös tässä opinnäytetyössä.

Komennoista esitetään mahdollisimman hyödyllisiä ja yksinkertaisia versioita, joilla tarvittavat tiedot saa nopeasti esille. Saadakseen täyden hyödyn näistä komennoista, kannattaa lukijan käydä läpi käyttämiensä komentojen manuaalisivut, koska työn laajuuden vuoksi voidaan esittää vain otos näiden ominaisuuksista.

Komennot esitetään seuraavalla formaatilla, jossa on ensin tummennetulla käytettävä komento, ja sen jälkeen sisennyksessä lyhyt seloste komennon tuottamasta tulosteesta tai sen ominaisuuksista.

Komento: **man komento**

> Tällä komennolla voit tarkastella käyttämäsi komennon manuaalisivua, jossa on tietoa komennon ominaisuuksista, kuten käyttötarkoituksesta ja käytettävissä olevista parametreista.

Komento olisi esimerkiksi muotoa **man cat**, jos haluttaisiin tietoa cat -komennosta.

Osa komennoista vaatii superuser do (sudo) -oikeudet toimiakseen, jolloin komennon eteen tulee tarvittaessa lisätä **sudo**. Tällöin komennon suorittaminen vaatii käyttäjän, jolla on komentoon vaadittava käyttöoikeustaso.

Tapauskohtaisesti ennen analyysia, tulee laitteen verkkoyhteydet katkaista, irrottamalla verkkokaapelit ja estämällä langattomat yhteydet laitteen asetuksista. Näin menettelemällä voidaan menettää tietoja hyökkääjästä, esimerkiksi aktiivisista verkkoyhteyksistä. Tällä kuitenkin estetään lisävahingot niissä tapauksissa, kun hyökkääjän on mahdollista seurata laitteen tapahtumia tai kontrolloida laitetta. (Baker ym. 2005, 4.)

Jos mahdollista, niin analyysia tulisi suorittaa pareittain, joissa toinen tallentaa tehdyt toimenpiteet ja toinen suorittaa teknisen analysoinnin. Lisäksi olisi hyvä olla jonkinlainen poikkeamatilanteen seurantajärjestelmä, johon kirjataan joka kerta samankaltaisesti ja järjestelmällisesti tapahtumaan liittyvät tiedot (Cichonski ym. 2012, 31).

Aluksi on hyvä kerätä perustiedot järjestelmästä. Mikäli hyökkääjä on päässyt järjestelmään, niin jatkoanalyysin kannalta on tärkeä selvittää, kohdistuuko kyseessä olevaan jakeluun sellaisia tunnettuja haavoittuvuuksia, joiden avulla hyökkäys on voitu toteuttaa (Thatipalli, A. 2023).

Komento: **cat /etc/os-release**

> Tulostaa järjestelmätiedot, kuten jakeluversion (Thatipalli, A. 2023).

Tulosteen tiedoilla voidaan hakea kaikki Common Vulnerabilities and Exporsures (CVE) – järjestelmään ilmoitetut Ubuntun-käyttöjärjestelmään vaikuttavat haavoittuvuudet. CVE-tiedot löytyvät Ubuntua ylläpitävän Canonical -yrityksen sivuilta. (Ubuntu 2024c.)

Tehdyt toimenpiteet tulee taltioida esimerkiksi kirjoittamalla, kuvaamalla, videoimalla, tai millä tahansa muulla hyväksi havaitulla menetelmällä, jotta myöhemmin voidaan palata löydettyihin tietoihin.

5 Prosessit

Tietoturvapoikkeamatilanteessa tulee selvittää, mitä prosesseja järjestelmässä on käynnissä, sekä mitä aliprosesseja suoritetaan ja mihin ne liittyvät. Analyysin kannalta on tärkeää selvittää prosessin lähde, suorituspaikka, suorittava käyttäjä ja suoritustapa (Thatipalli, A. 2023).

Jotta hyökkääjä voisi suorittaa haitallista koodia järjestelmässä, niin koodin pitää olla osana jotain prosessia. Prosessi voidaan ajatella säiliönä suoritettavalle koodille, joka käyttää järjestelmäresursseja. Prosessi käyttää ulkoisia resursseja, kuten tiedostoja, joita se tarvitsee tehtävänsä suorittamisessa. (Anson, S. 2020, 52.)

Jokaiselle prosessille annetaan käytön ajaksi ainutkertainen tunnus (PID, Process Identification Number), viittaus siihen prosessiin, joka loi sen (Engl. parent process), ja konteksti, jonka alaisuudessa se on suoritettu, yleensä prosessin käynnistänyt käyttäjä. Lisäksi prosessilla voi olla useita aliprosesseja (Engl. child process). (Anson, S. 2020, 52, 239.)

Haittaohjelmat kätkevät toimintansa usein järjestelmän prosessien taakse, jotka saattavat näennäisesti vaikuttaa harmittomilta. Haitallisten ohjelmien tunnistaminen voi olla hankalaa, sillä hyökkääjä voi esimerkiksi muuttaa prosessin nimestä vain yhden kirjaimen, piilottaen prosessin toiminnan sallitun ohjelmiston nimen taakse. (Johansen, G. 2020, 198.)

5.1 Analysointi

Prosessitietojen selvittämiseksi on käytettävissä kaksi peruskomentoa ps ja top (Domantas, G. 2024).

Komento: **ps**

> Tulostaa kysely hetken tilanteen käynnissä olevista prosesseista. Huomioitavaksi, että tulos on staattinen, eikä prosessit päivity reaaliaikaisesti. Tuloksessa on neljä saraketta: PID, terminal name (TTY), running time (TIME), ja prosessin käynnistäneen komennon nimi (CMD). (Domantas, G. 2024.)

Ps on yksinkertainen ja hyödyllinen komento, jota tulisi käyttää säännöllisesti, jotta järjestelmän ylläpitäjälle kehittyy kokonaiskuva siitä, että mitkä prosessit ovat järjestelmän normaaliin toimintaan liittyviä. Näin mahdollisten haitallisten prosessien tunnistaminen on helpompaa poikkeamatilanteessa.

Hyödyllisiä parametreja ps -komennon kanssa käytettäväksi:

Parametri **a** tuottaa kaikkien järjestelmässä olevien käyttäjien, kaikki käynnissä olevat prosessit (Domantas, G. 2024).

Parametri **u** antaa lisätietoja, kuten muistin ja suorittimen käyttöprosentin, prosessin tilakoodin ja prosessien omistajan (Domantas, G. 2024).

Parametri **x** listaa myös kaikki prosessit, joita ei ole suoritettu komentotulkista. Näihin kuuluvat esimerkiksi daemonit, jotka ovat järjestelmään liittyviä prosesseja, jotka toimivat taustalla, kun järjestelmä käynnistetään (Domantas, G. 2024).

Komento: **ps aux**

Antaa kattavat tiedot käynnissä olevista prosesseista.

Komento: **ps auxwf**

Järjestelee prosessit puukaavioon, jossa aliprosessit on niihin liittyvien prosessien alla (Sandfly Security 2018).

Komento: **ps -u käyttäjänimi**

Tulostaa listan tietyn käyttäjän käynnissä olevista prosesseista (Domantas, G. 2024).

Prosessi tietueiden määrä voi tuntua ylitsepääsemättömältä, mutta keskity aluksi tuloksissa ”Command” sarakkeeseen, jossa luetellaan prosessien nimet. Jos joku nimistä on sinulle täysin tuntematon, niin ota sen PID-tunnus talteen. Tämän jälkeen etsi prosessista tietoa Internet -hakukoneella ja yritä selvittää onko siinä jotain epäilyttävää vai liittyykö se esimerkiksi käyttöjärjestelmän normaaliin toimintaan. (Drake, N. 2024a.)

Komento: **ps -fp PID**

Tulostaa valitun prosessin PID-tunnuksella prosessiin liittyvän applikaation tietoja (Sheward, M. 2018, 163).

Komento: **lsof -p PID**

Jos prosessi vaikuttaa epäilyttävältä, niin tällä lsof -komennolla voit tarkastella mitä tiedostoja prosessi on käyttänyt (Drake, N. 2024a). Tarvitset prosessin PID-tunnuksen. Huomaa komennossa käytettävä pieni L.

Komento: **strace -p PID**

> Tällä komennolla voi tarkastella käynnissä olevan prosessin tekemiä järjestelmäkutsuja (Engl. System calls and signals) (Cooper, E. 2017a).

Erityisesti serveriympäristössä ja myös yksittäisissä laitteissa tulisi tarkastella runsaasti järjestelmän resursseja (CPU/RAM) käyttäviä prosesseja, tällaisia ovat esimerkiksi kryptovaluutan louhintaan suunnitellut haittaohjelmat (Drake, N. 2024a).

Kun serveri on vaarantunut kokemattoman hyökkääjän tai automatisoitujen bottien tekemien hyökkäyksien takia, niin on todennäköistä, että hyökkäyksessä käytetyt keinot kuluttavat suurimman osan järjestelmän resursseista. Resurssien kuluttaminen ilmenee järjestelmän hitautena, joka käyttäjän näkökulmasta näkyy esimerkiksi ongelmina web-sivun lataamisen kanssa tai sähköpostin lähettämiseen ja saapuvan sähköpostin vastaanottamiseen kuluvan ajan hidastumisena. (Cooper, E. 2017a.)

Prosessien resurssien käyttöä voidaan tarkastella seuraavilla komennoilla.

Komento: **top**

> Top -komento näyttää reaaliaikaisesti käynnissä olevat prosessit sen mukaan, miten paljon ne käyttävät suoritinta (Domantas, G. 2024).

Käytetään seuraavaksi htop -komentoa. Voit asentaa sen Ubuntussa komennolla **sudo apt-get install htop.**

Komento: **htop**

> Esittää prosessit reaaliaikaisesti käyttäjäystävällisessä näkymässä, jossa pystyt esimerkiksi suodattamaan ja järjestelemään prosesseja, sekä näet niiden täydelliset suoritussijainnit (Domantas, G. 2024).

Jos käytät prosessien tarkasteluun htop-komentoa, niin voit lopettaa prosessin suorittamisen suoraan käyttöliittymästä painamalla joko F9- tai k-näppäintä sen prosessin kohdalla, jonka haluat pysäyttää (de Koff, J. 2020). Htop sisältää myös muita hyödyllisiä toimintoja, kuten sen että pystyt helposti hakemaan prosessitietoja esimerkiksi tietyllä prosessinimellä painamalla F3-näppäintä.

Käytetään seuraavaksi atop-komentoa. Voit asentaa sen Ubuntussa komennolla **sudo apt install atop.**

Komento: **atop**

> Atop näyttää kaikkien prosessien järjestelmän resurssien käytön ja yksityiskohtaisia tietoja prosessien suorittamisesta (Domantas, G. 2024). Atop sisältää paljon ominaisuuksia, joista saat lisätietoja työkalun manuaalista.

Voidaan käyttää myös pidstat -komentoa, joka näyttää aktiiviset prosessit päivittyvällä näkymällä. Mikäli pidstat ei ole asennettuna, niin saat asennettua sen komennolla **sudo apt install sysstat**.

Komento: **pidstat 3**

> Tulostaa aktiiviset prosessit kolmen sekunnin välein päivittyvällä näkymällä (Boelen, M. 2024c).

Komento: **pidstat -C prosessin_nimi 3**

> Tällä komennolla voit seurata tiettyä prosessia ja sen CPU-käyttöä (Boelen, M. 2024c).

Komento**: pidstat -l -t**

> Tällä komennolla saat enemmän tietoa prosesseista ryhmiteltynä puukaavioon (Boelen, M. 2024c).

Pidstat -komentoa voidaan käyttää myös prosessin käynnistämiseen ja sen vaikutusten seuraamiseen.

Komento: **pidstat 3 -e käynnistettävä_prosessi**

> Komennolla voit käynnistää prosessin ja seurata sen vaikutuksia käynnistämisen alusta alkaen (Boelen, M. 2024c).

Proc -tiedostojärjestelmää käytetään tiedon välittämiseen ytimen (Engl. Kernel) ja prosessien välillä. Tiedostoon tallentuu myös linkki ajettavan ohjelman executable (exe)-tiedostoon. (Linux.fi 2021.)

Hyödyllisiä komentoja näiden tietojen analysointiin:

Komento: **ll /proc/PID/exe**

> Komennolla saat selville prosessiin liittyvän exe-tiedoston. Vaihda PID tilalle sen prosessin PID-tunnus, jota haluat tarkastella. Huomaa komennossa käytettävä pieni L.

Komento: **ls /proc/*exe -la**

Tulostaa listauksen kaikista prosesseista, exe-linkkeineen (de Koff, J. 2020).

Komento: **ls -al /proc/PID/exe**

Tulostaa PID-tunnuksella prosessin todellisen prosessipolun (Sandfly Security 2018).

Komento: **ls -alR /proc/*/exe 2> /dev/null |grep deleted**

Tulostaa näkyville poistetut binäärit, jotka ovat edelleen aktiivisina. (Sandfly Security 2018)

Komento: **strings /proc/PID/comm** ja **strings /proc/PID/cmdline**

Valitun prosessin PID-tunnuksella saadaan selville komennon nimi ja komentotulkki (Sandfly Security 2018).

Komento: **strings /proc/PID/environ**

Tulostaa prosessiin liittyvät ympäristötiedot (Engl. environment) (Sandfly Security 2018).

Komento: **ls -alR /proc/*/cwd**

Tulostaa listauksen prosessi hakemistosta (Sandfly Security 2018).

5.2 Prosessien lopettaminen

Jos havaitset epäilyttäviä prosesseja, voit lopettaa niiden suorittamisen seuraavilla komennoilla. Prosessin lopettamiseen tarvitset sen yksilöivän PID-tunnuksen tai prosessinimen.

Komento: **kill -9 PID**

Lähettää signaalin, jolla prosessi saadaan pysäytettyä välittömästi. Kaikki erilaiset prosessin lopettamiseen liittyvät signaalit saat esille **kill -l** komennolla.

Komento: **killall prosessinimi**

Jos ohjelma suorittaa useita prosesseja, voit lopettaa kaikki siihen liittyvät prosessit tällä komennolla (Drake, N. 2024a).

Voit myös lopettaa tietyn käyttäjän kaikki prosessit kerralla.

Komento: **killall -u käyttäjänimi**

Lopettaa käyttäjän prosessit (Linux-audit.com 2024).

5.3 Automaattiset ja ajoitetut

Linux-järjestelmissä on monia tapoja suorittaa koodia ilman käyttäjän suoraa osallistumista sen suorittamiseksi. Cron-tehtäviä voidaan käyttää tehtävien ajoittamiseen, systemd tai init.d voidaan määrittää suorittamaan koodia järjestelmän käynnistyksen yhteydessä ja komento-tulkkiprofiilit voidaan määrittää käynnistämään koodia käyttäjän kirjautuessa sisään. Näiden menetelmien tarkat sijainnit voivat vaihdella jakelun ja sen konfiguraation mukaan. (Anson, S. 2020, 60.)

Haittaohjelmista esimerkiksi madot (Engl. Worm) ja C2-yhteyksiin liittyvät toiminnot hyödyntävät ajoitettuja tehtäviä (Thatipalli, A. 2023).

Jos hyökkääjä on onnistunut saamaan oman haittaohjelmansa automaattisesti suoritettavien ohjelmien joukkoon, niin haitalliset ohjelmat suorittavat toimintojaan säännöllisesti ilman käyttäjän interaktiota (Databasemart.com 2023).

Automaattisesti suoritettavat cron -tehtävät tallennetaan tiedostoon **/etc/crontab**. Crontab on listaus komennoista, joita suoritetaan määritellyn aikataulun mukaisesti. Crontab tulee sanoista ”cron table” (de Koff, J. 2020).

Komento: **less /etc/crontab**

Tulostaa crontab tiedoston sisällön (de Koff, J. 2020).

Komento: **crontab -l**

Näyttää kaikki ajoitetut tehtävät nykyiselle käyttäjälle (Databasemart.com 2023). Huomaa komennossa käytettävä pieni L.

Komento: **crontab -u käyttäjänimi -l**

Näyttää valitun käyttäjän ajoitetut tehtävät (Databasemart.com 2023).

Nähdäksesi tietyille ohjelmille tehtyjä yksilöllisiä cron -tehtäviä, niin siirry ensin sijaintiin **/etc/cron.d/**, jonka jälkeen **ls -l** -komennolla saat tulosteen kansiossa olevista tiedostoista. Tarkastele valitsemaasi tiedostoa **cat** -komennolla. (de Koff, J. 2020.)

Muita sijainteja cron -tehtävien tarkasteluun (Ubuntu 2024b):

/etc/cron.daily
/etc/cron.weekly
/etc/cron.hourly
/etc/cron.monthly
/etc/cron.d

Systemd:n (kts. kohta 10.2) mukana tulee erillinen cron-toiminto nimeltä systemd.timer.

Komento: **systemctl list-timers --all**

Tulostaa listan cron-tehtävistä (de Koff, J. 2020).

Hyökkääjä voi yrittää varmistaa pääsyn vaarantuneeseen järjestelmään lataamalla kirjautumisen yhteydessä asettamiaan takaportteja (Engl. backdoor) komentotulkkiprofiilien yhteyteen. Tämän kaltaisia muutoksia voidaan etsiä seuraavista sijainneista (Ubuntu 2024b):

/home/username/.bash_profile
/home/username/.bash_rc
/home/username/.profile

Nämä ovat kolme yleisintä kohdetta, joihin hyökkääjät tekevät muutoksia. Esimerkiksi **/bin/sh** liittyvät tietueet voivat olla merkki hyökkääjän tekemistä muutoksista ko. tiedostoissa. Takaporttien avulla hyökkääjä saa aikaiseksi pysyvyyttä (Engl. persistence), sillä tiedostot ladataan toistuvasti uudelleen, komentotulkkiprofiilien lataamisen yhteydessä. (Ubuntu 2024b.)

Pysyvyyden saavuttaminen tarkoittaa, että vaikka hyökkäys saataisiinkin osittain torjuttua, niin pelkästään tietokoneen uudelleen käynnistäminen voi mahdollistaa hyökkäyksen jatkumisen.

Lisäksi hyökkääjä voi muokata käynnistysohjelmia suorittaakseen haitallisen ohjelman käyttöjärjestelmän käynnistämisen yhteydessä. Tämä on yleinen keino saada hyökkäykselle pysyvyyttä. Käynnistämisen yhteydessä suoritettava ohjelma, joka ei vaadi suoraan käyttäjän interaktiota onkin yleensä hyökkääjän kannalta haluttu tapa saada oma haitallinen prosessi suoritetuksi (Anson, S. 2020, 56).

Käynnistysohjelmien tiedot tallennetaan Ubuntussa init.d -tiedostoon, joka löytyy hakemistosta **/etc/init.d** (Thatipalli, A. 2023).

Komento: **ls /etc/init.d**

Tiedoston sisällön tarkastelu suoritetaan ls -komennolla (Thatipalli, A. 2023).

Pysyvyyden kannalta tarkastettavia sijainteja ovat lisäksi (Sandfly Security 2018):

```
/etc/rc.local
/etc/rc*.d
/etc/modules
/var/spool/cron/*
```

Mikäli epäilyttävä prosessi johdattaa haitallisen ohjelman äärelle, niin asiassa vaaditaan jatkoanalyysia, jossa takaisin mallinnetaan (Engl. reverse-engineering) ohjelman toiminta (Thatipalli, A. 2023). On myös mahdollista, että haittaohjelma on entuudestaan tuttu, jolloin siihen liittyen on todennäköisesti saatavilla informaatiota Internetistä.

6 Verkkotoiminta

Verkkotoiminta on yleensä osa mitä tahansa tietoturvapoikkeamaa. Hyökkääjällä on oltava jokin yhteys laitteeseen ja useimmat tavoitteet, joihin hyökkääjät pyrkivät, edellyttävät jonkinlaista mekanismia kommunikoida verkkojen ja muiden resurssien kanssa, joten verkkoyhteydet ovat looginen kohde analyysille (Anson, S. 2020. 49, 261).

Ulkoisiin yhteyksiin liittyvien verkkoyhteyksien tarkastelu voi esimerkiksi paljastaa järjestelmässä suoritetun haitallisen prosessin, joka yrittää kommunikoida ulospäin (Johansen, G. 2020, 273).

Yhteydet voivat olla myös organisaation sisäisten järjestelmien välisiä, kun hyökkääjä pyrkii tekemään sivuttaisliikkeitä (Engl. lateral movement) saavuttaakseen pääsyn syvemmälle organisaatioon tai tehdäkseen lisätiedustelua. Hyökkääjä pyrkii todennäköisesti siirtymään sivusuunnassa, kun se on saanut alustavan jalansijan organisaation verkossa sijaitsevaan järjestelmään. (Anson, S. 2020, 49.)

Usein haitalliset sovellukset tai järjestelmään sisälle päässyt hyökkääjä avaavat yhteyksiä jatkaakseen haitallista toimintaa tai laajentaakseen sitä. Analysointi kannattaa aloittaa selvittämällä nykyiset aktiiviset yhteydet. Hyökkäyksiin käytetään usein avoimia portteja, joten porttien tilan tarkastaminen järjestelmän turvallisuuden kannalta on tärkeää.

Suurin osa verkkotoimintaan liittyvistä komennoista vaatii toimiakseen sudo-oikeudet. Useat työkalut antavat samankaltaisia tietoja toistensa kanssa, mutta hieman eri formaatissa.

Aloitetaan analysointi käyttämällä netstat -komentoa. Netstat antaa lähes reaaliaikaisen kuvan järjestelmän saapuvista ja lähtevistä yhteyksistä (Ubuntu 2024b). Netstat voidaan asentaa käyttämällä komentoa: **sudo apt install net-tools**. Tämä asentaa net-tools paketin, jonka osana netstat on.

Komento: **netstat -rn**

Näyttää verkkoyhteyksien reititykset (Engl. routing table).

Komento: **netstat -ie**

Listaa tiedot käytettävissä olevista verkkosovittimista (Engl. Network Interface) (Verma, K. 2024).

Komento: **netstat -anp | more**

Näyttää yhteydet ja runsaasti lisätietoja niistä (Ubuntu 2024b). Käytä morea, jotta terminaali ei täyty tulosteesta.

Komento: **netstat -nalp**

Näyttää aktiiviset ja kuuntelevat portit ja runsaasti lisätietoja niistä, kuten prosessitiedot.

Komento: **netstat -plunt**

Listaa vain kuuntelevat TCP- ja UDP-yhteydet.

Komento: **watch sudo netstat -anp**

Antaa kahden sekunnin välein päivittyvän tilannekuvan aktiivisista yhteyksistä (Ubuntu 2024b).

Taitavat hyökkääjät osaavat asentaa ohjelmia, jotka käyttävät mahdollisimman vähän järjestelmän resursseja, mutta kuuntelevat verkkoyhteyksiä toimintaohjeiden saamiseksi (Drake, N. 2024a).

Komento: **netstat -l**

Listaa vain kuuntelevat portit (Verma, K. 2024).

Tällaisten prosessien tunnistaminen voi olla erityisen hankalaa, sillä ne eivät tee mitään muuta kuin odottavat ohjeita kuuntelemalla verkkoyhteyksiä. Tunnistamiseksi tulisi etsiä prosesseja, jotka joko odottavat yhteyttä (LISTEN) tai joilla on avoin yhteys (ESTABLISHED). (de Koff, J. 2020.)

Komento: **netstat -pt**

Tulostaa prosessin PID-tunnuksen ja ohjelman nimen aktiivisiin verkkoyhteyksiin liittyen (Verma, K. 2024).

Komento: **netstat -an | grep ':80'**

Etsi tiettyyn porttiin liittyvää prosessia (Verma, K. 2024).

Komento: **netstat -tulpn | grep LISTEN**

Tulostaa kaikki TCP- ja UDP-portit, jotka LISTEN-tilassa, näyttää lisäksi PID-numeron ja ohjelman tiedot (Gite, V. 2024).

Yhteyksiä voidaan tarkastella myös lsof -komennon avulla.

Komento: **lsof -i :443**

Listaa kaikki porttiin x (esimerkissä 443), liittyvät yhteydet (Sheward, M. 2018, 163).

Komento: **lsof -i**

Listaa verkkoyhteyksiä kuuntelevat prosessit. Jos listassa on outoja nimiä, selvitä löytyykö prosessinimestä tietoa Internet-haulla (Drake, N. 2024a). Voit käyttää myös komentoa **lsof | grep IPv** (Baker ym. 2005, 6).

IP-osoitteet ovat varsinkin tietoturvapoikkeaman myöhemmän selvittelyn kannalta oleellisia tietoja, sillä niillä voidaan päästä huolimattoman hyökkääjän jäljille.

Komento: **netstat -ntu|awk '{print $5}'|cut -d: -f1 -s|sort|uniq -c|sort -nk1 -r**

Tämä komento oikein syötettynä listaa kaikki IP-osoitteet, jotka ovat yhteydessä järjestelmään/serverille. Tulos näyttää lukumäärällisesti laskevassa järjestyksessä, kuinka monta eri yhteyttä kullakin IP-osoitteella on kyselyhetkellä. 1–50 yhteyttä per IP-osoite on vielä suhteellisen normaalia, mutta mikäli yhteyksiä per IP-osoite on yli 100, niin siinä on todennäköisesti jotain outoa. (de Koff, J. 2020.)

Yleinen järjestelmänvalvojien tehtävä on selvittää mistä epäonnistuneet SSH kirjautumisyritykset tulevat. SSH-yhteyksien lokitiedostot kannattaa tarkastaa sen varalta, että joku yrittää saada yhteyden serverille. Ubuntussa SSH yhteyksien lokitiedot tallentuvat **/var/log/auth.log** tiedostoon. (Cooper, E. 2017b.)

Analyysin kannalta on selvitettävä kahden tyyppiset tiedot (Cooper, E. 2017b):

1. Kirjautumisyritykset käyttäjillä, joita ei ole olemassa.
2. Olemassa olevat käyttäjät, mutta kirjautumiseen on käytetty väärää salasanaa.

Komento: **tail -n 300 /var/log/auth.log | grep sshd**

Tulostaa auth.log -lokitiedostosta 300 viimeistä riviä, listaten kaikki epäonnistuneet ja sallitut yhteydet, sekä niihin liittyvät käyttäjänimet (de Koff, J. 2020).

Esimerkiksi bottien suorittamissa hyökkäyksissä auth.log tiedostosta löytyisi todennäköisesti useita ”Invalid user” tietueita, joten rajataan tuloksia (Cooper, E. 2023a).

Komento: **tail -n 100 /var/log/auth.log | grep "Invalid user"**

Tulostaa auth.log -lokitiedostosta 100 viimeistä riviä, listaten kaikki epäonnistuneet kirjautumisyritykset käyttäjänimellä, jota ei ole olemassa.

Komento: **grep "Invalid user" /var/log/auth.log | awk '{print $8}'**

Tulostaa ainoastaan IP-osoitteet (Cooper, E. 2023a).

Komento: **cat /var/log/auth.log | grep "authentication failure"**

Tulostaa kaikki epäonnistuneet kirjautumiset väärällä salasanalla (Hussain, S. 2020).

Komento: **cat /var/log/auth.log | grep "failed - POSSIBLE BREAK-IN ATTEMPT"**

Tulostaa ilmoitukset mahdollisista murtautumisyrityksistä (Hussain, S. 2020). Tämän kaltainen tietue voi johtua monesta eri syystä.

Asennettavista ohjelmista erityisen hyödyllinen työkalu on **nmap**. Nmapin käytössä pitää huomioida, että sillä ei tule tehdä luvattomia skannauksia oman toimialueen ulkopuolella. Nmapin asennus onnistuu komennolla **sudo apt install nmap**.

Komento: **nmap localhost**

Tulostaa näkyville paikallisen järjestelmän avoimet portit ja niihin liittyvät palvelut.

Nmap:ssa on valtavasti erilaisia toiminnallisuuksia, joihin kannattaa tutustua. Sen käyttämisestä on myös paljon tutoriaaleja ja ohjeistuksia, joten tässä työssä ei esitellä sen ominaisuuksia tämän laajemmin

Palvelunestohyökkäyksiin liittyvät yhteydet saadaan nopeasti selville iftop komennolla, joka antaa tietoa datan lähettämisestä ja vastaanottamisesta verkkoyhteyksien välityksellä.

Iftop ei tule asennettuna, mutta voit asentaa sen komennolla **sudo apt install iftop**.

Komento: **iftop**

> Komento näyttää valitsemasi verkkosovittimen reaaliaikaisen verkkoyhteyden käytön. Oletuksena iftop komento ilman parametreja valitsee ensimmäisen sovittimen, joka vaikuttaa välittävän ulkoisia yhteyksiä. (Geeksforgeeks.org 2024b.)

Ss -komennon avulla voidaan selvittää merkkejä haitallisesta toiminnasta ja se on nopeampi ja antaa tarkempia tietoja kuin netstat. (Wake, T. 2023a)

Ss-komentoa voidaan käyttää esimerkiksi tilatietojen suodattamiseen.

Komento: **ss state established**

> Tulostaa kaikki muodostetut yhteydet, established voidaan korvata esim. listen, syn-sent jne. (Wake, T. 2023a).

Poikkeamatilanteeseen vastaavan on usein tunnistettava lähtevät yhteydet, jotka voivat viitata esimerkiksi C2-toimintaan. Ss:n avulla voidaan suodattaa portteihin, tai tuntemattomiin IP-osoitteisiin muodostettuja yhteyksiä (Wake, T. 2023a).

Komento: **ss -tp state established dport = :443**

> Esimerkki komento, jossa selvitetään muodostettuja yhteyksiä porttiin 443 (Wake, T. 2023a).

Myös listening-tilaisten porttien selvittäminen on tärkeää, jotta voidaan tunnistaa mahdolliset luvattomat palvelut ja takaportit.

Komento: **ss -tulpn**

> Listaa kaikki TCP- ja UPD-portit, jotka listening-tilassa (Wake, T. 2023a).

Komento: **ss -tan**

> Voidaan tarkastella sekä listening-, että non-listening-tilaisia TCP-yhteyksiä epätavallisen liikennöinnin tunnistamiseksi esimerkiksi C2-toiminnan ja datavuotojen yhteydessä (Wake, T. 2023a).

Hyökkääjä voi myös käyttää spooffattuja (Engl. spoofed) ARP-paketteja, esimerkiksi ohjatakseen verkkoliikenteen oman laitteensa kautta (Sheward, M. 2018, 154).

Komento: **arp -a**

Tulostaa ARP cachen -tiedot (Sheward, M. 2018, 154).

Mikäli epäilyttäviä yhteyksiä löytyy, niin haitallisen toiminnan lähteen jäljille voidaan päästä yhdistämällä prosessi verkkotoimintaan. Tämä voi olla ratkaisevaa tietoturvapoikkeaman laajuuden ja sen vaikutusten selvittämisessä. (Wake, T. 2023a.)

Komento: **ss -tp**

Yhdistää verkkotoiminnan siihen liittyvään prosessi nimeen ja PID-tunnukseen (Wake, T. 2023a).

Verkkotoimintaa ja siihen liittyvää statistiikkaa voidaan seurata reaaliaikaisesti esimerkiksi IPTraf -työkalulla. Asennus onnistuu komennolla **sudo apt install iptraf-ng**. IPTraf:ssa on graafinen käyttöliittymä, joka käynnistetään komennolla **iptraf-ng** ja se sisältää useita toiminnallisuuksia esimerkiksi tietyn verkkokortin tai portin yhteyksien seuraamiseen. Lisäksi saatavilla on helppolukuisia tilastoja sekä lähtevistä, että saapuvista yhteyksistä ja niihin liittyvistä porteista. (Cooper, E. 2021.)

Myös netstat ja ss komentoja voidaan käyttää statistiikkatietojen selvittämiseen.

Komento: **netstat -s**

Tulostaa statistiikka yhteyksistä (Verma, K. 2024).

Komento: **ss -a -e -i**

Ss -komentoa voidaan käyttää pistokkeiden (Engl. socket) -statistiikan näyttämiseen. Sen avulla on mahdollista saada enemmän tietoa, kuin netstatilla (Sandfly Security 2018).

Komento: **ss -m**

Näyttää pistokkeiden muistin käyttöön liittyvää statistiikkaa (Wake, T. 2023a).

Verkkotoiminnan osalta voidaan tarkastella myös Internet-selaamisesta jääneitä jälkiä, esimerkiksi verkkohakuihin ja vierailuihin liittyviä tietoja. Näistä tiedoista on erityisesti hyötyä, kun tutkitaan hyökkäyksiä, joissa käyttäjät siirtyvät haitalliselle verkkosivustolle, tai ovat käyneet sellaisella ja ladanneet esimerkiksi haitallisen ohjelman. Tarkasta siis myös käytössä olevan Internet-selaimen historiatiedot, sekä käyttäjän Downloads-hakemiston sisältö. Hakemistojen sisällön tarkastelusta tarkemmin seuraavassa luvussa.

7 Hakemistot

Suurin osa Linux -jakeluista käyttää samaa hierarkkista tiedostojärjestelmä rakennetta, jossa ensimmäiseltä tasolta löytyy seuraavat hakemistot (Sheward, M. 2018, 142):

- /bin, keskeiset Linux-komentojen binäärit;
- /boot, boot loader tiedostot;
- /dev, laitteiden raakatiedostot (Linux käsittelee esimerkiksi tallennusmedioita tiedostoina);
- /etc, järjestelmän konfiguraatio tiedostot;
- /home, käyttäjien kotihakemistot;
- /lib, jaetut kirjastot (Engl. shared libraries);
- /media, mount-pisteet irrotettaville tallennusmedioille;
- /mnt, käytetään yleensä tilapäisenä mount-pisteenä;
- /opt, valinnaisesti asennetut ohjelmistot;
- /proc, virtuaalinen tiedostojärjestelmä, jota käytetään prosessien ja ytimen tietojen säilyttämiseen;
- /root, root eli Linux-järjestelmän pääkäyttäjän kotihakemisto;
- /run, käytetään reaaliaikaisesti muuttuvien tietojen tallentamiseen, kuten siihen, että ketkä käyttäjistä on parhaillaan kirjautuneina sisään;
- /sbin, käytetään järjestelmän binäärejä ja laiteajureita koskevien tietojen säilyttämiseen;
- /tmp, väliaikaista tiedostotilaa, joka ei säily uudelleenkäynnistysten välillä;
- /usr, käytetään useita käyttäjiä koskevien apuohjelmien ja sovellusten toimesta;
- /var, käytetään muuttuvien (Engl. variable) tiedostojen tietojen tallentamiseen ja säilyttämiseen, toisin sanoen sellaisten tietojen tallentamiseen, jotka muuttuvat normaalin järjestelmän käytön yhteydessä, kuten lokitiedostot.

Rowlandin (2021) ja Sandfly Securityn (2018) mukaan yleiset kohteena olevat hakemistot, joita käytetään hyökkääjän toimesta haitallisten koodien suorittamiseksi, harhauttamiseen käytetyn ja varastetun datan piilottamiseksi, sekä pysyvyyden säilyttämiseksi ovat:

```
/tmp
/var/tmp
/dev
/dev/shm
/bin
/sbin
/usr/bin
/usr/sbin
/lib*
/usr/lib*
/etc
/var/log
/var/run
```

/var/spool
/home/ (Käyttäjien kotihakemistot)

Kolme erityisen turvatonta ja yleisesti hyökkäyksissä hyödynnettyä sijaintia tarkasteltavaksi ovat /tmp, /dev/shm ja /var/tmp (de Koff, J. 2020).

Esimerkiksi /dev hakemisto tarjoaa pääsyn käynnissä olevan järjestelmän kannalta tärkeisiin oheislaitteisiin, kuten kiintolevyihin ja edelleen niiden resursseihin kuten levyosioihin (Anson, S. 2020, 342).

Yleisimpien hyökkäyskohteiden sisältöä voidaan tarkastella ls -komennolla.

Komento: **ls -lhaRp**

Tulostaa nykyisen hakemiston sisällön ja antaa yksityiskohtaisesti tietoa, kuten tiedostojen koon. Näyttää lisäksi piilotetut tiedostot (Grundy, B. 2023, 47). Parametri p:llä saadaan lisäksi ”/” tiedostonimen perään, osoittamaan että kyseessä on kansio.

Komento: **ls /tmp -la**

Tulostaa tietyn kansion sisällön.

Hakemisto /tmp puhdistetaan uudelleen käynnistyksen yhteydessä. Hakemistosta saattaa löytyä esimerkiksi prosessin väliaikaistiedostoja, jotka katoavat, kun järjestelmä käynnistetään uudelleen. Hakemistoon on yleensä laajat kirjoitusoikeudet myös alemman tason käyttäjillä, joten järjestelmään päässyt hyökkääjä tai haittaohjelma voi käyttää hakemistoa hyväkseen, hyödyntäen sitä, että tiedot eivät tallennu pysyvästi. (Sheward, M. 2018, 154.)

Komento: **ls -alR /proc/*/cwd 2> /dev/null | grep tmp**

Tulostaa prosessit, jotka käynnissä tmp-hakemistossa (Sandfly Security 2018).

Find -komentoa voidaan hyödyntää tietyn tyyppisen tiedon tai tiedostojen etsimiseen.

Komento: **find / -type d -name ".*"**

Tulostaa listauksen piilotetuista hakemistoista (Sandfly Security 2018). Kattavimman tuloksen saat Sudo-oikeuksin.

Komento: **find / \(-nouser -o -nogroup \) -exec ls -lg {} \;**

Hakee tiedostot ja hakemistot, joilla ei ole käyttäjän tai käyttäjäryhmän nimeä (Sandfly Security 2018).

Komento: **find /hakemisto -type f -exec file -p '{}' \; | grep ELF**

Hakee suoritettavia tiedostoja valitusta hakemistosta (Sandfly Security 2018).

Komento: **tree**

Komento asennetaan **sudo apt install tree** -komennolla. Tulostaa hakemiston sisällön puukaavio näkymässä, jossa näkyvillä myös alikansioiden sisältö ja rakenne (Grundy, B. 2023, 200).

8 Tiedostot

Linux-järjestelmissä tiedostonimet ja hakemistojen nimet tukevat paljon suurempaa merkkimäärää kuin Windowsissa. Hyökkääjät voivat käyttää tätä hyväkseen ja yrittää piilottaa tiedostoja tiedostojärjestelmään nimeämällä ne mahdollisimman samankaltaisesti sinne kuuluvien tiedostojen sekaan. (Anson, S. 2020, 58–59.)

Tiedostonimet ovat merkkikokoriippuvaisia (Engl. case-sensitive), ja kaikki pisteellä alkavat tiedostot ovat oletusarvoisesti piilotettuja. Etsimällä epätavallisia tiedostonimiä, kuten sellaisia, jotka päättyvät välilyöntiin tai joiden alussa on useita pisteitä yms. piilotustekniikoita, voidaan paljastaa hyökkääjän käyttämiä ohjelmistoja, tiedostoja, tai muita hyökkäykseen liittyviä tietoja. Sallittua on esimerkiksi vain yhdestä välilyönnistä koostuvat tai välilyöntiin päättyvät tiedostonimet. (Anson, S. 2020, 342.)

Tiedostojen joukosta kannattaa etsiä tiedostoja, joita on käytetty, muutettu tai luotu lähellä epäillyn tietoturvapoikkeaman tapahtuma-aikaa tai sellaista aikaa, jolloin järjestelmässä ei pitäisi olla tapahtumia, kuten työaikojen ulkopuolella.

Ansonin (2020, 314) mukaan Linux-järjestelmissä tiedostoissa on käytössä kolmenlaisia aikaleimoja:

- Atime (Engl. Accessed) – Aika, jolloin tiedostoa käytettiin viimeksi;
- Mtime (Engl. Modified) – Aika, jolloin tiedostoa muokattu viimeksi;
- Ctime (Engl. Change) – Aika, jolloin tiedoston metatietoihin, kuten omistajuuteen, tiedostonimeen, sijaintiin tai käyttöoikeuksiin on tehty viimeisin muutos.

Lisäksi tiedostoille annetaan ”Birth” aika, joka on se hetki, kun tiedosto on luotu (Boelen, M. 2024b).

Jos tiedostoa vain käytetään, esimerkiksi luetaan sen sisältöä tekemättä siihen mitään muutoksia, vain atime muuttuu (Anson, S. 2020, 314).

Jos käyttöoikeuksia tai tiedoston nimeä muutetaan, vain ctime muuttuu, koska vain metatiedot muuttuvat, ei itse tiedoston sisältö (Anson, S. 2020, 314).

Jos tiedoston sisältöä muutetaan, eli tiedostoon itseensä tehdään muutoksia, niin sekä mtime että ctime muuttuvat (Anson, S. 2020, 314).

8.1 Analysointi

Tiedostoon liittyvät aikaleimat saadaan selville stat komennolla.

Komento: **stat tiedostonimi** tai suoraan sijainnilla **stat /etc/passwd**

> Tulostaa näkyville tiedostoon liittyvät aikaleimat ja muita tietoja tiedostosta (Boelen, M. 2024b).

Mikäli tapahtuma-aikaa on onnistuttu haarukoimaan tai hyökkäys on edelleen käynnissä, niin on hyödyllistä selvittää laajemmin tiedostoihin ja hakemistoihin tehtyjä muutoksia.

Komento: **ls -lart /**

> Komennolla saat root-hakemiston sisällön aika järjestyksessä niihin tehtyjen muutosten mukaisesti. (Baker ym. 2005, 5)

Komento: **ls -lart**

> Komennolla saat hakemiston sisällön aika järjestyksessä tiedostoihin tehtyjen muutosten mukaisesti. Näyttää myös piilotettu tiedostot.

Jos tiedetään ongelmien alkaneen esimerkiksi edellisenä päivänä, niin voidaan yrittää selvittää, että mitä muutoksia järjestelmässä on tapahtunut näiden kahden päivän aikana.

Komento: **sudo find /etc /var -mtime -2**

> Komennolla saat selville tiedot kaikista järjestelmätiedostoista, joita on muokattu kahden edeltävän päivän aikana. Jos järjestelmä on päivitetty näiden päivien aikana, niin komento tulostaa paljon tietueita. (de Koff, J. 2020.)

Komento: **find / -mtime -1**

> Hakee ja tulostaa luodut ja muokatut tiedostot viimeisen vuorokauden ajalta (Sandfly Security 2018).

Hakemistot /etc ja /dev sisältävät suuria määriä tiedostoja, joten sinne on helppo piilottaa tiedosto, joka sulautuu konfigurointi- ja järjestelmätiedostojen joukkoon. Hyökkääjät voivat

käyttää hyväkseen esimerkiksi arkistotiedostoja, kuten ZIP- tai TGZ-tiedostoja. Käyttäjän tiedostoista erityisen kiinnostavia ovat suoritettavat tiedostot, jotka sijaitsevat esimerkiksi käyttäjän Downloads-hakemistossa. (Anson, S. 2020, 58–59.)

Mikäli hakemistoista löytyy epäilyttäviä tiedostoja, niin niiden osalta on tulee selvittää tiedoston tyyppi, sisältö ja käyttöoikeudet.

Komento: **file tiedostonimi**

Komennolla voit selvittää tuntemattoman tiedoston tiedostotyypin. Komento vertaa tiedostoa tunnettuihin tiedostomuotojen header -tietoihin tiedostotyypin määrittämiseksi (Grundy, B. 2023, 50). Tulos ei aina ole täysin luotettava.

Komento: **file * -p**

Tulostaa tiedostonimet ja niiden tiedostotyypit sen hetkisestä hakemistosta (Sandfly Security 2018).

Komento: **strings tiedostonimi**

Tulostaa tiedostosta kaikki vähintään neljä merkkiä pitkät string-muotoiset tiedot luettavassa muodossa. Hyödyllinen erityisesti data-tiedostojen, kuten exe-tiedostojen sisällön tutkimiseen. (Grundy, B. 2023, 50, 203)

Komento: **ls -l tiedostonimi**

Mikäli ensimmäinen merkki on viiva (-), niin kyseessä on tiedosto. Jos ensimmäinen merkki on ”d”, niin kyseessä on kansio. Ensimmäinen merkki voi olla myös joku muu kuten ”l”, joka tarkoittaa linkkiä toiseen tiedostoon tai kansioon. Seuraavat 9 merkkiä kertovat tiedoston käyttöoikeuksista, joista r = lukuoikeus, w = kirjoitusoikeus ja x = suoritusoikeus. (Grundy, B. 2023, 51–52.)

Mikäli löydät jatkoanalyysin kannalta mielenkiintoisen tiedoston ja haluat suojella sitä kaikenlaisilta muutoksilta, niin voit suojata sen muuttamalla tiedoston attribuutteja. Tiedostojen käyttöoikeuksia voidaan tarvittaessa muokata **chmod** -komennolla. Lisäämällä tiedostoon ”immutable” attribuutin, siitä voidaan tehdä muuttumaton, jolloin sitä ei voida esimerkiksi poistaa vahingossa (Grundy, B. 2023, 56).

Komento: **lsattr tiedostonimi**

Tulostaa tiedoston attribuutit (Grundy, B. 2023, 56).

Komento: **sudo chattr +i tiedostonimi**

Tarvitse sudo-oikeudet. Komennolla lisäät haluttuun tiedostoon muutoksia estävän **i** attribuutin (Grundy, B. 2023, 56).

Komento: **sudo chattr -i tiedostonimi**

Poistaa **i** attribuutin, jotta voit jälleen tehdä tiedostoon muutoksia (Grundy, B. 2023, 56).

Attribuutteja voidaan vastaavasti muokata myös kokonaisiin kansioihin (Grundy, B. 2023, 56).

Mahdollisen haitallisen tiedoston osalta voidaan tehdä tarkistuksia selvittämällä sen hash, eli tiiviste ja vertaamalla sitä tunnettuihin haittaohjelmien tiivisteisiin.

Komento: **md5sum tiedostonimi**

Tulostaa valitun tiedoston MD5 -muotoisen tiivistelmän.

Kopioimalla tulosteesta saamasi tiivisteen esimerkiksi Virustotal palveluun, voit saada nopeasti selville, että onko kyseessä tunnettu haittaohjelma (Virustotal 2024).

Mikäli analysoitavassa järjestelmässä ei ole virustorjuntaohjelmistoa, mutta on epäilys tartunnasta, niin tiedostojen skannaamiseen voidaan käyttää ClamAV-nimistä avoimen lähdekoodin antivirus-ratkaisua (Grundy, B. 2023, 105–107).

ClamAV:n avulla voidaan suorittaa AV-skannaus esimerkiksi tietyssä hakemistossa, suoraan komentotulkin kautta. ClamAV tulostaa tuloksen suoraan terminaalin, kun skannaus on valmis. Lisää tietoa ja ajantasaiset asennus- ja käyttöohjeet ClamAV:sta löydät ohjelmiston kotisivulta. (ClamAV 2024.)

8.2 Paketinhallintajärjestelmä

Eri Linux jakeluilla on erilaisia menetelmiä ohjelmapakettien käsittelyyn. Paketinhallintajärjestelmän tarkoituksena helpottaa ohjelmistojen asennusta, poistoa ja riippuvuuksien hallintaa. (Linux.fi 2023.)

Ubuntussa on käytössä dpkg-paketinhallintajärjestelmä, jonka hallintaan käytetään Advanced Package Tool (APT) -nimistä paketinhallinta työkalua/edustaohjelmaa, joka huolehtii pakettien riippuvuussuhteista ja niiden päivittämistä. Suoritettavat ohjelmat tarvitsevat toimiakseen erinäisiä ohjelmakirjastoja ja apuohjelmia. Paketinhallinta on tärkeää järjestelmän turvallisuuden kannalta, sillä se mahdollistaa esimerkiksi tietoturvapäivitysten asentamisen. (Linux.fi 2023.)

Myös hyökkääjä voi hyödyntää paketteja ja asentaa ohjelmistoja ohi paketinhallinnan.

Komento: **sudo dpkg --verify**

> Tarkastaa asennettujen pakettien nimen ja eheyden vertaamalla asennettujen tiedostojen tietoja dpkg-tietokannassa oleviin metatietoihin. Komennon suorittaminen vie joitain minuutteja. Komennolla saadaan selville onko hyökkääjä mahdollisesti tehnyt muutoksia paketteihin. Komento tulostaa mahdolliset poikkeamat virhekoodin kera, mutta mikäli muutoksia ei havaittu, niin komennon tuloste on tyhjä. Lisätietoja virhekoodeista saat komennon manuaalista.

8.3 Rootkit

Rootkit-haittaohjelmat ovat vaarallisimpia uhkia Linux-järjestelmille ja Windows-keskeisille tietoturva-asiantuntijoille voi tulla yllätyksenä, miten usein niitä hyödynnetään Linux-järjestelmiin kohdistuneissa hyökkäyksissä. Koska Linux on avoimen lähdekoodin järjestelmä, niin siinä ei ole käytössä yhtenäistä, pakotettua ajurien allekirjoituskäytäntöä. Tämä aiheuttaa sen, että esimerkiksi kernel-ajureihin liittyvät rootkit ovat paljon yleisimpiä Linuxissa. (Wake, T. 2023b.)

Rootkit on käytännössä merkki siitä, että hyökkääjä on jo vaarantanut järjestelmän ja tietoturvaloukkaus on tapahtunut. Rootkitin olemassaolo edesauttaa hyökkääjän kykyä hallita järjestelmää ja toimii alustana tietojen anastamiseen ja muiden haitallisten toimintojen tekemiseen.

Rootkit on käytännössä koonti erilaisia työkaluja, joiden avulla hyökkääjä piilottaa jälkensä ja ylläpitää pääsyään takaisin järjestelmään. Sana rootkit tulee sanasta ”root”, mikä tarkoittaa järjestelmän pääkäyttäjää. Kit sana taas tulee sanasta ”toolkit”. (Boelen, M. 2024d.)

Tyypillisesti rootkitit käyttävät muunneltuja binääriohjelmia, jotka näyttävät tyypillisiltä komennoilta, mutta ovatkin osa rootkittiä. Jotkin rootkiteistä tarjoavat takaportin hyökkääjälle, jolloin järjestelmä voi vaikuttaa täysin palautetulta, mutta hyökkääjä pääsee sisään tämän piilotetun sisäänkäynnin kautta. (Boelen, M. 2024d.)

Rootkitteihin liittyy monia erilaisia merkkejä, joiden avulla voidaan epäillä järjestelmän saastuneen. Usein ensimmäinen merkki on järjestelmän toimintaan liittyvän palvelun, kuten daemonin kaatuminen. (Boelen, M. 2024d.)

Rootkittien havaitsemiseen on käytössä erilaisia ohjelmia. Esimerkiksi Rkhunter ja Chkrootkit-ohjelmat on suunniteltu auttamaan vaarantuneen järjestelmän tunnistamisessa (Boelen, M. 2024d).

Chkrootkitin voit asentaa Ubuntussa komennolla **sudo apt install chkrootkit**.

Komento: **chkrootkit**

Tarkastaa paikallisesti yleisimmät rootkitit (Chkrootkit 2023).

Rkhunterin voit asentaa Ubuntussa komennolla **sudo apt install rkhunter**.

Komento: **rkhunter --check**

Tarkastaa paikallisesti yleisimmät rootkitit (rkhunter 2018).

Molemmat ohjelmat toimivat siten, että ne tarkastavat tiettyjen järjestelmätiedostojen eheyden, koska nämä tiedostot ovat usein sellaisia, joihin hyökkääjä pyrkii luomaan takaportteja. Lisäksi sovellukset käyttävät tunnettujen rootkittien allekirjoituksia (Engl. signature), havaitakseen mahdollisia niihin liittyviä prosesseja ja tiedostoja. Lisäksi ne etsivät esimerkiksi piilotettuja TCP-portteja, jotka ovat usein merkki rootkitin toiminnasta. (Ubuntu 2024b.)

Mikäli järjestelmästä löytyy rootkit, niin sen poistaminen asentamatta koko järjestelmää uudelleen on käytännössä mahdotonta (Boelen, M. 2024d).

9 Käyttäjät

Yksi selvimmistä järjestelmän vaarantumisesta kertovista tiedoista on luvattoman käyttäjän löytyminen järjestelmästä.

Järjestelmään luotu tili tarjoaa hyökkääjälle erinomaisen keinon liikkua vapaasti järjestelmässä ja siihen liitetyissä ympäristöissä. Tilin luomisella hyökkääjä voi varmistaa, että pääsee takaisin järjestelmään silloinkin, mikäli alkuperäinen tietoturva-aukko on saatu selvitettyä ja sen hyödyntäminen on estetty. (Anson, S. 2020, 56.)

Hyökkääjät voivat luoda järjestelmään uusia käyttäjiä, joiden avulla on mahdollista päästä takaisin järjestelmään, siinäkin tapauksessa, että vaihtaisit yhden vaarantuneeksi tunnistetut käyttäjätilin salasanan (Ubuntu 2024b).

On tärkeää tunnistaa tilit, jotka eivät ole luotu organisaation tai luvallisen käyttäjän toimesta ja tilit, joita ei käytetä kuten niitä on tarkoitettu käytettäväksi. Tällaisia tilejä voi olla esimerkiksi päivitystehtäviä varten luodut järjestelmävalvoja tilit, joita on tarkoitus käyttää vain ajoittain. Järjestelmässä voi olla myös käytöstä poistettuja tilejä, joita hyökkääjä voi yrittää aktivoida. Tällaisia tilejä voi olla esimerkiksi organisaatiosta lähteneen työntekijän käytössä ollut tili. (Anson, S. 2020, 56.)

9.1 Analysointi

Käyttäjätunnuksiin liittyvät tiedot tallennetaan **/etc/passwd**-tiedostoon, joka on tarkistettava huolellisesti. Tiedostosta saadaan selville käyttäjät ja käyttöoikeustasot.

Komento: **cat /etc/passwd | column -t -s :**

> Tulostaa tiedot käyttäjätileistä ja järjestelee tiedot erillisiin sarakkeisiin, joissa muun muassa käyttäjänimi, käyttäjän id (UID), ryhmä id (GID), kuvaus tilistä ja tilin kotihakemisto (Thatipalli, A. 2023).

Tarkastetaan myös **/etc/group** tiedosto. Esimerkiksi erilaisia ryhmäkäytäntöjä voidaan määritellä tietyille käyttäjille.

Komento: **cat /etc/group**

> Tulostaa tiedot olemassa olevista ryhmistä ja niiden perässä yksittäiset käyttäjänimet, jotka kuuluvat kyseessä oleviin ryhmiin (Thatipalli, A. 2023).

Myös **/var/log/auth.log** tiedosto tulee tarkastaa ja kiinnittää huomiota erityisesti uusiin käyttäjiin.

Komento: **cat /var/log/auth.log | tail**

> Tulostaa tietoja käyttäjän istunnosta (Engl. session) (Thatipalli, A. 2023). Tail parametrilla saadaan tiedostosta vain kymmenen viimeisintä tietuetta. Jos haluat tarkastella koko tiedostoa, niin käytä komentoa ilman tail parametria.

Komento: **cat auth.log | grep 'new user'**

> Tulostaa auth.log tiedostosta rivit, jotka sisältävät tietueen ”new user” – uusi käyttäjä.

Komento: **lslogins ’käyttäjätunnus’**

> Näyttää valitusta käyttäjätunnuksesta kattavan koonnin, hakemalla tietoa esimerkiksi passwd -tiedostosta (Stocker, S. 2024).

Shadow-tiedostossa säilytetään salattuna käyttäjien salasanat. Tietoja voidaan hyökkääjän toimesta käyttää salasanojen murtamiseen. (Sheward, M. 2018, 142.)

Komento: **cat /etc/shadow**

> Tulostaa tiedoston sisällön.

Selvitetään kaikki analysointi hetkellä kirjautuneena olevat käyttäjät.

Komento: **w**

> Näyttää muut kirjautuneena olevat käyttäjät ja heidän kirjautumisaikansa. Komennon tulosteessa, kohdassa ”FROM”, voit nähdä kirjautuneen käyttäjän IP-osoitteen ja selvittää edelleen esimerkiksi whois-komennolla tietoja tästä IP-osoitteesta. (Drake, N. 2024a.)

Komento: **who -H -a**

> Listaa tietoja kirjautuneista käyttäjistä. Näyttää käyttäjänimen, komentotulkin johon käyttäjä on yhdistetty ja päivämäärän/kellonajan, jolloin kirjautuminen on tapahtunut, sekä viimeisenä joko komentotulkin nimen tai IP-osoitteen, riippuen siitä, että onko käyttäjä kirjautunut paikallisesti vai ei. Parametri -a antaa kaiken mahdollisen tiedon, mitä who-komennolla on saatavissa ja parametri -H parantaa luettavuutta antamalla sarakkeille otsikot. **(**Linuxconfig.org 2023.)

Mikäli järjestelmästä ei löydy tuntemattomia käyttäjiä, niin hyökkääjä on kuitenkin voinut olla kirjautuneena järjestelmään. Tarkastetaan kirjautumishistoria.

Komento: **last**

> Näyttää tiedot kirjautumisista aikajärjestyksessä järjestelmän asennuksesta alkaen. Tulosteessa käyttäjänimi, IP-osoite (tai paikallisesti komentotulkin tieto), ja kirjautumisaika (Drake, N. 2024a). Voit modifioida komentoa antamalla parametriksi esimerkiksi -10, jolloin näet vain 10 viimeisintä kirjautumista.

Käyttämällä last-komentoa voit selvittää kirjautuneena olleet käyttäjät ja tarvittaessa esimerkiksi sen, onko yksittäisen käyttäjän osalta normaalia, että kirjautuminen on tapahtunut tiettyyn aikaan.

Komento: **last käyttäjänimi**

> Jos havaitset oudon käyttäjän, niin voit tulostaa ainoastaan kyseessä olevan käyttäjän tiedot lisäämällä last-komennon perään tarkasteltavan käyttäjänimen (Drake, N. 2024a).

Komento: **last -aiF**

> Näyttää jokaisen käyttäjän kirjautumisen ja uloskirjautumisen, kuinka kauan istunto oli aktiivinen ja mistä yhteys tuli (Hussain, S. 2020).

Jos näet last-komennolla vain viimeisimmän kirjautumisen, niin se on todennäköinen merkki siitä, että hyökkääjä on peitellyt jälkiään ja poistanut historiatietoja.

Komento: **sudo pkill -U 'käyttäjänimi'**

> Mikäli toteat käyttäjän luvattomaksi, niin tällä komennolla voit lopettaa käyttäjän yhteyden serverille ja kaikki prosessit kyseessä olevaan käyttäjään liittyen. (Drake, N. 2024a.)

Tarkasta myös epäonnistuneet kirjautumisyritykset. Toistuvat kirjautumisyritykset saattavat kertoa siitä, että joku yrittää murtaa käyttäjätunnuksen salasanaa väsytyshyökkäyksellä (Engl. brute-force).

Komento: **subo lastb**

> Listaa epäonnistuneet kirjautumiset (Sandfly Security 2018).

Komento: **sudo lastb -adF käyttäjänimi**

> Listaa tietyn käyttäjän epäonnistuneet kirjautumisyritykset (Hussain, S. 2020).

Kirjautumisyrityksiä voidaan tarkastella auth.log tiedostosta.

Komento: **grep 'Failed password' /var/log/auth.log**

> Grep -komennolla poimitaan auth.log -tiedostosta kaikki epäonnistuneet kirjautumisyritykset.

Komento: **faillog -a**

> Komennolla voit tarkastella sijainnista /var/log/faillog löytyvän faillog -tiedoston tietoja (Ubuntu 2024b). Faillog sisältää tietoa epäonnistuneista kirjautumisyrityksistä. Parametrilla -a saat tiedot kaikista käyttäjistä. Faillog tulostaa näkyville ainoastaan epäonnistuneet kirjautumiset, mikäli ko. käyttäjä ei ole epäonnistuneen yritykseen jälkeen kirjautunut onnistuneesti järjestelmään (Tutorialspoint 2024).

Kirjautumistietojen osalta löytyy myös binääri-muotoisia analysoitavia tiedostoja. Tällaisia ovat esimerkiksi utmp, wtmp ja lastlog -tiedostot (Anson, S. 2020, 343). Lisäksi löytyy btmp -tiedosto, joka pitää kirjaa epäonnistuneista kirjautumisyrityksistä (Thatipalli, A. 2023). Näiden tiedostojen analysointiin ei voida käyttää esimerkiksi grep tai cat -komentoja.

Utmp sisältää tietoja käyttäjistä, jotka ovat tällä hetkellä kirjautuneena käynnissä olevaan järjestelmään. Wtmp sisältää tietoa käyttäjien kirjautumishistoriasta, kuten kaikki nykyiset ja aikaisemmat kirjautumiset ja informaatiota esimerkiksi järjestelmän uudelleen käynnistämisistä. Btmp sisältää kaikki epäonnistuneet kirjautumisyritykset. (Sandfly Security 2019.)

Koska utmp, wtmp ja btmp tiedostot sisältävät kaikkiin käyttäjiin liittyviä kirjautumistietoja, niin ne ovat yleensä hyökkääjien ja haittaohjelmien kohteena, joko niiden muuttamiseksi tai tuhoamiseksi. Monet haittaohjelmat yksinkertaisesti tuhoavat alkuperäisen tiedoston ja luovat tilalle uuden 0-bitin kokoisen tiedoston peittääkseen jälkensä. (Sandfly Security 2018.)

Tuhoamisen lisäksi tiedostoja voidaan muokata ylikirjoittamalla haluttu tietoa nollilla tai poistamalla tiedostosta vain valitut tietueet (Sandfly Security 2018).

Voit käyttää utmpdump-komentoa binäärimuotoisten lokitiedostojen raakadatan esille saamiseen (Sandfly Security 2019).

Komento: **utmpdump /var/log/wtmp** tai **utmpdump /var/run/utmp** tai **utmpdump /var/log/btmp**

Tulostaa sisällön (Sandfly Security 2019).

Utmpdump-komentoa voidaan käyttää myös sen tarkistamiseksi, että onko lokitiedostoja muokattu.

Komento: **utmpdump /var/run/utmp | grep " \[0\].*1970-01-01"**

Tulostaa rivit, joiden osalta on käytetty nollilla ylikirjoittamista (Sandfly Security 2019).

Mikäli tietue on poistettu ja päivämäärä sarakkeessa on päivämäärä 1970-01-01, niin tarkastelemalla aikaleimoja ennen ja jälkeen tämän tietueen, voit haarukoida tapahtuma-aikaa (Sandfly Security 2019).

Mikäli käyttäjiä on paljon, niin voi olla hyödyllistä selvittää wtmp-tiedostosta kirjautumistietoja. Analyysin kannalta voidaan havaita joko epäilyttävän harvoin, tai usein kirjautuneita käyttäjiä, jotka saattavat johtaa hyökkääjän jäljille.

Komento: **sudo last -f /var/log/wtmp**

Tulostaa onnistuneet kirjautumiset (Thatipalli, A. 2023).

Samalla komennolla voidaan tarkastella myös btmp -tiedoston sisältöä.

Komento: **sudo last -f /var/log/btmp**

Tulostaa epäonnistuneet kirjautumisyritykset (Thatipalli, A. 2023).

Wtmp -tiedoston analysoinnissa voidaan käyttää myös for -komentoa, jonka avulla voidaan toteuttaa useammasta eri vaiheesta koostuvia komentosarjoja.

Komento: **for user in 'ls /home'**

- **do**
- **echo -n "$user: "**
- **who /var/log/wtmp | grep "^$user " | wc -l**
- **done**

Tällä komennolla saadaan listattua käyttäjätunnuksien kirjautumislukumäärä lukumuodossa. Edellytyksenä on, että näillä käyttäjillä on kotihakemisto järjestelmässä. (Stocker, S. 2024.)

Nykyään vähemmän käytetty lastlog tiedosto tietoa siitä, että milloin kukakin käyttäjä on viimeksi kirjautunut sisään (Anson, S. 2020, 343.).

Komento: **lastlog**

Komennolla voit tarkastella sijainnista /var/log/lastlog löytyvän lastlog -tiedoston sisältöä (Ubuntu 2024b).

Komento: **lastlog -u käyttäjänimi**

Tarkastele tietyn käyttäjän tietoja (Hussain, S. 2020).

Voi olla hyödyllistä selvittää, että onko tiettyinä päivinä ollut tavallista enemmän kirjautumisia. Tällä voidaan joissain tilanteissa haarukoida tapahtuma-aikaa tietoturvapoikkeamalle.

Analyysiin voidaan hyödyntää ac-komentoa, jonka voit asentaa komennolla **sudo apt install acct**. Ac-komennolla voidaan tarkastella statistiikkaa tuntikohtaisesti wtmp -tiedostoon tallentuneiden kirjautumisten ja uloskirjautumisten perusteella (Kenlon, S. 2020).

Komento: **ac -d**

Parametrilla -d saadaan koonti käyttäjän päivittäisistä kirjautuneena olo ajoista lukumuodossa (Stocker, S. 2024).

Komento: **ac -p**

> Parametrilla -p saadaan koonti käyttäjäkohtaisesta kirjautuneena olo ajasta lukumuodossa (Stocker, S. 2024).

Kiinnitä huomiota kirjautumisaikoihin. Mikäli epäilyttävän käyttäjätilin kirjautumisaika on lähellä kuluvaa aikaa, niin on mahdollista, että hyökkääjä on poistanut wtmp-tiedoston, jossa säilytetään viimeisimmät kirjautumisajat (Drake, N. 2024a).

Lisää informaatiota käyttäjien kirjautumisista voidaan selvittää lslogins-komennolla.

Komento: **lslogins**

> Näyttää kaikkien järjestelmä- ja käyttäjätunnusten kirjautumistietoja (Stocker, S. 2024).

Komento: **lslogins -u**

> Näyttää saman kuin edellinen, mutta poistaa tuloksesta järjestelmätunnukset (Stocker, S. 2024).

Mikäli analysoitava kohde on serveri, niin kirjautumistietoja voidaan selvittää myös auditd lokien kautta (Hussain, S. 2020).

Komento: **aureport**

> Aureport -komento tulee auditd:n mukana ja on valmiina uusimmissa Linux -jakeluissa. Komento antaa tilastoja järjestelmä tapahtumista (Hussain, S. 2020). Mikäli tulosteen kohdissa ”Number of failed logins” ja ”Number of failed authentications” on suuria lukuja, niin saattaa se olla merkki hyökkäyksestä.

Auditd:n lokitiedosto sijaitsee yleensä sijainnissa **/var/log/audit/audit.log** (Hussain, S. 2020).

9.2 Sudo ja korotetut käyttöoikeustasot

Saatuaan haltuunsa alemman käyttöoikeustason käyttäjän, hyökkääjä todennäköisesti yrittää korottaa käyttäjänsä oikeuksia (Engl. priviledge escalation). Mikäli käyttäjätunnuksella on ID-numerona ”0”, tarkoittaa tämä sitä, että tilillä on järjestelmässä korkeimmat root-oikeudet (Anson, S. 2020, 344).

Komento: **grep ":0:" /etc/passwd**

> Etsii käyttäjätunnuksia, joilla on UID 0/GID 0 (Sandfly Security 2018).

Sudo -ohjelman avulla käyttäjä voi käynnistää ohjelmia pääkäyttäjänä ja pääsee siten esimerkiksi lukemaan tavallisilta käyttäjiltä rajattuja tietoja ja hakemistojen sisältöjä.

Sijainnista **/etc/sudoers** löytyvän sudoers tiedoston sisältö kannattaa tarkastaa mahdollisten muutosten löytämiseksi. Sudoers tiedostoa voidaan muokata, jotta käyttäjän on mahdollista suorittaa toimintoja sudo-ohjelman avulla (Anson, S. 2020, 344). Sudoers-tiedostosta selviää käyttäjät, joilla on root-tason oikeudet (Thatipalli, A. 2023).

Komento: **cat /etc/sudoers**

Tulostaa sudoers tiedot (Sandfly Security 2018). Tarkasta muutosten varalta kohdat (Thatipalli, A. 2023):

- User privilege specification
- Members of the admin group may gain root privileges
- Allow members of group sudo to execute any command

Komento: **cat /var/log/auth.log | grep "user NOT in sudoers"**

Tulostaa epäonnistuneet yritykset korottaa oikeuksia (Hussain, S. 2020).

Sudo-historiaa voidaan analysoida myös journalin avulla, mikäli systemd (kts. luku 10.2) on käytössä, käyttämällä komentoa **journalctl /usr/bin/sudo.**

Jotta näkisit myös muiden käyttäjien sudo-historian, niin sen käyttäjän, jolla teet kyselyn pitää kuulua käyttäjäryhmiin ’adm’ ja ’systemd-journal’. Voit tarkistaa mitkä käyttäjät kuuluvat ko. ryhmiin komennolla **grep -e adm -e systemd.journal /etc/group**. **(**Analysis of sudo logs 2024.)

Käyttämällä komentoa **journalctl /usr/bin/sudo –no-pager**, voit tulostaa tiedot sudo tapahtumista ilman rivien katkaisua **(**Analysis of sudo logs 2024). Tämä helpottaa tietojen lukemista suoraan komentotulkista.

9.3 Komentotulkki

Useimmat komentotulkkiohjelmat ylläpitävät historiatietoja suoritetuista komennoista. Mikäli hyökkääjä on päässyt sisälle järjestelmään saatuaan käyttäjän haltuunsa, niin komentotulkin historia saattaa paljastaa arvokasta tietoa hyökkääjän tekemisistä järjestelmässä (Anson, S. 2020, 342). Komentohistorian selvittely on myös erityisen hyödyllistä servereillä, joihin ei ole asennettu graafista käyttöliittymää (Sheward, M. 2018, 142).

Oletuksena auki olevan komentotulkin Bash-historia tallentuu RAM-muistiin, kunnes suljet komentotulkin. Tämän jälkeen tiedot kirjoitetaan .bash_history tiedostoon, joka löytyy jokaisen käyttäjän kotihakemistosta sijainnista **/home/käyttäjä/.bash_history**. (Levinas, M. 2022.)

Historia -tiedosto on yksinkertainen tekstimuotoinen tiedosto, joka koostuu maksimissaan 2000 tietueesta. Terminaalin historian osalta puskuri on vain 1000 tietuetta. Lisäksi hyökkääjä voi tyhjentää sekä RAM-muistiin, että historiatiedostoon tallentuneet tiedot yksinkertaisilla komennoilla. (Levinas, M. 2022.)

Tietoturvapoikkeamatilanteessa komentotulkin historia kannattaa tarkastaa mahdollisimman pian. Sisältö voi antaa viitteitä hyökkääjän toiminnasta, mikäli hyökkääjä on käyttänyt komentotulkkia, eikä ole poistanut historiatiedostoa.

Komento: **history**

> Tulostaa tiedot kuluvan session komennoista nykyisestä komentotulkista ja historiatiedostosta. Mikäli on useita avonaisia komentotulkkeja, niin ei näytä muiden avoinna olevien komentotulkkien RAM-muistiin tallentunutta historiaa (Levinas, M. 2022).

Komento: **cat ~/.bash_history**

> Tulostaa .bash_history -tiedoston sisällön, jossa jokainen rivi on yksittäinen suoritettu komento komentotulkissa.

Mikäli saat tuloksena vastauksen, että: ”No such file or directory”, on hyökkääjä mahdollisesti peittänyt jälkiään tuhoamalla historiatiedoston (Drake, N. 2024a).

Kiinnitä historiatiedoissa erityisesti huomiota komentoihin, joita voidaan käyttää tiedostojen lataamiseen ja haitallisten sovellusten asentamiseen, esim. **install, curl** ja **wget** (Drake, N. 2024a).

Mikäli historiatiedoista löytyisi esimerkiksi **wget http://haittaohjelma.tar.gz** kaltainen komento, niin tällä voitaisiin päästä hyökkääjän lataaman haittaohjelman jäljille. Huomioi myös pakettien purkamiseen liittyvät komennot kuten **gunzip** ja **ta**r, sekä tiedostojen poistamiseen liittyvät komennot, kuten **rm** (Baker ym. 2005, 6).

Voit myös tarkastella toisen paikallisen käyttäjän komentotulkin historiatietoja.

Komento: **cat ~käyttäjänimi/.bash_history**

> Vaihda tarkasteltava käyttäjä, käyttäjänimen kohdalle. Tarvitset sudo-oikeudet päästäksesi tarkastelemaan historiaa tällä tavoin.

Hyökkääjät voivat yrittää modifioida järjestelmän toimintaa tekemällä muutoksia ympäristömuuttujiin (Engl. environment variable). PATH muuttujat ylläpitävät tietoa esimerkiksi siitä, että mitä sijainteja haetaan, kun käyttäjä kirjoittaa komennon tai suoritettavan tiedoston nimen komentotulkkiin (Anson, S. 2020, 343).

Hyökkääjä voi lisätä esimerkiksi ”LD_PRELOAD=” attribuutin, jonka avulla on mahdollista ladata hyökkääjän muokkaamia jaettuja kirjastoja (Engl. shared libraries) etusijalla (Ubuntu 2024b).

Komento: **env**

> Käynnissä olevassa järjestelmässä nykyisen käyttäjän muuttujat voidaan tarkistaa **env** -komennolla (Anson, S. 2020, 343).

Lisäämällä epätyypillisiä sijainteja tai muuttamalla hakujärjestystä hyökkääjä voivat saada käyttäjän ajamaan haitallisia suoritettavia tiedostoja toisen komennon yhteydessä (Anson, S. 2020, 343). Ota tämä huomioon komentotulkin historiatietojen analysoinnissa.

10 Lokitiedostot

Erilaiset lokitiedot ovat tietoturvapoikkeaman selvittämisen kannalta keskeisessä roolissa. Lokitietueiden läpikäyminen voi olla turhauttavan hidasta ja pelkästään tietueiden ja erilaisten lokitiedostojen määrä ja niiden sisällön ymmärtäminen vaatii syvällistä perehtymistä aiheeseen.

Windowsista poiketen Linux-järjestelmät tallentavat suuren osan lokitiedoista tekstimuotoisessa formaatissa, mikä tarkoittaa, että niitä voidaan käydä läpi ilman erikoistyökaluja (Sheward, M. 2018, 143). Tekstimuotoiset lokitiedostot voivat olla todella isoja ja hankalasti analysoitavissa, lisäksi ne korvautuvat tasaisin väliajoin uusilla ja vanhat ei aktiiviset lokitiedostot pakataan (Anson, S. 2020, 194).

Aina kun esimerkiksi verkkosivua pyydetään, käyttäjä lähettää sähköpostia tai kirjautuu sisään järjestelmään, niin tapahtuma kirjataan lokitiedostoon. Jokaisen järjestelmänvalvojan olennainen taito on osata analysoida näitä lokitiedostoja ja parsia niistä hyödyllistä tietoa. (Cooper, E. 2017b)

Lokitietojen avulla voidaan saada tietoa siitä, miten hyökkäys toteutettiin ja miten syvälle hyökkääjä on päässyt järjestelmässä. Mikäli hyökkääjä on saavuttanut itselleen pääsyn järjestelmään ja onnistunut saamaan esimerkiksi pääkäyttäjäoikeudet haltuunsa, niin ensimmäisenä toimenpiteenä hyökkääjä yrittää peittää jälkiään lokitiedostoista. Yksi nopeimmista tavoista

selvittää onko järjestelmään päästy sisälle, on selvittää, onko keskeisiin lokitiedostoihin tehty muutoksia. (Ubuntu wiki 2020.)

Taitava hyökkääjä tekee muutoksia vain omien jälkiensä peittämiseksi. Selvitä lokitiedostoista, että löytyykö ajallisia aukkoja, jolloin järjestelmä on ollut käytössä, mutta lokitiedostoon ei ole muodostunut uusia tietueita. Mikäli tällaisia löytyy, niin on mahdollista, että hyökkääjä on pyrkinyt siivoamaan jälkiään. Etsi erityisesti yli viisi minuuttia kestäviä välejä, joissa ei ole muodostunut tietueita (Ubuntu wiki 2020).

Selvitä myös, että onko lokitiedostoissa outoja aikaleimoja tai tietueita. Hyökkääjä saattaa kopioida lokitiedoston tilalle, joko kokonaan tai osittain tietueita vastaavasta lokitiedostosta- (Ubuntu wiki 2020). Tällaisia muutoksia pystyy havaitsemaan analysoimalla aikaleimoja. Kesken lokitiedoston aikaleimat heittävät esimerkiksi menneeseen tai tulevaan aikaan, joka viittaa siihen, että lokitiedostoihin on tehty muutoksia.

Hyökkääjä saattaa myös poistaa kokonaisia lokitiedostoja, jolloin muutosten havaitseminen on helpompaa. Mikäli lokitiedosto on kokonaan poistettu, on syytä epäillä, että järjestelmä on vaarantunut, lokitiedostot eivät katoa itsestään (Ubuntu wiki 2020).

Mikäli hyökkääjä muokkaa lokitiedostoja, niin se ei ainoastaan ole huono asia, sillä se antaa myös varmuuden siitä, että hyökkäys on tapahtunut (Sandfly Security 2019).

Suurin osa käyttöjärjestelmän toimintaan liittyvistä lokitiedostoista löytyy hakemistosta **/var/log**, joka sisältää sekä järjestelmä, että applikaatio tason lokitietoja (Sheward, M. 2018, 143).

Voit selvittää nopeasti, että onko lokitiedostoja tyhjennetty hyökkääjän toimesta hakemalla kaikki tyhjät tiedostot.

Komento: **ls -al /var/log/***

> Hae kaikki nollakokoiset lokitiedostot valitusta sijainnista (Sandfly Security 2018).

Oleellisia lokitiedostoja /var/log sijainnissa:

/var/log/secure

> Sisältää järjestelmän laajuiset todennustapahtumat (Sheward, M. 2018, 143).

/var/log/auth.log

Authorization log, joka ylläpitää tietoa järjestelmän todennuksista, kuten salasanojen syötteistä, sudo-komennon käytöstä ja etäyhteyksillä kirjautumisista (Ubuntu 2024b).

Auth.log tiedosto sisällöstä voi olla apua sen määrittämisessä, että onko joku päässyt luvatta järjestelmään joko paikallisesti tai etänä. Myös brute force -väsytyshyökkäyksien jälkiä esimerkiksi SSH-kaltaista etähallintapalvelua vastaan voidaan selvittää auth.log tiedostosta. Jos hyökkääjä on päässyt sisään järjestelmään, niin tämä on ensimmäisiä tiedostoja, jonka hyökkääjä pyrkii siistimään omien jälkiensä peittämiseksi. Mikäli tähän tiedostoon on tehty muutoksia, se on lähes varma merkki siitä, että järjestelmä on vaarantunut. (Ubuntu 2024b.)

/var/log/daemon.log

Sisältää daemon-ohjelmien tietoja (Ubuntu 2024b)

/var/log/debug

Sisältää järjestelmän ja sovellusten virheenkorjaustietoja (Engl. debugging) (Ubuntu 2024b).

/var/log/kern.log

Linuxin ytimeen (Engl. kernel) liittyvät lokit (Ubuntu 2024b).

Syslog-ng on Linux-järjestelmissä oleva lokienhallinta, joka kerää, käsittelee ja välittää lokiviestejä erilaisista lähteistä. Sen lokit tallennetaan yleensä hakemistoihin **/var/log/messages** tai **var/log/syslog.** Syslog-ng kirjoittaa tekstipohjaisten lokien lisäksi lokitiedostoja myös binääri ja JSON formaateissa. (Wake, T. 2023c.)

Syslog-ng on erityisen hyödyllinen, kun yrität selvittää, onko järjestelmä vaarantunut. Se kirjaa kaikki järjestelmätason tapahtumat. Mikäli joku järjestelmän palveluista on vaarantunut, niin se saattaa ilmetä esimerkiksi kyseisen palvelun kaatumisena. (Ubuntu 2024b.)

Syslog-ng lokitiedostot toimivat runsaana tiedonlähteenä, kun yritetään ymmärtää tietoturvapoikkeaman edeltäneitä, sen aikaisia ja sen jälkeisiä tapahtumia. Sen tuottamat lokitiedostot ovat tärkeitä tapahtumakulun selvittämisen kannalta. (Wake, T. 2023c.)

Syslog-konfiguraatio tiedosto rsyslog.conf löytyy sijainnista /etc ja oletussäännöt löytyvät sijainnista /etc/rsyslog.d/50-default.conf.

/var/log/syslog

> Tietoa järjestelmästä, jos et löydä jotain tietoa muista lokitiedostoista, niin se löytyy todennäköisesti täältä (Ubuntu 2024b).

Syslog -tiedostosta selviää kaikki sallitut ja luvattomat muutokset ja siihen tallentuu tietueita jatkuvalla syötöllä, kunnes tiedosto on täynnä, jonka jälkeen luodaan uusi syslog -tiedosto (Thatipalli, A. 2023).

Syslog on yksi yleisimmistä lokitiedostoista, johon hyökkääjät tekevät muutoksia, joten jos havaitset puuttuvia aikaleimoja/aikavälejä tai poikkeavuuksia, on mahdollista, että järjestelmä on vaarantunut (Ubuntu 2024b).

Syslog -tiedosto on hyödyllinen, myös esimerkiksi sellaisten palveluiden toiminnan selvittämisessä, jotka ovat alttiita buffer overflow -tyyppisille hyökkäyksille, jolloin lokitiedostosta voidaan etsiä segmentointi virheitä (Engl. segmentation fault). Nämä virheet eivät aina tarkoita, että järjestelmä olisi vaarantunut, mutta antavat viitteitä vähintään lisäselvitysten tarpeelle. (Ubuntu 2024b.)

Järjestelmälokien lisäksi myös palomuurien lokitiedostot voivat olla hyödyllisiä.

Sijainti: **/var/log/ufw.log**

> Sijainnista löytyy palomuurin lokitiedosto, mikäli ufw (Uncomplicated Firewall) on käytössä. Kyseessä on Ubuntun oletus palomuurin konfigurointiin tarkoitettu työkalu.

Ufw.log -tiedostosta on hyvä analysoida ainakin sääntöjen perusteella estetyt liikennöinnit erilaisiin portteihin. Estetty liikenne löytyy tietueista UFW BLOCK merkinnällä. Oudot yhteysyritykset voivat kertoa joko haitallisen sovelluksen toiminnasta tai yrityksistä löytää heikkouksia järjestelmästä. Jos käytät iptables -ohjelmaa ufw:n sijaan, niin tiedot löytyvät syslog tai kern.log tiedostoista (Ubuntu 2024b).

Myös monet applikaatiot käyttävät /var/log sijaintia lokitiedostojen tallentamiseen. Esimerkkinä Apache, jonka lokitiedostot tallentuvat oletuksena sijaintiin **/var/log/apache2/** (Ubuntu 2024b). Apachen lokien osalta mielenkiintoisin on access.log -tiedosto, josta löytyy tietoja esimerkiksi käyttäjien latauspyynnöistä ja IP-osoitteista (Sheward, M. 2018, 143).

10.1 Tekstimuotoisten lokitiedostojen manuaalinen tarkastelu

Ubuntussa on oletuksena asennettu graafinen käyttöliittymä järjestelmän lokitiedostojen tarkasteluun. Löydät sen joko nimellä **Logs** tai vanhemmissa versioissa nimellä **System Log** (Ubuntu 2024b). Graafinen käyttöliittymä on nopea tapa yleisimpien lokitiedostojen tarkasteluun.

Logs -käyttöliittymä monitoroi myös tiedostoihin tehtäviä muutoksia, joten voit avata sillä lokitiedoston tarkasteltavaksi ja näet kaikki avaamisen jälkeen tehdyt muutokset tummennetulla tekstillä (Ubuntu 2024b).

Logs -käyttöliittymällä voit selata näitä lokitiedostoja helposti ja nopeasti, mutta samoja tiedostoja voidaan analysoida myös manuaalisesti. Lokitiedostot löytyvät oletuksena hakemistosta /var/log (Ubuntu wiki 2020). Sijaintiin tallentuu useita eri **.log** -päätteisiä tiedostoja, jotka ylläpitävät historiaa erityyppisistä tapahtumista järjestelmässä.

Ansonin (2020, 194–195) mukaan tekstipohjaisten lokitiedostojen tarkastelu voidaan jakaa kolmeen vaiheeseen:

1. Hae tiedot lokitiedostosta ja tulosta ne ruudulle.
2. Tutki tiedoston sisältöä, ymmärrä mitä se sisältää ja millä tavalla data esitetään.
3. Suodata näkyviin oleellisia tietoja, jotta tutkittavien tietojen määrä vähenee.

10.1.1 Tiedostojen haku

Lokitiedostot saat esille siirtymällä komentotulkin kautta kyseessä olevan lokitiedoston hakemistoon, esim. **cd /var/log**.

Voit selata lokitiedostojen sisältöä yksinkertaisesti **cat** -komennolla, jos tiedosto on pakkaamaton.

Komento**: cat lokitiedosto**

> Tulostaa tiedot valitusta lokitiedostosta (Anson, S. 2020, 196).

Tulosta tiedot **zcat** -komennolla, jos tiedosto on pakattu. Zcat mahdollistaa tiedon tarkastelun suoraan pakatusta tiedostosta sen purkamisen sijaan (Anson, S. 2020, 194).

10.1.2 Tietojen tutkinta

Tiedostojen sisällön tutkimiseen voit käyttää esimerkiksi **head** -komentoa, jolla voit tarkastella ensimmäistä 10 riviä tiedostosta (Anson, S. 2020, 195). Jos tarvitset enemmän kuin 10 riviä, niin muuta tarkasteltavien rivien määrää muokkaamalla komennon parametreja.

Komento: **head -n 20 kohdetiedosto**

> Tämä esimerkki komento tulostaa 20 ensimmäistä riviä tiedostosta (Anson, S. 2020, 195). Näin saat nopeasti kuvan tiedoston sisällöstä.

Jos taas haluat tarkastella lokitiedoston loppua, voit käyttää **tail** -komentoa, joka näyttää vastaavasti 10 viimeistä riviä. Tail komentoa voi muuttaa head komennon tapaan käyttämällä **-n ”luku”** parametria. (Anson, S. 2020, 195.)

Voit hyödyntää tail -komentoa myös live -järjestelmän analysoinnissa.

Komento: **tail -f tiedostonimi**

Voit seurata reaaliajassa lokitiedoston loppuun tulevia uusia rivejä. (Anson, S. 2020, 195.)

10.1.3 Tietojen suodattaminen

Tietojen suodattamiseen voidaan käyttää **grep** -komento, jonka käyttö perustuu erilaisten regular expressioneiden käyttöön. Näiden käyttö kuluttaa paljon järjestelmän resursseja varsinkin isojen tiedostojen analysoinnissa.

Grep palauttaa kaikki lokitiedoston rivit, jotka vastaavat määritettyä suodatusta. Esimerkiksi IP-osoitteen grep-haku palauttaa kaikki lokitiedoston rivit, jotka sisältävät määritetyn IP-osoitteen, olipa se sitten lähde- tai kohdeosoite (Anson, S. 2020, 195).

Esimerkkejä grep -komennon käytöstä:

Komento: **grep "haettava_tieto" tiedostonimi**

Voit hakea tiedostosta string -muotoista tietoa, esimerkiksi IP-osoitetta (Anson, S. 2020, 195).

Komento: **grep -i "haettava_tieto" tiedostonimi**

Voit hakea string -muotoista tietoa, jossa haettavan tiedon isoilla ja pienillä kirjaimilla ei ole merkitystä, vaan hakee kaikki vastaavuudet (Anson, S. 2020, 195).

Komento: **cat lokitiedosto | grep -F ip_osoite**

Tulostaa valitun IP-osoitteen sisältävät rivit lokitiedostosta (Anson, S. 2020, 196).

Komento: **grep -F ”ip_osoite” lokitiedosto**

Sama kuin edeltävä komento, mutta käyttää tiedoston lukemiseen ja filteröintiin grep -komentoa (Anson, S. 2020, 196).

Komento: **fgrep ”ip_osoite” lokitiedosto**

Sama kuin edeltävä komento, mutta yksinkertaistaa komennon käyttämällä fgrep -komentoa -F parametrin sijaan (Anson, S. 2020, 196).

Komento: **grep -oE "\b((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\b" lokitiedosto**

Listaa kaikki IP-osoitteet valitusta lokitiedostosta (Sheward, M. 2018, 63).

Komento: **grep -oE "\b((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\b" lokitiedosto | sort | uniq -c | sort -n -r**

Listaa kaikki uniikit IP-osoitteet valitusta lokitiedostosta ja ryhmittelee ne sen mukaan, kuinka usein ne esiintyvät ko. lokitiedostossa (Sheward, M. 2018, 63).

Voit myös suodattaa tietoja suoraan pakatusta tekstimuotoisesta tiedostosta käyttämällä grepin sijaan **zgrep** -komentoa (Anson, S. 2020, 195).

Komento: z**grep -F ip_osoite ***.gz

Hakee haun hetkellä valittuna olevan hakemiston kaikista .gz -päätteisistä tiedostoista valittua IP-osoitetta. Tulostaa rivit, joista löytyy ko. IP-osoite (Anson, S. 2020, 196).

Yleensä lokitiedostojen tiedot ovat jossain määrämuodossa, joten jos esimerkiksi lokitiedostossa on 10 kenttää per rivi, joista jokainen on rajattu rivinvaihdoilla, ja kiinnostava IP-osoite on jokaisen rivin kolmannessa kentässä, voit käyttää cut-komentoa IP-osoitekentän poimimiseen (Anson, S. 2020, 196).

Komento: **cut -f 3 lokitiedosto**

Tulostaa edeltävään esimerkkiin liittyen vain IP-osoitekentän ja hylkää loput tiedot (Anson, S. 2020, 196).

Cut -komentoa voidaan käyttää yhdessä grepin kanssa.

Komento: **grep -F ip_osoite lokitiedosto | cut -f 3**

Tulostaa kaikki kolmannet kentät, jokaiselta lokitiedoston riviltä, mikäli ne sisältävät määritetyn IP-osoitteen (Anson, S. 2020, 196).

Komento: **grep -F ip_osoite lokitiedosto | cut -f 3 | uniq |wc -l**

Sama kuin edeltävä komento, mutta uniikkien tuloksien sijaan tulostaa numeerisen arvon siitä, että kuinka monta tulosta lokitiedostosta löytyi. Tämä tapahtuu komennolla wc -l (Anson, S. 2020, 196).

Komento: **cut -d ', ' -f 3,6 lokitiedosto**

Cut komento, jossa määritellään tiedot erottelevaksi merkiksi ”,”. Hyvä komento esimerkiksi tietojen hakemiseen CSV-tiedostosta. Tämä komento palauttaa kolmannen ja kuudennen kentän (Anson, S. 2020, 196).

Komento: **cut -d ' ' -f 4 logfile | sort**

Tulostaa jokaisen rivin neljännen kentän lokitiedostosta ja esittää tuloksen nousevassa ASCII järjestyksessä (Anson, S. 2020, 196).

Läpikäytävän datan määrän suodattamiseen voidaan käyttää myös sort -komentoa.

Komento: **sort -u**

Poistaa kaikki kaksoiskappaleet ja palauttaa vain uniikit arvot tiedostosta.

Analyysin kannalta suodattaminen on nopea tapa saada haluttu tieto esille, mutta suodattamiseen liittyy aina riski siitä, että ei ymmärrä mitä suodattaa. Siksi olisi hyvä testata tietojen suodattamista erilaisilla lokitiedostoilla etukäteen, siten että ymmärtää tietojen esitystavan ja sen, että missä tapauksissa tietoja suodattamalla voi jäädä jotain oleellista huomioimatta.

Komennoista erityisesti grepin käytön opetteleminen on oleellinen taito lokitiedostojen tehokkaaseen analysointiin.

10.2 Systemd

Systemd on yleisesti käytössä oleva järjestelmää ja ajettavien palveluiden hallintaa ohjaava toiminto. Systemd:n osana tulee journal niminen keskitetty lokien hallinta.

Journal sisältää tietoja erilaisista järjestelmätapahtumista, kuten laitteistoon liittyvistä tapahtumista ja ytimen sekä eri palveluiden ja sovellusten viesteistä. Tiedot tallentuvat sijaintiin **/var/log/journal**, mutta toisinkuin edellä mainitut tekstipohjaiset lokitiedostot, niin journalin tiedostot eivät ole tekstimuotoisia, eikä niitä voida käsitellä samalla tavalla. Sen sijaan tietoja tarkastellaan komentotulkista journalctl -komennolla. (Linux.fi 2020.)

Komento **journalctl** ilman mitään parametreja tulostaa kaikki journaliin tallennetut viestit, ja tuottaa siten valtavan määrän informaatiota. Koska tämä on harvoin hyödyllistä, niin seuraavassa käydään läpi muutamia hyödyllisiä komentoja tietojen esille saamiseksi.

Komento**: journalctl -b**

Tulostaa tiedot laitteen viimeisestä käynnistämisestä alkaen (Linux.fi 2020).

Komento**: journalctl -b -1**

Lisäämällä –”luku” parametri, voidaan valita näytettäväksi, joku muu edeltävistä käynnistyskierroksista (Linux.fi 2020).

Komento**: journalctl --list-boots**

Komennolla voidaan selvittää edelliset käynnistyskierrokset ja niitä vastaavat numerot, joita voidaan käyttää aikaisemman komennon yhteydessä (Linux.fi 2020).

Komento**: journalctl –since "YYYY-MM-DD HH:MM:SS"**

Tulostaa kaikki journaliin tallennetut viestit tietystä päivämäärästä ja kellonajasta alkaen (Geeksforgeeks.org 2024a). Huomaa, että komennossa käytetään lainausmerkkejä ja muodon on oltava kuten yllä.

Komento**: journalctl --since "YYYY-MM-DD HH:MM:SS" --until "YYYY-MM-DD HH:MM:SS"**

Tulostaa kaikki journaliin tallennetut viestit tietyllä päivämäärä ja kellonaika välillä (Geeksforgeeks.org 2024a).

Komento**: journalctl -u service**

Käyttämällä -u parametria, voidaan tulostaa kaikki journaliin tallennetut viestit tietyn palvelun osalta (Geeksforgeeks.org 2024a). Muuta kohtaan service, tarkasteltavan palvelun nimi.

Komento**: journalctl -f**

Käyttämällä -f parametria voidaan monitoroida journalia reaaliaikaisesti (Geeksforgeeks.org 2024a). Erityisen hyödyllinen järjestelmän muutosten seuraamisessa, esimerkiksi epäillyssä haittaohjelma tartunnassa.

Komento: **journalctl _SYTEMD_UNIT=sshd.service | grep error**

Komennolla voidaan etsiä SSH-palvelun lokitiedoista viestejä esimerkiksi epäonnistuneista tunnistautumisista (Hussain, S. 2020).

11 Jälkitoimet

Mikäli analysoinnin yhteydessä selviää, että järjestelmä on vaarantunut, niin reagointiprosessissa siirrytään edelleen eristäminen, hävittäminen ja palauttaminen vaiheisiin.

Ensimmäinen toimenpide on eristää laite verkosta ja saada järjestelmä takaisin hallintaan, kun hyökkääjän yhteys laitteeseen saadaan katkaistua. Yhteyksien katkaiseminen estää myös mahdollisen haittaohjelman leviämisen muihin laitteisiin. (Drake, N. 2024b.)

Myös mahdolliset yhteydet varmuuskopiointijärjestelmiin tulee katkaista, jotta mahdollinen haittaohjelma ei pääse leviämään niihin.

Jos olet samassa tilassa laitteen tai serverin kanssa, niin yhteyden katkaiseminen voidaan toteuttaa poistamalla fyysinen verkkokaapeli (Drake, N. 2024b).

Jos saat yhteyden esimerkiksi SSH:n kautta, niin voit käyttää ifconfig -komentoa verkkosovittimen käytöstä poistamiseen (Drake, N. 2024b).

Komento: **ls /sys/class/net/**

Tulostaa saatavilla olevat sovittimet (Drake, N. 2024b).

Komento: **sudo ifconfig verkkosovitin down**

Poistaa valitun sovittimen käytöstä, arvo verkkosovittimeen voisi olla esimerkiksi ”eth0”.

Jos serverillä on käytössä langattomia yhteyksiä, niin myös ne pitää ottaa pois käytöstä.

Komento: **sudo rfkill block all**

Estää langattomat yhteydet (Drake, N. 2024b).

Jos kyseessä on esimerkiksi etäyhteyden päässä oleva serveri ja epäilet organisaation tietojen edelleen vuotavan, niin ainoa ratkaisu lisävahingoilta välttymiseksi voi olla serverin sammuttaminen.

Komento: **sudo shutdown -h now**

Sammuttaa välittömästi serverin (Drake, N. 2024a).

Mikäli kyseessä on järjestelmä, jonka vaarantuminen ei analyysin perusteella ole johtanut siihen, että organisaation tai sen asiakkaiden tietoturvallisuus on vaarantunut, niin paras vaihtoehto on asentaa se ja siihen liittyvät laitteet uudelleen.

Uudelleenasennus tulee tehdä tunnetusta luotettavaksi tiedetystä varmuuskopiosta. Varmuuskopion luotettavuuden arviointia helpottaa, mikäli analyysissa voidaan määritellä tarkka tapahtuma-aika poikkeamalle. (Baker ym. 2005, 3.)

Tärkeää on muistaa, että missään tapauksissa ei tule korjata vaarantunutta järjestelmää tai serveriä esimerkiksi poistamalla hyökkääjän tekemää käyttäjätunnusta ja ottaa järjestelmää sen jälkeen uudelleen käyttöön. Et voi koskaan olla täysin varma, että järjestelmä on jälleen turvallinen. Ainoa järkevä toimintamalli on palauttaa järjestelmä puhtaaksi varmennetusta varmuuskopiosta tai asentaa se täysin uudelleen. (Cooper, E. 2024.)

Mikäli laitteella oli haittaohjelma, niin se voidaan hävittää käyttämällä antivirus-ohjelmaa. Tämän jälkeen palautetaan järjestelmä puhtaaksi arvioidusta varmuuskopiosta, tai sen puuttuessa asennetaan se uudelleen. Palauttamisen jälkeen suoritetaan uusi antivirus-skannaus ja varmistetaan, että järjestelmä on todella puhdas. Tietoturvapoikkeamasta riippuen voidaan joutua poistamaan ja muuttamaan muitakin tietoa poistettavan haittaohjelman lisäksi. Ennen järjestelmän verkkoon laittamista tulisi ainakin vaihtaa esimerkiksi käyttäjätilien salasanat. (Drake, N. 2024b.)

Mikäli analyysissa selviää, että hyökkääjä on saanut haltuunsa suojattavia tietoja, tulee asian selvittämistä jatkaa digitaalisen forensiikan keinoin. Digitaalisella forensiikalla tarkoitetaan rikosteknistä osa-aluetta, jossa keskitytään digitaalisiin laitteisiin tai muihin digitaalisiin tallennusvälineisiin tallentuneiden tietojen hankkimiseen, käsittelyyn, analysointiin ja raportointiin (Interpol 2019). Rikosasioissa tulee aina konsultoida viranomaisia jatkotoimenpiteistä.

Ensitoimena voidaan ottaa talteen katoavaa todistusaineistoa kuten aktiivisten prosessien tiedot.

Komento: **ps aux > aktiiviset_prosessit.txt**

> Tallentaa tekstitiedostoon sen hetkisen listauksen aktiivisista prosesseista (Drake, N. 2024b).

Dokumentaatiot prosesseista voivat olla arvokasta todistusaineistoa, mikäli haitallisia prosesseja on edelleen käynnissä (Drake, N. 2024b).

Voidaan myös ottaa forensiikkatyötä varten taltioita (Engl. Image) laitteen tallennustiloista, jotta järjestelmän sen hetkistä tilaa voidaan myöhemmin analysoida siten, että dataan ei ole

tehty, eikä tehdä muutoksia. Linux forensiikan osalta suosittelen tutustumaan Barry Grundyn ylläpitämään kattavaan käsikirjaan (Grundy, B. 2023).

Tässä vaiheessa tulisi myös viimeistään ottaa yhteyttä kyberturvallisuuteen erikoistuneihin ammattilaisiin. Tämä vaihe on erityisen tärkeä, mikäli vaarantunut laite sisältää suojattavia tietoja, kuten asiakastietoja (Drake, N. 2024b). Tähän liittyen on organisaatiosta ja sen käsittelemästä datasta riippuen erilaisia ilmoitusvelvollisuuksia. Ajantasaista tietoa tietoturvaloukkauksien vaatimista toimenpiteistä löydät Traficomin Kyberturvallisuuskeskuksen sivuilta (Kyberturvallisuuskeskus 2024).

Jälkitoimiin kuuluu myös kiinteästi se, että analysoidaan, miten ja miksi tietoturvapoikkeama tapahtui ja tehdään tarvittavat muutokset, jotta sama ei tapahtuisi enää uudelleen (Drake, N. 2024b).

12 Tulokset ja pohdinnat

Opinnäytetyön aikana täsmentyi miten hankala on löytää kootusti tietoa Linuxin analysoinnista tietoturvapoikkeamatilanteissa. Tietoa kyllä löytyi, mutta hyvin hajanaisesti. Varmasti tähän vaikuttaa se, että perinteisesti Linux on nähty turvallisena käyttöjärjestelmänä.

Mielenkiintoista on kuitenkin se, että on useita alustoja, joissa voidaan kuukausimaksua vastaan harjoitella Linux-pohjaisiin järjestelmiin tunkeutumista. Avoimuus ja haavoittuvuuksien testaamineen lisää kiistatta myös turvallisuutta, mutta voidaanko olettaa, että jokainen Linuxin kanssa töitä tekevä harrastaa eettistä hakkerointia, jotta ymmärtää mistä etsiä merkkejä järjestelmän vaarantumisesta.

Opinnäytetyön tavoitteena oli kehittää vaihe vaiheelta etenevä kirjallinen prosessi tietojen analyysista. Opinnäytetyön laajuuden vuoksi pystyin tuomaan esille vain yleisimmät analyysikohteet, mutta kuitenkin siten, että lähteisiin perustuen tärkeimmät kohteet tuli käsitellyksi. Kirjallisen prosessin lisäksi tuloksena syntyi myös työhön ja läpikäytyyn materiaaliin perustuva muistilista, joka toimii apuna järjestelmän nopeaan tarkistukseen (Liite 1: Muistilista – nopeat tarkistukset).

Tiedon etsiminen ja yhdisteleminen toi itselle valtavasti uutta tietoa järjestelmän toiminnasta ja erityisesti lokitiedostojen sisällöstä. Toivon, että myös opinnäytetyön lukija saa työstä apua oman ammattitaidon kasvattamiseksi.

Lähteet

Analysis of sudo logs 2024. Who gained root access on my Linux system - an analysis of sudo logs. Video. BlueMonkey 4n6. Katsottu 16.9.2024. https://www.youtube.com/watch?v=ZBfZs-gqRcc

Anson, S. 2020. Applied Incident Response. E-kirja. Wiley Data and Cybersecurity.

Baker, S. & Green, P. 2005. Checking UNIX/LINUX Systems for Signs of Compromise. Oxford University, University College London. Viitattu 1.10.2024. https://www.first.org/resources/guides/Checking-UNIX_LINUX-Systems-for-Signs-of-Compromise.pdf

Boelen, M. 2024a. Linux-audit.com. How to deal with a compromised Linux system. Viitattu 7.8.2024. https://linux-audit.com/dealing-with-a-compromised-linux-system/#incident-response-plan

Boelen, M. 2024b. Linux-audit.com. Understanding the output of the stat command. Viitattu 22.10.2024. https://linux-audit.com/filesystems/understanding-the-output-of-the-stat-command/

Boelen, M. 2024c. Linux-audit.com. Troubleshooting CPU usage. Viitattu 22.10.2024. https://linux-audit.com/system-performance/cpu/

Boelen, M. 2024d. Linux-audit.com. Detecting Linux rootkits. Viitattu 22.10.2024. https://linux-audit.com/intrusion-detection-linux-rootkits/

Chkrootkit 2023. Locally checks for signs of a rootkit. Viitattu 21.9.2024. https://www.chkrootkit.org/

Cichonski, P., Millar, T., Grance, T. & Scarfone, K. 2012. Computer Security Incident Handling Guide. NIST Special publication 800-61, Revision 2. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

ClamAV 2024. Open-source antivirus engine for detecting trojans, viruses, malware & other malicious threats. Viitattu 30.10.2024. https://www.clamav.net/

Cooper, E. 2017a. How To Tell If Your Linux Server Has Been Compromised. Viitattu 23.9.2024. https://bash-prompt.net/guides/server-hacked/

Cooper, E. 2017b. Extracting Information From Logs - Part 1. Viitattu 22.10.2024. https://bash-prompt.net/guides/using-logs-1/

Cooper, E. 2021. How To Monitor Network Activity With IPTraf-ng. Viitattu 23.10.2024. https://bash-prompt.net/guides/iptraf/

Cooper, E. 2023a. Bash Oneliners - Count, Sort, and Print Objects. Viitattu 23.10.2024. https://bash-prompt.net/guides/bash-oneliners-sort/

Databasemart.com 2023. Database Mart LLC. How to Check if a Linux Server is Hacked. Viitattu 23.9.2024. https://www.databasemart.com/blog/how-to-check-if-a-linux-server-is-hacked

de Koff, J. 2020. Hackingpassion.com. Determine if Your Linux Computer or Server Is Hacked. https://hackingpassion.com/determine-if-your-linux-computer-or-server-is-hacked/

Domantas, G. 2024. Hostinger. How to check running processes in Linux using ps, top, htop, and atop commands. Viitattu 8.9.2024. https://www.hostinger.com/tutorials/vps/how-to-manage-processes-in-linux-using-command-line

Drake, N. 2024a. Linuxinsider.com. How To Check if Your Linux Server Has Been Hacked. Luettu 17.9.2024. https://www.linuxinsider.com/story/how-to-check-if-your-linux-server-has-been-hacked-177287.html

Drake, N. 2024b. Linuxinsider.com. What To Do if Your Linux Server Has Been Hacked. Luettu 22.10.2024. https://www.linuxinsider.com/story/what-to-do-if-your-linux-server-has-been-hacked-177303.html

Geeksforgeeks.org 2024a. How To Use Journalctl to View and Manipulate Systemd Logs. Viitattu 16.9.2024. https://www.geeksforgeeks.org/how-to-use-journalctl-to-view-and-manipulate-systemd-logs/

Geeksforgeeks.org 2024b. Iftop command in Linux with Examples. Viitattu 23.9.2024. https://www.geeksforgeeks.org/iftop-command-in-linux-with-examples/

Gite, V. 2024. How to check open ports in Linux using the CLI. Viitattu 23.10.2024. https://www.cyberciti.biz/faq/how-to-check-open-ports-in-linux-using-the-cli/

Grundy, B. 2023. The Law Enforcement and Forensic Examiner's Introduction to Linux. A Comprehensive Practitioner's Guide to Linux as a Digital Forensics Platform. Viitattu 15.10.2024. https://www.linuxleo.com/Docs/LinuxLeo_4.97.pdf

Hussain, S. 2020. Xplg.com. Linux Security Must-Read Guide 2020: How to Investigate Suspected Break-in Attempts in Linux. Viitattu 22.10.2024. https://www.xplg.com/linux-security-investigate-suspected-break-in/

Johansen, G. 2020. Digital Forensics and Incident Response. Incident response techniques and procedures to respond to modern cyber threats. 2.painos. E-kirja. Birmingham – Mumbai: Packt Publishing.

Interpol 2019. Global Guidelines for Digital Forensics Laboratories. Viitattu 7.8.2024. https://www.interpol.int/content/download/13501/file/INTERPOL_DFL_GlobalGuidelinesDigitalForensicsLaboratory.pdf

Kenlon, S. 2020. Red Hat. User status and activity monitoring in Linux with GNU acct. Viitattu 17.9.2024. https://www.redhat.com/sysadmin/linux-system-monitoring-acct

Kokonaisturvallisuuden sanasto 2017. Helsinki: Sanastokeskus TSK ry. Viitattu 5.8.2024. https://turvallisuuskomitea.fi/wp-content/uploads/2018/02/Kokonaisturvallisuuden_sanasto.pdf

Kinger, P., Bharti, S., Oliveira, M. 2023. The Linux Threat Landscape Report. Trendmicro.com https://www.trendmicro.com/vinfo/ie/security/news/cybercrime-and-digital-threats/the-linux-threat-landscape-report

Kyberturvallisuuskeskus 2024. Ohjeet ja oppaat organisaatioille ja yrityksille. Viitattu 30.10.2024. https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/ohjeet-ja-oppaat/ohjeet-ja-oppaat-organisaatioille-ja-yrityksille.

Levinas, M. 2022. Cherryservers.com. A Complete Guide to Linux Bash History. Viitattu 24.9.2024. https://www.cherryservers.com/blog/a-complete-guide-to-linux-bash-history

Linux-audit.com 2024. How to stop all processes of a single user. Viitattu 22.10.2024. https://linux-audit.com/processes/faq/how-to-stop-all-processes-of-a-single-user/

Linuxconfig.org 2023. Who Linux command: Explained. Viitattu 17.9.2024. https://linuxconfig.org/who-linux-command-explained

Linux.fi 2020. Systemd. Viitattu 16.9.2024. https://www.linux.fi/wiki/Systemd

Linux.fi 2021. Proc-tiedostojärjestelmä. Viitattu 23.9.2024. https://www.linux.fi/wiki/Proc-tiedostoj%C3%A4rjestelm%C3%A4

Linux.fi 2023. Paketinhallintajärjestelmä. Viitattu 7.10.2024. https://www.linux.fi/wiki/Paketinhallintaj%C3%A4rjestelm%C3%A4

Rkhunter 2018. The Rootkit Hunter project. Viitattu 30.10.2024. https://rkhunter.sourceforge.net/

Rowland, G. 2021. Sandfly Security. Command Line Compromise Detection for Linux. Viitattu 8.9.2024. https://sandflysecurity.com/linux-compromise-detection-presentation.pdf

Sandfly Security 2018. Linux Compromise Assessment Command Cheat Sheet. Viitattu 7.10.2024. https://sandflysecurity.com/linux-compromise-detection-command-cheatsheet.pdf

Sandfly Security 2019. Using Linux utmpdump for Forensics and Detecting Log File Tampering. Viitattu 22.10.2024. https://sandflysecurity.com/blog/using-linux-utmpdump-for-forensics-and-detecting-log-file-tampering/

Sandfly Security 2021. Linux Command Line Forensics and Intrusion Detection Cheat Sheet. Viitattu 8.10.2024. https://sandflysecurity.com/blog/compromised-linux-cheat-sheet/

Sheward, M. 2018. Hands-on Incident Response and Digital Forensics. E-kirja. BCS Learning & Development Limited, 2018.

Stocker, S. 2024. Networkworld.com. Tracking user logins on Linux. Viitattu 17.9.2024. https://www.networkworld.com/article/3523730/tracking-user-logins-on-linux.html

Thatipalli, A. 2023. Security.packt.com. How to perform Digital Forensic analysis on Linux Machines. Viitattu 19.10.2024. https://security.packt.com/how-to-perform-digital-forensic-analysis-on-linux-machines/

Tutorialspoint 2024. Faillog – Unix, Linus command. Viitattu 22.9.2024. https://www.tutorialspoint.com/unix_commands/faillog.htm.

Ubuntu 2024a. Choose the cloud architecture that suits you best. Viitattu 11.8.2024. https://ubuntu.com/cloud

Ubuntu 2024b. Viewing and monitoring log files. Viitattu 23.9.2024. https://ubuntu.com/tutorials/viewing-and-monitoring-log-files#1-overview

Ubuntu 2024c. CVE reports. Viitattu 19.10.2024. https://ubuntu.com/security/cves

Ubuntu wiki 2020. Did I Just Get Owned?. Viitattu 23.9.2024. https://wiki.ubuntu.com/BasicSecurity/DidIJustGetOwned

Verma, K. 2024. Geeksforgeeks .org. Netstat command in Linux. Viitattu 18.10.2024. https://www.geeksforgeeks.org/netstat-command-linux/

Virustotal 2024. Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches. Viitattu 30.10.2024. https://www.virustotal.com/gui/home/upload

Wake, T. 2023a. Linux Incident Response - Using ss for Network Analysis. Viitattu 24.10.2024. https://www.sans.org/blog/linux-incident-response-using-ss-for-network-analysis/

Wake, T. 2023b. Linux Intrusions – A Growing Problem. Viitattu 24.10.2024. https://www.sans.org/blog/linux-intrusions-a-growing-problem/

Wake, T. 2023c. Linux Incident Response - A Guide to syslog-ng. Viitattu 24.10.2024. https://www.sans.org/blog/linux-incident-response-a-guide-to-syslog-ng/

Liitteet

Liite 1: Muistilista – nopeat tarkistukset

Muistilista - nopeat tarkistukset

Taltioi tekemäsi toimenpiteet

Katkaise verkkoyhteydet jos mahdollista

Lisätiedot ja parametrit komennon manuaalisivulta **man komento**

Prosessit - Selvitä runsaasti resursseja käyttävät ja tuntemattomat

Komento	**Mitä tekee**
ps -auxwf	Käynnissä olevat
top	Reaaliaikainen resurssien käyttö
killall	Prosessin ja aliprosessien pysäyttäminen

Verkkotoiminta - Saapuvat ja lähtevät yhteydet - epätavallinen liikenne

Komento	**Mitä tekee**
netstat -anp	Yleiskuva verkkoyhteyksistä
iftop	Verkkosovittimen reaaliaikainen käyttö

Hakemistot - Käytetty, muutettu tai luotu lähellä tapahtuma-aikaa

Komento	**Mitä tekee**
ls -lharRp	Tiedot hakemiston sisällöstä

Tarkista yleisesti kohteena olevat hakemistot

/tmp	/lib*
/var/tmp	/usr/lib*
/dev	/etc
/dev/shm	/var/log
/bin	/var/run
/sbin	/var/spool
/usr/bin	/home/user (Käyttäjien kotihakemistot)

Tiedostot - Käytetty, muutettu tai luotu - suoritettavat - epätavallisesti nimetyt

Aikaleimat	
Accessed	Tiedostoa käytettiin viimeksi
Modified	Tiedostoa muokattu viimeksi
Change	Tiedoston metatietojen viimeisin muutos
Birth	Tiedosto luotu
Komento	**Mitä tekee**
file	Selvitä tuntemattoman tiedoston tiedostotyyppi
stat	Aikaleimat

Käyttäjät - Uudet, tuntemattomat, uudelleen aktivoidut, kirjautumistiedot

Komento	**Mitä tekee**
cat /etc/passwd	Tiedot käyttäjätileistä
w	Muut kirjautuneena olevat käyttäjät
who -H -a	Tietoa kirjautuneista käyttäjistä
last	Kirjautumishistoriatietoja

Lokitiedostot - Sijainti **/var/log** - tapahtuma-ajalta - muokatut -poistetut

Vaiheet
1. Hae tiedot lokitiedostosta ja tulosta ne ruudulle.
2. Tutki sisältöä, ymmärrä mitä se sisältää ja millä tavalla data esitetään.
3. Suodata oleellisia tietoja (**grep**).

Suorita virusskannaus, antivirus esim. **https://www.clamav.net/**

cat /etc/os-release, tarkista haavoittuvuudet **https://ubuntu.com/security/cves**

md5sum tiedostonimi, lataa tiedoston tiiviste vertailuun **https://www.virustotal.com/gui/home/upload**