Linux-palvelinten tietoturva

LuK-tutkielma
Turun yliopisto
Tulevaisuuden teknologioiden laitos
Tietojenkäsittelytiede
2020
Maks Turtiainen

TURUN YLIOPISTO
Tulevaisuuden teknologioiden laitos

Maks Turtiainen: Linux-palvelinten tietoturva

LuK-tutkielma, 25 s.
Tietojenkäsittelytiede
Marraskuu 2020

Miten Linux-palvelinylläpitäjä voi suojautua kyberhyökkäyksiltä? Tutkielman haasteena on tarkastella keinoja suojautua tyypillisimmiltä kyberhyökkäyksiltä Linux-palvelinympäristössä.
Palvelimeen kohdistuneen kyberhyökkäyksen seuraukset voivat olla erittäin tuhoisat. Tyypillisesti seuraukset ovat palveluntarjoajille rahallisia ja palveluiden käyttäjille palvelun käyttökatkoja, mutta erityisen tuhoisassa hyökkäyksessä voidaan puhua ihmisten terveyteen kohdistuvasta vaarasta.
Linux on yleisin käyttöjärjestelmä palvelimissa. Tästä syystä tutkielma tarkastelee keinoja suojautua kyberhyökkäyksiltä nimenomaan Linux-palvelinten näkökulmasta.
Tutkielmassa esitellään muutamia yleisimpiä kyberhyökkäyksiä ja millaisia resursseja nämä uhkaavat. Tämän jälkeen esitellään muutamia yleisimpiä menetelmiä vahventaa tietoturvaa Linux-palvelimilla ja miten ne auttavat suojautumaan tutkielmassa esitellyiltä kyberhyökkäyksiltä. Menetelmiä suojautua kyberhyökkäyksiltä käsitellään ensin yleisellä tasolla, jonka jälkeen käydään käytännönläheisesti läpi miten menetelmää voi soveltaa Linux-palvelinympäristössä.

Asiasanat: Tietoturva, Linux, Palvelin

Sisällys

Kuvat

1 Johdanto

Palvelimen vaarantunut tietoturva voi asettaa alttiiksi tuhansien, jopa miljoonien ihmisten tietoja. Liiketoiminnan kontekstissa palvelimen tietoturvan pettäminen voi johtaa miljoonien eurojen menetykseen liiketoimintakriittisen sovelluksen ollessa pois käytöstä. Pahimmillaan voidaan puhua ihmishenkien menetyksen vaarasta kun kyseessä on esimerkiksi terveydenhuollon infrastruktuuriin kuuluva palvelin. Palvelinten tietoturvaa voi edellä mainittujen seikkojen vuoksi pitää huomattavasti tärkeämpänä kuin esimerkiksi tavallisen työpöytätietokoneen.

Palvelinten tietoturva on tärkeää myös palvelinten luonteen vuoksi. Palvelimen on oltava helposti saatavilla asiakassovelluksilleen. Palvelimen suora saatavuus internetissä ja palvelut, joita palvelin ajaa, luovat palvelimen hyökkäyspinta-alasta korkeamman kuin esimerkiksi tavallisen työpöytätietokoneen.

Valtaosa palvelimista käyttää käyttöjärjestelmänään Linuxia. W3Cookin analyysin mukaan jopa 96,4 % julkisista web-palvelimista käyttää Linux-käyttöjärjestelmää[1]. Tästä syystä tutkielma keskittyy nimenomaan Linux-palvelinten tietoturvaan.

Tutkielmani haasteena on esitellä kuinka Linux-palvelimen ylläpitäjä voi suojautua tyypillisimmiltä tietoturvauhilta. Luvussa 2 johdattelen Linuxin ja palvelinten perusteisiin, jonka jälkeen luvussa 3 käyn läpi tärkeimpiä konsepteja tietoturvasta puhuttaessa ja tyypillisimpiä tietoturvauhkia sekä kyberhyökkäyksiä. Luvussa 4 esittelen kuinka näiltä tietoturvauhilta ja kyberhyökkäyksiltä voi suojautua.

2 Linux ja palvelimet

2.1 Linux

Linux on perhe käyttöjärjestelmiä, jotka perustuvat Linux-käyttöjärjestelmäytimeen eli kerneliin. Linux-kernelin ensimmäisen version julkaisi Linus Torvalds vuonna 1991 opiskellessaan Helsingin Yliopiston Tietojenkäsittelytieteen laitoksella. Linux-käyttöjärjestelmällä viitataan mihin tahansa Linux-kerneliä käyttävään käyttöjärjestelmään. Linux on UNIX-tyyppinen käyttöjärjestelmä, mutta ei jaa samaa koodikantaa UNIX-käyttöjärjestelmien kanssa, ei ole sertifioitu eikä noudata Single UNIX Specification -standardia. Linux-kerneliä käyttävät käyttöjärjestelmät paketoidaan yleensä Linux-jakeluksi, jotka useimmiten sisältävät kernelin lisäksi kokoelman ohjelmistoja sekä paketinhallinnan. Linux-kerneli ja useimmat Linux-jakelut ovat vapaata lähdekoodia. [2]

Työpöytäkäytössä Microsoft Windows on suosituin käyttöjärjestelmä, Linuxia käyttävät vuonna 2020 vain 1,53% työpöytäkäyttäjistä [3]. Mobiililaitteissa suosituin käyttöjärjestelmä on Googlen Android, jonka voi katsoa olevan Linux-jakelu sen käyttäessä muokattua versiota Linux-kernelistä. [4] Mobiililaitteista Androidia käytti 85 % vuonna 2018 [3]. Vuosina 2017–2019 maailman 500 tehokkaimmasta supertietokoneesta kaikki käyttävät Linuxia. [5]. Internetin julkisista web-palvelimista vuonna 2015 Linuxia käyttää 71,6%-96,4% riippuen lähteestä. [6] [1]

Linux-jakeluiden kotisivujen kävijämäärien perusteella suosituimmat 3 Linux-

jakelua ovat MX Linux, Manjaro ja Mint (21.10.2020). [7] Useimpien Linux-jakeluiden vapaan saatavuuden vuoksi on vaikea arvioida todellisia käyttäjämääriä, mutta suosituimpia Linux-jakeluita palvelinkäytössä lienevät Red Hat Enterprise Linux, SuSE, Ubuntu, Debian sekä CentOS. [8]

Perusteita suurelle Linuxin käytölle palvelimissa on arvioitu olevan vakaus ja luotettavuus, turvallisuus, muokattavuus, lähdekoodin avoimuus sekä kustannukset. [9]

2.2 Palvelimet

Palvelin on tietokonejärjestelmä, joka tarjoilee palveluja, dataa tai muita resursseja asiakastietokoneilleen tai –sovelluksilleen, useimmiten internetin välityksellä. Palvelimet koostuvat yleensä palvelinkäyttöön tarkoitetusta tietokoneesta sekä palvelinkäyttöön tarkoitetusta käyttöjärjestelmästä. Erityyppisiä palvelimia ovat mm. tiedostopalvelimet, tulostinpalvelimet, sovelluspalvelimet, DNS-palvelimet, sähköpostipalvelimet, tietokantapalvelimet ja web-palvelimet.

Järjestelmän arkkitehtuuria, jossa palvelin palvelee asiakaskonetta, kutsutaan asiakas–palvelin malliksi. Tyypillisesti palvelimet ja asiakkaat keskustelevat keskenään pyyntö– ja vastausperiaatteella. Asiakas lähettää pyynnön palvelimelle, johon palvelin vastaa. Esimerkiksi asiakkaan web-selain lähettää HTTP-pyynnön palvelimelle, johon palvelin vastaa HTML:n muodossa.

Käytännössä mikä tahansa tietokone voi olla palvelin, mutta yleensä palvelimet ovat palvelinkäyttöön tarkoitettuja tietokoneita, jotka sijaitsevat palvelinsalissa. Palvelintietokoneet koostuvat osista, jotka ovat luotettavampia kuin kuluttajatietokoneissa. Useimmiten palvelintietokoneet ovat räkkiin sopivassa vaakamallisessa kotelossa ja räkissä palvelimia voi olla useita kymmeniä. Suurimmissa palvelinsaleissa voi olla satoja räkkejä. Palvelin ei tarvitse näyttöä tai syöttölaitteita kuten näppäimistöä tai hiirtä muuta kuin huoltotoimenpiteissä, joten tilan ja kustannusten säästämiseksi näitä harvemmin on palvelimissa. Palvelinta kontrolloidaan etänä esimer-

kiksi SSH:n välityksellä, web-pohjaisesta käyttöliittymästä tai jollakin kaupallisella ratkaisulla kuten Microsoft Managment Consolella.

Palvelintietokoneissa osien kokoonpano pyrkii mahdollisemman suureen toimintavarmuuteen. Tekniikoita, joilla toimintavarmuutta pyritään takaamaan, ovat muun muassa virheenkorjaava muisti (ECC), osien lennosta vaihto, kriittisten osien tuplana saatavilla oleminen, RAID–levyjärjestelmät sekä virransyötön takaaminen akustolla (UPS) tai jopa generaattoreilla. Tyypillinen palvelin pysyy toimintakykyisenä vaikka siitä hajoaisi virtalähde tai tallennuslaite kuten kovalevy tai vaikka koko rakennuksesta katkeaisivat sähköt. [10]

Palvelin voi olla myös toisen palvelimen tarjoama virtuaalipalvelin. Tässä tapauksessa fyysinen palvelin toimii virtuaalipalvelinalustana ja voi ylläpitää useita kymmeniä virtualisoituja käyttöjärjestelmiä. Nykyisin virtuaalipalvelinalusta koostuu useista fyysisistä palvelimista tai jopa palvelinsaleista ja resursseja pystyy allokoimaan virtuaalipalvelimille joustavasti (klusterointi).

Käytännössä varsinaisten fyysisten palvelinten ja palvelinsalien ylläpito on keskittynyt muutamille suurille palveluntarjoajille, joilla on käytössään useita kymmeniä palvelinsaleja. Harvat palvelinresursseja tarvitsevat ylläpitävät itse omia fyysisiä palvelimiaan omissa tiloissaan. Tyypillisesti resurssit vuokrataan palveluntarjoajalta. Palvelinresurssit voivat olla virtuaalipalvelimia, fyysisiä palvelimia palveluntarjoajan tiloissa tai pääsy yhteisessä käytössä olevalle palvelimelle. [11]

2.3 Tyypillinen Linux-palvelinkonfiguraatio

Esittelen seuraavaksi kuvitteellisen, mutta realistisen esimerkin palvelinarkkitehtuurin toteutuksesta laitteistosta ohjelmistoihin, loppukäyttäjästä ylläpitoon. Päämääränä on tarjota palvelinresurssit keskisuuren yrityksen web-sovellukselle.

Yritys vuokraa palveluntarjoajalta virtuaalipalvelimen. Palveluntarjoajalla on useita suuria palvelinsaleja. Kyseisen virtuaalipalvelimet tarjoillaan yhdestä pal-

velinsalista, jossa on 100 kpl räkkejä, joissa jokaisessa on 10 palvelintietokonetta. Yksittäisen räkin varavirtalähteenä on akusto, joka sijaitsee räkin alaosassa. Koko palvelinsalin varavirranlähteenä toimii diesel-aggregaatti. Palvelinsalin palvelintietokoneista on allokoitu virtuaalipalvelinten vuokraamiseen 10 räkin eli 100 palvelintietokoneen verran. Palvelintietokoneet ovat identtisiä keskenään. Suuren kapasiteetin tarpeen vuoksi niissä on useampi prosessori sekä runsaasti virheenkorjaavaa keskusmuistia. Tallennustilana toimii 10 SSD-levyä. Levyt ovat kytketty RAID 6 järjestelmään tarjoten näin tallennustilaa 80% levykapasiteetista kahden levyn redundanssilla. Levyt on lennosta vaihdettavia, joten levyn rikkoontuessa palvelimen toiminta ei keskeydy. Yhdessä palvelinkoneessa on 2 lennosta vaihdettavaa virtalähdettä.

Virtuaalipalvelinalustat käyttävät Red Hat Enterprise Linuxia käyttöjärjestelmänään. Virtualisointiin käytetään vapaan lähdekoodin QEMU–projektia. Virtuaalikoneita on keskimäärin 10 yhdellä fyysisellä palvelimella.

Asiakasyritys vuokraa yhden virtuaalipalvelimen, jolle allokoidaan 1/10 fyysisen palvelimen resursseista. Virtuaalipalvelimen käyttöjärjestelmänä on Ubuntu Linux. Virtuaalipalvelinta ohjataan SSH-yhteiden välityksellä. Yrityksen web-sovellus on Python–ohjelmointikielellä kirjoitettu. Web-sovelluskirjastona on käytetty Djangoa. Djangon sisäinen HTTP-palvelinsovellus tarjoilee sisällön ainoastaan IP:lle 127.0.0.1 porttiin 3000. Samalla virtuaalipalvelimella ajetaan myös Nginx–nimistä HTTP-palvelinta, joka toimii käänteisenä välityspalvelimena paikallisen HTTP-palvelimen ja ulkoverkon välillä. Nginx HTTP-palvelin välittää Django-palvelimen portin 3000 ulkoverkkoon portteihin 80 ja 443.

Palvelinresursseja vuokraava yritys tarjoaa myös DNS-nimipalveluita. Virtuaalipalvelimen IP:lle allokoidaan domain *esimerkki.fi*.

Asiakasyrityksen web-sovellus on nyt saatavilla HTTP (S) -protokollan ylitse osoitteesta *esimerkki.fi* portista 80 tai 443. Asiakasyrityksen asiakkaat vierailevat

web-selaimellaan osoitteessa *esimerkki.fi*. Asiakkaan tietokone lähettää ensin DNS-tiedustelun ja saa vastaukseksi virtuaalipalvelimen IP-osoitteen. Tämän jälkeen web-selain lähettää HTTP–pyynnön kyseiseen IP:seen porttiin 80. Virtuaalipalvelimella pyörivä Nginx välittää pyynnön Djangon HTTP-palvelimelle, joka vastaa pyyntöön HTML-koodilla. Nginx välittää tämän HTML:n takaisin asiakkaan web-selaimelle ja web-selain renderöi HTML-koodista web-sivuston.

3 Tietoturva

Tietoturvalla tarkoitetaan tietokonejärjestelmien ja verkkojen suojelemista elektronisten resurssien varkauksilta, ohjelmistojen ja laitteiden vahingoilta sekä tahallaan aiheutetuilta häiriöiltä palvelujen toimintakyvyssä. Suojautumaan pyritään myös palvelujen väärinkäytöksiltä. Tietoturvan merkitys on kasvanut nopeasti digitalisaation myötä. [12]

3.1 CIA-malli

Tärkeimpiä konsepteja tietoturvasta puhuttaessa on luottamuksellisuus, eheys sekä saatavuus. Tämän johdosta yksi tärkeimmistä malleista kuvata tietoturvan osa-alueita on CIA-malli (Confidentality, Integrity, Availability). Malli antaa viitekehyksen keskustellessa siitä, millainen jokin tietoturvauhka on. Mallin mukaisesti jokin tietoturvauhka kohdistuu aina yhteen tai useampaan CIA-mallin osa-alueista.

ISO/IEC 27000:2018(E) -standardi [13] määrittelee CIA-mallin kohdat seuraavasti:

- **Luottamuksellisuus**: "Property that information is not made available or disclosed to unauthorized individuals, entities, or processes"
- **Eheys**: "Property of accuracy and completeness"
- **Saatavuus**: "Property of being accessible and usable on demand by an authorized entity"

Luottamuksellisuudella tarkoitetaan siis sitä, että tietoa voivat nähdä vain ne, joilla on siihen oikeus. Eheys tiedon oikeuden ja täydellisyyden varmistamista siltä, ettei tietoa ole muokattu luvattomasti. Saatavuudella puolestaan tarkoitetaan sitä, että tieto on saavutettavissa siihen oikeutettujen tahojen toimesta. [14]

3.2 Haavoittuvuudet ja kyberhyökkäykset

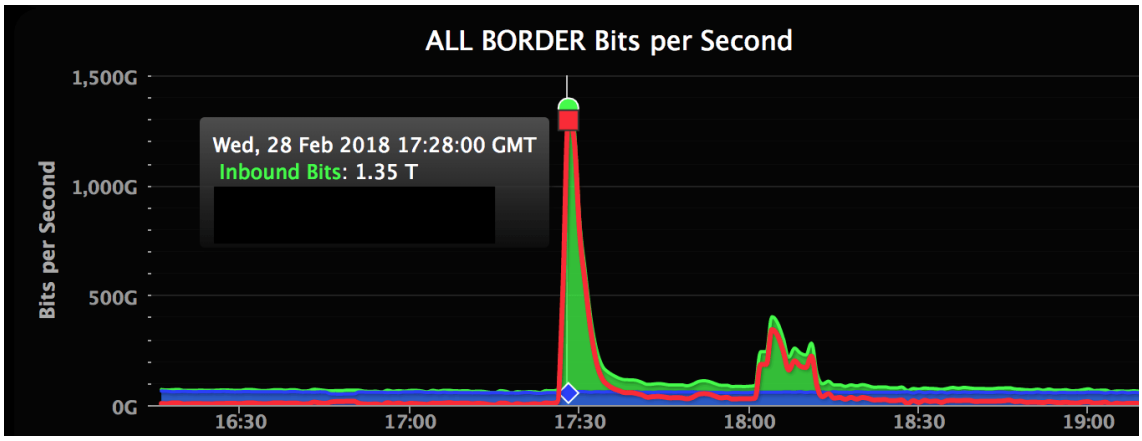
Tietoturva-aukolla eli haavoittuvuudella tarkoitetaan heikkoutta tietokonejärjestelmässä, jonka avulla hyökkääjän on mahdollista päästä tekemään järjestelmässä jotakin mitä hänen ei pitäisi. Kyberhyökkäys tarkoittaa varsinaista toimenpidettä, jossa hyökkääjä käyttää haavoittuvuutta päästäkseen järjestelmään. [12]

Esittelen seuraavaksi yleisimpiä kyberhyökkäyksiä, mitä haavoittuvuutta ne hyödyntävät ja mitä CIA-mallin kohtaa ne vaarantavat. Merkitsen jokaisen esittelemäni hyökkäyksen tunnisteella ja numeroinnilla Hx, jotta niihin on helpompi palata myöhemmissä luvuissa.

3.2.1 H1: Palvelunestohyökkäys

Palvelunestohyökkäyksen (eng. *Denial of Service, DoS*) tarkoitus on saada jokin verkossa oleva resurssi pois käytöstä häiritsemällä tätä internetin välityksellä. Tyypillisesti tämä saavutetaan häiritsemällä kohdetta lukuisilla palvelupyynnöillä, joiden tarkoitus on saada resurssi ylikuormitettua, jonka jälkeen tavalliset resurssin käyttäjät eivät enää pääse tähän käsiksi. Palvelunestohyökkäys estää palvelun saatavuuden, joten se hyökkää CIA-mallin A-kohtaan.

Palvelintietokoneella on lukuisia rajallisia resursseja kuten kaista, levytila tai suoritinaika. Hyökkääjä voi esimerkiksi lähettää toistuvasti palvelupyyntöjä ladatakseen suuren tiedoston palvelimelta ja näin ollen tukkia kaistan tai hyökkääjä voi lähettää toistuvia palvelupyyntöjä johonkin resurssiin, jonka tietää olevan raskas



Kuva 3.1: Github DDoS 2018 [16]

suorittimelle ja näin ollen ylikuormittaa suorittimen.

Todellisuudessa harvalla hyökkääjällä on käytössään sellaisia resursseja, joilla olisi mahdollista tukkia jonkin kaupallisen toimijan palvelinresurssit. Tämän vuoksi nykyään yleisempi tapa toteuttaa palvelunestohyökkäys on tehdä se hajautetusti (DDoS, Distibuted Denial of Service). Hajautetussa palvelunestohyökkäyksessä hyökkääjällä on käytössään useista internetiin yhdistetyistä tietokoneista muodostuva bottiverkko. [15]

Kuvassa 3.1 toistaiseksi toiseksi suurimman palvelunestohyökkäyksen piikki kaistankäytössä hyökkäyksen aikana. Hyökkäys on vuodelta 2018 ja se kohdistui GitHubiin. Poikkeuksellisesti hyökkäyksessä ei käytetty bottiverkkoa vaan hyökkääjä vahvisti omia häirintäpyyntöjään useilla tuhansilla väärinkonfiguroiduilla Memcached-palvelimilla. [16]

3.2.2 H2: Takaovet

Takaovi (eng. *Backdoor*) on keino ohittaa järjestelmän tyypilliset autentikointimenetelmät. Tyypillisesti takaovi on tietokoneohjelma joka antaa hyökkääjälle hallinnan kohdejärjestelmästään. Esimerkiksi Linux-järjestelmässä takaovella yritetään saada tilanne aikaan, jossa hyökkääjä voi kirjautua sisään kohdejärjestelmälle *root-*

oikeuksilla käymättä kuitenkaan läpi normaalia autentikaatioprosessia. Takaovi on ylimääräinen ohjelma, jonka ei kuuluisi olla kohdejärjestelmässä, joten lähtökohtaisesti takaovi vaarantaa CIA-mallin I-osan. Mikäli takaovi mahdollistaa hyökkääjälle *root*-käyttäjän oikeudet, hyökkääjällä on täysi hallinta kohdejärjestelmästä. Edellä mainitun tilanteen voi katsoa vaarantavan kaikki CIA-mallin osa-alueet.

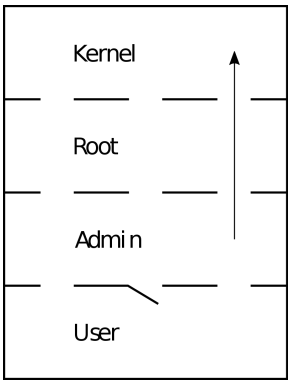
Takaovi asennetaan kohdetietokoneelle esimerkiksi niin kutsutun troijalaisen mukana. Troijalainen on viattomaksi naamioitu haittaohjelma, esimerkiksi peli, jonka mukana on kuitenkin ohjelmakoodia, joka ei kuulu peliin kuten esimerkiksi takaovi.

Takaovia asennetaan myös jo muilla keinoin onnistuneen hyökkäyksen päätteeksi taatakseen hyökkääjällä pääsyn kohdejärjestelmään tulevaisuudessa. [17]

3.2.3 H3: Verkon kuuntelu

Verkon kuuntelu (eng. *Network Eavesdropping*) on keino salakuunnella jotakin tahoa analysoimalla verkkoliikennettä. Useimmiten liikenne on salattu, joten ensin on onnistuttava murtautumaan salauksen läpi. Liikenne voi myös kulkea kaapeleita pitkin, joihin pääsy on fyysisesti estetty ja itse verkkoliikenne on salaamatonta. Verkon kuuntelulla lähtökohtaisesti hyökätään CIA-mallin C-kohtaan, sillä hyökkääjä näkee tietoja, jotka eivät ole hänen katseltavakseen tarkoitettu. Verkon kuuntelun tuloksena voidaan saada käsiin tietoja, joilla onnistutaan tekemään jokin muu kyberhyökkäys.

Langattomien verkkojen aikana verkon kuuntelu ei vaadi enää fyysistä pääsyä kaapeleihin, joten tarpeeksi suurella vastaanottimella hyökkääjä voi toteuttaa hyökkäyksen hyvinkin kaukaa. Tätä ongelmaa on korjattu vahvoilla salauksilla langattomassa verkkoliikenteessä. Varsinkin WLAN:n alkuaikoina salaukset olivat heikkoja tai niitä ei ollut lainkaan ja tämä oli suuri ongelma. [12]



Kuva 3.2: Diagrammi eskalaatiohyökkäyksestä [19]

3.2.4 H4: Eskalaatiohyökkäys

Eskalaatiohyökkäys (eng. *Privilege escalation*) on keino saada ylimääräisiä käyttäjäoikeuksia ja näin päästä käsiksi resursseihin, joihin kyseinen käyttäjä ei normaalisti olisi oikeutettu. Tyypillisesti tämä toteutetaan hyödyntämällä bugia, suunnitteluvirhettä tai konfigurointivirhettä käyttöjärjestelmässä tai sovelluksessa.

Käyttöoikeuksien eskalaatio voi tapahtua joko vertikaalisesti tai horisontaalisesti. Vertikaalisessa eskalaatiohyökkäyksessä pyritään saamaan korkeampia käyttöoikeuksia kun taas horisontaalisessa eskalaatiohyökkäyksessä pyritään saamaan jonkun toisen saman tason käyttäjän oikeuksia. Linux-järjestelmissä vertikaalisessa eskalaatiohyökkäyksessä yritetään tyypillisesti saada *root*-käyttäjän oikeudet. [18]

3.2.5 H5: Injektiohyökkäys

Injektiohyökkyksessä (eng. *Code Injection*) hyökkääjä onnistuu haavoittuvuuden avulla suorittamaan kohdejärjestelmässä koodia, jota hänen ei olisi tarkoitus pystyä suorittamaan. Haavoittuvuus johtuu ohjelmointivirheestä ja tyypillisesti tilanteissa, jossa palvelin prosessoi käyttäjän syötettä ja tätä syötettä ei sarjoiteta oikein. Injektiohyökkäys voi pahimmillaan ja varsinkin hyökkäyksen H4 kanssa johtaa kohteen täyden kontrollin päätymiseen hyökkääjälle, joten injektiohyökkäys vaarantaa CIA-mallin jokaisen kohdan. Erilaisia injektiohyökkäyksiä ovat mm. SQL-injektio, web-

sovelluksen ohjelmointikielen (esim. PHP) injektio ja sivustojen välinen skriptaus (eng. *Cross-site scripting*).

Verkkosovellukset ovat usein injektiohyökkäyksen kohteena. Haavoittuvuus on useimmiten web-sovelluksessa, mutta saattaa olla myös palvelinsovelluksessa. Tyypillisessä haavoittuvuudessa web-sovellus prosessoi verkkosivuston käyttäjän syötettä ilman minkäänlaista validointia tai hyvin vähäisellä sellaisella. [20]

3.2.6 H6: Väsytyshyökkäys

Väsytyshyökkäyksessä (eng. *Brute-force attack*) hyökkääjä käy systemaattisesti läpi kaikki mahdolliset vaihtoehdot salasanan tai salausavaimen arvaamiseksi. Salasana, jota yritetään arvata, voi olla esimerkiksi Linuxin käyttöjärjestelmän käyttäjän salasana tai salauksen purkuun tarvittava salasana. Väsytyshyökkäyksellä pyritään arvaamaan jokin tieto, joka ei ole tarkoitettu hyökkääjän nähtäväksi, joten väsytyshyökkäys vaarantaa CIA-mallin kohdan C.

Jotta väsytyshyökkäys voitaisiin toteuttaa kohteen ei tulisi asettaa rajoituksia sille kuinka monta kertaa salasanaa voidaan kokeilla ja jotta hyökkäys voitaisiin toteuttaa kohtuullisessa ajassa, eri salasanojen kokeilu pitäisi onnistua tekemään nopeasti. Käytännössä mikä tahansa salasana on arvattavissa mikäli salasanan kokeilukertoja ei ole rajoitettu. Tämä voi tosin viedä epäkäytännöllisen pitkän ajan kuten esimerkiksi tuhansia vuosia.

Väsytyshyökkäystä voidaan tehostaa monin tekniikoin kuten esimerkiksi sanalistoilla. Ihmisten valitsemat salasanat ovat yleensä luonnollisten kielien sanoja tai niiden yhdistelmiä tai muunnelmia. Väsytyshyökkäyksessä voidaan kokeilla ensin nämä ja jos tämä ei onnistu niin jatkaa satunnaisten merkkijonojen kokeiluun. [21]

3.2.7 H7: Laitteen varkaus

Tietoturvan kontekstissa laitteen varkaudella pyritään saamaan haltuun tietoa, jota laitteilla säilytetään. Tällainen laite voi olla esimerkiksi kannettava tietokone, muistitikku tai palvelinsalin hävitettäväksi menossa oleva kovalevy. Tieto johon hyökkääjä haluaa päästä käsiksi voi olla jo suoraan varastettavalla laitteella tai varastettava laite voi sisältää vain esimerkiksi salasanan tai salausavaimen, jota voi käyttää murtautumaan jollekin muulle resurssille. Tyypillisesti laitteen varkaus vaarantaa CIA-mallin kohdan C-kohdan, sillä hyökkääjä pyrkii pääsemään käsiksi tietoon johon hänellä ei ole oikeuksia. Toisaalta, mikäli laite varastetaan sen toivossa, että se sisältäisi esimerkiksi jonkin palvelimen käyttäjän salasanan niin laitteen varkaus voi vaarantaa CIA-mallin kaikki kohdat. [21]

4 Kyberhyökkäyksiltä suojautuminen

Esittelen seuraavaksi keinoja suojautua aikaisemmin mainittuja kyberhyökkäyksiä vastaan Linux-palvelinten näkökulmasta. Tarjoan myös konkreettisia komentoesimerkkejä näiden toteutukseen Linuxilla.

4.1 Salaus

Tallennustila ja tietoliikenne voidaan kryptografisesti salata. Salaamalla sekokielistä tietoa muutetaan muotoon, jossa vain tarvittavan avaimen haltija pystyy lukemaan tietoja selkokielisinä.

Salaamalla tietoa voidaan estää päätymästä sellaisiin käsiin mihin se ei ole tarkoitettu. Salauksella vahvennetaan tietoturvaa CIA-mallin C-kohdan osalta. Salaamisella voidaan suojautua tietojen päätymiseltä vääriin käsiin esimerkiksi tietokoneen varkauden tai verkon kuuntelun yhteydessä. [14]

4.1.1 Tallennustilan salaaminen

Tallennustilan tai yksittäisten tiedostojen salaamisella pyritään ehkäisemään tietojen päätyminen vääriin käsiin tilanteessa, jossa laite, jolle tieto on tallennettu, päätyy tahoille, joiden ei ole tarkoitus nähdä tietoja. Tilanne on tämä esimerkiksi

hyökkäyksessä H7. Tallennustilan tai tiedostojen salaamisella voidaan myös pyrkiä suojelemaan jotakin erittäin salaista osaa tiedoista, jonka salausta pidetään purettuna vain silloin kun tietoja tarvitaan. Tällaisessa tapauksessa tietojen on mahdollista pysyä pois vääristä käsistä jopa silloin kun hyökkääjä on saanut muutoin täyden hallinnan tietoja säilyttävästä palvelimesta, kuten hyökkäyksissä H2, H4 ja H6. [14]

DMCrypt on Linux-kernelin levysalausjärjestelmä. Järjestelmä on osa laajempaa Device Mapper rajapintaa, josta järjestelmän nimikin on johdettu (**D**evice **M**apper **crypt**). Device Mapper on rajapinta, jolla voi osoittaa virtuaalisia laitetiedostoja fyysisiin laitetiedostoihin. Tätä teknologiaa DMCrypt hyödyntää salauksessaan. Varsinainen levyosion tai laitteet laitetiedosto on salattu ja sellaisenaan käyttökelvoton. Purkaessaan salauksen DMCrypt luo virtuaalisen laitetiedoston, jossa tieto on selkokielisenä ja tämä virtuaalinen osio on valmis liitettäväksi järjestelmään.

Cryptsetup on työkalu levysalauksien hallintaan DMCryptillä. Useita standardeja siitä, missä muodossa salatut osiot tulee olla, on useita ja Cryptsetup tukee näistä LUKS:ia, loop-AES:ia, TrueCryptiä ja Microsoftin BitLockeria. Salatun osion muoto määrittelee esimerkiksi sen, millainen osion alun salaamaton ylätunniste on. Ylätunniste kertoo lyhyesti ohjeet salauksen purkamiseen, esimerkiksi sen, mitä salausalgoritmia salaukseen on käytetty. Cryptsetup tukee myös ylätunnisteettomia niin kutsuttuja paljaita DMCrypt-osioita. LUKS:ia tyypillisesti suositellaan Linux-järjestelmän osioita salatessa.

Ohjelmalistauksessa 1 esimerkki levyosion salaamisesta Cryptsetupilla käyttäen salausavaimena salasanaa. Salattava osio on lohkolaitetiedosto */dev/sda2* ja se on tarkoitus liittää hakemistopuun sijaintiin */home*. [22]

Ohjelmalistaus 1 Levyosion salaus Cryptsetupilla.

```
# Ensin alustetaan osio LUKS-muotoon (syötä haluttu salasana)
cryptsetup luksFormat /dev/sda2
# Puretaan salatun osion salaus ja annetaan sille virtuaalinen nimi
cryptsetup open /dev/sda esim
# Nyt virtuaalinen laitetiedosto on saatavilla polussa /dev/mapper/esim
# Alustetaan tiedostojärjestelmä virtuaaliselle laitetiedostolle
mkfs.ext4 /dev/mapper/esim
# Nyt virtuaalisella laitetiedostolla on tiedostojärjestelmä
# ja sen voi liittää normaalisti hakemistopuuhun
mount /dev/mapper/esim /home
```

Tietoa voi salata myös tiedostotasolla. Tiedostotason salaukseen on useita työkaluja kuten eCryptFS ja EncFS. eCryptFS on toteutettu Linuxin kerneliin kuten DMCrypt. EncFS on käyttäjätilassa toimiva erillinen sovellus ja huomattavasti helppokäyttöisempi. EncFS:n käyttö on varsin suoraviivaista, ohjelmalistauksessa 2 salataan hakemisto */home/user/salattava* ja säilötään salattu data hakemistoon */home/user/.salattu*. [23]

Ohjelmalistaus 2 Levyosion salaus EncFS:llä.

```
# Salatun hakemiston luonti ja olemassa olevan salatun
# hakemiston salauksen purku tapahtuu samalla komennolla
encfs /home/user/.salattu /home/user/salattava
```

4.1.2 Tietoliikenteen salaaminen

Tietoliikenteen salaamisella pyritään ensisijaisesti suojautumaan hyökkäykseltä H3. Useimmiten tietoliikenteen salaaminen on jonkin tiedonvälitykseen käytettävän protokollan tehtävä (esim. HTTP vs. HTTPS) ja merkityksellisintä turvallisen tiedonvälityksen kannalta on tehdä turvallisia sovellusvalintoja. Suositeltavaa on esimerkiksi käyttää ennemmin salattua SFTP:tä kuin salaamatonta FTP:tä siirrettäessä tiedostoja verkon ylitse.

On myös mahdollista toteuttaa kokonaisvaltaisempaa tietoliikenteen salaamista tunneloimalla tietoliikenne jonkin salatun teknologian lävitse. Tällöin voidaan varmistua, että liikenne on salattua, vaikka sovellustasolla käytettäisiinkin protokollaa, joka ei tue salausta. Yleisin tähän tarkoitukseen käytetty teknologia on salattu VPN. Implementaatioita VPN:stä on lukuisia ja toiminta niiden välillä eroaa paljonkin. Käytännöllisimpiä VPN:n implementaatioita Linux-palvelinten näkökulmasta lienevät WireGuard ja OpenVPN. WireGuardin ja OpenVPN:n avulla voidaan TCP/UDP-tasolla tunneloida kaikki tietoliikenne salatun IP-tunnelin ylitse. [18]

4.2 Palomuurit

Palomuuri (eng. *Firewall*) on järjestelmä, jonka tarkoitus on estää asiaton pääsy verkkojen välillä. Useimmiten tämä toteutetaan kokoelmalla erilaisia sääntöjä. Palomuureilla voidaan ennaltaehkäistä useita uhkia vastaan pienentämällä hyökkäyspintaalaa esimerkiksi sulkemalla liikenne kokonaan tiettyihin portteihin tai tietyistä lähteistä. Palomuuria voi myös konfiguroida dynaamisesti, jolloin esimerkiksi kun hyökkäys H1, H6 tai H5 havaitaan, liikenne voidaan sulkea näistä lähteistä. [18]

Netfilter on ohjelmistokehys Linux-kernelissä, joka on tarkoitettu moniin verkkoyhteyksiin liittyvien asioiden hallintaan ja tällä kernelin palomuuri on implementoitu. IPTables on käyttäjätason sovellus hallinnoimaan kernelin palomuurin sääntöjä

ja tulee useimpien Linux-jakeluiden mukana. IPTablesin seuraaja on NFTables, mutta siihen, että NFTables otettaisiin laajempaan käyttöön kuin IPTables voi mennä vielä vuosia. IPTablesin syntaksi on vasta-alkajille usein sekava, joten palomuurin hallintaan on myös käyttäjäystävällisempiä ratkaisuja kuten UFW (**U**ncomplicated **F**ire**w**all). UFW:n käyttö on varsin suoraviivaista ja esittelen ohjelmalistauksessa 3 muutaman hyödyllisen komennon palomuurin hallintaan UFW:llä. [24] [25]

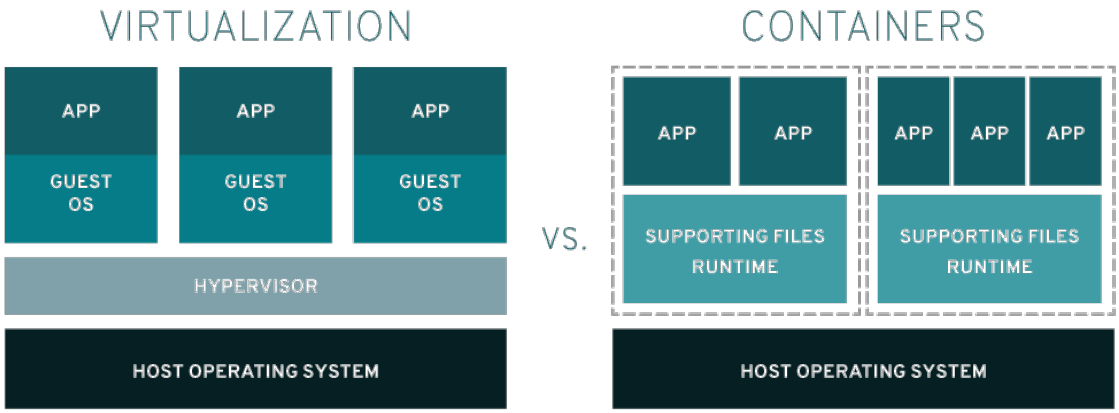
Ohjelmalistaus 3 UFW:n käyttö.

```
# Estetään liikenne tietystä IP-osoitteesta
ufw deny from "80.186.199.108"
# Estetään kaikki liikenne SSH:n porttiin 22
ufw deny ssh
# Sallitaan kuitenkin SSH-yhteydet tietystä lähteestä
ufw allow from "192.168.1.12" to any port 22
```

4.3 Eristys ja virtualisointi

Vaikka Linux-kerneli tarjoaa keinoja eristää sovelluksia, on usein tarpeen toteuttaa eristys virtualisoinnin avulla. Palvelimen jokaista sovellusta voidaan ajaa suljetussa virtuaalikoneessa, josta ei ole pääsyä isäntäpalvelimelle tai muille virtuaalipalvelimille. Mikäli jostakin sovelluksesta löytyy tietoturva-aukko ja aukkoa hyödynnetään hyökkäyksessä niin hyökkääjä saa hallinnan vain virtualisoidusta järjestelmästä. Tämäkään ei ole hyvä asia, mutta vahingot jäävät huomattavasti pienemmiksi kuin tilanteessa, jossa hyökkääjä saisi hallinnan koko järjestelmästä ja kaikista sovelluksista. Tämä rajoittaa huomattavasti hyökkäysten H2, H5, H6 sekä H4 aiheuttamaa vahinkoa. [11]

Virtualisointitekniikat jaetaan useimmiten kahteen pääkategoriaan; kokonaiseen virtualisointiin, jossa koko tietokoneen laitteisto simuloidaan sekä käyttöjärjestel-



Kuva 4.1: Täysi virtualisointi vs. käyttöjärjestelmätason virtualisointi [26]

mätason virtualisointiin, jossa ohjelmistot ajetaan omassa eristetyssä ympäristössä, niin kutsutuissa *konteissa* simuloimatta kuitenkaan tietokoneen komponentteja. Alustoja laitteiston virtualisointiin Linuxilla ovat mm. KVM (**K**ernel-based **V**irtual **M**achine), VMWare, VirtualBox, XEN, QEMU sekä helpottamaan virtualisoinnin ylläpitoa libvirtd. Käyttöjärjestelmätason virtualisoinnin alustoja ovat mm. Docker, Vagrant ja Linuxin chroot-ympäristö. [11]

4.4 Järjestelmän ja sovellusten konfiguraatio

4.4.1 Autentikaatio

Autentikaatioprosessia vahventamalla pyritään suojautumaan ensisijaisesti hyökkäykseltä H6. Perinteiset keinot suojautua, kuten tarpeeksi vahvan salasanan valitseminen ovat toki tärkeitä, mutta autentikaatioprosessia on myös mahdollista vahventaa järjestelmän konfiguraatioilla ja työkaluilla.

Käyttäjien salasanoille voidaan asettaa rajoitteita. Tällaiset rajoitteet voivat olla liian lyhyiden, liian yksinkertaisten tai luonnollisten kielten sanoista muodostuvien salasanojen estäminen. Kirjautumisyrityskerroille voidaan asettaa yläraja tai salasanoille voidaan asettaa vanhenemisaikoja. Nämä tekniikat heikentävät huomat-

tavasti hyökkäyksen H6 onnistumismahdollisuuksia. Linuxilla näitä tekniikoita voi ottaa käyttöön autentikaatiojärjestelmää konfiguroimalla tiedostosta */etc/login.defs* tai PAM:n avulla (**P**luggable **A**uthentication **M**odules). [27]

SSH on yleisin protokolla hallinnoida Linux-palvelimia etänä ja estämällä *root*-käyttäjän kirjautumisen SSH:n ylitse pienentää huomattavasti hyökkäyspinta-alaa. Pääkäyttäjän *root*-käyttäjätunnus on aina sama useimmissa UNIX-tyyppisissä järjestelmissä, joten hyökkääjän tarvitsee enää arvata vain salasana hyökkäyksen onnistumiseksi. Ennen *root*-käyttäjän kirjautumisen estoa jokin toinen käyttäjä tulisi lisätä */etc/sudoers*-tiedostoon, jotta operointi pääkäyttäjäoikeuksilla onnistuisi silti SSH:n välityksellä. [28] [29]

Ohjelmalistaus 4 Root-käyttäjän kirjautumisen estäminen

```
# Ensin lisätään "esimerkki"-käyttäjä /etc/sudoers:n
# ja annetaan tälle täydet oikeudet
echo 'esimerkki ALL=(ALL) ALL' >> /etc/sudoers
# Tämän jälkeen avataan SSH-palvelimen konfiguraatiotiedosto
vi /etc/ssh/sshd_config
# Muutetaan rivi:
PermitRootLogin Yes
# Riviksi:
PermitRootLogin No
# Tämän jälkeen SSH-palvelin tulisi käynnistää uudelleen
# esimerkiksi komennolla (riippuu jakelusta)
systemctl restart sshd
```

4.4.2 Käyttöoikeudet

Käyttöoikeuksien hallinnalla on tarkoitus hallita sitä kuka saa käyttää tai nähdä resursseja. Linux-palvelinten näkökulmasta on tietoturvan kannalta järkevää antaa esimerkiksi HTTP-palvelimelle oikeudet päästä käsiksi vain HTTP-palvelimelle relevantteihin tiedostoihin eikä koko järjestelmään. Tällöin esimerkiksi onnistuneen HTTP-palvelimeen kohdistuneen hyökkäyksen H5 seuraukset ovat pienemmät.

Perinteisesti UNIX-tyyppisissä käyttöjärjestelmissä resurssien käyttöoikeuksia hallinnoidaan tiedostojärjestelmätasolla niin, että jokaisella tiedostolla on omistajakäyttäjä sekä omistajaryhmä. Tämän lisäksi tiedostolle voidaan asettaa kirjoitus-, luku-, ja suoritusoikeudet kolmelle eri käyttäjäryhmälle erikseen: omistajakäyttäjälle, omistajaryhmälle sekä muille. Tiedoston omistajuus ilmoitetaan yleensä notaatiolla *käyttäjä:ryhmä* ja käyttäjäryhmäkohtaiset oikeudet joko numeerisella kolmen numeron sarjana (esim. *755*) josta jokainen numero on väliltä 0-7 tai symbolisella 10 merkin merkkijonolla, joka ensimmäistä merkkiä lukuun ottamatta koostuu merkeistä **r** (**r**ead, lukuoikeus), **w** (**w**rite, kirjoitusoikeus, **x** (e**x**ecution, suoritusoikeus) tai **-** (ei oikeutta). Esimerkin numeerinen oikeuksien esitys *755* kääntyisi symboliseksi esitykseksi *-rwxr-xr-x*. Symbolisen esityksen ensimmäinen merkki kertoo tiedoston tyypin eikä suoranaisesti liity käyttöoikeuksiin. Tiedoston omistajuuksia voi hallinta komennolla *chown* ja käyttöoikeuksia komennolla *chmod*.

Numeerisen esityksen ensimmäinen numero ja symbolisen esityksen kolme ensimmäistä merkkiä (tiedoston tyyppimerkin jälkeen) kertovat mitä oikeuksia tiedoston omistajakäyttäjällä on. Edellä mainitun esimerkin tapauksessa omistajalla on täydet oikeudet eli kirjoitus-, luku-, sekä suoritusoikeus. Numeerisen esityksen keskimmäinen numero ja symbolisen esityksen kolme seuraavaa merkkiä kertovat mitä oikeuksia omistajaryhmällä on. Esimerkin tapauksessa omistajaryhmällä on oikeudet lukea ja suorittaa tiedostoa. Viimeinen numero sekä viimeiset kolme merkkiä edustavat sitä millaisia oikeuksia kaikilla muilla käyttäjillä on. Esimerkin tapauksessa heilläkin

on oikeudet lukea sekä suorittaa tiedostoa.

Useimmat tiedostojärjestelmät tukevat tämän lisäksi tiedostojen ominaisuuksia (eng. *File Attributes*), joilla voidaan määritellä lisäasetuksia tiedostojen oikeuksille. Esimerkiksi tiedostoille voidaan määrittää, että niitä ei voi poistaa tai muunnella. Komennolla *chattr* voidaan hallita tiedostojen ominaisuuksia.

Tiedostojen käyttöoikeuksista päättää tiedoston omistajakäyttäjä tai *root*-käyttäjä. Perinteinen UNIX-tyyppinen käyttöoikeuksien hallinta on vuosikymmeniä sitten kehitetty järjestelmä ja se saattaa olla moderniin palvelinylläpitoon liian vanhanaikainen. Tämän vuoksi on kehitetty esimerkiksi laajennetut tiedostojen ominaisuudet (eng. *Extended File Attributes*) tai kokonaan erilaisia lähestymistapoja käyttöoikeuksien hallintaan kuten SELinux. [30]

SELinux

SELinux (Security Enhanced Linux) on alun perin NSA:n kehittämä laajennus Linux-kerneliin, joka on ollut osa Linux-kerneliä vuodesta 2003. SELinux antaa työkalut hallita käyttöoikeuksia tarkemmin. Tärkein konseptillinen ero perinteiseen UNIX-tyyppiseen käyttöoikeuksien hallintaan on se, että SELinux arkkitehtuurillisesti pakollista käyttöoikeuksien hallintaa (eng. *Mandatory Access Control, MAC*) kun taas perinteinen UNIX-tyyppinen käyttöoikeuksien hallinta valinnaista käyttöoikeuksien hallintaa (eng. *Discretionary Access Control, DAC*). SELinux pyrkii erottamaan tietoturvan toteutuksen tietoturvaan liittyvistä päätöksistä.

Tuki SELinuxille on Linux-kernelissä ja useimpiin Linux-jakeluihin SELinuxin hallinnointityökalut ovat saatavilla. Useimmissa RPM-pohjaisissa jakeluissa kuten Red Hat Enterprice Linuxissa hallinnointityökalut tulevat vakiona. [31]

4.5 Monitorointi

Monitoroinnin tarkoitus ei ole niinkään suojautua kyberhyökkäykseltä vaan ennemminkin keino havaita se. Kun hyökkäys havaitaan, se voidaan keskeyttää. Linux-jakelut ja sovellukset säilövät lokinsa yleensä hakemistoon */var/log/*. Epäilyttäviä HTTP-pyyntöjä kuten hyökkäyksessä H5 usein muodostuu voi yrittää etsiä HTTP-palvelimen lokista. Käynnissä olevan hyökkäyksen H6 voi tunnistaa etsimällä epäilyttäviä kirjautumisyrityksiä Systemd:n lokista komennolla *journalctl* tai tiedostoista */var/log/secure* tai */var/log/auth* riippuen jakelusta.

Linux Audit Framework on yksi tärkeimmistä työkaluista tietoturvan monitorointiin Linuxilla. Audit Frameworkin käyttäjätilan työkalulla *aureport* voidaan laatia raportti tietoturvapoikkeamista. Mikäli SELinux on käytössä se tallentaa lokia tietoturvapoikkeamista omiin lokitiedostoihinsa. [32]

Koodilistauksessa 5 tunnistan epäilyttävän kirjautumisyrityksen parsimalla Systemd:n lokia, jonka jälkeen käytän palomuuria sulkeakseni liikenteen kyseisestä lähteestä keskeyttääkseni hyökkäysyrityksen.

Ohjelmalistaus 5 Epäilyttävän kirjautumisyrityksen tunnistus.

```
# Parsitaan Systemd:n lokia

journalctl -xe | grep -i "failed password"

# Komento tulostaa seuraavat rivit

marras 19 04:56:40 r5 sshd[2256116]: Failed password for invalid user

epailyttava from 192.168.8.124 port 19204 ssh2

marras 19 04:56:44 r5 sshd[2256116]: Failed password for invalid user

epailyttava from 192.168.8.124 port 19204 ssh2

marras 19 04:56:47 r5 sshd[2256116]: Failed password for invalid user

epailyttava from 192.168.8.124 port 19204 ssh2

# Estetään liikenne kyseisestä lähteestä palomuurilla

ufw deny from "192.168.8.124"
```

5 Yhteenveto

Tarkastelun kohteena oli muutamia yleisimpiä kyberhyökkäyksiä ja keinoja vahventaa Linuxin tietoturvaa. Esitellyt keinot vahventaa Linux-palvelinten tietoturvaa tarjoavat vähintään vähäistä suojaa esitellyiltä kyberhyökkäyksiltä.

Taulukkoon 5.1 on koottu käsitellyt hyökkäykset suhteessa suojautumismenetelmiin. Iso **X** tarkoittaa sitä, että suojautumismenetelmä auttaa merkittävästi suojautumaan tietoturvahyökkäykseltä. Pieni **x** tarkoittaa puolestaan sitä, että menetelmä auttaa suojautumaan hieman tietoturvahyökkäykseltä. Pienen **x**:n merkintää käytetään myös silloin, jos suojautumismenetelmä auttaa merkittävästi vähentämään tietoturvahyökkäyksen aiheuttamaa vahinkoa.

Kuten taulukosta 5.1 huomaa, esitellyt suojautumismenetelmät tuntuvat jakautuvan kahden pääryhmän välillä. Ensimmäisen ryhmän suojautumismenetelmät auttavat tehokkaasti, mutta vain yhteen tiettyyn kyberhyökkäykseen. Kun taas toisen ryhmän menetelmät ovat kokonaisvaltaisempia ratkaisuja, jotka auttavat monilla osa-alueilla hieman.

Taulukosta 5.1 huomataan myös, että joltakin hyökkäyksiltä on hankala suojautua. Tällainen hyökkäys on esimerkiksi palvelunestohyökkäys (H1). Hyökkäykseltä H1 pystytään palomuurien (luku 4.2) avulla suojautumaan vain mikäli hyökkäys tulee yhdestä lähteestä, ennalta arvattavista lähteistä tai muutoin ennalta-arvattavalla tavalla.

Taulukko 5.1: Hyökkäykset vs. suojautumismenetelmät

	H1	H2	H3	H4	H5	H6	H7
Tallennustilan salaaminen (4.1.1)							**X**
Tietoliikenteen salaaminen (4.1.2)			**X**				
Palomuurit (4.2)	**x**				**x**	**x**	
Eristys ja virtualisointi (4.3)		**x**		**x**	**x**	**x**	
Autentikaatioprosessin vahvennus (4.4.1)						**X**	
Käyttöoikeuksien vahvennus (4.4.2)		**x**		**x**	**x**		

Lähdeluettelo

[1] W3Cook. (n.d.). ”Os Market Share and Usage Trends”. [06.08.2015 arkistoitu verkkolähde; Viitattu 04.11.2020], url: https://web.archive.org/web/20150806093859/http://www.w3cook.com/os/summary/.

[2] Open Book Project Contributors. (n.d.). ”Operating Systems”. [Verkkolähde; Viitattu 04.11.2020], url: http://openbookproject.net/courses/intro2ict/system/os_intro.html.

[3] StatCounter. (n.d.). ”Operating System Market Share Worldwide”. [Verkkolähde; Viitattu 04.11.2020], url: https://gs.statcounter.com/os-market-share/.

[4] Google. (n.d.). ”Platform Architecture (Android Developer Documentation)”. [Verkkolähde; Viitattu 04.11.2020], url: https://developer.android.com/guide/platform.

[5] A. Prakash. (2019). ”Linux Runs on All of the Top 500 Supercomputers, Again!” [Verkkolähde; Viitattu 04.11.2020], url: https://itsfoss.com/linux-runs-top-supercomputers/.

[6] W3Techs. (n.d.). ”Usage statistics of operating systems for websites”. [Verkkolähde; Viitattu 04.11.2020], url: https://w3techs.com/technologies/overview/operating_system.

[7] DistroWatch. (n.d.). ”DistroWatch Page Hit Ranking”. [Verkkolähde; Viitattu 21.10.2020], url: `https://distrowatch.com/dwres.php?resource=popularity`.

[8] Linux.com. (2019). ”Top 5 Linux Server Distributions”. [Verkkolähde; Viitattu 04.11.2020], url: `https://www.linux.com/topic/desktop/top-5-linux-server-distributions/`.

[9] P. Koutoupis. (2017). ”Why the Largest Companies in the World Count on Linux Servers”. [Verkkolähde; Viitattu 04.11.2020], url: `https://www.linuxjournal.com/content/why-largest-companies-world-count-linux-servers`.

[10] Paessler. (n.d.). ”IT Explained: Server”. [Verkkolähde; Viitattu 04.11.2020], url: `https://www.paessler.com/it-explained/server`.

[11] M. Portnoy, *Virtualization Essentials*. John Wiley & Sons, 2016.

[12] J. Andress, *The basics of information security: understanding the fundamentals of InfoSec in theory and practice*. Syngress, 2014.

[13] ”Information technology — Security techniques — Information security management systems — Overview and vocabulary”, International Organization for Standardization, Standard ISO 27000:2018(E), 2018.

[14] M. Stamp, *Information security: principles and practice*. John Wiley & Sons, 2011.

[15] NIST CSRC. (2009). ”Understanding Denial-of-Service Attacks”. [Verkkolähde; Viitattu 04.11.2020], url: `https://us-cert.cisa.gov/ncas/tips/ST04-015`.

[16] S. Kottler. (2018). ”February 28th DDoS Incident Report”. [Verkkolähde; Viitattu 04.11.2020], url: `https://github.blog/2018-03-01-ddos-incident-report/`.

[17] H. F. Tipton ja M. Krause, *Information Security Management Handbook, Sixth Edition*. CRC press, 2007.

[18] M. Ciampa, *Security+ guide to network security fundamentals*. Cengage Learning, 2012.

[19] Wikipedia, the free encyclopedia, *Privilege Escalation Diagram*, [Verkkolähde; Viitattu 04.11.2020], n.d. url: `https://en.wikipedia.org/wiki/Privilege_escalation#/media/File:Privilege_Escalation_Diagram.svg`.

[20] M. McDonald, *Web Security for Developers: Real Threats, Practical Defense*. No Starch Press, 2020.

[21] K. Beaver ja R. Stiennon, *Hacking for Dummies*. John Wiley & Sons, 2015.

[22] J. Saout, C. Fruhwirth, A. Wagner, M. Broz ja Red Hat, Inc., *Cryptsetup*, versio 2.3.4. url: `https://gitlab.com/cryptsetup/cryptsetup/`.

[23] V. Gough, *EncFS*, versio 1.9.5. url: `https://vgough.github.io/encfs/`.

[24] C. Binnie, *Linux Server security: hack and defend*. Wiley Online Library, 2016.

[25] Canonical Ltd., *UFW*, versio 0.36. url: `https://launchpad.net/ufw`.

[26] Red Hat, Inc., *Virtualization vs. Containers*, [Verkkolähde; Viitattu 27.11.2020], n.d. url: `https://www.redhat.com/cms/managed-files/styles/wysiwyg_full_width/s3/virtualization-vs-containers_transparent.png`.

[27] J. Kemp, *Linux System Administration Recipes: A Problem-solution Approach*. Apress, 2009.

[28] T. Ylönen, A. Campbell, B. Beck, M. Friedl, N. Provos, T. de Raadt ja D. Song, *OpenSSH*, versio 8.4p1. url: `https://www.openssh.com/`.

[29] T. C. Miller ja sudo contributors, *Sudo*, versio 1.8.3p1. url: `https://www.sudo.ws/`.

[30] T. Kalsi, *Practical Linux Security Cookbook: Secure your Linux environment from modern-day attacks with practical recipes*. Packt Publishing Ltd, 2018.

[31] Red Hat, Inc. (n.d.). "What is SELinux?" [Verkkolähde; Viitattu 19.11.2020], url: `https://www.redhat.com/en/topics/linux/what-is-selinux`.

[32] S. Hussain. (n.d.). "Linux Security Must-read Guide 2020: How to Investigate Suspected Break-in Attempts in Linux". [Verkkolähde; Viitattu 19.11.2020], url: `https://www.xplg.com/linux-security-investigate-suspected-break-in/`.