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, 20 s., 1 liites.
Tietojenkäsittelytiede
Kesäkuu 2020

Linux-käyttöjärjestelmä on käytetyin alusta palvelinylläpidossa.
Tutkielmassa käyn lähdekirjallisuuden avulla läpi miten tyypillisimmiltä tietoturvauhilta voidaan suojautua Linux-palvelimilla.

Asiasanat: Tietoturva, Linux, Palvelin

UNIVERSITY OF TURKU
Department of Future Technologies

Maks Turtiainen: Linux-palvelinten tietoturva

Bachelor's Thesis, 20 p., 1 app. p.
Computer Science
June 2020

Second abstract in english (in case the document main language is not english)

Keywords: here, a, list, of, keywords

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 96,3 % 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. Toisessa luvussa johdattelen Linuxin ja palvelinten perusteilla jonka jälkeen luvussa 3 käyn läpi tärkeimpiä konspeteja tietoturvasta puhuttaessa ja tyypillisimpä 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 kokoilman ohjelmistoja sekä paketinhallinnan. Linux-kerneli ja useimmat Linux-jakelut ovat vapaata lähdekoodia. [2] [3]

Työpöytäkäytössä Microsoft Windows on suosituin käyttöjärjestelmä, Linuxia käyttävät vuonna 2020 vain 1,74% työpöytäkäyttäjistä. [4] Mobiililaitteissa suosituin käyttöjärjestelmä on Googlen Android, jonka voi katsoa olevan Linux-jakelu sen käyttäessä muokattua versiota Linux-kernelistä. [5] Mobiililaitteista Androidia käytti 85 % vuonna 2018 [6]. Vuosina 2017–2019 maailman 500 tehokkaimmasta supertietokoneesta kaikki käyttävät Linuxia. [7]. Internetin julkisista web-palvelimista vuonna 2015 Linuxia käyttää 69,7%-96,4% riippuen lähteestä. [8] [1]

Linux-jakeluiden kotisivujen käyttäjämäärien perusteella suosituimmat 3 Linux-

jakelua ovat MX Linux, Manjaro ja Mint (21.10.2020). [9] 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. [10]

Perusteita suurelle Linuxin käytölle palvelimissa on arvioitu olevan vakaus ja luotettavuus, turvallisuus, muokattavuus, lähdekoodin avoimuus sekä kustannukset. [11]

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 sääs-

tämiseksi näitä harvemmin on palvelimissa. Palvelinta kontrolloidaan etänä esimerkiksi SSH:n välityksellä, web-pohjaisesta käyttöliittymästä tai jollakin kaupallisella ratkaisulla kuten Microsoft Managment Consolella. [12]

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.

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. [13]

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 palvelinsalista 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. Yriyksen web-sovellus on Python–ohjelmointikielellä kirjoitettu. Web-applikaatiokirjastona 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 3000 portin ulkoverkkoon porttiin 80 ja 443.

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

Asiakasyrityksen web-sovellus on nyt saatavilla HTTP (S) -protokollan ylitse

esimerkki.fi osoitteesta 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 tahallen aiheutetuilta häiriöiltä palvelujen toimintakyvyssä. Suojautumaan pyritään myös palvelujen väärinkäytöksiltä. Tietoturvan merkitys on kasvanut nopeasti digitalisaation myötä. [14]

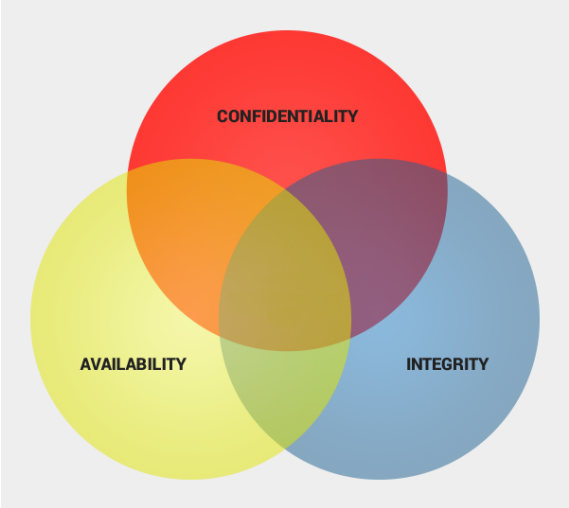
3.1 CIA-malli

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. [15]

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. [16]

Esittelen seuraavaksi yleisimpiä kyberhyökkäyksiä, mitä haavoittuvuutta ne hyö-



Kuva 3.1: CIA-malli

dyntävät 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. [17]

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 suorittemelle ja näin ollen ylikuormittaa suorittimen.

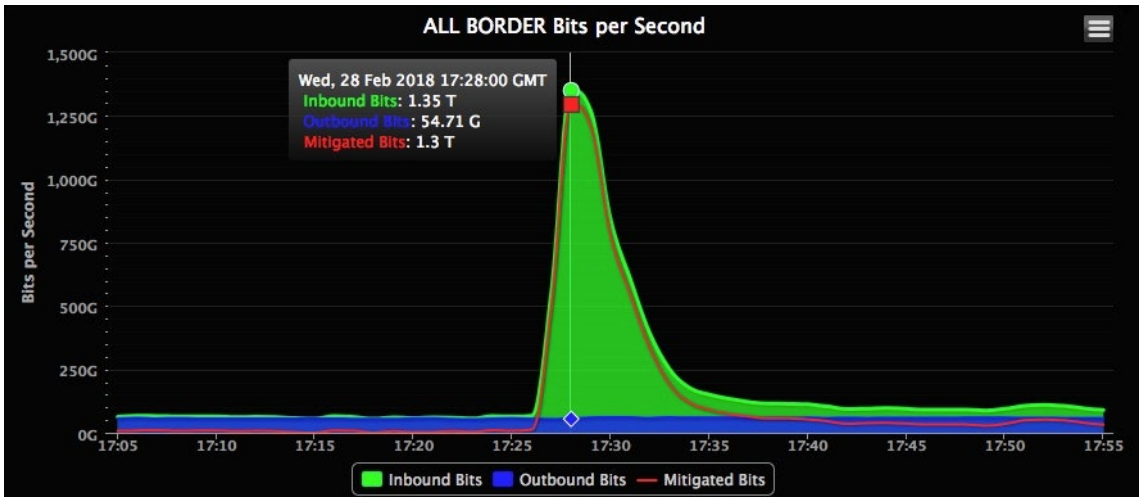
Todellisuudessa harvalla hyökkääjällä on käytössään sellaisia resursseja, joilla

olisi mahdollista tukkia jonkin kaupallisen toimijan palvelinresurssit. Tämän vuoksi nykyään yleisempi tapa on toteuttaa palvelunestohyökkäys on tehdä se hajautetusti (DDoS, Distibuted Denial of Service). Hajautetussa palvelunestohyökkäyksessä hyökkääjällä on käytössään useista internetiin yhdistetyistä tietokoneista muodostuva bottiverkko. Näiden tietokoneiden hallinnan hyökkääjä on saanut aikaisemmin jollakin muulla kyberhyökkäyksellä. Bottiverkossa voi olla mukana jopa satoja tuhansia tietokoneita ja tällaisen verkon haltija voi lähettää toistuvia pyyntöjä kaikilta verkon tietokoneilta samaan aikaan. Näin massiivisen verkon haltija voisi ylikuormittittaa esimerkiksi tyypillisen verkkosivuston vain yksinkertaisesti lataamalla jokaisella verkon koneella toistuvasti verkkosivustoa. [18]

Tällaisiä "raakaan voimaan" perustuvia palvelunestohyökkäyksiä vastaan voi olla vaikea suojautua. Useimmat nykyaikaiset reitittimet osaavat jo jättää hyökkääjän liikenteen huomiotta, mikäli toistuvat palvelupyynnöt tulevat samasta lähteestä. Tilanne onkin toinen kun pyynnöt tulevat useista tuhansista lähteistä samaan aikaan. Tyypillisesti isoilla palveluntuottajilla on vain yksinkertaisesti niin paljon resursseja, että näitä on hyvin vaikea kenenkään tukkia. Toinen vaihtoehto on tehdä suunnitelma sen varalta, jos palvelunestohyökkäys tapahtuu. Tällaiseen suunnitelmaan voi kuulua replikaa palvelusta joka otetaan käyttöön hyökkäyksen sattuessa.

"Raa'an voiman" käyttö palvelunestohyökkäyksissä on yksinkertaisin ja yleisin tapa, mutta palvelunestohyökkäys voidaan toteuttaa myös muilla keinoilla. Muita keinoja ovat osoitintietojen muuttaminen ja virheellisten palvelupyyntöjen lähetys sen toivossa, että palvelu kaatuu.

Kuvassa 3.2 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 tuhansilla väärinkonfiguroiduilla Memcached-palvelimilla. [19]



Kuva 3.2: Github DDoS 2018

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 normaalia autentikaatioprosessia läpi. Tässä tapauksessa hyökkääjällä on täysi kontrolli kohdejärjestelmästä, joten hyökkäys voi kohdistua kaikkiin CIA-mallin osa-alueisiin. [20]

Takaovi asennetaan kohdetietokoneelle esimerkiksi nk. 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.

3.2.3 H3: Verkon kuuntelu

Verkon kuuntelu 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 mistä vain. 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.

3.2.4 H4: Näppäilytallennin

Näppäilytallennin (eng. Keylogger) on laite tai tietokoneohjelma joka tallentaa kaikki kohteen näppäinpainallukset ja joko lähettää ne reaaliajassa hyökkääjälle tai hyökkääjä voi hakea esimerkiksi piilotetun laitteen kohteesta. Näppäinpainallusten tallennuksella tarkoituksena on useimmiten hankkia salaiseksi tarkoitettuja tietoja, kuten salasanoja. Näppälytallennin kohdistuu CIA-mallin C-kohtaan, sillä hyökkääjä yrittää saada tietoja, joita hänen nähtäväkseen ei oltu tarkoitettu.

3.2.5 H5: Koodin etäsuoritus

Koodin etäsuoritus (eng. Remote Code Execution)

3.2.6 H6: Eskalaatiohyökkäys

Eskalaatiohyökkäys (eng. Privilege escalation)

3.2.7 H7: Väsytyshyökkäys

Väsytyshyökkäys (eng. Brute-force)

3.2.8 H8: Laitteen varkaus

Laitteen varkaus

4 Kyberhyökkäyksiltä suojautuminen

Vaikka keskityn tutkielmassani ensisijaisesti Linux–palvelinten tietoturvan parantamiseen, monet yleiset keinot tietoturvan parantamiseen pätevät myös Linux–palvelinten kontekstissa joten käsittelen myös niitä lopuksi lyhyesti.

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.

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 hyökkäyk-

sessä H8. Tallennustilan tai tiedostojen salaamisella voidaan myös pyrkiä suojelemaan jotakin erittäin salaista osaa tiedoista jonka salausta pidetään purettuva vain silloin kun tietoja tarvitaan. Tälläisessä tapauksessa tietojen on mahdollista pysyä pois vääristä käsistä jopa silloin kun hyökkääjä on saanut muutoin täyden hallinnan tietoja säilyttävästä palvelimesta, kuten hyökkäyksissä H2, H5 ja H7.

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 nk. 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*.

Ohjelmalistaus 1 Levyosion salaus Cryptsetupilla.

```
# Ensin alustetaan osio LUKS-muotoon (syötä haluttu salasana)
cryptsetup luksFormat /dev/sda2
# Puretaan juuri salatun osion salaus ja anna 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*.

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. Esimerkiksi on suositeltavaa tehdä Linux-palvelimen ylläpitotoimia etäyhteydellä ennemmin salatun SSH:n kuin salaamattoman Telnetin välityksellä.

On myös mahdollista toteuttaa kokonaisvaltaisempaa tietoliikenteen salaamista tunneloimalla tietoliikenne jonkin salatun teknologian lävitse. Tällöin voidaan varmistua, että liikenne on salattua, vaikka käyttäjätasolla 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. Linux-palvelinylläpidon näkökulmasta käytännöllisin implementaatio lienee OpenVPN.

OpenVPN:n avulla voidaan TCP/UDP tasolla tunneloida kaikki tietoliikenne salatun IP-tunnelin ylitse.

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 tai H7 havaitaan, liikenne voidaan sulkea näistä lähteistä.

Netfilter on ohjelmistokehys Linux-kernelissä joka on tarkoitettu moniin verkkoyhteyksiin liityvien 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 kute UFW.

4.3 SELinux

SELinux (Security Enhanced Linux) on Red Hat Enterprise Linuxin...

4.4 Kernelitason suojaus

Linuxin kerneli tarjoaa monia työkaluja suojella...

4.5 Eristys ja virtualisointi

Vaikka Linux-kerneli tarjoaa keinoja eristää sovelluksia, on usein tarpeen toteuttaa eristys virtualisoinnin avulla.

4.5.1 Virtualisoinnin työkaluja

Virtualisointi jaetaan useimmiten kahteen pääkategoriaan; kokonaiseen virtualisointiin jossa koko tietokoneen laitteisto simuloidaan sekä ns. paravirtualisointiin jossa ohjelmistot ajetaan omassa eristetyssä ympäristössä simuloimatta kuitenkaan tietokoneen komponentteja. Alustoja laitteiston virtualisointiin Linuxilla ovat mm. KVM (Kernel-based Virtual Machine), VMWare, VirtualBox, XEN, QEMU sekä helpottamaan virtualisoinnin ylläpitoa libvirtd. Paravirtualisointiin alustaja ovat mm. Docker, Vagrant ja Linuxin chroot-ympäristö.

4.6 Järjestelmän ja sovellusten konfiguraatio

4.6.1 Autentikaatio

Autentikaatioprosessia vahventamalla voidaan suojautua esimerkiksi hyökkäykseltä H7. Perinteiset keinot suojautua, kuten tarpeeksi vahvan salasanan valitseminen ovat toki tärkeitä, mutta autentikaatioprosessia on myös mahdollista vahventaa järjestelmän konfiguraatioilla ja työkaluilla. Liian heikot salasanat voi olla myös mahdollisia purkaa mikäli hyökkääjä on saanut salasanojen tiivisteet käsiinsä */etc/shadow*-tiedostosta.

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.

Ohjelmalistaus 3 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
```

Järjestelmän käyttäjien salasanoille on myös mahdollista asettaa minimivaatimuksia. PAM.cracklib.

4.7 Monitorointi

Monitoroinnin tarkoitus ei ole niinkään suojautua kyberhyökkäykseltä vaan ennemminkin keino havaita se. Kun hyökkäys havaitaan, se voidaan estää.

5 Yhteenveto

wip

Lähdeluettelo

[1] W3Cook. (2015). ”Os Market Share and Usage Trends”, url: https://web.archive.org/web/20150806093859/http://www.w3cook.com/os/summary/.

[2] O. B. P. Contributors. (2017). ”Operating Systems”, url: http://openbookproject.net/courses/intro2ict/system/os_intro.html.

[3] T. I. Valley. (2019). ”What Is Linux: An Overview of the Linux Operating System”, url: https://medium.com/@theinfovalley097/what-is-linux-an-overview-of-the-linux-operating-system-77bc7421c7e5.

[4] StatCounter. (2020). ”Desktop Operating System Market Share Worldwide”, url: https://hostingtribunal.com/blog/operating-systems-market-share/.

[5] J. Lynch. (2013). ”Is Android really a Linux distribution?”, url: https://web.archive.org/web/20140205165359/http://www.itworld.com/open-source/369810/android-really-linux-distribution.

[6] G. Gottsegen. (2018). ”Apple is rethinking the hearing aid – and now Android is, too”, url: https://www.cnet.com/news/apple-is-rethinking-the-hearing-aid-and-now-android-is-too/.

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

[8] W3Techs. (2020). "Usage statistics of operating systems for websites", url: https://w3techs.com/technologies/overview/operating_system.

[9] DistroWatch. (2020). "DistroWatch Page Hit Ranking", url: https://distrowatch.com/dwres.php?resource=popularity.

[10] L. E. Staff. (2019). "Top 5 Linux Server Distributions", url: https://www.linux.com/topic/desktop/top-5-linux-server-distributions/.

[11] A. Kili. (2017). "6 Reasons Why Linux is Better than Windows For Servers", url: https://www.tecmint.com/why-linux-is-better-than-windows-for-servers/.

[12] Paessler. (2020). "IT Explained: Server", url: https://www.paessler.com/it-explained/server.

[13] B. Golden, *Virtualization for Dummies*. Wiley, 2011. url: https://books.google.fi/books?id=2ppZkdmpSlgC&lpg=PA54&hl=fi.

[14] J. W. Daniel Schatz Rabih Bashroush, "Towards a More Representative Definition of Cyber Security", *Journal of Digital Forensics, Security and Law*, vol. 12, nro 2, 2017. url: https://commons.erau.edu/jdfsl/vol12/iss2/8/.

[15] J. Andress, *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice*. Elsevier, 2014. url: https://books.google.fi/books?id=9NI0AwAAQBAJ&lpg=PP1&hl=fi.

[16] N. CSRC. (2020). "Definition of Vulnerability", url: https://csrc.nist.gov/glossary/term/vulnerability.

[17] ——, (2009). "Understanding Denial-of-Service Attacks", url: https://us-cert.cisa.gov/ncas/tips/ST04-015.

[18] M. Stamp, *Information Securiy: Principles and Practices*. Wiley, 2011. url: https://books.google.fi/books?id=UW3SS9P9hdEC&lpg=PA12&ots=0XK6AbvcRA&dq=information%20security&lr&hl=fi.

[19] L. H. Newman. (2018). ”GitHub Survived the Biggest DDoS Attack Ever Recorded”, url: https://www.wired.com/story/github-ddos-memcached/.

[20] K. Zetter. (2014). ”Hacker Lexicon: What Is a Backdoor?”, url: https://www.wired.com/2014/12/hacker-lexicon-backdoor/.

Liite A Liitedokumentti 1

Esimerkki