Touko Ala-Savikota

Sähköpostipalvelimen auditointityökalu

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinöörityö

10.4.2023

Tiivistelmä

Tekijä:	Touko Ala-Savikota
Otsikko:	Sähköpostipalvelimen auditointityökalu
Sivumäärä:	34 sivua + 2 liitettä
Aika:	10.4.2023
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Pelisovellukset
Ohjaaja:	Lehtori Miikka Mäki-Uuro

Insinöörityössä kehitettiin sähköpostipalvelimen auditointityökalu, jolla voidaan tarkastaa sähköpostipalvelimen toimivuus ja paljastaa tietoturvaongelmia. Tarkoituksena oli kehittää yleispätevä työkalu kaikkien olemassa olevien sähköpostipalvelimien automaattiseen auditointiin ja luoda kattavia raportteja tehtyjen auditointitestien perusteella.

Sähköpostipalvelimen auditointityökalun kehittämiseen käytettiin Python 3.6 -ohjelmointikieltä ja Linux-pohjaisen käyttöjärjestelmän alustaa. Kaikissa työkalun tavoitteissa onnistuttiin, ja työkalu toimii luotettavasti eri sähköpostipalvelimien auditoinnissa. Auditointityökalua testatessa sähköpostipalvelimista löydettiin ongelmia, joita ei välttämättä olisi ollut helppoa selvittää ilman työkalua. Auditointityökalun avulla luodut raportit antavat hyödyllistä tietoa löydettyjen ongelmien ratkaisuun ja auttavat parantamaan sähköpostin kuljetusketjun luotettavuutta ja turvallisuutta.

Avainsanat: sähköpostipalvelin, auditointi, smtp

Abstract

Author:	Touko Ala-Savikota
Title:	Mail server audit tool
Number of Pages:	34 pages + 2 appendices
Date:	10 April 2023
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communication Technology
Professional Major:	Game applications
Supervisor:	Miikka Mäki-Uuro, Senior Lecturer

The aim of the thesis was to create a mail server audit tool that can be used to verify the functionality of a mail server and unveil potential security problems. The primary goal was to create a tool that can automatically audit all kinds of mail server implementations and create reports based on tests done.

The mail server audit tool was created using the Python programming language version 3.6 and a Linux-based operating system. The mail server audit tool was a success, and it can work reliably with different kinds of mail servers. During practical tests the audit tool was able to reveal problems with the mail servers that wouldn't necessarily have been founded without using a specific audit tool. The reports created by the mail server audit tool give helpful insight to the problems that arise during testing, and they can help improve the reliability and security of mail delivery.

Keywords: mail server, audit, smtp

Sisällys

1 Johdanto

Sähköposti on viestintämenetelmä, joka poikkeaa perinteisestä kirjepostista, sillä se kulkee digitaalisesti tietoverkkoa pitkin jonkin ihmiselle silminnähtävän muodon sijaan. Sähköpostin avulla voi viestiä toiselle puolelle maailmaa, kunhan sähköpostilla on olemassa oleva vastaanottaja ja sähköpostin välityspalvelin, joka suostuu välittämään sähköpostin.

Tavallisen kirjepostin avulla on mahdollista viestiä myös arkaluontoisia asioita, joiden ei kuuluisi päätyä kenenkään muun kuin kirjeelle tarkoitettujen vastaanottajien tietoon. Tämä on hyvä peruste pitää huolta kirjepostin turvallisuudesta, jotta voidaan varmistaa kirjepostin sisällön luottamuksellisuus. Sähköpostin avulla on myös mahdollista lähettää sisältöä, joka voi olla luonteeltaan arkaluonteista. Tämä tekee sähköpostin luottamuksellisuuden huolenpidosta tietoturva-asian.

Insinöörityön tavoitteena oli luoda sähköpostipalvelimen auditointityökalu. Työkalun tarve perustui huoleen sähköpostin tietoturvasta ja tämän tietoturvan ylläpitämisestä. Sähköpostipalvelimen auditointityökalulla pitäisi pystyä tehokkaasti valvomaan niitä tahoja, jotka ovat vastuussa sähköpostin kuljetusketjuun osallistumisesta. Työkalun kuuluisi olla helppokäyttöinen tietokoneen suorittama ohjelma, ja sen kuuluisi soveltua yleiseen tapaukseen eli työkalulla kuuluisi pystyä valvomaan mahdollisimman laaja-alaisesti sähköpostin tietoturvaa. Työkalun pitäisi olla sekä manuaali- että automaattikäyttöön sopiva.

Vaikka sähköpostin turvallisuuteen liittyy sähköpostipalvelimen tietoturvan lisäksi kuluttajille suunnattujen sähköpostiohjelmien tietoturva, tämä insinöörityö kuitenkin käsittelee vain sähköpostipalvelimen auditointia. Sähköpostipalvelimen tietoturvan auditointi on sekin laaja aihe, ja insinöörityön tarkoituksena olikin löytää ne avainkohdat, joiden avulla luoda tehokas työkalu sähköpostipalvelimien auditointiin.

Insinöörityöraportin toisessa luvussa käydään läpi aiheeseen liittyviä teknisiä termejä, joita raportin lukijan kuuluisi ymmärtää ennen muun tekstin lukemista. Kolmannessa luvussa annetaan yleinen kuvaus siitä, miten työkalu toimii ja mitä siltä voidaan odottaa. Neljännessä luvussa käydään läpi auditointityökalun tekemiä testejä, ja lukija voi odottaa selvennystä, miksi testejä tehdään. Auditointityökalun teknistä toteutusta ja yksityiskohtia avataan enemmän viidennessä luvussa.

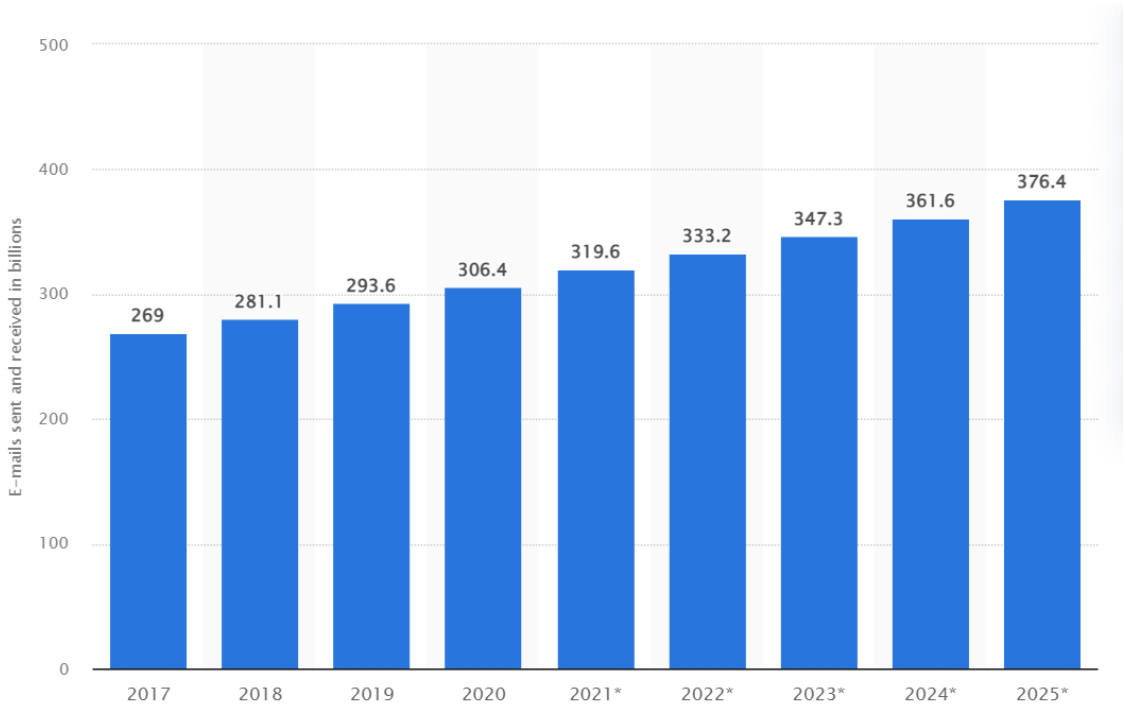
Insinöörityöraportissa käytetään runsaasti Internet Engineering Task Force -organisaation dokumentteja. Internet Engineering Task Force eli IETF vastaa useiden tietoliikenteessä käytettävien protokollien standardoinnista (1). Insinöörityöraportissa käytetään erityisesti IETF:n ehdotettuja standardeja tai standardiluonnoksiksi luokiteltuja dokumentteja. Tämä johtuu siitä, että vain harva IETF:n dokumentti on luokiteltu Internet-standardiksi ja nämä standardit voivat olla vanhentuneita. Esimerkiksi IETF RFC 821: Simple Mail Transfer Protocol on Internet-standardi, mutta se on ominaisuuksiltaan vanhentunut. Standardi ei esimerkiksi kerro myöhemmissä RFC-versioissa määritellyistä SMTP-protokollan laajennuksista, jotka ovat välttämättömiä muun muassa viestin enimmäiskoon ja sähköpostipalvelimen TLS-tuen (Transport Layer Security) selvittämiseksi (2, s. 9).

2 Tekninen pohjustus

Sähköpostin kuljetusketju on teknisesti monimutkainen kokonaisuus. Sen hallitseminen on kuitenkin välttämätöntä, jotta itse auditointiprosessi voidaan ymmärtää. Sähköpostipalvelimen auditointityökalun toimintojen ymmärtämiseksi kuuluisi vähintäänkin olla tietoinen sähköpostipalvelimen toiminnasta ja useiden palvelimien muodostamasta kuljetusketjusta. On myös välttämätöntä tietää auditointityökalun valvomista protokollista, joita sähköpostipalvelimet käyttävät Internetissä keskustelemiseen. Tämän luvun tarkoitus onkin antaa lukijalle pohjatietoa näihin asioihin, jotta insinöörityön kokonaisuuden ymmärtäminen on mahdollista.

2.1 Sähköpostipalvelimen toiminta ja kuljetusketju

Sähköpostipalvelimen tehtävä on välittää sähköpostia (2, s. 4). Jokainen lähetetty sähköposti kulkee yhden tai useamman sähköpostipalvelimen läpi, kunnes sähköposti saapuu päätepisteenä toimivalle sähköpostipalvelimelle, josta vastaanottaja voi hakea postinsa (2, s. 8). Vuonna 2021 tehdyn tutkimuksen mukaan vuonna 2020 lähetettiin ja vastaanotettiin maailmanlaajuisesti 306,4 miljardia sähköpostia päivässä (ks. kuva 1) ja tutkimus ennustaa määrän vain kasvavan (3). Pelkästään yhdessä päivässä kulkee siis hyvin merkittävä määrä sähköpostia, ja tämä luo vastuuta sähköpostipalvelimille, jos sähköpostien sisältö on arkaluonteista.



Kuva 1. Vuosittain lähetettyjen sähköpostien määrä vuosina 2017–2025. Diagrammin vaaka-akselilla on vuosiluku ja pystyakselilla sähköpostien lukumäärä miljardeina. Tähdellä merkityt arvot ovat ennustuksia, sillä tutkimus on toteutettu vuonna 2020. (3.)

Sähköpostipalvelin välittää sähköpostia sähköpostipalvelinohjelman avulla. Näitä ohjelmia on olemassa useille eri käyttöjärjestelmille, kuten Windowsille ja Linux- ja BSD-jakeluille. Sähköpostipalvelinohjelmia ovat esimerkiksi Microsoft

Exchange Server, Exim, OpenSMTPD, Sendmail ja Postfix. Nämä ohjelmat tyypillisesti toteuttavat IETF:n määrittelemän SMTP-protokollan, jossa määritellään, miten sähköpostin välittämisen kuuluisi käytännössä toimia. SMTP-protokollaa käydään luvussa 2.2 tarkemmin läpi. SMTP ei ole kuitenkaan ainoa tehtävä, jonka sähköpostipalvelinohjelma voi ottaa hoitaakseen. Sähköpostin jakeluverkostossa liikkuu paljon roskapostia, ja tämän takia on tullut aiheelliseksi suodattaa sähköpostia sähköpostipalvelimelta käsin. Sähköpostipalvelinohjelmat voivatkin tarjota ratkaisuja sähköpostin suodatukseen, jotta roskapostiksi epäiltyä sisältöä ei kuljeteta perille saakka.

Kuva 2 havainnollistaa tyypillistä reittiä, jonka sähköposti kulkee, kun sähköposti lähetetään asiakasohjelman kautta. Asiakasohjelma lähettää ensin sähköpostin lähettävälle sähköpostipalvelimelle ja tämä välittää sähköpostin vastaanottavalle sähköpostipalvelimelle. Kun vastaanottava palvelin saa sähköpostin haltuunsa, se välittää sähköpostin vastaanottajan postilaatikkoon, josta vastaanottajan asiakasohjelma voi noutaa sen. Sähköpostin kuljetusta mutkistaa se, että sähköposti ei kulje jokaisessa vaiheessa samalla tavalla. Lähettävä ja vastaanottava sähköpostipalvelin keskustelevat keskenään SMTP-protokollaa käyttäen, mutta asiakasohjelmat keskustelevat sähköpostipalvelimille erillisellä ”Message Submission” -protokollalla (2, s. 6). ”Message Submission” -protokolla mainitaan vain sen vuoksi, että sen olemassaolo tiedostetaan, mutta protokolla ei kuulu sähköpostipalvelimen auditointityökalun piiriin toisin kuin SMTP-protokolla.

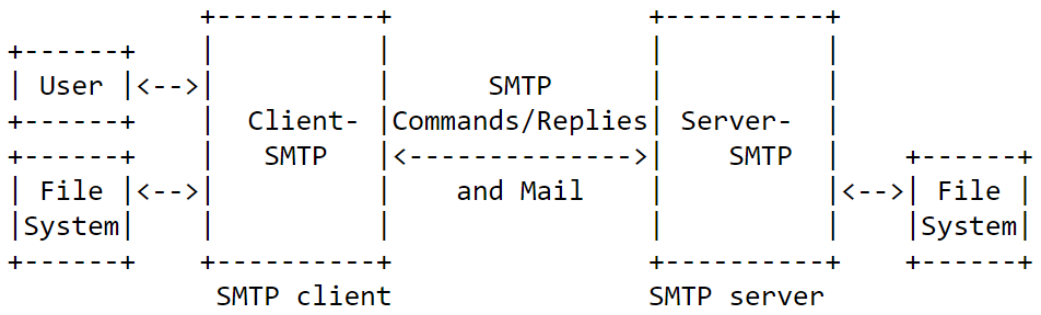
Kuva 2. Esimerkki sähköpostin kulkemasta reitistä, kun se lähetetään asiakasohjelman kautta (lähteen 2 tietojen pohjalta).

On tärkeää ymmärtää, että sähköposti ei välttämättä kulje suoraan yhdeltä palvelimelta toiselle, vaan välissä voi olla useita muitakin palvelimia. Tämän voi mieltää kirjepostin kulkuun, jossa kirje voi siirtyä usealta välittäjältä toiselle, ennen kuin se saapuu vastaanottajan postilaatikkoon. Sähköpostin

välittämiseen käytettävä tekniikka voi muuttua, kuten kirjettä voidaan kuljettaa eri kuljetustavoilla, ennen kuin se saavuttaa vastaanottajansa.

2.2 SMTP-protokolla

Simple Mail Transfer Protocol eli SMTP on IETF:n määrittelemä protokolla, jonka päämäärä on välittää sähköpostia luotettavasti ja tehokkaasti (2, s. 5). Se perustuu kaksisuuntaiseen tiedonsiirtoyhteyteen SMTP-asiakkaan ja -palvelimen välillä (ks. kuva 3). SMTP-asiakkaan velvollisuus on siirtää sähköposteja yhdelle tai useammalle SMTP-palvelimelle ja kertoa onnistumisesta tai epäonnistumisesta tehdä niin. (2, s. 7.) Käytännössä SMTP-protokollalla sähköpostipalvelimet viestivät keskenään välittääkseen sähköpostia.



Kuva 3. SMTP-protokollan toimintaperiaate (2).

Sähköpostin lähetys SMTP:llä alkaa tiedonsiirtoyhteyden avaamisella jollakin tietoliikenneprotokollalla (2, s. 8). SMTP-protokollan määritellyssä RFC 5321 -dokumentissa mainitaan TCP-protokolla, mutta SMTP voi toimia muullakin tietoliikenneprotokollalla, kunhan tiedonsiirtoyhteys on luotettava (2, s. 5). Tietoliikenneyhteyden muodostuttua SMTP-asiakas voi aloittaa postin siirron. Postin siirto koostuu komennoista, joita SMTP-asiakas suorittaa ja joihin SMTP-palvelin vastaa. Komennoilla on tiukka syntaksi, ja jokaiseen komentoon kuuluu vastata kolminumeroisella koodilla. Esimerkkikoodi 1 havainnollistaa SMTP-istuntoa, jossa sähköpostipalvelimelle avataan yhteys ja käyttäjä lähettää

palvelimelle komennon QUIT, joka merkitsee pyyntöä lopettaa istunto. Palvelimen vastaus kertoo asiakkaalle, oliko suoritettu komento hyväksytty, tapahtuiko komennon suorittamisessa väliaikainen tai pysyvä virhe tai vaaditaanko asiakasta suorittamaan lisäkomentoja. (2, s. 8, 16.)

```
[user@localhost seminaari]$ telnet 192.168.1.25 25
Trying 192.168.1.25...
Connected to palvelin.localhost.
Escape character is '^]'.
220 localhost ESMTP Postfix
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

Esimerkkikoodi 1. Yksinkertainen SMTP-istunto sähköpostipalvelimen kanssa Telnet-komentorivityökalua käyttäen. Telnet-ohjelmalla avataan yhteys sähköpostipalvelimeen portista 25, ja istunto alkaa toimimaan SMTP-protokollan mukaisesti.

Yksi SMTP:n tärkeistä ominaisuuksista on myös sen kyky välittää sähköpostia eri verkkojen välillä, esimerkiksi laajaverkon (engl. WAN) ja sisäverkon välillä (engl. LAN) (2, s. 5). Tällainen sähköpostipalvelimen erikoistyyppi on yhdyskäytäväpalvelin (engl. gateway). Se toimii siten, että SMTP-asiakas luovuttaa sähköpostin palvelimelle, joka lupaa huolehtia sähköpostin perillepääsystä tai mahdollisesti raportoida tilanteesta, jossa sähköpostia ei voida kuljettaa perille saakka. Kun sähköpostipalvelin saa sähköpostin hoitaakseen, se ottaa itse asiakkaan roolin ja ottaa yhteyden taas seuraavaan sähköpostipalvelimeen. Tätä voi toistua niin kauan, kunnes sähköpostin päätepiste saavutetaan tai kunnes jokin sähköpostipalvelin toteaa kuljettamisen mahdottomaksi. Palvelinta, joka voi ottaa SMTP-asiakkaan roolin, kutsutaan välityspalvelimeksi (engl. relay server). Nämä sähköpostipalvelimien erityismuodot tekevät sähköpostin kuljetusekosysteemistä monipuolisen. Yksi sähköposti saattaa kulkea jokaisen mainitun sähköpostipalvelimen erikoistyypin läpi, ennen kuin se saavuttaa kohteensa. Jokaisen sähköpostipalvelimen kuuluu toteuttaa SMTP-protokolla huolellisesti, jotta sähköpostin kuljetusketju säilyy mahdollisimman luotettavana.

2.3 Nimipalvelujärjestelmä

Nimipalvelujärjestelmä (engl. Domain Name System) eli DNS toimii osana sähköpostin kuljetusketjua. Sähköpostipalvelimen verkkotunnus liitetään palvelimen IP-osoitteeseen nimipalvelujärjestelmän avulla. Kun pyritään avaamaan yhteys sähköpostipalvelimeen, tarvitaankin vain palvelimen verkkotunnus eikä IP-osoitetta tarvitse muistaa. Tämä on erityisen hyödyllistä, jos palvelimen IP-osoite sattuu vaihtumaan. Sähköpostin lähetyksessä kohdesähköpostipalvelin määritellään vastaanottajan sähköpostiosoitteen verkkotunnusosan kautta (2, s. 69). Ilman toimivaa nimipalvelujärjestelmää ei voida tietää, minne sähköposti kuuluisi lähettää, jotta se saapuisi vastaanottajalleen. Vaikka nimipalvelujärjestelmä on osa sähköpostipalvelimen auditointityökalun toimintaa, aiheen laajuuden takia insinöörityöraportissa avataan DNS-testejä vain pintapuolisesti luvussa 3.

2.4 Sähköpostipalvelimen auditoinnin haasteet

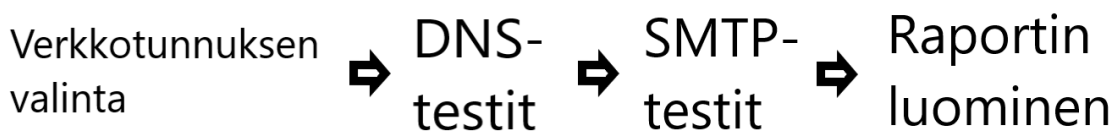
Sähköpostipalvelimen auditointi ei ole yksinkertaista, sillä mahdollista auditoitavaa on paljon. Sähköpostipalvelinohjelma saattaa olla päivittämätön ja täten haavoittuvainen (4), tai palvelimen alustana toimiva käyttöjärjestelmä itse voi olla alttiina verkkorikollisten hyökkäyksille. Verkon palomuuriasetukset saattavat vaarantaa sähköpostin kuljetuksen, tai palvelimen käyttämä tietokanta saattaa olla määritelty tietoturvaltaan puutteellisin keinoin. Itse sähköpostipalvelinohjelmistossa on myös mahdollista olla konfigurointioptioita, jotka saattavat sähköpostipalvelimen tai sitä kautta kulkevat sähköpostit vaaran alaiseksi. Auditoinnista tulee ongelma, kun auditoitavan kohteen mittakaava kasvaa. Voi olla perusteltua väittää, että kaikkea sähköpostinvälitysekosysteemiin liittyvää osaa ei ole mielekästä tarkistaa ajanpuutteen tai taloudellisten syiden vuoksi. Tämän takia tässä insinöörityössä sähköpostipalvelimen auditointi rajataankin palvelimen suorittamaan sähköpostipalvelinohjelmaan.

Sähköpostipalvelinohjelmien toteutukset voivat poiketa toisistaan. Sähköpostia kuljettaakseen sähköpostipalvelinohjelmien kuuluu kuitenkin toteuttaa protokollia, jotka mahdollistavat sähköpostin kuljetuksen. SMTP-protokolla on yksi näistä protokollista. Poikkeuksena SMTP-protokollan toteutukseen on sähköpostipalvelin, joka toimii ”message submission” -roolissa (2, s. 9; 5). Toinen protokolla, jonka sähköpostipalvelimen voidaan olettaa toteuttavan, on TLS-salausprotokolla. Tämän voi perustella tietoturvalla, sillä ilman TLS-protokollaa sähköpostia ei salata kolmansilta osapuolilta (6) ja tietosuojan saavuttamiseksi voidaan todeta sähköpostin salauksen olevan aiheellista toteuttaa.

Kun sähköpostipalvelinta auditoidaan sen toteuttamien protokollien mukaan, on mahdollista seurata suoraan IETF:n määrittelemiä standardeja ja luoda vaatimuksia, jotka ovat yhteisiä jokaiselle sähköpostipalvelimelle. Luvussa 4 käydään läpi muun muassa sähköpostipalvelimen SMTP-protokollan ja TLS-salausprotokollalaajennuksen toteutuksen tarkistamista. Lisäksi otetaan viitteitä muistakin IETF:n dokumenteista, joiden on tarkoitus parantaa sähköpostipalvelimen tietoturvaa ja toiminnallisuutta.

3 Sähköpostipalvelimen auditointityökalu

Insinöörityössä tehty työkalu on Python-ohjelmointikielellä kirjoitettu sovellus, joka tekee DNS- ja SMTP-testejä kohdeverkkotunnukseen ja luo raportin mahdollisista tietoturvauhkista ja käytettävyyden ongelmista. Ohjelma ottaa syötteenään kohteen, joka on se verkkotunnus, jonka sähköpostipalvelin tai sähköpostipalvelimet aiotaan auditoida. Verkkotunnus voi olla mikä tahansa, kunhan se on todellinen verkkotunnus. Kuva 4 havainnollistaa tätä vaiheittain suoritettavaa auditointiprosessia.



Kuva 4. Auditointityökalun suorittaman auditointiprosessin vaiheet.

Verkkotunnuksen saatuaan ohjelma alkaa ensimmäiseksi tehdä DNS-kyselyjä. DNS-kyselyillä selviää, hoitaako mikään palvelin kohdeverkkotunnuksen nimipalveluja. Nimipalvelimien toiminta on kriittistä, sillä ilman toimivaa nimipalvelujärjestelmää ei sähköpostia voida kohdistaa verkkotunnukseen. DNS-kyselyillä varmistetaan myös, että jokainen nimipalvelin on ajan tasalla, jotta jokainen niistä antaisi varmasti paikkansa pitävää tietoa. Testeihin kuuluu muun muassa nimipalvelimien kohdeverkkotunnukselle antamien tietueiden yhdenmukaisuuden testaus sekä rDNS eli käänteisnimipalvelun testaus (ks. kuva 5). Työkalun suorittamat DNS-testit ovat oleellinen osa sähköpostipalvelimen toiminnan auditointia, mutta aiheen laajuuden takia insinöörityöraportissa ei selvennetä paremmin testien yksityiskohtia.

DNS tests	Result
Name servers	PASS
Name server IP addresses	PASS
Zone serial number	PASS
Mail exchanges	PASS
Mail exchange IP addresses	PASS
Mail exchange reverse address records	PASS
Spamcop blacklist	PASS
SURBL blacklist	PASS
URIBL blacklist	PASS
Spamhaus blacklist	PASS
SORBS blacklist	PASS
Spamhaus Zen blacklist	PASS

Kuva 5. Ote auditointityökalun luomasta raportista, jossa esitellään tehdyt DNS-testit ja niiden tulokset.

DNS-kyselyillä saadaan selville, mitkä palvelimet ovat vastuussa kohdeverkkotunnuksen sähköpostipalveluista. Niitä voi olla useampia, ja ne ovat usein prioriteettijärjestyksessä. Koska työkalu selvittää verkkotunnuksen sähköpostipalvelimet DNS-kyselyjen avulla, se toimii täsmälleen niin kuin oikea sähköpostia lähettävä sähköpostipalvelin, ja täten työkalun löytämät virheet vaikuttavat myös todellisiin lähettäviin sähköpostipalvelimiin. Työkalu testaa jokaisen löydetyn sähköpostipalvelimen erikseen erojen varalta. Testeillä varmistutaan sähköpostipalvelimen SMTP-protokollatoteutuksen toiminnasta. Joukkoon kuuluu monta testiä, joista esimerkkejä ovat satunnaisen sähköpostiosoitteen ja palvelimen TLS-salausprotokollan testaus. Tätä auditointiprosessia käydään tarkemmin läpi luvussa 4. Työkalun erityisominaisuus on sen häiritsemättömyys, sillä työkalulla voidaan testata kaikki SMTP-protokollaan liittyvä toiminnallisuus ilman, että ainuttakaan oikeaa sähköpostia lähetetään. Tällä vältetään sähköpostiliikenteeseen osallistuvien

palvelimien turha kuormittaminen ja helposti roskapostiksi tulkittavien viestien lähettäminen.

Työkalu on vahvasti Internet-yhteydestä riippuvainen, sillä testattavat palvelimet voivat sijaita maantieteellisesti missä vain. Internet-yhteyden nopeus ei ole tässä niinkään merkityksellistä, vaan enemmänkin alhainen viive. Alhainen viive mahdollistaa reaaliaikaisen kommunikoinnin työkalua suorittavan tietokoneen ja palvelimen välillä. Alhainen viive kasvattaa työkalun suoriutumisnopeutta. Työkalun toiminnan nopeuttamiseksi on sen tietoverkkoyhteyksiä käyttävät osuudet toteutettu rinnakkaislaskennan avulla, ja näitä teknisiä yksityiskohtia avataan enemmän luvussa 5.

Testien jälkeen työkalu luo raportin, jossa kuvataan selkeästi auditoinnin aikana tapahtuneet asiat. Raportin alkupäässä on testien yhteenveto, joka antaa nopealla vilkaisulla yleiskuvan auditoinnin lopputuloksesta (ks. liite 1). Tämän jälkeen raportissa esitellään auditoinnissa tehdyt testit tarkemmin ja annetaan huomautuksia, jos tämä on tarpeen (ks. kuva 6). Huomautukset jaetaan kolmeen kategoriaan: Ensimmäinen niistä on ”ok”, joka kuvaa onnistunutta operaatiota. Toinen kategoria on varoitus, joka tarkoittaa sitä, että sähköpostipalvelin tai verkkotunnusta hoitavat DNS-palvelimet eivät toimi odotusten mukaisesti, mutta järjestelmän toiminta ei ole kuitenkaan estynyt. Kolmas kategoria on virhe, joka merkitsee virheellistä käytettävyyteen vaikuttavaa ongelmaa, kohonnutta tietoturvariskiä tai näitä molempia.

Mail to abuse@[redacted].fi	OK | Mail from [redacted] accepted. Response: 250 2.1.0 Ok WARNING | "RCPT TO:" expected response code 250 but got a temporary error. Response: 451 4.7.1 : Recipient address rejected: Greylisting in effect, please come back later OK | Server resetting the session Email to user "abuse" was not accepted. This email alias must be valid and accepted. It can be a real account or an alias to system's administrator. See RFC 2142 for more information.

Kuva 6. Ote erään verkkotunnuksen auditoinnin raportista, jossa annetaan varoitus "abuse"-sähköpostiosoitteen testissä tapahtuneesta tilanteesta. Varoitus sisältää sekä teknisen että selkokielisen selityksen huomautuksen sisällöstä. Tarkemmat tiedot testattavasta kohteesta on peitetty yksityisyyden suojaamiseksi.

4 Sähköpostipalvelimen auditointi

4.1 SMTP-istunnon avaaminen

Ensimmäinen askel sähköpostin lähettämiseen on avata yhteys sähköpostipalvelimelle. Kuten luvussa 2.2 mainitaan, yhteys avataan SMTP-protokollaa käyttäen. Sähköpostipalvelimen auditointityökalussa käytetään tähän BSD-pistoketta (engl. socket). Pistoke mahdollistaa eri tietokoneilla suoritettavien prosessien välisen kommunikoinnin tietoliikenneyhteyksiä käyttäen (7). Pistoke on ohjelmointirajapinnoissa matalan tason käsite, ja se antaa paljon vastuuta ohjelmoijalle ja samalla myös paljon varaa virheisiin. Pistokkeen avulla voidaan määritellä hyvinkin tarkasti, miten prosessit keskustelevat keskenään, ja virhe määrittelyssä voi tehdä prosessien välisestä kommunikoinnista mahdotonta. Auditointityökalu käyttää pistokkeita sen takia, että niiden avulla voidaan avata suora SMTP-protokollaa käyttävä istunto

sähköpostipalvelimelle. Näin auditointityökalu pysyy sähköpostin lähetykseen liittyvän prosessin joka vaiheessa mukana.

Insinöörityönä tehdyssä auditointityökalussa käytetään Pythoniin sisäänrakennettua socket-ohjelmakirjastoa, joka taas käyttää käyttöjärjestelmässä toimivia järjestelmäkutsuja pistokkeiden hallinnointiin. Työkalu avaa pistokkeen avulla auditoitavaan sähköpostipalvelimeen yhteyden portista 25, joka on SMTP-protokollaan yleisesti käytetty portti. Palveluiden käyttämät tietoliikenneprotokollien portit määrittelee IANA-järjestö, ja SMTP:n käyttämä portti on yksi näistä (8).

Jos yhteyttä ei kyetty avaamaan, koetaan tämä sähköpostipalvelimen virheelliseksi toiminnaksi. Mahdollisia syitä virheelliseen toimintaan on useita. Yksi todennäköinen syy on hetkellinen tietoverkkokatkos. Toinen mahdollinen syy on sähköpostipalvelimen palomuurin virheellinen konfiguraatio, joka estää yhteyden muodostamisen. Joka tapauksessa virhe yhteyden muodostamisessa kirjataan työkalun tekemään raporttiin.

Jos yhteyden muodostaminen onnistui, voi työkalu alkaa kommunikoimaan sähköpostipalvelimen kanssa SMTP-protokollaan suunniteltujen sääntöjen mukaan. Yhteyden muodostamisen jälkeen sähköpostipalvelin tervehtii yhteyden muodostajaa ilmoittamalla oman identiteettinsä, joka tässä tapauksessa tarkoittaa sähköpostipalvelimen verkkotunnusta (2). SMTP-protokollan mukaisesti sähköpostipalvelin voi tervehdysviestissään antaa hyödyllistä tietoa, kuten esimerkiksi käyttämänsä sähköpostipalvelinohjelmansa nimen. Yhteyden muodostamisen auditoinnin osalta on merkitystä vain sillä, tervehtiikö sähköpostipalvelin vai ei. Jos sähköpostipalvelin tervehtii jollain tavalla, tulkitaan yhteyden muodostamisen onnistuneen ja testi on suoritettu. Vastaavasti jos sähköpostipalvelimelta ei vastaanoteta minkäänlaista tervehdystä, palvelin toimii virheellisesti ja tästä tehdään merkintä raporttiin.

4.2 Tervehdyskomennot

Sähköpostipalvelimen täytyy tukea tervehdyskomentoa EHLO (2, s. 9). EHLO on tervehdyskomento, joka on alun perin määritelty dokumentissa IETF RFC 1425: SMTP Service Extensions osana SMTP-protokollan laajennuksia (9). EHLO-komento on nykyään osa SMTP-protokollaa ja määritelty dokumentissa RFC 5321: Simple Mail Transfer Protocol. Sähköpostipalvelimen tulee tukea EHLO-komentoa, jotta asiakas voi tietää, mitä laajennuksia palvelin tukee. (2, s. 9.) Laajennuksia ovat esimerkiksi palvelimen EHLO-komennon vastauksessa ilmoittama TLS-tuki (6) ja viestin enimmäiskoko (10). Historiallisen yhteensopivuuden takia sähköpostipalvelimen tulee myös tukea vanhempaa HELO-komentoa (2, s. 9). Nämä molemmat tervehdyskomennot, eli HELO ja EHLO, testataan.

Tervehdyskomentotestissä työkalu lähettää sähköpostipalvelimelle SMTP-protokollan syntaksin mukaisen EHLO-komennon. Komento on muotoa ”EHLO [verkkotunnus]” (2, s. 33), jossa verkkotunnus on työkalua suorittavan laitteen verkkotunnus. Palvelimen vastaus on se, mitä työkalu tässä testissä tarkastelee. Palvelin vastaa SMTP-protokollan syntaksin mukaisesti koodilla ja merkkijonolla. Vastausta arvioidessa koodi on tärkein, sillä se merkitsee operaation onnistumista tai epäonnistumista. Tervehdyskomennon kuuluisi aina onnistua, ja onnistunutta operaatiota kuvaa numerolla 2 alkava koodi. Jos koodi alkaa millään muulla numerolla, operaatio ei onnistunut ja testi hylätään. Esimerkkikoodi 2 havainnollistaa EHLO-komennon käyttöä sähköpostipalvelimella, minkä tuloksen auditointityökalu hyväksyisi. Komentosuoriteotteessa käyttäjä avaa yhteyden sähköpostipalvelimelle ja antaa komennon ”EHLO user.localhost”, jossa EHLO:n jälkimmäinen osa viittaa käyttäjän omaan verkkotunnukseen. Palvelin vastaa koodilla 250, joka merkitsee onnistunutta operaatioita, sekä listalla laajennuksia, joita palvelin tukee.

```
[user@localhost seminaari]$ telnet 192.168.1.25 25
Trying 192.168.1.25...
Connected to palvelin.localhost.
Escape character is '^]'.
220 localhost ESMTP Postfix
EHLO user.localhost
250-palvelin.localhost
250-PIPELINING
250-SIZE 30720000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

Esimerkkikoodi 2. Telnet-komentorivityökalulla käyty keskustelu, jossa asiakas tervehtii palvelinta asianmukaisesti EHLO-tervehdyksellä.

EHLO-komennon lisäksi testataan HELO-komento. Molemmat testit suoritetaan samalla tavalla. Ainoa ero testien välillä on palvelimen vastauksen merkkijonolla, joka HELO-komennon tapauksessa ei sisällä hyödyllistä tietoa palvelimen tukemista laajennuksista. Jos sähköpostipalvelimella on käytössään palvelimen tietoja peittävä palvelu, on EHLO-testin suorittaminen huomattavasti vaikeampaa. Joissakin sähköpostipalvelimissa on käytännön testeissä huomattu käytettävän palveluja, jotka antavat mahdollisimman vähän tietoa SMTP-istunnon aikana. Tämä ilmenee testissä siten, että palvelin antaa lukukelvottomia merkkijonoja vastauksissaan ja on mahdotonta arvioida sen tukemia laajennuksia. Tämä muun muassa estää palvelimen TLS-protokollatuen selvittämisen. Jos tällaisen palvelun todetaan olevan päällä, kirjataan tästä huomautus.

4.3 Viestin enimmäiskoko

SMTP-protokollan laajennus viestin enimmäiskoon rajoittamiseksi määritellään dokumentissa RFC 1870: SMTP Service Extension for Message Size Declaration (10). Multimediatuen ansiosta sähköpostien koot ovat kasvaneet merkittävästi, ja täten voi olla aiheellista rajoittaa viestin kokoa, jonka

sähköpostipalvelin sallii. On huomioitava, että SMTP-standardin määrittelemässä dokumentissa RFC 5321 kehotetaan olemaan rajoittamatta viestin kokoa. (2, s. 63.) Tästä kehotuksesta huolimatta on aiheellista tarkistaa sähköpostipalvelimen tuki viestin enimmäiskoon asettamiseen. Sähköpostit voidaan tallettaa, jotta niiden vastaanottaja voi lukea ne silloin kun haluaa. Sähköposteille varatut säilöt eivät kuitenkaan ole loputtoman kokoisia. Jos esimerkiksi sallitaan teratavun, mikä vuonna 2023 on paljon, kokoisten liitteiden lähettäminen, miltä tahansa palvelimelta voi loppua tila kesken, sillä palvelinten infrastruktuuria ei ole rakennettu sietämään näin suuria määriä dataa. Ongelma pahenee, jos liitteitä kantavalle sähköpostille on määritelty useita eri vastaanottajia, sillä sähköpostipalvelin voi tallettaa jokaiselle vastaanottajalle kopioita liitteistä.

Toinen ongelma viestin enimmäiskoon rajoittamatta jättämisessä piilee tiedonsiirrossa. Sähköpostipalvelinta ei nimittäin ole tarkoitettu tiedostopalvelimeksi, eikä standardissa esimerkiksi suositella protokollan toteuttajaa hyödyntämään tehokkaita menetelmiä suurien datamäärien siirtämistä varten. Puute sähköpostipalvelimen valmiudessa siirtää isoja datamääriä kerralla voi aiheuttaa palvelimen jumiutumisen suurien sähköpostien käsittelemisen ajaksi, ja tämä riski voidaan välttää asettamalla kohtuullinen rajoitus viestin koolle.

Viestin enimmäiskoon tarkistamiseksi täytyy ensin varmistua siitä, tukeeko sähköpostipalvelin kyseistä ominaisuutta. Tähän voi käyttää EHLO-tervehdyskomennon tulosta, sillä siinä palvelimen antama merkkijono paljastaa palvelimen tukemat SMTP-protokollan laajennukset. Jos ”SIZE” löytyy merkkijonosta, palvelin tukee viestin enimmäiskoon rajoittamista. ”SIZE”-merkkijonon esiintymisen lisäksi palvelin ilmoittaa asettamansa viestin enimmäiskoon tavuina. Jotta voidaan varmistua sähköpostipalvelimen oikeasta toiminnasta, työkalulla pyritään lähettämään sähköpostia, joka ylittää palvelimen asettaman rajan. Jos sähköpostipalvelin suostuu välittämään tämän viestin, voidaan todeta palvelimen toimivan virheellisesti, sillä se ei noudata omia rajoituksiaan. Työkalu ei siis ota kantaa siihen, mikä enimmäiskoko on

palvelimelle oikea, vaan se tarkistaa vain, noudattaako sähköpostipalvelin sille asetettua rajaa. Jos sähköpostipalvelimelle on asetettu iso raja sähköpostin koolle, on sen ylläpitäjien vastuulla pitää huolta tarpeeksi suuresta kapasiteetista sähköpostin tallentamiseen. Sähköpostipalvelimen auditointityökalua ei suunniteltu pääsemään käsiksi sähköpostipalvelimen tietokantoihin tai muihin siihen liitettyihin tiedostojärjestelmiin, jotta se voisi tarkastella palvelimelle asetettuja säilöjä.

Esimerkkikoodi 3 havainnollistaa viestin enimmäiskoon rajoittamisen testaamista käytännössä. Esimerkissä avataan yhteys sähköpostipalvelimelle ja tervehditään asianmukaisesti EHLO-komennolla, jonka avulla selviää palvelimen asettama rajoitus viestin koolle. Esimerkissä palvelin ilmoittaa tukevansa laajennusta ”SIZE 30720000”, jossa numero-osuus merkitsee viestin kokorajoitusta tavuina. Käyttäjä antaa tämän jälkeen ”MAIL FROM”-komennon, jonka parametriksi annetaan ”SIZE=30900000”. Käyttäjän antama koko on huomattavasti palvelimen antamaa rajoitusta suurempi, joten sähköpostipalvelin vastaa virhekoodilla 552 ja hylkää sähköpostin. Esimerkissä sähköpostipalvelin siis toimii oikein ja auditointityökalu antaisi käydystä keskustelusta hyväksytyn tuloksen.

```
[user@localhost seminaari]$ telnet 192.168.1.25 25
Trying 192.168.1.25...
Connected to palvelin.localhost.
Escape character is '^]'.
220 localhost ESMTP Postfix
EHLO user.localhost
250-palvelin.localhost
250-PIPELINING
250-SIZE 30720000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
MAIL FROM: <user@localhost> SIZE=30900000
552 5.3.4 Message size exceeds fixed limit
```

Esimerkkikoodi 3. Telnet-komentorivityökalulla käyty keskustelu, jossa se estää viestin lähettämisen kokorajoituksen takia.

Jos sähköpostipalvelin ei aseta minkäänlaista rajaa sähköpostin enimmäiskoolle, työkalu valitsee itse perustellun rajan ja kokeilee, antaako sähköpostipalvelin lähettää kyseisen kokoista sähköpostia. Työkalussa asetettu raja on 2 gibitavua, jonka oletetaan olevan kohtuullinen rajoitus. Jos palvelin suostuu välittämään sähköpostin, joka on kokoa 2 gibitavua tai sen yli, testi hylätään.

4.4 Sähköpostiosoitteet ”postmaster” ja ”abuse”

Sähköpostipalvelimen täytyy tukea sähköpostin välitystä osoitteeseen ”postmaster” (2, s. 61; 11, s. 1). ”postmaster”-osoite vaaditaan muun muassa virheilmoitusten takia (2, s. 68), ja täten osoitteen olemassaolo kuuluu testata sähköpostipalvelimella. Sähköpostipalvelinta vaaditaan myös hyväksymään postia ”abuse”-osoitteeseen (11, s. 2−3). Jos sähköpostipalvelin ei tue näitä osoitteita, se ei pysty vastaanottamaan ilmoituksia muun muassa toimittamatta jääneistä viesteistä.

”postmaster”- ja ”abuse”-sähköpostiosoitteita testatessa täytyy toimia SMTP-protokollan mukaisesti. Kuten luvuissa 4.1 ja 4.2 kerrotaan, sähköpostipalvelimelle täytyy ensin avata toimiva yhteys sekä tervehtiä sitä, ennen kuin voidaan aloittaa itse sähköpostin lähetys. Palvelin voi estää sähköpostin lähettämisen, jos asianmukainen tervehtiminen jää suorittamatta. Tämä toiminta on kuitenkin sähköpostipalvelimen toteutuksesta riippuvainen, eivätkä kaikki palvelimet välttämättä toimi niin.

Sähköpostiosoitteen olemassaolo voidaan testata tavanomaisilla komennoilla, joilla lähettää sähköpostia. Palvelimelle voidaan syöttää komento ”RCPT TO”, jolla sähköpostipalvelimelle ilmoitetaan, mihin sähköpostiosoitteeseen halutaan lähettää postia. Jos ”RCPT TO” -komennon parametrina on käytetty palvelimen ”abuse”- tai ”postmaster”-osoitetta, kuuluisi sähköpostipalvelimen vastata komentoon numerolla 2 alkavalla koodilla. Jos palvelin tekee niin, tarkoittaa tämä onnistunutta operaatiota ja verkkotunnuksella on olemassa kyseinen osoite. Jos palvelin vastaa millä tahansa muulla koodilla, on tapahtunut virhe ja

tämä merkitään raporttiin. ”postmaster”- tai ”abuse”-sähköpostiosoitteen hylkäämisen joko pysyvällä tai väliaikaisella virheellä havaittiin olevan tyypillinen vika sähköpostipalvelimella, kun auditointityökalulla suoritettiin käytännön kokeiluja. Vika havaittiin ainoastaan Microsoft Exchange Server -sähköpostipalvelimilla, mutta minkäänlaisten johtopäätösten tekeminen havaintojen pohjalta vaatii tarkempia testejä.

Esimerkkikoodi 4 havainnollistaa testin suorittamista. Esimerkissä käyttäjä avaa yhteyden sähköpostipalvelimelle, tervehtii asianmukaisesti ja aloittaa sähköpostin lähettämisen valmistelun ”MAIL FROM” -komennolla, jolla kerrotaan, kuka sähköpostia on lähettämässä. Palvelimen hyväksyttyä lähettäjän käyttäjä antaa komennon ”RCPT TO” ja sen parametrina ”postmaster”-osoitteen. Palvelin hyväksyy vastaanottajan asettamisen koodilla 250, joka auditointityökalun arvioinnissa merkitsee hyväksyttyä testiä.

```
[user@localhost seminaari]$ telnet 192.168.1.25 25
Trying 192.168.1.25...
Connected to palvelin.localhost.
Escape character is '^]'.
220 localhost ESMTP Postfix
EHLO user.localhost
250-palvelin.localhost
250-PIPELINING
250-SIZE 30720000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
MAIL FROM:<user@localhost>
250 2.1.0 Ok
RCPT TO:<postmaster@localhost>
250 2.1.5 Ok
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

Esimerkkikoodi 4. Telnet-komentorivityökalulla käyty keskustelu, jossa palvelin hyväksyy ”postmaster”-osoitteen määrittelemisen sähköpostin vastaanottajaksi.

Testin luotettavaa suorittamista vaikeuttaa palvelimen reagoiminen olemattomiin sähköpostiosoitteisiin. Sähköpostipalvelin voi antaa virheellisen

viitteen ”postmaster”- tai ”abuse”-sähköpostiosoitteen olemassaolosta, jos sähköpostipalvelin suostuu välittämään postia satunnaisiin sähköpostiosoitteisiin. Jos luvussa 4.5 selitetyssä satunnaisen sähköpostiosoitteen testissä tapahtuu virhe, se voi tarkoittaa virhettä myös tässä testissä. Virhettä voi olla vaikea saada selville ilman, että osoitteeseen oikeasti lähetetään sähköpostia. Kuten luvussa 3 on kerrottu, työkalu ei koskaan lähetä sähköpostia. Se vain asettaa lähettäjän ja vastaanottajan mahdolliselle sähköpostille.

4.5 Satunnainen sähköpostiosoite

Sähköpostipalvelin ei saa vastaanottaa sähköpostia osoitteeseen, jota ei ole olemassa. Kun vastaanottava sähköpostipalvelin ottaa vastaan sähköpostin, jonka vastaanottajaa ei ole olemassa, lähettävä sähköpostipalvelin voi sen asetusten mukaan lähettää virheilmoituksen takaisin sähköpostin lähettäjälle (12, s. 27). Verkkorikolliset kuitenkin käyttävät tekaistuja sähköpostiosoitteita lähettääkseen postia, joten sähköpostipalvelimen lähettämä virheilmoitus voi mennäkin toiselle sähköpostipalvelimelle, jolla ei ole mitään tekemistä alkuperäisen viestin kanssa. Tätä kutsutaan takaisinsironnaksi (engl. backscattering), ja tällä verkkorikolliset voivat hyödyntää muita sähköpostipalvelimia levittääkseen roskapostia. Takaisinsironta voi myös aiheuttaa palvelunestohyökkäyksen osoitteisiin, joihin sähköpostipalvelimet lähettävät virheilmoituksen toimittamatta jääneestä viestistä (12, s. 37).

Testi suoritetaan suurimmilta osin samoin tavoin kuin luvun 4.4 testi ”abuse”- ja ”postmaster”-sähköpostiosoitteista. Sähköpostipalvelimelle kuuluu ensin avata yhteys ja tervehtiä sitä asianmukaisesti. Tämän jälkeen voidaan käyttää ”RCPT TO” -komentoa ja antaa sille parametrina jokin kohdeverkkotunnuksen sähköpostiosoite, jonka tiedetään olevan olematon. Jos sähköpostipalvelin vastaa positiivisella eli 2-alkuisella koodilla, tiedetään palvelimen toimivan virheellisesti, sillä se hyväksyi sähköpostiosoitteen, jota ei ole olemassa. Toisin sanoen, jos sähköpostipalvelin vastaa komentoon 5-alkuisella koodilla, palvelin estää sähköpostiosoitteen lähettämisen ja toimii oikein.

Esimerkkikoodi 5 havainnollistaa auditointityökalun suorittamaa testiä, jossa sähköpostipalvelin torjuu satunnaisen sähköpostiosoitteen asettamisen vastaanottajaksi virhekoodilla 550. Auditointityökalu asettaa ensin viestille lähettäjän ”testaaja@domain.invalid”, joka hyväksytään koodilla 250. Tämän jälkeen auditointityökalu asettaa viestille vastaanottajan ”rnd89274@foobar.fi”, jonka olemassaoloa pidetään epätodennäköisenä. Sähköpostipalvelin hylkää vastaanottajan asettamisen koodilla 550 ja perustelee hylkäystä sillä, että vastaanottajaa ei löydy virtuaalinimitaulukosta. Sähköpostipalvelin toimii testissä oikein, joten auditointityökalu antaa siitä hyväksytyn tuloksen.

```
------------------------------------------------
Test: Mail to rnd89274@foobar.fi Result: PASS
OK | Mail from testaaja@domain.invalid accepted. Response: 250 2.1.0
Ok

OK | "RCPT TO:<rnd89274@foobar.fi>" got the expected response code 5.
Response: 550 5.1.1 <rnd89274@foobar.fi>: Recipient address rejected:
undeliverable address: host mappi.foobar.fi[2001:998:2e::101] said:
550 5.1.1 <rnd89274@foobar.fi>: Recipient address rejected: User un-
known in virtual alias table (in reply to RCPT TO command)

OK | Server resetting the session
------------------------------------------------
```

Esimerkkikoodi 5. Tekstipohjaisen raportin ote työkalun suorittamasta ”foobar.fi”-verkkotunnuksen auditoinnista. Ote sisältää auditointityökalun suorittaman testin välitulokset ja sen lopullisen tuloksen.

On huomioitava, että testiä suorittaessa palvelin voi vastata 4-alkuisella koodilla, joka tarkoittaa väliaikaista virhettä. Auditointityökalun käytännön kokeilussa tätä havaittiin tapahtuvan varsinkin Microsoft Exchange Server -sähköpostipalvelintoteutuksilla. Tässä tapauksessa päätös palvelimen toiminnan oikeellisuudesta ei olekaan yksinkertaista. Koska satunnaisen sähköpostin vaaroja ei ole merkitty SMTP-protokollaa määrittelevään dokumenttiin RFC 5321, ei ole yleistä ohjelinjaa, miten sähköpostipalvelimen kuuluisi toimia osoitteen kanssa, jota ei ole olemassa. Täten sähköpostipalvelin voikin vastata 4-alkuisella koodilla joka kerta, kun yritetään lähettää sähköpostia osoitteeseen, jota ei ole olemassa. Auditointityökalua suunniteltaessa kuitenkin todettiin selkeät ongelmat dokumentin RFC 5321 määrittelyssä, sillä se ei vastaa todellisiin uhkiin, joita sähköpostin lähettämisessä on käytännössä

ilmennyt. Työkalun virheentarkastuksessa katsotaankin virheen tapahtuneen, jos palvelin vastaa 4-alkuisella koodilla.

4.6 TLS-salausprotokolla

Transport Layer Security eli TLS on salausprotokolla. Se auttaa SMTP-asiakasta ja -palvelinta suojaamaan kommunikointinsa salakuuntelijoilta tai hyökkääjiltä joko osittain tai kokonaan. TLS ei ole suoraan osa SMTP-protokollaa, vaan se tulee laajennuksen kautta, joka on määritelty dokumentissa RFC 3207: SMTP Service Extension for Secure SMTP over Transport Layer Security. (6, s. 1.) Koska TLS on laajennus, sähköpostipalvelimen ei ole pakko tukea sitä. Tämän takia sähköpostipalvelimen TLS-tuki täytyy testata, jotta voidaan varmistua siitä, että sähköpostit säilyvät luottamuksellisina. Luottamuksellisuus tarkoittaa tässä yhteydessä sitä, että mikään kolmas osapuoli ei pääse lukemaan sähkö- postien sisältöä tai muokkaamaan sitä.

TLS-tuen lisäksi tulisi selvittää, mitä versiota TLS-salausprotokollasta sähköpostipalvelin tukee. Viimeisin TLS-salausprotokollaversio on 1.3, ja se on määritelty dokumentissa RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3 (13). Vanhoja versioita TLS-salausprotokollasta tulee välttää, sillä ne ovat alttiina salauksen murtamiselle. Erityisesti TLS 1.0- ja TLS 1.1 -versioita on kehotettu välttämään muun muassa SHA-1-tiivistealgoritmin vuoksi, jonka turvallisuus ei vastaa moderneja vaatimuksia. TLS 1.2 on TLS-salausprotokollan alin versio, jota sähköpostipalvelin saa käyttää salauksen muodostamiseen. (14.)

TLS-istunnon aloittamiseksi työkalu syöttää komennon STARTTLS. Jos sähköpostipalvelin tukee operaatiota, sen kuuluisi vastata 2-alkuisella koodilla. Jos palvelin vastaa millä tahansa muulla koodilla, palvelimen voidaan todeta toimivan virheellisesti, koska se ei suostu aloittamaan keskustelun salaamista. Vaikka työkalun luokittelemia virheitä ei ohjelmassa luokitella kriittisyysjärjestykseen, on virhe TLS-istunnon muodostamisessa vakava. Jos palvelin hyväksyy salatun istunnon muodostamisen, voi työkalu jatkaa. Tällöin

työkalu itse asiassa toistaa aikaisemmin tekemänsä SMTP-testit, jotka on selitetty luvuissa 4.2–4.5. Tämä tehdään, koska TLS-salattu istunto on lopulta SMTP-istunto; se on vain salattu kryptografisella algoritmilla. Virheet näissä testeissä huomioidaan muuten samalla tavalla, mutta raporttiin kirjataan erillinen huomautus, että virhe tapahtui TLS-istunnossa.

TLS-salausprotokollan testaamiseen kuuluu enemmänkin: Vaikka sähköpostipalvelin tukisikin TLS-yhteyksiä ja viimeisintä TLS-salausprotokollaa, tämäkään ei takaa yhteyden luotettavuutta. Ennen TLS-salauksen käyttöä tulee nimittäin varmistua palvelimen identiteetistä (13, s. 6). Jos esimerkiksi verkkotunnuksen nimipalveluita hoitava järjestelmä on joutunut verkkorikollisen haltuun, tämä voi korvata nimipalvelujärjestelmän sisältämät viitteet sähköpostipalvelimiin omilla palvelimillaan. Kun asiakas kysyy kaapatulta nimipalvelimelta, mitkä palvelimet hoitavat verkkotunnuksen sähköpostipalveluja, saatu vastaus voi johdattaakin verkkorikollisen omille palvelimille. Näin verkkorikollinen voi saada mahdollisesti luottamuksellista tietoa haltuunsa, jos lähettävä sähköpostipalvelin suostuu keskustelemaan vihamielisen vastaanottavan sähköpostipalvelimen kanssa. Sähköpostipalvelinta auditoidessa voidaankin varmistaa sähköpostipalvelimen TLS-varmenteet, jotta tiedetään, mille taholle sähköpostipalvelin kuuluu.

TLS-varmenne on palvelimelle myönnetty varmistusmerkintä siitä, että se on tunnustettu. Sähköpostipalvelimen omistavalle taholle myöntää TLS-varmenteen yleensä jokin luotettu varmennemyöntäjä, joka allekirjoittaa sähköpostipalvelimen verkkotunnukselle kuuluvan varmenteen omalla varmenteellaan. (13, s. 135.) Varmennemyöntäjä nauttii yhteisön luottamusta, mikä käytännössä tarkoittaa sitä, että varmennemyöntäjä myöntää varmenteita vain rehellisiin tarkoituksiin eikä esimerkiksi verkkorikollisille, jotka pyrkivät luomaan verkkotunnuksestaan näennäisesti luotettavan. Sähköpostipalvelimen varmenne ja varmennemyöntäjän varmenne luovat jo pienen varmenneketjun. Varmenteiden luotettavuutta kasvatetaan edelleen siten, että varmennemyöntäjän varmenteen allekirjoittaa juurivarmenne. Juurivarmennetta ylläpitää taho, jonka luottamuksen kuuluu olla absoluuttinen. Verkkorikolliset

voivat varastaa huolimattomasti talletettuja allekirjoitusavaimia ja esiintyä näin varmenteen omistajana ja allekirjoittaa itse lisää varmenteita. Tämän takia varmenteiden allekirjoitusavaimia tulee suojata ja siksi varmenneketjun juurella toimii kansainvälisesti tunnustettu järjestö, jonka ammattitaitoon voidaan luottaa.

Sähköpostipalvelimen TLS-varmenteet voidaan tarkistaa samalla, kun avataan TLS-salattu SMTP-istunto. Yhteyden muodostuksessa palvelin antaa kopion varmenneketjustaan, josta esimerkkinä toimii esimerkkikoodi 6. Sähköpostipalvelimen auditointityökalu tarkistaa varmenneketjun varmentamalla jokaisen ketjuun kuuluvan varmenteen. Työkalulla on käytössään ennalta määritelty lista luotetuista juurivarmenteista, joilla voidaan tarkistaa palvelimen antama juurivarmenne. Jos varmentaminen onnistuu, siirrytään ketjussa eteenpäin seuraavaan varmenteeseen ja tarkistetaan, onko palvelimen antama juurivarmenne allekirjoittanut kyseisen varmenteen. Onnistuneen varmennuksen jälkeen siirrytään taas seuraavaan varmenteeseen ja prosessi jatkuu niin kauan, kunnes saavutaan sähköpostipalvelimen omaan varmenteeseen ja varmennetaan se. Varmenneketjun testi on hyväksytty, jos varmenneketjun luottamus säilyy juurivarmenteesta ketjun päähän saakka. Jos jokin palvelimen antamista varmenteista on virheellisesti allekirjoitettu, testistä annetaan hylätty ja huomautus tästä sisällytetään raporttiin.

```
[user@localhost seminaari]$ openssl s_client -starttls smtp -tls1_3 -
crlf -connect smtp.metropolia.fi:25
CONNECTED(00000003)
depth=2 C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST
Network, CN = USERTrust RSA Certification Authority
verify return:1
depth=1 C = NL, O = GEANT Vereniging, CN = GEANT OV RSA CA 4
verify return:1
depth=0 C = FI, ST = Uusimaa, O = Metropolia Ammattikorkeakoulu Oy, CN
= smtp.metropolia.fi
verify return:1
---
```

Esimerkkikoodi 6. OpenSSL-komentorivityökalulla avataan yhteys Metropolian sähköpostipalvelimeen, jossa yhteys salataan TLS-salausprotokollan versiolla 1.3. Ohjelman suoritus palauttaa listan varmenteista, joiden varaan luottamus palvelimeen rakentuu.

Esimerkkikoodi 7 sisältää otteen ”foobar.fi”-verkkotunnuksen tekstipohjaisesta auditointiraportista. Ote sisältää 5 hyväksyttyä testiä, jotka liittyvät TLS-salatun yhteyden aloittamiseen ja sen toiminnan varmistamiseen. Esimerkissä auditointityökalu antaa komennon STARTTLS, jolla siirrytään TLS-salattuun SMTP-istuntoon. Sähköpostipalvelin hyväksyy komennon koodilla 220. Tämän jälkeen suoritetaan niin kutsuttu kättely, jossa keskusteluparit neuvottelevat salausavaimet, joilla suorittaa SMTP-istunnon salaus. Onnistuneen kättelyn jälkeen auditointityökalu tarkistaa sähköpostipalvelimen varmenneketjun juuresta loppuun. Koska varmenteiden varmistusprosessissa ei ilmaannu virheitä, auditointityökalu hyväksyy testin varmenneketjuista. Tämän jälkeen auditointityökalu suorittaa vielä kaksi testiä, joissa varmistutaan siitä, mille verkkotunnukselle palvelimen varmenne on annettu, ja siitä, onko palvelimelle annettu varmenne päässyt vanhentumaan.

```
------------------------------------------------
Test: STARTTLS     Result: PASS
OK | STARTTLS got the expected response code 220. Response: 220 2.0.0
Ready to start TLS

------------------------------------------------
Test: TLS handshake     Result: PASS
OK | TLS handshake successful
------------------------------------------------
Test: Certificate chain Result: PASS
OK | Certificate "ISRG Root X1" successfully verified.
OK | Certificate "R3" successfully verified.
OK | Certificate "ikijono.foobar.fi" successfully verified.
------------------------------------------------
Test: Certificate subject    Result: PASS
OK | Subject and hostname match "bar-jono.foobar.fi".
------------------------------------------------
Test: Certificate dates     Result: PASS
OK | Certificate is valid.
------------------------------------------------
```

Esimerkkikoodi 7. Tekstipohjaisen raportin ote työkalun suorittamasta ”foobar.fi”-verkkotunnuksen auditoinnista. Ote sisältää auditointityökalun suorittamien testien välitulokset ja niiden lopulliset tulokset.

5 Auditointityökalun toteutus

5.1 Auditointityökalun perustus

Auditointityökalu kehitettiin toimimaan Foobar Linux 8 -käyttöjärjestelmällä, joka on kaupallinen korkean käytettävyyden Linux-jakelu. Työkalu käyttää Python 3.6 -ohjelmointikieltä ja useita avoimen lähdekoodin ohjelmakirjastoja toteuttamaan vaadittavat ominaisuudet sähköpostipalvelimien auditoimiseen.

Komentokehotteelta suoritettavat ohjelmat ovat tyypillisiä ohjelmoijan apuvälineitä. Auditointityökalu rakennettiinkin suoritettavaksi komentokehotteelta käsin, ja sille on mahdollista antaa parametreja suoritustavan muuttamiseksi. Esimerkiksi ohjelman suoritusajan tapahtumista on mahdollista saada yksityiskohtaisempaa tietoa asettamalla lokikirjauksen parametri. Työkalulle ei suunniteltu graafista käyttöliittymää, sillä sen ensisijainen suoritustapa on automatisoitu. Merkittävä etu komentokehotetyökalussa on sen joustavuus muiden komentokehoteoperaatioiden kanssa. Esimerkiksi Unix-järjestelmissä on mahdollista putkittaa ohjelman tuloste, jonka avulla ohjelman palaute voidaan syöttää toiselle ohjelmalle tai vaikka tekstitiedostoon. Putkittamalla voidaan siirtää auditointityökalun luoma raportti esimerkiksi verkkosivujen esittämisessä käytettyyn HTML-tiedostoon, jonka voi avata verkkoselaimessa.

5.2 Testien tulosten tallentaminen ja tarkistaminen

Kattavat testitulokset määrittelevät auditointiprosessin hyödyllisyyden toiminnallisia ongelmia ja tietoturvahaavoittuvuuksia ratkottaessa. Auditointityökalulla pyritään tallentamaan kaikki testien aikana tapahtuvat asiat, joista voi olla hyötyä. Tämä tarkoittaa nimipalvelukyselyissä monipuolisten tietorakenteiden hallinnointia tietueiden varastointiin. SMTP-protokollan testeissä auditointityökalu tallentaa kaikki sen käymät keskustelut sähköpostipalvelimien kanssa.

Auditointityökalu tarkastaa automaattisesti testien lomassa niiden tuloksia. Tuloksista etsitään virheellisiä tietoja, jotka paljastavat testattavan kohteen toimivan väärin. Esimerkiksi SMTP-protokollan testeissä verrataan sähköpostipalvelimen vastauksissaan antamia koodeja. Niitä verrataan niihin koodeihin, joita auditointityökalu odottaa saavansa virheettömässä keskusteluistunnossa. Jos auditointityökalu ei pysty osoittamaan testattavan kohteen toimivan virheellisesti, testi merkitään hyväksytyksi. Tällä periaatteella toimivat kaikki auditointityökalun tekemät testit.

Testien tuloksista selviää, toimiiko testattava kohde väärin. Huomautus virheellisestä toiminnasta on sekin tärkeä tieto, mutta tärkeämpää on tietää, minkä takia toimitaan väärin. Työkalulla on käytössään kattava tietopankki testattavien asioiden yksityiskohdista ja luonnollisella kielellä kerrotuista selityksistä, minkä takia jotain pidetään virheellisenä toimintana. Tietopankkina itse asiassa toimii kielitiedosto, jonka avulla voidaan kirjoittaa raporttiin käyttäjän kielen mukaisia selityksiä testien yksityiskohdille.

5.3 Nimipalvelukyselyt

Auditointityökalu käyttää Python-ohjelmointikielelle kehitettyä dnspython-ohjelmakirjastoa nimipalvelukyselyjen suorittamiseen. Työkalu selvittää ensimmäiseksi, mitkä palvelimet hoitavat verkkotunnuksen nimipalvelua. Tähän työkalu käyttää käyttöjärjestelmään asetettua DNS-ratkaisijaa (engl. DNS resolver). Käytännössä mikä tahansa DNS-ratkaisija käy, kunhan voidaan luottaa siihen, että ratkaisija kykenee löytämään testattavan verkkotunnuksen.

Kun nimipalvelimet ovat selvillä, työkalu alkaa tehdä varsinaisia testejä nimipalvelimille. Nimipalvelukyselyt ovat Internet-yhteydestä riippuvaisia, joten peräkkäisten kyselyiden suorittaminen on hidasta. Työkalun nopeuttamiseksi nimipalvelukyselyt suoritetaan asynkronisesti eli kyselyt lähetetään sarjassa mutta niihin ei odoteta vastausta, ennen kuin seuraava kysely lähetetään. Työkalulla on Internet-yhteyden muodostamisessa käytettyihin IPv4- ja IPv6-protokollaan täysi tuki. Asynkronisista kyselyistä tulee erityisen tärkeitä, jos

verkkotunnuksen nimipalvelimilla on käytössään molemmat versiot IP-protokollista, sillä tässä tapauksessa testit suoritetaan kahteen kertaan. Auditointityökalu ei kuitenkaan vaadi nimipalvelimen tukevan IPv6-protokollaa, joten sen toteuttamatta jättämisestä ei anneta virhettä.

Koska nimipalvelukyselyt kulkevat Internet-yhteydellä, yhteysongelmat ovat mahdollisia. Auditointityökaluun on toteutettu toiminnallisuus, jolla yhteysongelman sattuessa auditointityökalu voi yrittää tehdä nimipalvelukyselyn uudelleen. Väliaikaisista yhteysongelmista voidaan palautua asettamalla työkaluun aika, jonka työkalu odottaa yhteysongelman sattuessa, ennen kuin se tekee nimipalvelukyselyn uudestaan. Lopuksi auditointityökaluun on asetettu säädettävä aikaraja, jonka ylittyessä työkalu ei enää yritä tehdä nimipalvelukyselyä. Ratkaisemattomasta ongelmasta tehdään raporttiin merkintä.

5.4 Yhteys sähköpostipalvelimelle

Kuten luvussa 4.1 mainitaan, työkalu käyttää BSD-pistokkeita SMTP-yhteyden muodostamiseen sähköpostipalvelimelle. Työkalua käytetään Linux-pohjaisella käyttöjärjestelmällä, jossa pistokkeita voidaan käyttää järjestelmäkutsujen avulla. Näitä järjestelmäkutsuja käytetään Python-koodissa tähän sisäänrakennetun socket-ohjelmakirjaston avulla. Pistokkeet ovat joustavia, ja ohjelmoija voi itse päättää, miten niillä kommunikoidaan. Ongelmaksi tulee se, että pistokkeiden avulla kommunikointi on vuoropuhelua ja keskustelijoiden kuuluu keskustella samoilla säännöillä. Jos sääntöjä ei ole, prosessien on mahdotonta keskustella keskenään. Sähköpostipalvelimien välisessä viestinnässä IETF:n määrittelemä SMTP-protokolla laatiikin sääntöjä keskusteluun (2). Keskustelun säännöt toteuttamalla auditointityökalu voi tekeytyä lähettäväksi sähköpostipalvelimeksi ja pysyä mahdollisimman autenttisena sähköpostin välitysprosessissa.

Auditointityökalun pistokkeet käyttävät AF_INET-rajapinnan IPv4- ja IPv6-protokollan versioita tiedonvälitykseen. Pistokkeiden kommunikointityyliksi on

asetettu SOCK_STREAM, joka tarkoittaa tiedon välittämistä virtana. Virtauspistokkeella käyttöjärjestelmä varmistaa tiedon eheyden. Virtauspistokkeilla kommunikoivien prosessien kuuluu silti sopia tavoista, joilla merkitä lähetettävän ja vastaanotettavan tiedon pituutta tai sitä, millä tietovirta päätetään. (7.) SMTP-protokollan tapauksessa käytetään ASCII-merkistön CRLF-merkkiä merkitsemään lähetettävän ja vastaanotettavan tiedon päättymistä (2, s. 14). Jos auditoinnin yhteydessä työkalu vastaanottaa viestin, jossa ei ole CRLF-merkkiä, SMTP-istunnon todetaan olevan rikkinäinen ja testien teko lopetetaan. Päättävän CRLF-merkin lisäksi jokainen viesti muutetaan binääriksi ennen lähettämistä.

On kaksi vallitsevaa tyyliä hallinnoida pistokkeisiin lukemista ja kirjoittamista: lukkiutuvat ja lukkiutumattomat pistokkeet. Lukkiutuvat pistokkeet vangitsevat koko ohjelman siksi aikaa, kun tehdään luku- ja kirjoitusoperaatioita. Lukkiutumattomat pistokkeet taas antavat ohjelman tehdä muuta samalla kun pistokkeet toimivat. Auditointityökalussa lukkiutumattomien pistokkeiden hallinnointiin käytetään select-järjestelmäkutsua, jolla varmistutaan siitä, että yhteysongelmat eivät lukitse koko työkalua.

Pistokkeiden yhteysongelmia ratkaistaan samalla tavalla kuin nimipalvelukyselyjen yhteysongelmien ratkaiseminen, joka on selitetty luvussa 5.3. Jos yhteysongelmat katkovat auditointityökalun suoritusta, se voi asetusten mukaan yrittää yhteyden muodostamista uudelleen tai odottaa tietyn ajan ennen uudelleen yrittämistä. Jos pistokkeella luotu SMTP-istunto todetaan käyttökelvottomaksi, istunto katkaistaan ja raporttiin tehdään tästä merkintä.

5.5 TLS-salausprotokollan käyttäminen

Auditointityökalun Python-koodissa TLS-yhteyden muodostamiseen käytetään avoimen lähdekoodin pyOpenSSL-ohjelmakirjastoa. pyOpenSSL taas käyttää käyttöjärjestelmän OpenSSL-ohjelmakirjastoa suorittamaan kryptografisia funktioita. pyOpenSSL-ohjelmakirjastolla kääritään SMTP-istuntoon käytetty

pistoke, kun auditointityökalu ja sähköpostipalvelin sopivat aloittavansa salatun TLS-istunnon.

Toteutusyksityiskohtana auditointityökalu ei käytä lukkiutumattomia pistokkeita salatun TLS-istunnon aikana, sillä yhteyden muodostamisessa havaittiin ongelmia käytännön kokeiden aikana. select-järjestelmäkutsu kertoi virheellisesti pistokkeen olevan valmis kommunikointiin, vaikka tämä ei pitänyt paikkaansa. Ongelma kierrettiin käyttämällä TLS-istunnon aikana lukkiutuvia pistokkeita. Ratkaisun ei havaittu vaikuttavan auditointityökalun tehokkuuteen.

Auditointityökalun pitää tarkastaa sähköpostipalvelimen TLS-varmenteet, kun salattu TLS-istunto alkaa. Tähän käytetään myös OpenSSL-ohjelmakirjastoa, ja auditointityökalu seuraa varmenteiden tarkistusprosessia keräten merkinnät virheistä. Tätä toteutettaessa huomattiin ongelma OpenSSL-ohjelmakirjaston käyttämässä OpenSSL-ohjelmakirjaston funktiokutsussa, jonka tarkoitus on palauttaa yhteyden varmentamiseen käytetty varmenneketju. Funktiokutsu palautti varmenneketjun, jonka juurivarmenne oli jo vanhentunut. Vanhentuneeseen varmenteeseen ei voi luottaa. Kyseessä oli Let’s Encrypt -varmennemyöntäjän käyttämä DST Root CA X3 -varmenne, ja OpenSSL-kirjasto palautti varmenneketjun, jossa tämä vanhentunut varmenne oli virheellisesti ketjun juurena. Ongelma kierrettiin käyttämällä toista OpenSSL-kirjaston funktiokutsua, jossa ongelmaa ei esiintynyt.

5.6 Raportin luominen

Testien päätteeksi auditointityökalu luo raportin käyttäjän toiveiden mukaan. Työkalulla on mahdollista luoda PDF-, HTML- tai tekstiraportti. Raportti luodaan testien päätteeksi niiden tulosten perusteella. Tekstipohjainen raportti on kaikista yksinkertaisin, ja se on suunnattu esittämään vain teknistä tietoa (ks. liite 2). Sen luonnissa ei hyödynnetä kielitiedostoa ja täten sitä on myös hankalampi lukea. Tekstipohjainen raportti onkin suunniteltu vain ammattilaisten luettavaksi. Tekstipohjaisen raportin luontiin käytetään käskyjä, jotka piirtävät

suoraan työkalua suorittavaan terminaaliin, ja täten työkalun tulostama teksti voidaan helposti ohjata tekstitiedostoon arkistointia varten.

Auditointityökalulla on mahdollista luoda verkkosivujen esitykseen käytettyä HTML-koodia, johon sisällytetään tehtyjen testien tulokset. HTML-koodi taas sisältää CSS-koodia raportin tyylittelyä varten. CSS-koodi on upotettu luotuun HTML-koodiin raportin kuljettamisen helpottamista varten, sillä koko raportti säilyy yhdessä tiedostossa. HTML-koodi sisältää useita esityksiä taulukoista, joihin testejä ja niiden tuloksia asetetaan. CSS-koodilla taas tyylitellään luotu raportti, jotta sitä on helpompi lukea. Tyylittelyyn kuuluu muun muassa tulosten värikorostus, joka auttaa silmää erottamaan huomioitavat testit.

PDF- ja HTML-raportit käyttävät erillistä kielitiedostoa, johon on tallennettu selkokielisiä selityksiä testien mahdollisista huomautuksista. Kun auditointityökalu antaa testistä huomautuksen, selkokielinen selitys liitetään automaattisesti raporttiin (ks. kuva 6). Selkokielisten selitysten tarkoitus on antaa teknisiin yksityiskohtiin perehtymättömälle henkilölle parempi kuva testien tilanteesta ja auttaa lukijaa ymmärtämään auditoinnin tulosten kokonaiskuvaa paremmin.

6 Auditointityökalun arviointi

Insinöörityönä tehdyn auditointityökalun lopputulokseen ollaan hyvin tyytyväisiä, sillä se selviytyy täsmälleen niistä tehtävistä, joihin se on suunniteltu. Työkalua testattiin kattavasti, ja sen toimintavirheitä ratkaistiin tehokkaasti. Työkalu kykenee testaamaan sähköpostipalvelimia hyvinkin luotettavasti, ja sen antamat raportit ovat kattavia ja antavat jokaiseen raportoituun ongelmaan kehitysratkaisuja. Erityisesti työkalun automaattinen testaus, joka on työkalun raskain toimintamuoto, toimii nopeasti ja sen testaukseen käyttämät resurssit ovat kohtuullisia.

Sähköpostipalvelimen auditointityökalun suunnittelu ja toteutus ei kuitenkaan ollut yksinkertaista, ja työkalusta kirjoitettiin useita eri versioita. Ensimmäisten

versioiden toimintavirheet hidastivat kehitystyötä. eikä ongelmiin aina ollut yksiselitteistä ratkaisua. Työkalusta tehtiin useita iteraatioita, koska yleisten tapojen löytäminen kaikkien palvelimien luotettavaan testaukseen ei ollut yksinkertaista. Tietoa työkalun käyttämistä protokollista SMTP, DNS ja TLS on olemassa, mutta kaikki tämä tieto on käytännössä sidottu IETF:n standardeihin tai sen muihin dokumentteihin. Vaikka IETF:n dokumentit selittävät protokollien toimintaa, pitää muistaa tiedon olevan abstraktia ja vain määritelmiä sille, miten protokollan toteuttavan ohjelman kuuluisi toimia. Sähköpostipalvelimet toteuttavat nämä protokollat seuraamalla dokumentteja tulkinnanvaraisesti. Itse auditointityökalu seuraa myös näitä protokollia määritteleviä dokumentteja ja toteuttaa ne oman tulkintansa mukaisesti. Ongelmaksi tulevat juuri nämä tulkinnan varaan jäävät asiat, joissa kaikki protokollien ominaisuudet eivät olekaan täysin yksiselitteisiä. Yksi esimerkki yksiselitteisyyden puutteesta on satunnaisten sähköpostiosoitteiden käsitteleminen, mikä voi saada palvelimen antamaan harhaanjohtavia väliaikaisia virheitä pysyvien virheiden sijaan. Ongelmaa kärjistää työkalun tähtääminen yleiseen tapaukseen, jossa se voi luotettavasti auditoida useita eri sähköpostipalvelintoteutuksia. Aiheen haastavuuteen nähden työkalu onkin menestys.

Auditointityökalun käytännön kokeissa pystyttiin huomaamaan ongelmia sähköpostipalvelimissa. Yksi havaittu ongelma oli luvussa 4.4 kerrottu ”postmaster”- tai ”abuse”-sähköpostiosoitteen puuttuminen, joka vaikuttaa sähköpostin kuljetusketjun toimivuuteen. Vaikka ongelmien löytyminen ei ole sähköpostin kuljetusketjun kannalta hyvä asia, on tärkeää, että työkalu kykeni paljastamaan ongelman, jotta se voidaan ratkaista mahdollisimman nopeasti. Työkalun kyky löytää ongelmia on sen suurin valtti sähköpostin kuljetusketjun turvallisuuden ja luotettavuuden varmistamisessa. On kuitenkin tärkeää olla realistinen auditointityökalun kanssa toimiessa, sillä sen tehokkuus määrittyy sen testipankin perusteella. On mahdollista, että on kokonaan uusia ongelmia, joita auditointityökalu ei huomaa, koska niille ei ole vielä suunniteltu testejä. On epätodennäköistä, että auditointityökalu kykenisi paljastamaan kaikki mahdolliset ongelmat sähköpostipalvelimella, mutta työkalun kehittäjällä on

suuri luottamus siihen, että se kykenee paljastamaan ainakin suurimman osan niistä.

7 Yhteenveto

Sähköposti on paljon käytetty digitaalinen viestintämenetelmä, ja sen suosion voidaan olettaa vain kasvavan. Sähköpostin avulla on myös mahdollista lähettää arkaluontoista sisältöä, mikä nostaa esiin aiheeseen liittyviä tietoturvakysymyksiä. Sähköpostin turvallisuus tulee taata, jotta sen käytettävyys ja luotettavuus säilyvät. Sähköpostipalvelimen auditointi on tähän hyvä keino, sillä jokainen sähköposti kulkee sähköpostipalvelimen läpi jossain vaiheessa kuljetustaan. Sähköpostipalvelinta auditoidessa on hyvä kiinnittää huomiota sen toteuttamaan SMTP-protokollaan, jonka määrittely on standardi ja joka on yleistettävissä lähes kaikkiin sähköpostipalvelimiin. Näin sähköpostipalvelimia voi testata samoilla keinoilla ja välttää ongelmat, joita on mahdollista tulla esimerkiksi eri käyttöjärjestelmille suunnatuista ohjelmistoista.

SMTP-protokollasta kuuluisi testata IETF:n viimeisimmän Simple Mail Transfer Protocol -dokumentin määrittelemät toiminnallisuudet. Sähköpostipalvelimen kyvyttömyys toimia protokollan mukaisesti voi aiheuttaa ongelman käytettävyydessä tai tietoturvassa. Esimerkiksi sähköpostipalvelimen virhe toteuttamisessa TLS-salausprotokollaan liittyvä SMTP-laajennus vaarantaa sähköpostien luottamuksellisuuden. IETF:n dokumenttien mukaiset toiminnallisuudet eivät kuitenkaan ole ainoat asiat, joihin kiinnittää huomiota sähköpostipalvelinta auditoidessa. Tietyt verkkorikollisten käyttämät toiminnallisuudet ovat nousseet esiin vasta vuosien käytännön kokeilun kautta, eikä esimerkiksi RFC 5321 varoita satunnaisten sähköpostiosoitteiden vaarasta tai viestien kokorajoitusten tärkeydestä. Sähköpostipalvelinta auditoidessa täytyykin käyttää käytännön kokemusta apuna itse standardien noudattamisen ohella, jotta tiedetään, mihin kiinnittää huomiota.

Insinöörityönä tehty auditointityökalu onnistuu tehtävässään, ja sen avulla on mahdollista auditoida luotettavasti ja tehokkaasti useita eri

sähköpostipalvelintoteutuksia. Auditointityökalun tärkeyttä määrittelevä tekijä on halu ylläpitää sähköpostien luottamuksellisuutta ja turvallisuutta. Käytännön kokeissa auditointityökalun avulla olikin mahdollista löytää todellisia ongelmia oikeista sähköpostipalvelimista. Esimerkiksi sähköpostipalvelimen virhe sähköpostin vastaanottamisessa ”postmaster”- tai ”abuse”-osoitteeseen oli tyypillinen esiintyvä virhe, kuten myös väliaikaisen virheen ilmoittaminen satunnaisen sähköpostiosoitteen asettamisesta.

Lähteet

1 Internet standards. Verkkoaineisto. Internet Engineering Task Force (IETF). <https://www.ietf.org/standards/>. Luettu 30.10.2022.

2 RFC 5321. Simple Mail Transfer Protocol. 2008. Internet Engineering Task Force (IETF).

3 Daily number of e-mails worldwide. 2022. Verkkoaineisto. Statista. <https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/>. Luettu 31.10.2022.

4 Murphy, C. 2006. Email--the killer application how safe are your mail servers. Accountancy Ireland. Vol. 38, s. 74−75.

5 RFC 6409. Message Submission for Mail. 2006. Internet Engineering Task Force (IETF).

6 RFC 3207. SMTP Service Extension for Secure SMTP over Transport Layer Security. 2002. Internet Engineering Task Force (IETF).

7 Sockets. Verkkoaineisto. FreeBSD Foundation. <https://docs.freebsd.org/en/books/developers-handbook/sockets/>. Luettu 16.3.2023.

8 Service Name and Transport Protocol Port Number Registry. Verkkoaineisto. Internet Assigned Numbers Authority (IANA). <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>. Luettu 16.3.2023.

9 RFC 1425. SMTP Service Extensions. 1993. Internet Engineering Task Force (IETF).

10 RFC 1870. SMTP Service Extension for Message Size Declaration. 1995. Internet Engineering Task Force (IETF).

11 RFC 2142. Mailbox Names for Common Services, Roles and Functions. 1997. Internet Engineering Task Force (IETF).

12 RFC 8601. Message Header Field for Indicating Message Authentication Status. 2019. Internet Engineering Task Force (IETF).

13 RFC 8446. The Transport Layer Security (TLS) Protocol Version 1.3. 2018. Internet Engineering Task Force (IETF).

14 RFC 8996. Deprecating TLS 1.0 and TLS 1.1. 2021. Internet Engineering Task Force (IETF).

Esimerkki auditointiraportin yhteenvedosta

MAIL SERVER AUDIT REPORT

Domain: foobar.fi

Report date: 2023-03-20

Tests passed: 100 %

Name servers
bar-ns.foobar.fi.
baz-ns.foocast.net.
foo-ns.foobar.fi.
qux-ns.foocast.net.
xyzzy-ns.foocast.net.

Zone serial
2023-03-19 13:01:34

Mail target	Priority
foo-jono.foobar.fi.	0
bar-jono.foobar.fi.	0

Test category	Passed tests	Failed tests
DNS	12	0
Mail to bar-jono.foobar.fi. 188.127.201.227	28	0
Mail to bar-jono.foobar.fi. 2a00:1190:c00a:f00::227	28	0
Mail to foo-jono.foobar.fi. 193.65.3.99	28	0
Mail to foo-jono.foobar.fi. 2001:998:2e::99	28	0

Ote tekstipohjaisesta auditointiraportista

```
------------------------------------------------
Test: SMTP: Open mail connection to 188.127.201.227   Result: PASS
OK | SMTP server is service ready. Banner: 220 bar-jono.foobar.fi
ESMTP Postfix

------------------------------------------------
Test: SMTP greeting     Result: PASS
OK | Greet successful. Response: 250 bar-jono.foobar.fi

------------------------------------------------
Test: Mailguard    Result: PASS
OK | No mailguard detected.
------------------------------------------------
Test: Mail to postmaster@foobar.fi    Result: PASS
OK | Mail from testaaja@domain.invalid accepted. Response: 250 2.1.0
Ok

OK | "RCPT TO:<postmaster@foobar.fi>" got the expected response code
250. Response: 250 2.1.5 Ok

OK | Server resetting the session
------------------------------------------------
Test: Mail to abuse@foobar.fi    Result: PASS
OK | Mail from testaaja@domain.invalid accepted. Response: 250 2.1.0
Ok

OK | "RCPT TO:<abuse@foobar.fi>" got the expected response code 250.
Response: 250 2.1.5 Ok

OK | Server resetting the session
------------------------------------------------
Test: Mail to rnd89274@foobar.fi Result: PASS
OK | Mail from testaaja@domain.invalid accepted. Response: 250 2.1.0
Ok

OK | "RCPT TO:<rnd89274@foobar.fi>" got the expected response code 5.
Response: 550 5.1.1 <rnd89274@foobar.fi>: Recipient address rejected:
undeliverable address: host mappi.foobar.fi[2001:998:2e::101] said:
550 5.1.1 <rnd89274@foobar.fi>: Recipient address rejected: User un-
known in virtual alias table (in reply to RCPT TO command)

OK | Server resetting the session
------------------------------------------------
Test: Close mail connection  Result: PASS
OK | Server closing connection
------------------------------------------------
```