jamk.fi

Tietojenkalastelutestaus

Automatisoidun alustan luominen

Pekka Sivusuo

Opinnäytetyö
Toukokuu 2020
Tekniikan ala
Insinööri (AMK), tieto- ja viestintätekniikka

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Sivusuo, Pekka	Opinnäytetyö, AMK	Toukokuu 2020
	Sivumäärä 72	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi
Tietojenkalastelutestaus
Automatisoidun alustan luominen

Tutkinto-ohjelma
Insinööri (AMK), tieto- ja viestintätekniikka

Työn ohjaaja(t)
Antti Häkkinen, Karo Saharinen

Toimeksiantaja(t)
Isoweli Oy

Tiivistelmä

Opinnäytetyön toimeksiantajana toimi jyväskyläläinen ICT-alan yritys Isoweli Oy. Työn toimeksiantona oli luoda mahdollisimman automatisoitu tietojenkalastelutestaus-alusta. Tietojenkalastelutestausta on Isoweli Oy:llä aikomus myydä palveluna asiakkailleen.

Tietojenkalastelu, joka myös tunnetaan toiselta nimeltä verkkourkintana, on tapa kerätä sensitiivistä tietoa kohteeltaan. Tietojenkalastelussa pyritään harhauttamaan uhria esimerkiksi tekeytymällä joksikin auktoriteettia omaavaksi tahoksi ja pyytää häneltä tämän varjolla sensitiivistä informaatiota joko uhrista itsestään tai yritykseltä. Tietojenkalastelua voidaan myös harjoittaa niin puhelimitse kuin kasvokkain.

Tietojenkalastelutestauksessa tulee ottaa huomioon testauksessa kerättä data, sen sensitiivisyys ja myös testauksen eettisyys. Tärkein asia tietojenkalastelutestauksessa on, että ketään yksilöä ei saa nolata testauksella. Tietojenkalastelutestauksen on tarkoitus herättää loppukäyttäjässä mielenkiintoa oman tietoturvan parantamiseen. Mikään tekninen ratkaisu ei voi suojata täydellisesti tietojenkalastelulta, joten tehokkain tapa suojautua on loppukäyttäjäkoulutus.

Koska Isoweli Oy tarjoaa palveluitaan pääsääntöisesti PK-yrityksille, tietojenkalastelutestaus täytyy pyrkiä hinnoittelemaan sellaiselle tasolle, että sen ostaisi myös pienemmän kassan omaava yritys. Täten alusta täytyi rakentaa mahdollisimman automatisoiduksi, jotta käsin tehtyä työtä on mahdollisimman vähän. Tämän mahdollistaa muun muassa Docker-konttitekniikka sekä Python ohjelmointikielen Pandas-kirjasto.

Työn tuloksena syntyi toimiva ja toimeksiantajaa hyvin palveleva tuote. Tuote herätti yrityksessä paljon mielenkiintoa ja sai hyvän vastaanoton.

Avainsanat (asiasanat)
Tietojenkalastelu, Phishing, Testaus, Docker, Python, Pandas, PHP, Automatisointi

Muut tiedot (Salassa pidettävät liitteet)

jamk.fi

Description

Author(s)	Type of publication	Date
Sivusuo, Pekka	Bachelor's thesis	May 2020
		Language of publication: Finnish
	Number of pages: 72	Permission for web publication: x

Title of publication

Phishing test

Creating an automated solution

Degree programme

Information Technology

Supervisor(s)

Antti Häkkinen, Karo Saharinen

Assigned by

Isoweli Oy

Abstract

This bachelor's degree was assigned by Isoweli Oy, a Jyväskylä-based ICT-company. The assignment was to create automated solution for a phishing test. Isoweli Oy is planning to sell phishing tests as a product for their clients.

Phishing is a method to collect sensitive information from their targets. Victim of phishing is usually tricked by impersonating high authority personnel and trying to collect sensitive information about the victim or company. Phishing can be conducted by email, phone or in person.

When conducting phishing test, notes needs to be taken about the data that is collected, sensitivity of data and ethical questions about the test. Most important thing in phishing test is that no one should be embarrassed by it. Phishing test is supposed to create interest in end-user about their personal cyber hygiene. There is no technical solution to prevent phishing perfectly, so the most efficient way is end-user training.

Isoweli Oy mainly produces ICT-services to small and medium sized enterprises so phishing test needs to be affordable for businesses with not so much capital. For this reason, solution for phishing test needs to be as automated as possible so it will not take many hours for a specialist to conduct the test. Docker-container technique and Python-programming language Pandas are some of the techniques that can help to achieve it.

As the result of assignment, Isoweli Oy received working and easy to use solution. Product got a lot of attention in the company and received good feedback from its employees.

Keywords/tags (subjects)

Phishing, Testing, Docker, Python, Pandas, PHP, Automated

Miscellaneous (Confidential information)

Sisältö

1 Johdanto 5

1.1 Toimeksiantaja 5

1.2 Työn tavoitteet 5

2 Tietojenkalastelu 7

2.1 Yleistä tietojenkalastelusta 7

2.2 Domain 8

2.3 Sähköposti 10

2.4 Verkkosivu 13

2.5 Haitalliset liitetiedostot 15

2.6 Uhrin hyväksikäyttö 15

3 Toteutuksen suunnittelu 16

3.1 Ympäristö 16

3.2 Web- sekä tietokantapalvelin 17

3.2.1 Verkkosivu 17

3.2.2 Tietokanta ja kerättävä data 17

3.3 Sähköposti 18

3.4 Domain sekä https 19

3.5 Tuloksien visualisointi ja esittäminen 19

4 Toteutus 20

4.1 Luvat 20

4.2 Palvelimen luonti 21

4.2.1 Yleistä 21

4.2.2 Front-end 21

4.2.3 Back-end 25

4.3 Datan visualisointi 27

4.4 Dockerin hyödyntäminen automatisoinnissa 29

4.4.1 Yleistä Dockerista 29

4.4.2 Toteutus 30

4.5 Testaus julkiverkossa 35

4.5.1 Julkiverkon palvelimen asennus 35
4.5.2 Domain-nimen hankinta ja DNS 36
4.5.3 Sähköpostin hankinta ja konfigurointi 36
4.5.4 Palvelun pystytys julkiverkkoon ja HTTPS 37

5 Yhteenveto 39
5.1 Tulokset 39
5.2 Suurimmat haasteet 40
5.3 Jatkokehitys 42

Lähteet 45

Liitteet 48
Liite 1. docker-compose.yml 48
Liite 2. master1.css 50
Liite 3. master2.css 53
Liite 4. config.php 57
Liite 5. index.html 58
Liite 6. verify.php 59
Liite 7. sqltograph.py 61
Liite 8. web.dockerfile 63
Liite 9. 000-default.conf 64
Liite 10. CreateTable.sql 65
Liite 11. Graafit sekä Excel 66
Liite 12. fail2ban-sovelluksen jail.local 67
Liite 13. lsoweli.fi DNS-konfiguraatio 68
Liite 14. Tietojenkalastelusivun ulkoasu 69
Liite 15. Tuotantoympäristön topologia 70
Liite 16. Testiympäristön topologia 71
Liite 17. Tuotantoympäristön Maltego-raportti 72

Kuviot

Kuvio 1. Domain-hierarkia 9
Kuvio 2. Tietojenkalasteluviesti 11
Kuvio 3. Tietojenkalasteluviesti 12
Kuvio 4. Tietojenkalasteluviesti suomeksi 12
Kuvio 5. Siisti tietojenkalastelusivusto 13
Kuvio 6. Huolimattomasti tehty tietojenkalastelusivusto 14
Kuvio 7. Microsoftin kirjautumissivusto 22
Kuvio 8. index.html- sekä verify.php-tiedostojen ylimmät rivit 22
Kuvio 9. jQuery-skripti animointia varten 23
Kuvio 10. Sisäänkirjautumislaatikon form-parametrit 24
Kuvio 11. verify.php-sivun salasanan syöttö 24
Kuvio 12. Tunnistautuminen ja yhteys tietokantaan 25
Kuvio 13. Tietojen keruu ja syöttö kantaan 26
Kuvio 14. Käyttäjän ohjaus pois sivustolta 26
Kuvio 15. Tietokannan users-taulun rakenne 27
Kuvio 16. Datan näkyminen tietokannassa 27
Kuvio 17. Tarvittavat Pythonin kirjastot 28
Kuvio 18. Yhteys MySQL-tietokantaan 28
Kuvio 19. Graafin tallennus levylle 29
Kuvio 20. Excel-datan tulostus ja loppu 29
Kuvio 21. Dockerin kansiorakenne 31
Kuvio 22. Docker-composen webkontin määrittely 32
Kuvio 23. Docker-composen mysql-määritykset 33
Kuvio 24. web.dockerfile-tiedoston sisältö 34
Kuvio 25. Tietokannan datan lataaminen isäntäkoneelle 34
Kuvio 26. users-taulun tyhjennys 35
Kuvio 27. Domain nimen A-tietue 36
Kuvio 28. Microsoftin vahvistuskoodi TXT-tietueena 37
Kuvio 29. Postitietueet Microsoft 365 -hallinnassa 37
Kuvio 30. Sivuston varmenne 38

Lyhenteet

APT	Advanced Persistent Threat
Bash	Bourne Again Shell
CSS	Cascading Style Sheets
DNS	Domain Name System
DPI	Dots Per Inch
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Tecnhology
ID	Identifier
IP-osoite	Internet Protocol -osoite
LAN	Local Area Network
PAM-tunnistautuminen	Pluggable Authentication Modules
PK-yritys	Pieni ja keskisuuri yritys
PHP	PHP: Hypertext Preprocessor
RGB	Red Green Blue -värimalli
SPF	Sender Policy Framework
SSH	Secure Shell
VLAN	Virtual LAN
YML/YAML	YAML Ain’t Markup Language

1 Johdanto

1.1 Toimeksiantaja

Toimeksianto opinnäytetyölle on saatu Isoweli Oy:ltä, joka on vuonna 1999 perustettu ICT-alan yritys. Isoweli Oy tarjoaa asiakkailleen ICT-ratkaisuja tarpeiden mukaisesti räätälöitynä. Isoweli Oy:n keskeisimpiin palveluihin kuuluvat ICT-ympäristön ylläpito, laitemyynti, loppukäyttäjätuki sekä verkkosivujen suunnittelu ja ylläpito (Isoweli Oy n.d).

Vuoden 2019 aikana Isoweli Oy:n asiakkaat saivat viikoittain tietojenkalasteluviestejä ja heidän järjestelmiinsä myös päästiin näiden avulla. Vuonna 2019 Isoweli Oy sai viikoittain analysoida ja vähentää tietojenkalastelusta aiheutuvaa vahinkoa tutkimalla lokitietoja ja konsultoimalla asiakkaitaan, miten heidän täytyy hoitaa viranomaisasiat ja liikesuhteet tietomurron jälkeen.

1.2 Työn tavoitteet

Verkkorikoksista tietojenkalastelu on todennäköisesti yleisin, johon normaali käyttäjä on törmännyt konkreettisesti käyttäen verkkopalveluita. Tietojenkalastelu on usein hyvin automatisoitua ja lisäksi se hyödyntää ihmisen tekemiä virheitä todella tehokkaasti.

Tietojenkalastelua voidaan vähentää teknisillä ratkaisuilla, jotka skannaavat haitallisia linkkejä tai analysoivat sähköpostin rakennetta. Kuitenkaan kohdennettua tietojenkalastelua ei voida teknisesti estää, vaan tehokkaimpana on käyttäjien koulutus.

Toimeksiantaja on saanut kyselyitä asiakkailtaan, onnistuisiko heidän henkilökunnalle tehdä tietojenkalastelutestausta. Tietojenkalastelutestauksessa lähetettäisiin kohdennettuja tietojenkalasteluviestejä esittäen joko itse kohdeorganisaatiota tai toista

yhteistyökumppania luvalla, esimerkiksi Isoweli Oy:tä. Tietojenkalasteluviestissä jaettaisiin linkkiä Isoweli Oy:n hallinnoimalle palvelimelle, joka kerää sinne syötetyn datan ja testauksen lopuksi kerää datan, analysoi sen ja esittää työn tilaajalle.

Isoweli Oy tarjoaa asiakkailleen tietoturvakoulutusta ja tietojenkalastelutestaus olisi lisäpalvelu koulutukseen. Yritykseen kohdistuneesta testauksesta saadaan mielenkiintoista dataa esiteltäväksi ja tämän toivotaan lisäävän myös koulutuksen tilaajan sekä henkilökunnan mielenkiintoa koulutusta kohtaan.

Isoweli Oy:n asiakkaina on paljon PK-yrityksiä joilla ei välttämättä ole suurta budjettia ICT-ympäristön ylläpitoon. Täten toimeksiantaja on pyytänyt automatisoimaan alustan siten, että käyttöönotto- sekä ylläpitokustannuksia on minimaalinen määrä, jotta testauksen hinta tulee houkuttelevaksi myös pienemmille yrityksille.

Opinnäytetyön toimeksiantona oli siis tutkia, mitä tekniikoita sekä tapoja tietojenkalasteluun tulisi käyttää siten, että se on kustannustehokasta, näyttää aidolta sekä kohdennetulta ja antaa arvokasta dataa työn tilaajalle.

Opinnäytetyöraportissa selvitetään, millaista tietojenkalastelu on ja miten sitä ovat toteutettaneet niin yksittäiset krakkerit kuin valtiolliset toimijatkin. Tämän lisäksi käydään läpi tyypillinen tietojenkalasteluskenaario niin tekniikoiden kuin motiivin osalta.

Työn toteutuksessa luotiin aluksi testiympäristö suljettuun verkkoon, jossa eri tekniikoita päästiin testaamaan tehokkaasti. Kun testaus oli suljetussa verkossa todettu toimivaksi, toteutus siirrettiin julkiseen verkkoon testaukseen julkisen verkon ominaisuuksia hyödyntäen.

2 Tietojenkalastelu

2.1 Yleistä tietojenkalastelusta

Tietojenkalastelu (engl. phishing) joka tunnetaan myös termillä ”verkkourkinta” on tapa manipuloida käyttäjää luovuttamaan sähköistä tietoa sähköisillä viestintämenetelmillä kuten sähköpostilla tai verkkosivulla. Tietojenkalastelussa pahantekijä yleensä esittäytyy jonain luotettavana instituutiona, joka yrittää saada uhrinsa luovuttamaan sensitiivistä informaatiota kuten luottokorttitietoja tai kirjautumistunnuksia (What Is Phishing? n.d).

Tietojenkalastelu on osa sosiaalista hakkerointia (engl. social engineering) jonka tarkoituksena on manipuloida uhria niin kasvokkain kuin sähköisesti. Sosiaalinen hakkerointi on tehokas tapa saada sensitiivistä tietoa uhrista eikä vaadi välttämättä laajaa teknistä osaamista. Tyypillisesti pahantekijä pyrkii manipuloimaan uhriaan esittämällä tärkeää henkilöä tai esimerkiksi ICT-osaston ylläpitäjää. Tapauksessa pahantekijä voi esimerkiksi tekaista skenaarion, että yrityksessä on epäilty tietomurto ja ICT-ylläpitäjänä hän tarvitsee käyttäjän tunnukset, jotta pääsee tutkimaan mitä hänen työasemallaan on tapahtunut. Jos uhri alkaa vastustelemaan, pahantekijä alkaa käyttäytyä aggressiivisemmin ja painottamaan, että on uhrin vika, jos jotain sattuu.

Uhrin manipulointi tunteisiin vedoten on tehokasta. Tehokkaimmat tunteet ovat empatia, ahneus ja kateus. Kun esimerkiksi hoputetaan kiireellä, ihmisen luontainen empatiakyky pyrkii auttamaan toista hädässä. Ahneus ja kateus ovat tehokkaita siksi, koska ne ovat voimakkaita tunteita. Uhrille voidaan lähettää huijausviesti esimerkiksi isosta lottovoitosta tai että hänet on poistettu yrityksen vapaa-ajan Facebook-ryhmästä. (Vatanen 2017.)

Tietojenkalastelu perustuu kaikkeen sähköisessä viestinnässä perustuvaan tietojen urkintaan, mutta tässä opinnäytetyössä keskitytään sähköpostissa ja verkkosivuilla tapahtuvaan tietojen urkintaan.

Suomessa tietojenkalastelu nousi suuresti esiin vuosina 2018 ja 2019 kun Traficomin alainen Kyberturvallisuuskeskus julkaisi varoituksen Office 365 -sähköpostin tietojenkalastelusta ja tietomurroista. Varoitus julkaistiin ensimmäisen kerran 11.06.2018 ja Kyberturvallisuuskeskus on sen jälkeen nostanut kriittisyyttään keltaisesta punaiseksi ja takaisin sekä julkaissut erilaisia ohjeita organisaatioille tietojenkalastelun ehkäisemiseksi. Kyberturvallisuuskeskus poisti varoituksen 16.9.2019 (Office 365 -sähköpostin tietojenkalastelu ja tietomurrot erittäin yleisiä – havaitse, suojaudu, tiedota! 2019).

2.2 Domain

Jotta tietojenkalastelua voidaan toteuttaa sähköpostitse ja verkkosivulla, sitä varten tarvitaan sähköposti- sekä palvelintilan lisäksi domain-nimi. Domain-muunnokset ovat yleinen ja tehokas tapa huijata uhria ja se on yksi yleinen tapa toteuttaa tietojenkalastelua. Domain-muunnoksessa jostakin jo käytössä olevasta, esimerkiksi uhrin organisaation domain-nimestä tehdään muunnos, joka nopealla silmäyksellä muistuttaa alkuperäistä. Pahantekijä voi esimerkiksi varata @lsoweli.fi-nimisen domain-nimen, joka muistuttaa @isoweli.fi domain-nimeä, joka on aito Isoweli Oy:n omistama domain. Kyseisessä muunnoksessa ensimmäinen i-kirjain on muutettu pieneksi L-kirjaimeksi.

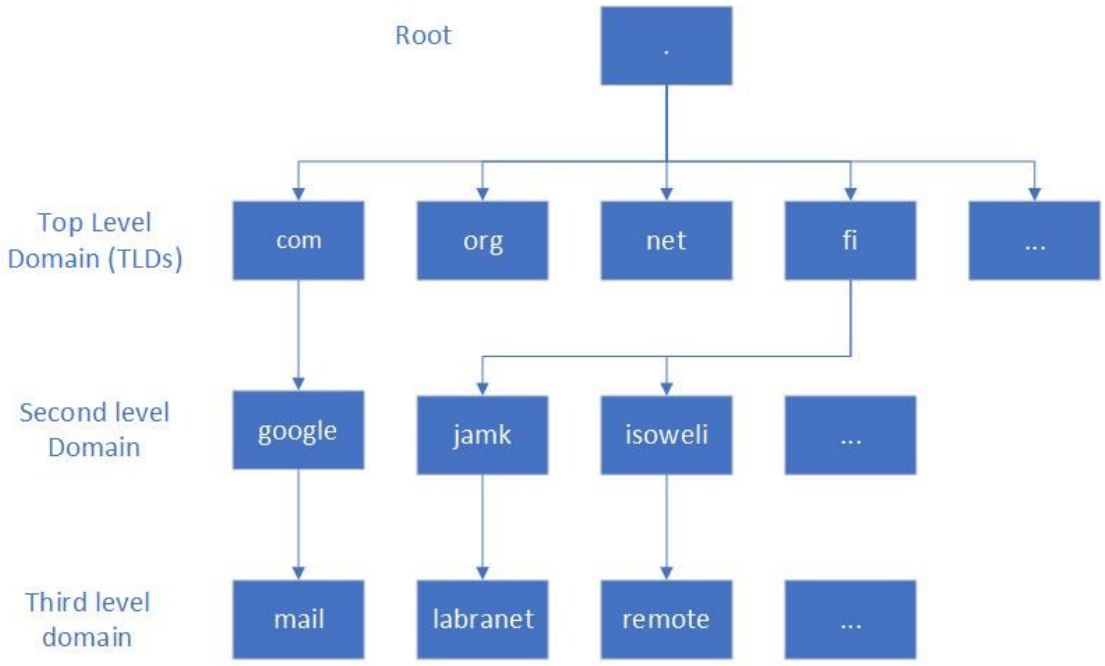
Domain-nimen varaaminen on nykypäivänä hyvin automatisoitua ja domain-nimien väärennöksien valvonta jää pitkälti organisaatioiden omille harteille. Kuitenkin ICANN (Internet Corporation for Assigned Names and Numbers), joka ylläpitää maailman DNS-hierarkiaa, valvoo suurimpien domain-nimien varaamista. Esimerkiksi vuoden 2020 Covid-19 pandemian vuoksi ICANN vahtii ketkä varaavat siihen liittyviä domain-nimiä ja reagoi, mikäli ne vaikuttavat epäilyttäviltä (ICANN Org’s Multifaceted Response to DNS Abuse 2020).

Domain-muunnoksia on käytetty tietojenkalastelussa monia vuosia, ja niitä ovat käyttäneet niin pienemmät tekijät kuin valtiolliset toimijat. Esimerkiksi Venäläiseksi valtiolliseksi toimijaksi epäilty ryhmittymä ”APT-28” on varannut seuraavanlaisia domain-

nimiä omien toimintojensa hyväksi (APT28: A WINDOW INTO RUSSIA'S CYBER ESPIONAGE OPERATIONS? 2014.):

- kavkazcentr[.]info – kavkazcenter.com
- rnil[.]am – mil.am
- login-osce[.].org – osce.org

Domain-nimen muunnelma ei myöskään ole ainoa tapa huijata loppukäyttäjää. Toinen tehokas tapa on hyväksikäyttää domain-hierarkiaa (ks. kuvio 1).



Kuvio 1. Domain-hierarkia

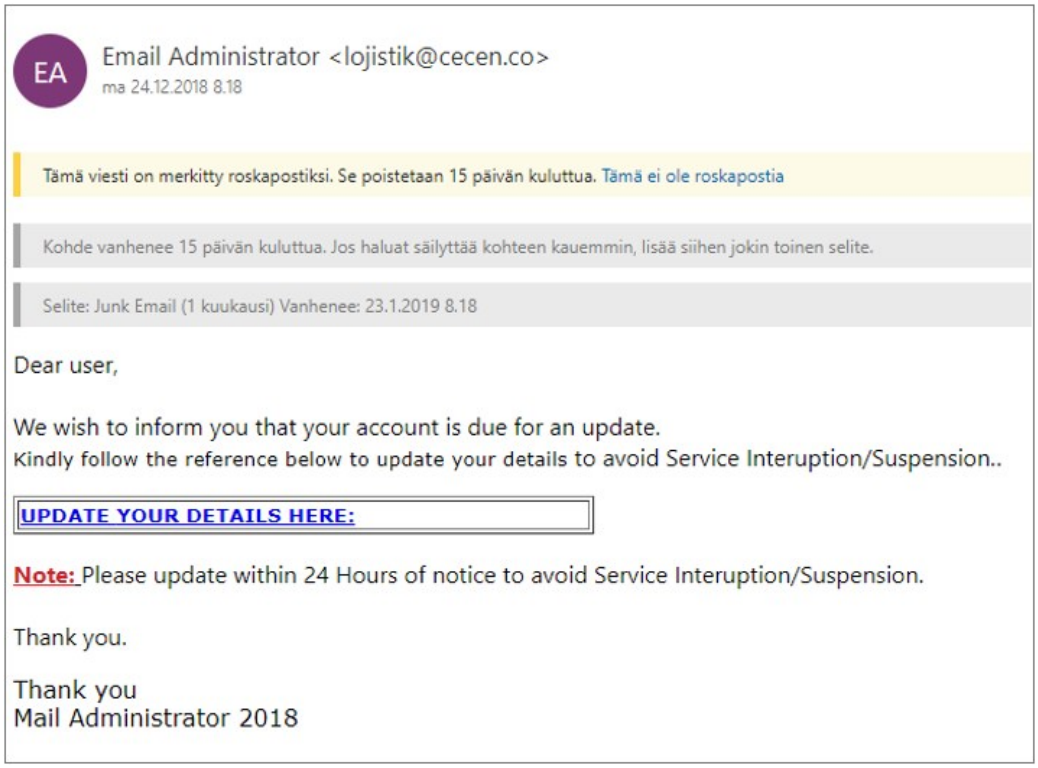
Domain-hierarkia on puumainen järjestelmä, johon sisältyvät Top level domainit, Second level domainit ja siitä eteenpäin numerojärjestyksessä, montako alidomainia on käytössä. Domainit erotetaan toisistaan pisteillä (Aitchison 2011, luku 1).

Normaali käyttäjä yleensä osaa järjestellä osoitteen Top level domainin ja Second level domainin mukaan, mutta siitä pidemmälle mentäessä tarvitaan enemmän syvempää ymmärrystä domain-hierarkiasta. Esimerkiksi ”student.labranet.jamk.fi” saattaa olla ensikertalaiselle hankala sisäistää.

Täten myös domain-hierarkiaa voidaan tehokkaasti käyttää hyödyksi varatessa domain-nimiä. Esimerkiksi voisi varata domainin ”labranet.fi” ja täten tehdä oman kalastelusivuston nimellä ”student.jamk.labranet.fi” jos haluttaisiin ohjata Jyväskylän ammattikorkeakoulun käyttäjiä omalle sivustolle.

2.3 Sähköposti

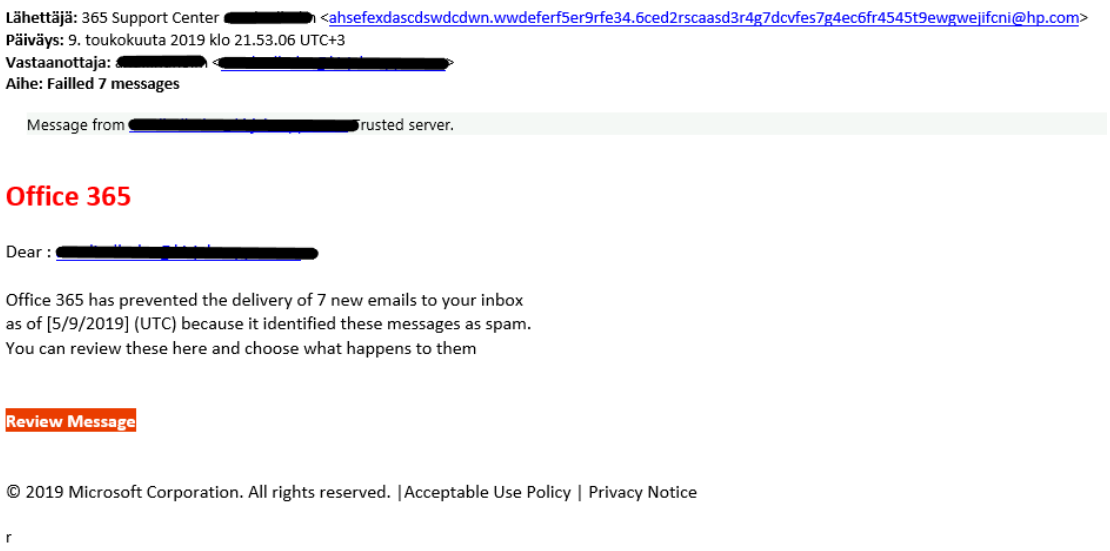
Tietojenkalastelukampanjassa pahantekijä yleensä aloittaa kampanjansa sillä, että hän luo luotettavan oloisen sähköpostin, jolla yrittää saada uhrinsa luovuttamaan tietoja (Ks. kuvio 2).



Kuvio 2. Tietojenkalasteluviesti

Tyypillisessä tietojenkalasteliviestissä lähettäjäksi asetetaan yleensä jokin luotettava taho. Kuten kuviosta 2 huomataan, viestin lähettäjäksi on nimetty ”Email Administrator”. Viestin todellisen lähettäjän kuitenkin huomaa sen sähköpostiosoitteesta, joka on ”lojistik@cecen.co”, joka ei tämän tapauksen kontekstissa mitenkään liittynyt kyseiseen organisaatioon.

Viestissä pyritään saamaan vastaanottaja avaamaan linkin, jonka perässä on tietojenkalastelusivu. Viestissä yleensä viitataan palvelun käyttöön, kuten esimerkiksi kuviossa 3 käyttäjälle kerrotaan, että hän on saanut useita viestejä, jotka pääsevät katsomaan viestin linkistä. Myös usein viesteissä pyritään painostaa käyttäjää kiireellä, kuten kuvion 2 esimerkissä, jossa sähköposti uhataan sulkea, jos käyttäjä ei vahvista tiliään 24 tunnin sisään.



Lähettäjä: 365 Support Center [redacted] <ahsefexdascdswdcdwn.wwdeferf5er9rfe34.6ced2rscaasd3r4g7dcvfes7g4ec6fr4545t9ewgwejifcni@hp.com>
Päiväys: 9. toukokuuta 2019 klo 21.53.06 UTC+3
Vastaanottaja: [redacted]
Aihe: Failled 7 messages

Message from [redacted] rusted server.

Office 365

Dear : [redacted]

Office 365 has prevented the delivery of 7 new emails to your inbox
as of [5/9/2019] (UTC) because it identified these messages as spam.
You can review these here and choose what happens to them

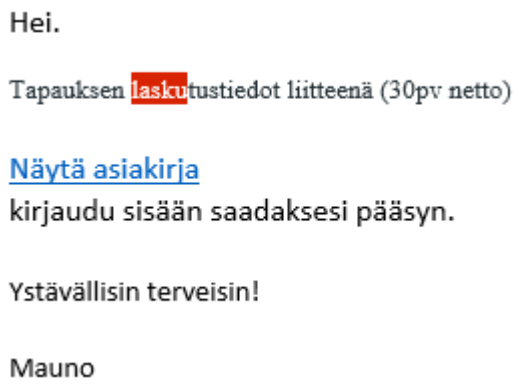
Review Message

© 2019 Microsoft Corporation. All rights reserved. |Acceptable Use Policy | Privacy Notice

r

Kuvio 3. Tietojenkalasteluviesti

Yleinen tapa tunnistaa tietojenkalasteluviesti on myös yleensä kehno kieliasu etenkin, jos ne lähetetään esimerkiksi suomenkielisenä. Myös englanniksi lähetetyt tietojenkalasteluviestit sisältävät usein paljon kieliopillisia virheitä (ks. kuvio 4).



Hei.

Tapauksen laskutustiedot liitteenä (30pv netto)

Näytä asiakirja
kirjaudu sisään saadaksesi pääsyn.

Ystävällisin terveisin!

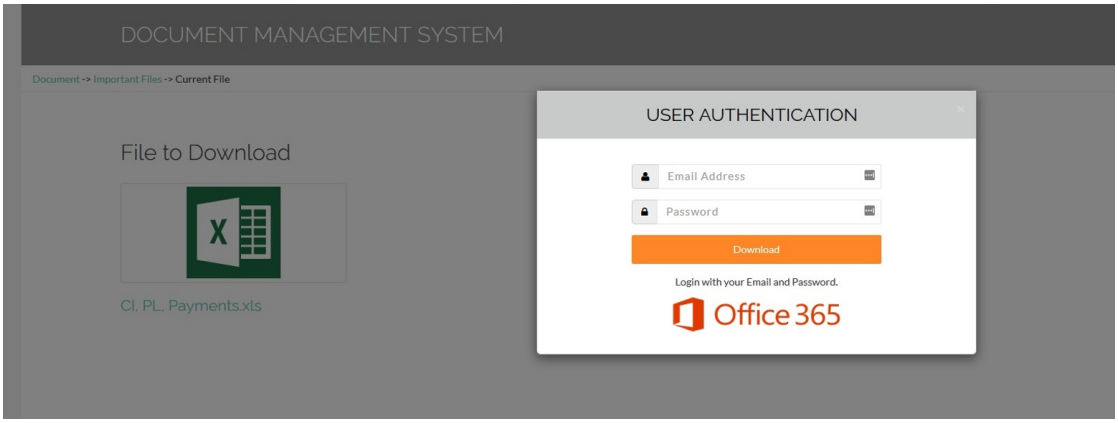
Mauno

Kuvio 4. Tietojenkalasteluviesti suomeksi

2.4 Verkkosivu

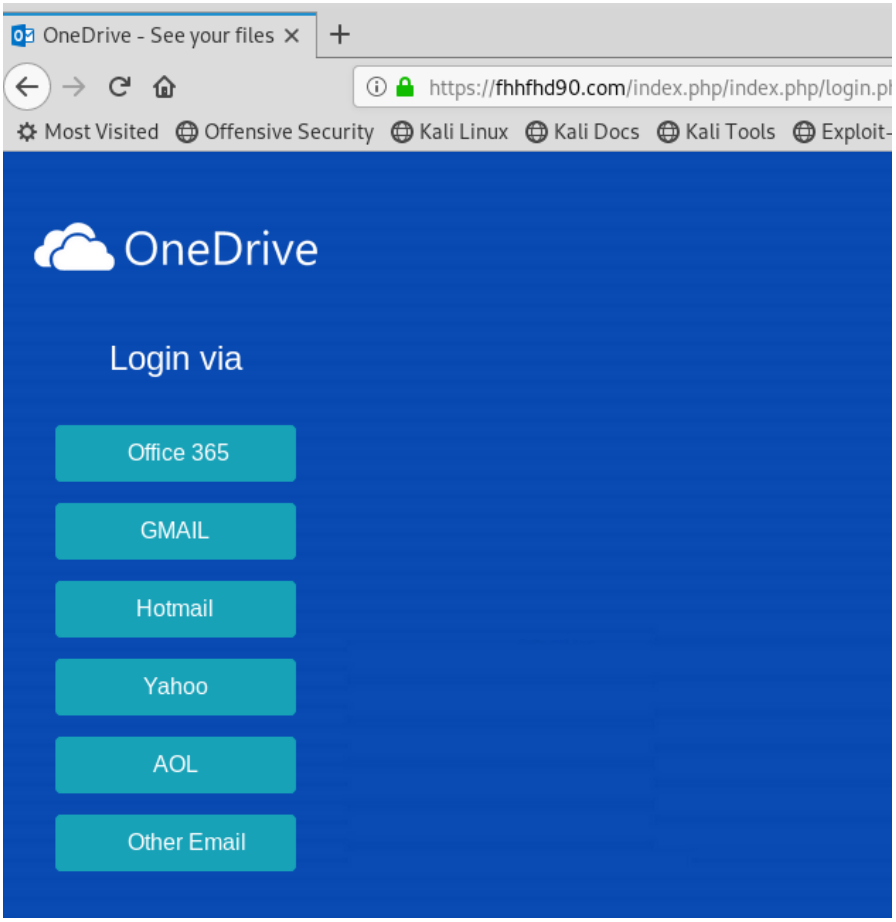
Pahantekijät luovat tietojenkalastelusivuston yleensä imitoimaan oikeaa kirjautumissivua. Esimerkiksi Kali Linux jakelupaketin SEToolkit- sovelluksessa sovellus itse rakentaa sille annetusta sivustosta oman kopion, josta se kerää tunnukset. Sivustojen rakenteessa on kuitenkin huomattavia eroja niin laadullisesti kuin toimivuudeltaan.

Monessa tietojenkalastelusivussa pyydetään lataamaan jokin tiedosto, useinmiten lasku. Toistaiseksi aidon näköisiä esimerkiksi OneDrive- tai SharePoint-palvelun kopioita ei ole tullut vastaan, mutta osa saattaa olla hyvinkin siististi toteutettuja (ks. kuvio 5).



Kuvio 5. Siisti tietojenkalastelusivusto

Usein kuitenkin vastaan tulee nopeasti ja hyvin karulla ulkoasulla toteutettuja kirjautumissivustoja. Näissä sivustoissa usein on yhteistä, että siellä käyttäjä voi ”valita”, millä tilillä hän haluaa liitetiedoston ladata. Usein tietojenkalastelusivustot perustetaan omille domain-nimilleen, jotka voivat olla myös nopeasti tekaistuja (ks. kuvio 6).



Kuvio 6. Huolimattomasti tehty tietojenkalastelusivusto

Monissa tapauksissa myös tietojenkalastelusivustoa voidaan ylläpitää esimerkiksi jonkun toisen organisaation murretun Wordpress-alustan päällä. Myös välillä vastaan tulee tietojenkalastelusivustoja, jotka on kytketty julkiverkkoon pelkällä IP-osoitteella.

Tietojenkalastelusivustojen toiminallisuudet myös eroavat toisistaan. Usein tietojenkalastelusivusto antaa herjan väärästä salasanasta muutaman kerran ja tämän jälkeen menee läpi. Tällä taktiikalla ilmeisesti pyritään saamaan mahdollisimman paljon salasanoja käyttäjätunnuksesta. Yleensä tietojenkalastelusivustot myös ohjaavat käyttäjän Microsoftin sivuille tunnuksien syötön jälkeen.

2.5 Haitalliset liitetiedostot

Verkkosivutyylisen tietojenkalastelun rinnalla toinen yleinen tapa on käyttää haitallisia liitetiedostoja. Liitetiedostot yleensä naamioidaan joksikin harmittomaksi, kuten Word-dokumentiksi, jonka taustalla on VisualBasic-makroja, jotka Wordin avatessa syöttävät uhrin koneeseen haitallista koodia (Spam Is Still the Choice of Online Criminals, 40 Years Later 2018).

Haittaohjelmat voivat sisältää mitä tahansa, riippuen hyökkääjän motiivista. Rahan takia motivoitunut krakkeri voi esimerkiksi ujuttaa kiristyshaittaohjelman uhrin työasemalle ja verkkoon liitteen mukana. Valtiollinen toimija voi taas esimerkiksi asentaa vakoiluohjelmia kohteen verkkoon.

Esimerkiksi APT28-ryhmittymä on käyttänyt usein Word-dokumentteja asentaakseen heidän GAMEFISH-takaportin uhrin verkkoon. GAMEFISH-takaportti mahdollistaa ryhmittymän pääsyn järjestelmän sisään ja hyväksikäyttämään ympäristöä haluamallaan tavalla (APT28: AT THE CENTER OF THE STORM 2017).

2.6 Uhrin hyväksikäyttö

Kun pahantekijä on saanut esimerkiksi sähköpostin tunnukset käyttöönsä hän alkaa monitoroimaan käyttäjän sähköpostiliikennettä. Noin puolissa omakohtaisissa tapauksissa, joita olen tutkinut, pahantekijät ovat tehneet sähköpostin uudelleenohjauksia eteenpäin omiin osoitteisiinsa.

Se mitä pahantekijät tekevät tunnuksilla, riippuu heidän motiivista. Esimerkiksi jos pahantekijät pyrkivät taloudelliseen hyötyyn, he yleensä pyrkivät etsimään sähköpostilaatikosta laskutusliikennettä. Kun he huomaavat esimerkiksi huomattavan laskun, he voivat tehdä sähköpostisäännön, jossa laskun maksavan osapuolen viestit menevät automaattisesti roskakoriin ja heidän uudella ja mahdollisesti väärennetyllä domain-nimellä lähetetyt viestit jatkavat viestin keskustelua. Pahantekijät pyrkivät

muuttamaan yleensä maksun saajan tilinumeron heidän omaksi vedoten verotukseen tai pankin vaihtoon (Suojautuminen Microsoft Office 365 -tunnusten kalastelulta ja tietomurroilta 2019).

Mikäli pahantekijä ei löydä mitään mielenkiintoista sisältöä käyttäjän sähköpostista, hän yleensä lähettää tämän sähköpostista lisää tietojenkalastelupostеja kaikille yhteystiedoille, joita sähköpostista löytyy. Tämä metodi on vaarallinen, koska silloin uusi uhri saa viestin tutun lähettäjän sähköpostista.

Mikäli kyseessä on esimerkiksi valtiollinen toimija, heitä voi kiinnostaa enemmän tiedonkeruu uhrin ympäristöstä. Keskimääräinen aika kuinka kauan pahantekijä säilyttää jalansijan verkossa huomaamatta on maailmanlaajuisesti 56 vuorokautta (M-TRENDS 2020).

3 Toteutuksen suunnittelu

3.1 Ympäristö

Opinnäytetyön toteutukseen toimiksiantaja on antanut mahdollisuuden käyttää yrityksen vanhaa VMware ESXi- virtualisointialustaa omalla erillisellä verkolla. Toimeksiantaja on antanut oikeudet virtualisointiympäristön hallintaan ja täten mahdollistanut laitekannan ja testipalvelimien itsenäisen hallinnan.

Virtuaaliympäristöön on luotu oma VLAN-verkko, josta on ollut pääsy ulkoverkkoon. Mahdollisuus ulkoverkolle on luotu siksi, jotta tarvittavat ohjelmistot voidaan ladata palvelimelle.

3.2 Web- sekä tietokantapalvelin

Alkuperäinen suunnitelma luoda alusta oli käyttää yhteistyökumppanin tarjoamaa webhotelli-palvelua, johon on helppo vain kopioida webpalvelimen tiedostot ja pystyttää tietokantapalvelin. Opinnäytetyön edetessä päädyttiin kuitenkin ratkaisuun, jossa oman palvelimen ylläpitäminen omassa laitesalissa tulee kustannustehokkaammaksi ja Docker-palvelulla palvelut voidaan pystyttää myös nopealla aikataululla ja kustannustehokkaasti.

Opinnäytetyötä varten on päädytty asentamaan CentOS 8-palvelin, koska sen käytöstä on ollut kokemusta jo ennen opinnäytetyötä. Palvelut on ensin rakennettu lokaalisti samalla kyseisellä palvelimella ja näiden toimivuuden toteamisen jälkeen lähdetty rakentamaan toimivaa ratkaisua Dockerin avulla.

3.2.1 Verkkosivu

Verkkosivun ulkoasuun oli alussa monta eri vaihtoehtoa. Yksi vaihtoehto, jota harkittiin ja testattiin, oli asentaa Wordpress, koska sen eri lisäosilla olisi ollut helppo rakentaa verkkosivu ja muokata sitä tarpeen vaatiessa. Kuitenkin pikaisen testauksen jälkeen todettiin, että Wordpress ei tuo lisäarvoa toteutukselle ja aukaisee vain enemmän hyökkäyspinta-alaa mahdollisille verkkohyökkäyksille.

Wordpress-testauksen jälkeen päädyttiin lähteä rakentamaan verkkosivua alusta asti itse. Koska verkkosivu ei ole monimutkainen eikä sisällä montaa sivua, kokonaisuus on helppo ja yksinkertainen toteuttaa.

3.2.2 Tietokanta ja kerättävä data

Tietokanta ja kerättävä data on suunniteltu miettien asiaa työn tilaajan kannalta. Normaalissa tietojenkalastelussa kerätään käyttäjätunnus sekä salasana, jotta niiden avulla päästään väärinkäyttämään tunnuksia.

Kuitenkin päädyttiin ratkaisuun, jossa kerätään vain käyttäjätunnus selkokielisenä sekä salasana salatussa muodossa. Vaihtoehto olisi ollut myös olla keräämättä salasanaa, mutta siitä, että saadaan jotain dataa salasanakenttään samalle riville kuin käyttäjätunnus, voidaan päätellä, että käyttäjä on syöttänyt siihen oikean salasanansa.

Salasana on myös henkilökohtainen asia, jota ei pitäisi luovuttaa kellekään ja usein se voi olla sellainen, jota ei haluaisi kertoa, esimerkiksi kirosana tai jokin muu alatyylinen. Lisäksi jos kerättäisiin selkokieliset salasanat, ne säilyisivät tietokannassa ja olisivat täten tietoturvariski. Näiden syiden takia on päädytty keräämään salasanat salatussa muodossa.

Tietokanta kerää myös tiedon minä päivänä ja kellonaikana syötetty data saapuu tietokantaan. Täten voidaan analysoida, kuinka moni on esimerkiksi heti viestin saatuaan käynyt tietojenkalastelusivustolla ja kuinka moni vasta päiviä myöhemmin.

Muita tietoja, joita olisi voinut kerätä olisivat olleet vierailijoiden lähde IP-osoite, sivustolla vierailtu aika sekä moni muu tieto, jota normaalit verkkosivut keräävät. Kuitenkaan nämä tiedot eivät tuo esiteltävään dataan mitään lisäarvoa, joten niistä päätettiin luopua.

3.3 Sähköposti

Microsoft 365 -ympäristön käyttö on toimeksiantajalle tuttua ja sen vuoksi se on valittu alustaksi sähköpostin käyttöä varten. Myös sähköpostin asettaminen omalla domain-nimellä on Microsoftin palveluissa varsin helppoa verrattuna moniin muihin palveluihin (Change your email address to use your custom domain 2020).

Sähköpostin käyttöön on saatu toimeksiantajalta lupa ostaa ALSO Cloud Markkinapaikan kautta käyttöön oma Microsoft 365-tenant sähköpostilisenssillä. Sähköpostin käyttö tämän kautta on luonnollisin vaihtoehto toimeksiantajalle, koska toimeksiantajan omat sekä asiakkaiden lisenssit hoidetaan tämän avulla.

3.4 Domain sekä https

Domain-nimien ostoon on pyydetty kirjallinen lupa Online Solutions Oy:ltä. Kirjallinen pyyntö on tehty, koska kyseessä on normaalista poikkeavaa liiketoimintaa ja voi pahimmassa tapauksessa aiheuttaa tiedustelupyyntöjä domain-nimen myyjää kohtaan. Domain-nimen tietoja päästään hallitsemaan Online Solutionsin tarjoamalla alustalla.

Verkkosivuston https-salaus toteutetaan EFF-järjestön (Electric Frontier Foundation) tarjoamalla Certbot-sovelluksella. Certbot on helppokäyttöinen sovellus, joka valmiiksi konfiguroi verkkopalvelimelle https-sertifikaatin ja konfiguroi verkkosivustolle menevän liikenteen suoraan salatulle sivustolle siten, että selkokieliselle http-sivulle ei enää päästä (Certbot n.d).

3.5 Tuloksien visualisointi ja esittäminen

Tuloksien visualisointia suunniteltiin aluksi tehtävän joko Excelillä, Ajax-tekniikalla tai Python 3 -ohjelmointikielen Pandas-kirjastolla. Myös datan visualisoinnissa painotuksena oli automatisointi, jonka vuoksi Excel poistettiin vaihtoehdoista nopeasti.

Ajax-tekniikalla datan visualisointi onnistuu helposti suoraan tietokannasta visualisoitavaksi esimerkiksi verkkosivulle. Kuitenkin dataa ei ollut tarkoitus visualisoida reaaliajassa sekä itselläni ei ollut juurikaan kokemusta Ajax-tekniikan hyödyntämisestä, joten poistin sen vaihtoehdoista myös.

Python 3 -ohjelmointikielen Pandas-kirjastosta on ollut jo ennestään kokemusta, joten datan visualisointi on päätetty toteuttaa sillä. Python 3 -ohjelmointikielellä voidaan tehdä suoraan ajettava skripti, joka hakee tiedon tietokannasta ja valmistaa siitä halutut diagrammit ilman, että käyttäjän tarvitsee niitä erikseen muokata (Visualization 2014).

Tuloksissa tulee myös ottaa huomioon kohteena olevien henkilöiden reaktiot ja toimet tietojenkalasteluviesteille. Esimerkiksi yhteydenotot helpdeskiin tai yrityksen

omalle ICT-ylläpitäjälle tulee dokumentoida ja ottaa huomioon tuloksien esittämisessä.

Tuloksia tuottaessa tulee ottaa huomioon testauksen eettisyys. Testauksen tarkoituksena ei ole nolata ketään käyttäjää sillä, että hän on mennyt tietojenkalastelusivustolle ja syöttänyt tunnuksensa sinne. Täten tuloksissa ei yksilöidä ketään henkilöä, vaan testattava kohdeorganisaatio esitetään anonyymeina käyttäjinä.

Tulokset esitellään työn tilaajalle riippuen tilauksesta PowerPoint-esityksenä, johon on liitetty automaattisesti luodut diagrammit sekä muut dokumentoidut tiedot. Tulokset ja niiden esittely voidaan myös yhdistää loppukäyttäjien koulutukseen.

4 Toteutus

4.1 Luvat

Kuten luvussa 3.4 on todettu, domain-nimien ostamiseen on kysytty kirjallinen lupa Online Solutions Oy:ltä. Toimeksiantajalla ja Online Solutions Oy:llä on pitkä historia yhteistyökumppaneina, joten domain-nimien osto heidän kautta on luonnollinen vaihtoehto.

Olen myös ollut yhteydessä Kyberturvallisuuskeskukseen aiheesta. Olen heiltä kysynyt, miten he reagoivat tapauksiin, jos joku ilmiantaa sivustoni tietojenkalastelusivustona ja pitävätkö he esimerkiksi listaa domain-nimistä, joita käytetään tämänkaltaiseen testaukseen. Kyberturvallisuuskeskukselta vastattiin, että he eivät pidä erillistä listaa testaavista organisaatioista tai domain-nimistä. Kun Kyberturvallisuuskeskus aloittaa ilmoituksen tutkinnan he tarkastavat esimerkiksi domain-nimen omistajuuden sekä IP-osoitteet ja niiden perusteella päättävät jatkotoimenpiteistä.

Kyberturvallisuuskeskuksen viestin perusteella ei ole sitä vaaraa, että tietojenkalastelutestauksesta joutuisi ongelmiin heidän puolesta. Domain-nimen varauksessa käytetään yrityksen oikeita tietoja ja mahdollisissa jatkotoimenpiteissä yhteydenotot osataan ohjata oikeaan suuntaan.

4.2 Palvelimen luonti

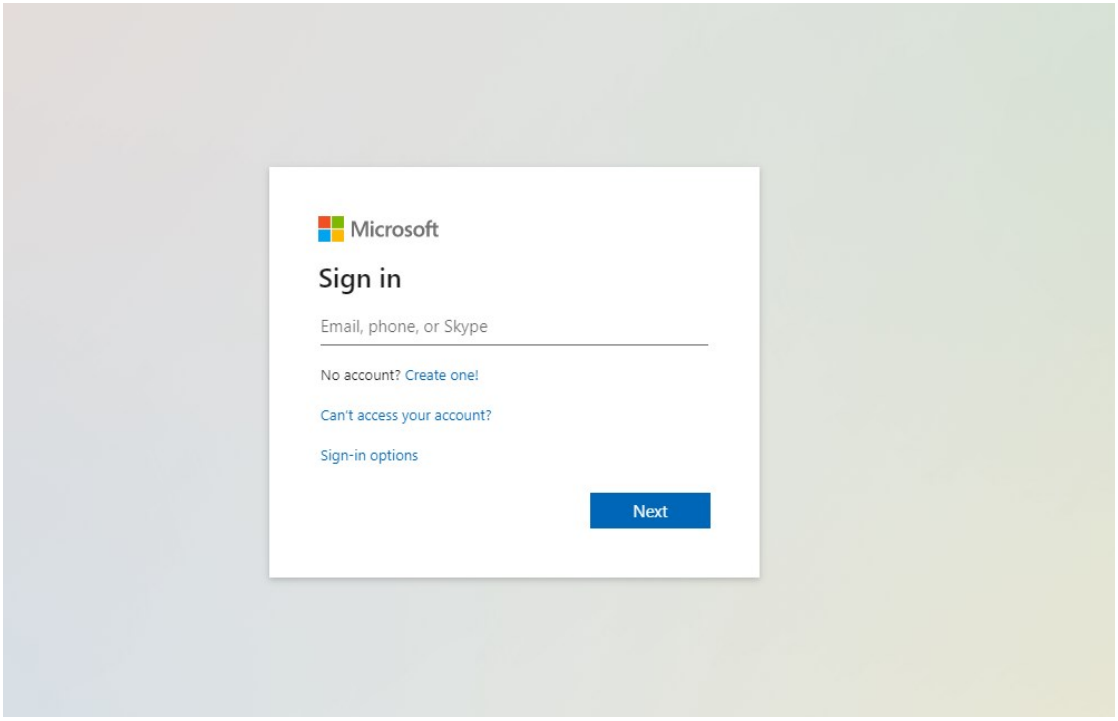
4.2.1 Yleistä

Työ on aloitettu luomalla CentOS 8- palvelin toimeksiantajan tarjoamaan testiympäristöön, jossa on alettu rakentamaan kirjautumissivustoa, tietokantaa sekä PHP-koodia, joka syöttää syötetyt tiedot tietokantaan. Testiympäristön topologia on kuvattu liitteessä 16.

Palvelimelle on asennettu testausta varten viimeisin Apache http Server -palvelinohjelmisto sekä viimeisin MySQL-relaatiotietokantaohjelmisto. Näiden lisäksi palvelimelle on myös asennettu PHP-ohjelmointikielen versio 7.3 jotta sillä voidaan ajaa PHP-koodia.

4.2.2 Front-end

Tietojenkalastelusivun suunnittelu on aloitettu imitoimalla aitoa Microsoft 365 -palvelun etusivua. Microsoftin oikea kirjautumissivusto on tyyliltään yksinkertainen ja sisältää pieniä animointeja. Sivuston kirjautumislomakkeelle syötetään aluksi käyttäjätunnus, jonka jälkeen päästään seuraavalle sivulle, jossa päästään asettamaan salasana. Kirjautumissivun ulkoasu saattaa muuttua, kun salasanaa syötetään, jos organisaatiolla on käytössä oma kustomoitu kirjautumissivu (ks. kuvio 7).



Kuvio 7. Microsoftin kirjautumissivusto

Koska kustomoitujen kirjautumissivustojen takia en voi saada kirjautumissivustoa täysin identtiseksi, olen päättänyt imitoida mahdollisimman paljon normaalia ulkoasua animaatioiden kanssa. Microsoftin omat HTML- ja CSS-määrittelyt ovat todella laajoja, joten niiden tutkimisesta on myös ollut apua toteutuksessa.

Molempien näkyvien sivujen HTML-dokumenttien alkuun on kirjoitettu info, joka aukeaa heti näkyville, kun sivulla avataan sovelluskehittäjän asetukset. Tämä tieto on lisätty siksi, että jos sivustolle eksyy valveutuneempi käyttäjä, hän ei lähtisi raportoimaan sivustoa viranomaisille (ks. kuvio 8).

```
<!DOCTYPE html>
<!--
  THIS IS AUTHORIZED PHISHINGTEST BY ISOWELI OY.
  PLEASE DONT REPORT THIS SITE OR DOMAIN.
  FOR MORE INFORMATION, CONTACT pekka.sivusuo@isoweli.fi or tuki@isoweli.fi
-->
```

Kuvio 8. index.html- sekä verify.php-tiedostojen ylimmät rivit

Header-osiossa olen myös luonut Javascriptin jQuery-kirjastolla kirjautumissivuston animoinnin, jossa sisäänkirjautumislaatikko tulee ruudun oikealta reunalta keskelle, kun sivu ladataan (ks. kuvio 9).

```
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script>
      $(function(){
        $(".loginboxcontent").animate({left: '0px'});
      });
</script>
```

Kuvio 9. jQuery-skripti animointia varten

Sisäänkirjautumislaatikon sisältö toimii HTML:n form-elementeillä. Ensimmäisellä sivulla näkyy pelkästään käyttäjätunnuksen syöttämisen vaihtoehto, sillä salasana kirjoitetaan vasta toisella sivulla. Formissa input-parametrilla otetaan käyttäjätunnus talteen ja se syötetään seuraavalle sivulle staattiseksi kuten Microsoftin omalla kirjautumissivulla.

Jotta vältytään mahdollisilta turhilta syötteiltä, käyttäjätunnuksille varattu kenttä on pakotettu kenttä, eli seuraavalle sivulle ei pääse ilman, että käyttäjätunnuksen kenttään asetetaan dataa. Tämä on toteutettu input-tunnisteen required-attribuutilla. Myös käyttäjätunnuksen kenttä on tyypiltään ”email”, jotta vältytään syötteiltä, jotka eivät ole sähköpostimuotoa tai pelkkiä välilyöntejä.

Kirjautumissivulle on myös asetettu hyperlinkit ”Create account” sekä ”Can’t access your account”, jotka löytyvät myös Microsoftin omilta sivuilta. Tässä tapauksessa hyperlinkit ohjautuvat vain Microsoftin etusivulle. (ks. kuvio 10)

```
  <h1>Sign in</h1>
  <!--POST-muuttujalla syotetaan kayttajatunnus seuraavalle sivulle -->
  <form action="/verify.php" method="post">
    <!-- Syotetaan kayttajatunnus -->
    <label for="username">Username</label>
    <input type="email" name="username2" id="username2" placeholder="Email, phone, or Skype" required>
    <a href="https://portal.lsoweli.fi/verify.php">
      <input type="submit" value="Next">
    </a>
<a href="https://www.microsoft.com/">Create account <br><br></a>
    <a href="https://www.microsoft.com/">Can't access your account?</a>
  </form>
```

Kuvio 10. Sisäänkirjautumislaatikon form-parametrit

Kun käyttäjätunnus on syötetty, päästään toiselle sivulle, jossa kysytään käyttäjän salasanaa. Toiselle sivustolle on tuotu ensimmäisellä sivulla syötetty käyttäjätunnus PHP-skriptillä ja tästä formin osasta on muokkaaminen estetty input-tunnisteen attribuutilla disabled. Myös PHP-skriptin sisään on täytynyt luoda echo-attribuutti, joka syöttää käyttäjätunnuksen config.php-tiedoston tietoihin. Rivi esiintyy kahteen kertaan, koska testausvaiheessa huomasin, että HTML-attribuutti disabled estää tiedon lähetyksen.

Myös salasanan syöttö on pakotettu input-tunnisteen attribuutilla required, jotta tyhjiä syötteitä ei tulisi tietokantaan. Myös jos käyttäjä vahingossa painaa ”Log In” painiketta tyhjällä salasanalla, hän voi tajuta kyseessä olevan epäilyttävä sivu. Toiselle sivulle on myös lisätty hyperlinkki ”Forgot Password?”, joka ohjaa myös Microsoftin etusivulle. (ks. kuvio 11)

```
  <!-- formi kayttaa config.php-tiedostoa ja keraa siihen parametrit-->
  <form action="config.php" method="post">
    <label for="username">Username</label>
<!-- scripti estaa sahkopostikentan muuttamisen ja tuo sen edelliselta sivulta nakyville-->
<?php
$email_address = $_POST['username2'];
echo '<input type="email" name="username" id="username" value="' . $email_address . '" disabled />';
    echo '<input type="hidden" name="username" id="username" value="' . $email_address . '" />';
?>
    <label for="password">Password</label>
    <input type="password" placeholder="Enter Password" name="password" required>
    <input type="submit" value="Log In">
<a href="https://www.microsoft.com/">Forgot password?</a>
  </form>
```

Kuvio 11. verify.php-sivun salasanan syöttö

Sivustojen css-määrittelyt löytyvät liitteistä (ks.liitteet 2 ja 3). Css-määrittelyt ovat hyvin yksinkertaisia ja koskevat pääasiassa kirjautumislaatikkoa, joten näihin määrityksiin ei perehdytä tässä osiossa tarkemmin.

4.2.3 Back-end

Palvelinpuoli (back-end) on toteutettu PHP-koodikielellä sekä MySQL-tietokannalla. PHP-skripti syöttää datan MySQL-kantaan automaattisesti, kun dataa on syötetty verkkosivun kautta.

Yhteyden MySQL-tietokantaan luo verify-php (liite 6). Tiedoston alussa määritellään tietokannan osoite, tunnistautuminen ja mitä tietokantaa käytetään. Sivusto antaa myös virheilmoituksen, mikäli yhteys kantaan ei onnistu (ks. kuvio 12).

```
#maaritellaan mysql-yhteyden parametrit
define('DB_HOST', 'db');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '*DATABASEPASSU*');
define('DB_NAME', 'phishingdb');

#Luodaan mysql-yhteys
$link = mysqli_connect(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_NAME);

#tarkistetaan mysql-yhteys
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
```

Kuvio 12. Tunnistautuminen ja yhteys tietokantaan

Kuviossa 11 tietojen keruu ja syöttö kantaan tapahtuu PHP:n $_POST-taulukkoon tallennettuja username- ja password-arvoja käyttäen. Koska toteutuksessa ei ole tarkoitus kerätä salasanoja, syötetyt arvot salataan PHP:n vakio salauksella, joka käyttää bcrypt-algoritmia luodessaan tiivistefunktiota (password_hash 2020).

Kun tiedot on syötetty tietokantaan, yhteys suljetaan kutsumalla funktiota mysqli_close. Vaikka yhteyttä ei testivaiheessa suljettu, se toimi moitteitta. Kuitenkin

hyvän käytänteen mukaisesti yhteys on hyvä sulkea, kun se on valmis, jotta yhteydelle varattu portti vapautetaan palvelimen päästä (ks. kuvio 13) (mysqli_close 2020).

```
#kerataan sivun tiedot post-parametrille talteen
$username = $_POST['username'];
$password = password_hash($_POST['password'], PASSWORD_DEFAULT);

#valmistellaan tietojen syotto kantaan ja syotetaan
$sql = "INSERT INTO users (username,password) VALUES ('$username','$password')";
mysqli_query($link,$sql);

#suljetaan mysql-yhteys
mysqli_close($link);
```

Kuvio 13. Tietojen keruu ja syöttö kantaan

Kun käyttäjä on syöttänyt tunnuksensa, hänet ohjataan pois sivustolta Microsoftin omalle kirjautumissivustolle. Olen tarkoituksella valinnut uudelleenohjauksen osoitteeksi Microsoftin kirjautumissivun, koska jos käyttäjä on jo selaimessaan kirjautunut Microsoftin tililleen, hän pääsee sen etusivulle luullen, että tietojenkalastelusivusto kirjasi hänet sisään (ks. kuvio 14).

```
#ohjataan kayttaja pois sivulta
header("location: https://portal.office.com/");
?>
```

Kuvio 14. Käyttäjän ohjaus pois sivustolta

Tietokannan rakenne on hyvin yksinkertainen. Tietokantaan on tarvinnut ainoastaan luoda yksi tietokanta ”phishingdb” jonka sisään on luotu taulu ”users”, johon kerätään syötettyjen tietojen lisäksi ID-numero sekä päivämäärä ja kellonaika jokaisesta syötteestä (ks. kuvio 15).

```
mysql> DESCRIBE users;
+------------+--------------+------+-----+-------------------+----------------+
| Field      | Type         | Null | Key | Default           | Extra          |
+------------+--------------+------+-----+-------------------+----------------+
| id         | int(11)      | NO   | PRI | NULL              | auto_increment |
| username   | varchar(50)  | NO   |     | NULL              |                |
| password   | varchar(255) | NO   |     | NULL              |                |
| created_at | datetime     | YES  |     | CURRENT_TIMESTAMP |                |
+------------+--------------+------+-----+-------------------+----------------+
4 rows in set (0.00 sec)
```

Kuvio 15. Tietokannan users-taulun rakenne

Tietokannan suunnittelussa tärkein huomioitava piirre on antaa salasanalle tarpeeksi merkkejä syöttöä varten. PHP:n vakio salausalgoritmi saattaa muuttua PHP:n päivityksen yhteydessä ja täten muuttaa tapaansa generoida tiivistefunktioita (password_hash 2020).

Data tulee kuvion 16 mukaisesti kantaan. Palvelimelle voisi esimerkiksi asentaa phpMyAdmin-sovelluksen, jolla tietoja voisi seurata graafisesti, mutta se ei tuo toteutukseen lisäarvoa ja samalla lisää hyökkäyspinta-alaa.

```
mysql> select * from users;
+----+-------------------+--------------------------------------------------------------+---------------------+
| id | username          | password                                                     | created_at          |
+----+-------------------+--------------------------------------------------------------+---------------------+
|  1 | sampo@testi.fi    | $2y$10$XICX.57bHd5bUR9kihW6euF4t96RiHMO9SYJqvh0y3zxkKF5vUrm6 | 2020-05-11 14:15:50 |
|  2 | pekka@testi.fi    | $2y$10$RVW8pEMd2RXdJe1YLm0xS.jBM847ZT1FbcEDu6ZB2qylySxzXg2vO | 2020-05-11 17:05:32 |
```

Kuvio 16. Datan näkyminen tietokannassa

4.3 Datan visualisointi

Kun verkkosivuston toiminta todettiin toimivaksi, aloitettiin datan visualisoinnin suunnittelu. Jotta palvelimella voidaan ajaa Python-koodia, siihen täytyy asentaa Python 3 kirjastot. Nämä löytyvät valmiiksi CentOS 8 -jakelupaketin Yum-repositoriosta (Tagliaferri 2017).

Kun Python on asennettu, voidaan halutut lisäkirjastot asentaa pip3-komennolla. Kuviossa 17 on listattuna kirjastot, joita skripti tarvitsee ajamiseen. Myös skripti tarvitsee Openpyxl-kirjaston tulostaessaan tulokset xslx-muotoon, mutta sitä ei tarvitse erikseen kutsua skriptissä.

```
#ladataan tarvittavat kirjastot
import pandas as pd
import sqlalchemy
import pymysql
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
```

Kuvio 17. Tarvittavat Pythonin kirjastot

Skripti aloittaa toimintansa ottamalla yhteyden tietokantaan ja lukemalla sen tiedot datakehykseksi. Skriptissä slqalchemy-kirjasto yhdessä pymysql-kirjaston kanssa luo yhteyden tietokantaan ja valitsee oikean tietokannan sen sisältä. Kun yhteys on luotu, valitaan mikä taulu tietokannasta halutaan ottaa käyttöön ja se muutetaan datakehykseksi. Kun tietokannan data on muutettu datakehykseksi, sitä voidaan muokata kuten mitä tahansa muuta datakehystä (ks. kuvio 18).

```
#Luodaan yhteys tietokantaan ja tuodaan tietokannan data dataframeksi
engine = sqlalchemy.create_engine('mysql+pymysql://root:*DATABASEPASSU*@localhost/phishingdb')
df = pd.read_sql_table('users', engine,)
```

Kuvio 18. Yhteys MySQL-tietokantaan

Kun data on saatu muotoiltua halutun näköiseksi graafiksi, se voidaan tallentaa levylle suoraan savefig-funktiolla. Savefig-funktiolle täytyy määrittää polku, jonne kuva halutaan tallentaa, sekä tarvittaessa muita arvoja, kuten tässä DPI-arvo (ks. kuvio 19).

```
#tallennetaan pylvasdiagrammi PNG-muodossa
plt.savefig('/home/pekka/phish/pylvas.png', dpi=400)
```

Kuvio 19. Graafin tallennus levylle

Kun halutut graafit on tulostettu, skripti vielä tulostaa datakehyksen tiedot Excel-muotoon, jos niitä haluaa tarkastella tarkemmin tai luoda Excelin puolella mahdollisesti laajempia graafeja muun datan avulla. Kun skripti on valmis, se tulostaa komentoriville viestin, että ajo on valmis (ks. kuvio 20).

```
#Luodaan uusi dataframe df4 jossa tulostetaan sen sisalto xlsx-muotoon
df4 = pd.read_sql_table('users', engine,)
df4.to_excel('/home/pekka/phish/sqldata.xlsx')

#Kun skripti ajettu, tulostetaan komentoriville siita ilmoitus
print('Ready?')
```

Kuvio 20. Excel-datan tulostus ja loppu

Kun skripti on onnistuneesti ajanut itsensä, se tallentaa valittuun polkuun tiedostot pylvas.png, piirakka.png ja sqldata.xlsx (ks. liite 11). Tämän jälkeen tiedostot voidaan siirtää palvelimelta työasemalle esityksen valmistelua varten esimerkiksi WinSCP-ohjelmalla.

4.4 Dockerin hyödyntäminen automatisoinnissa

4.4.1 Yleistä Dockerista

Docker on palvelu, joka virtualisoi eri sovelluksia käyttöjärjestelmän päällä. Dockerin eri palvelut jaetaan kontteihin, jotka ovat toisistaan eristettyjä prosesseja ja jakavat yhteisen isäntäkoneen kernelin. Verrattuna virtuaalikoneisiin, Docker virtualisoi käyttöjärjestelmän, kun taas virtuaalikone virtualisoi tietokoneen laitteistoa.

Dockerin etuja ovat sen toiminnallisuus sekä turvallisuus. Dockerin kontteja on laajasti saatavilla ja ne toimivat jokaisella alustalla, jolla on Docker asennettu. Dockerin kontit eivät tarvitse virtuaalikoneen lailla omaa hypervisor alustaa ja Dockerin kontit vievät paljon vähemmän tilaa kuin virtuaalikoneiden levykuvat ja käynnistyvät huomattavasti nopeammin kuin virtuaalikoneet. (What is a Container? 2020.)

Docker tuo toteutukseen myös tietoturvaa. Verrattuna ratkaisuun, jossa palvelua pyöritettäisiin suoraan CentOS 8-palvelimen päällä, mahdollinen pahantekijä pääsee koko työasemalle käsiksi. Koska tässä tapauksessa palvelut pyörivät eristetyissä konteissa, niiden läpi on vaikeampi päästä saastuttamaan itse isäntäkonetta (Vase 2015, 19).

Toiminnallisuuden ja turvallisuuden lisäksi Docker mahdollistaa opinnäytetyössä nopean ja automatisoidun tavan pystyttää palvelu. Palvelu käynnistyy oikeilla konfiguraatioilla minuuteissa ja konttien konfiguraatioita on helppo muuttaa ja testata Dockerin avulla.

4.4.2 Toteutus

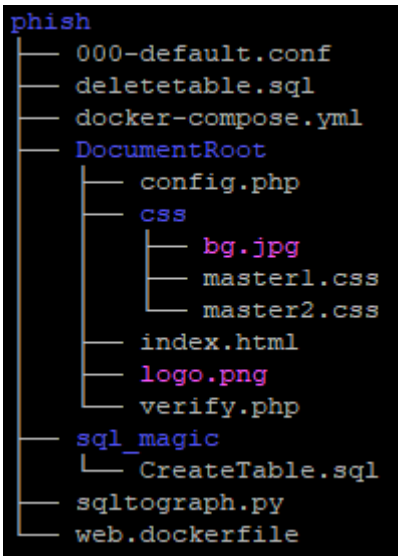
Toteutus on päätetty lähteä tekemään hyödyntäen Dockerin compose-ominaisuutta. Docker-compose -ominaisuudella voidaan luoda yksi YML-tiedosto, joka sisältää konfiguraatiot konteille ja osaa hakea mahdolliset lisäkonfiguraatiot määrittelytiedostoista (dockerfile) (Overview of Docker Compose 2020).

Docker CE:n (Community Edtition) asennusmateriaalit CentOS 8 -palvelimelle löytyvät Dockerin omasta repositoriosta, joka täytyy lisätä DNF-pakettihallinnan hakemistoon. Tämän jälkeen Docker CE voidaan asentaa DNF:n avulla.

Docker-compose täytyy myös asentaa erikseen palvelimelle, jos sitä halutaan käyttää. CentOS 8 -palvelimelle docker-compose löytyy Dockerin omasta Githubista josta se voidaan Curl-komentoa käyttäen asentaa palvelimelle. Ladatulle tiedostolle täytyy antaa lataamisen jälkeen suoritusoikeudet palvelimella. (Kumar 2019.)

Dockerin lisäksi, jotta palvelua voidaan käyttää, palvelimelle tulee asentaa oma MySQL-tietokanta sekä Python3 tarvittavilla kirjastoilla. Nämä asennetaan siksi, että kerätty data voidaan suoraan visualisoida samalla palvelimella.

Docker-composea varten eri tiedostot täytyy jakaa omiin kansiopolkuihin, jotta se osaa toimia oikein. Kuviossa 21 on kuvakaappaus kansiorakenteesta, joka palvelimella on.



Kuvio 21. Dockerin kansiorakenne

Liiteessä 1 on docker-compose.yml-tiedoston sisältö. Tiedostossa ensin määritellään web-kontin asetukset. Dockerfile käyttää web-kontin asennukseen sen omaa määrittelytiedosta, jonka lisäksi konttiin ohjataan portit 80 (http) sekä 443 (https) isäntäkoneelta ja määritellään käsin kontti käyttämään Googlen DNS-palvelinta. Kontti konfiguroidaan käyttämään omaa sisäistä verkkoa ”network1” aliaksella ”web” jotta se osaa kommunikoida tietokantakontin kanssa. Lisäksi kontin /var/www/html-kansioon kopioidaan DocumentRoot-kansiosta verkkosivun sisältö (ks. kuvio 22).

```
#maaritetaan palvelu "web" joka kayttaa web.dockerfilea luomiseen
services:
  web:
    build:
      context: ./
      dockerfile: web.dockerfile
    #maaritellaan portit ja DNS asetukset kontille
    ports:
      - 80:80
      - 443:443
    dns:
      - 8.8.8.8
    #maaritetaan kontti kayttamaan "network1" sisaista verkkoa, web-kontti aliaksella "web"
    networks:
      network1:
        aliases:
          - web
    #kopioi tiedostot lokaalilta hostilta konttiin
    volumes:
      - ./DocumentRoot:/var/www/html
  #luodaan mysql-kontti kayttaen mysql:5.7.25 imagea
```

Kuvio 22. Docker-composen webkontin määrittely

Konfiguraation toisessa osassa määritellään tietokantakontti toimimaan. Tietokantakontille ei tullut tarvetta luoda omaa määrittelytiedostoa, joten sen kaikki määritykset löytyvät tästä tiedostosta.

Tietokantakontti määritellään käyttämään mysql:5.7.25 levykuvaa (image) ja sille aukaistaan portti 3306 (mysql) käyttöön. Kontti määritellään myös toimimaan samassa suljetussa verkossa web-kontin kanssa aliaksella ”db”. Kontin tietokannalle annetaan käyttäjätunnukset ja tietokannan nimi valmiiksi tiedostossa. Kontti määritellään käynnistämään itsensä uudestaan automaattisesti, mikäli se sammuu ja siihen asetetaan general-lokitus päälle mahdollisten vikatilanteiden vuoksi. Lopussa myös kontti määritetään suorittamaan users-taulun luonti tietokantaan kopioimalla sql-tiedosto (ks. liite 10) ”docker-entrypoint-initdb.d”-kansioon. Kaikki tähän kansioon lisätyt suoritettavat tiedostot suoritetaan aakkosjärjestyksessä, kun palvelua pystytetään (ks. kuvio 23) (mysql 2020).

```
mysql:
  image: mysql:5.7.25
  #maaritellaan portit ja DNS asetukset kontille
  ports:
    - 3306:3306
  dns:
    - 8.8.8.8
  #maaritetaan kontti kayttamaan "network1" sisaista verkkoa, mysql-kontti aliaksella "db"
  networks:
    network1:
      aliases:
        - db
  #maaritetaan muuttujat tietokannalle
  environment:
    TZ: "Europe/Helsinki"
    MYSQL_ALLOW_EMPTY_PASSWORD: "no"
    MYSQL_ROOT_PASSWORD: "*DATABASEPASSU*"
    MYSQL_USER: 'root'
    MYSQL_PASSWORD: '*DATABASEPASSU*'
    MYSQL_DATABASE: 'phishingdb'
  #maaritetaan kontti kaynnistymaan jos se sammuu automaattisesti
  restart: always
  #maaritetaan general loki paalle mysql-palvelulle mahdollisia vikatilanteita varten
  command: mysqld --general-log=1 --general-log-file=/var/log/mysql/general-log.log
  #luodaan users-table tietokantaan luonnin yhteydessa. scripti on "sql_magic" kansiossa hostilla
  volumes:
    - ./sql_magic:/docker-entrypoint-initdb.d
```

Kuvio 23. Docker-composen mysql-määritykset

Docker-composen käyttämä web.dockerfile määrittää tarkemmin sille tehtävät parametrit. Web-kontti käyttää php:7.3-apache levykuvaa pohjana. Kontti päivitetään ajon aikana ja siihen asennetaan Nano-tekstieditori mahdollisten vikatilanteiden varalta, tarvittavat PHP-lisäosat sekä Certbot HTTPS-sertifikaatin generoimista varten. Dockerfile myös kopioi muokatun Apachen konfiguraatiotiedoston valmiiksi Apachen konfiguraatioon, jossa määritetään palvelimen domain-nimi verkkosivulle (ks. liite 9). Dockerfilen lopussa avataan myös varmuuden vuoksi portit 80 ja 443 toiminnan varmistamiseksi (ks. kuvio 24).

```
#kaytetaan php:7.3-apache imagea
FROM php:7.3-apache
#asennetaan nano seka tarvittavat PHP-liitannaiset
RUN apt-get update && apt-get install -y \
        nano \
    && docker-php-ext-install pdo_mysql mysqli pdo
#asennetaan certbot valmiiksi palvelimelle
RUN apt-get install -y certbot python-certbot-apache
#siirretaan muokattu apachen konfiguraatiotiedosto palvelimelle hostilta
COPY 000-default.conf /etc/apache2/sites-available/000-default.conf
#maaritellaan avonaiset portit
EXPOSE 80
EXPOSE 443
```

Kuvio 24. web.dockerfile-tiedoston sisältö

Kontit lähtevät käyntiin komennolla ”docker-compose up -d” samassa kansiossa, jossa docker-compose.yml-tiedosto sijaitsee. Parametri -d jättää komentokehotteen käyttöön konttien pystytyksen jälkeen. Kun kontit on ajettu kerran päälle ja tarvitsee esimerkiksi tehdä muutoksia konttien konfiguraatioon, kontit saadaan pysäytettyä ”docker-compose stop” -komennolla. Muutosten jälkeen, kun kontit ajetaan takaisin päälle, ne nousevat ylös nopeammin, koska Dockerin ei tarvitse hakea levykuvaa verkosta (olettaen, että levykuvaa ei vaihdeta konfiguraatiossa) (docker-compose up 2020).

Kun tietojenkalastelukampanja on ohi, tietokannan tiedot voidaan ladata kontista isäntäkoneelle omalla komennolla. Tiedot haetaan käyttäen ”mysqldump”-komentoa, joka suoritetaan kontissa ja tallennetaan lokaalille isäntäkoneelle (ks. kuvio 25).

```
docker exec linuxconfig_mysql_1 /usr/bin/mysqldump -u root -p phishingdb > phishdump.sql
```

Kuvio 25. Tietokannan datan lataaminen isäntäkoneelle

Kun tiedot on saatu talteen isäntäkoneelta, sammutetaan kontit. Tämän jälkeen käynnistetään isäntäkoneen MySQL-tietokanta, joka on rakenteeltaan vastaava kuin

kontissa oleva. Isäntäkoneen tietokantaan tuodaan sisälle kontista ladattu phish-dump.sql. Tämän jälkeen dataa voidaan tarvittaessa vielä käsin siivota, mikäli sivustolle on joku syöttänyt sinne kuulumatonta dataa. Kun data on valmis, ajetaan aiemmin luotu python-skripti, joka luo graafit sekä Excel-tiedoston tuloksista. Tämän jälkeen tietokanta voidaan tyhjentää ajamalla deletetable.sql-komento, joka sisältää komennon ”TRUNCATE TABLE users”, jolla tyhjennetään users-taulun data (ks. kuvio 26).

```
[pekka@phishsrv ~]$ sudo mysql -u root -p phishingdb < deletetable.sql
```

Kuvio 26. users-taulun tyhjennys

Ennen seuraavaa tietojenkalastelukampanjaa, Dockerin välimuisti ja olemassa olevat kontit kannattaa tyhjentää komennolla ”docker system prune -a”. Tällöin kun uusi kampanja alkaa, kontit hakevat viimeisimmät saatavilla olevat päivitykset kontteihin, kun se suorittaa määrittelytiedoston parametrit uudestaan.

4.5 Testaus julkiverkossa

4.5.1 Julkiverkon palvelimen asennus

Tuotantoympäristöön toimeksiantaja on asentanut CentOS 8 -palvelimen TNNET Oy:n hallinnoimaan datakeskukseen Jyväskylän Kanavuoreen. Palvelin on asennettu VMware ESXi -virtualisointialustan päälle ja se on saanut yhden kiinteän IPv4-osoitteen. Palvelimen ja julkiverkon välissä on WatchGuardin palomuuri, josta on aukaistu portit 22 (SSH), 80 sekä 443 palvelimelle (ks. liite 15).

Hallintatunnuksien saamisen jälkeen palvelinta on välittömästi kovennettu seuraavilla toimilla:

- root-käyttäjän salasana generoitu 128-merkkiseksi
- Hallintatunnuksen tunnistautuminen muutettu avainperäiseksi Ed25519-avaimilla käyttäen yksityisen avaimen salasanaa lisäksi

- Tunnistautuminen estetty salasanalla
- root-käyttäjän SSH-tunnistautuminen estetty
- ChallengeReponseAuthentication sekä PAM-tunnistautuminen myös estetty
- fail2ban-sovelluksen asentaminen ja konfigurointi (ks. liite 12)

Palvelimen kovennuksien jälkeen sille on asennettu kaikki komponentit, joita se tarvitsee Dockerin ajamiseen. Palvelut on mainittu luvussa 4.4.2.

4.5.2 Domain-nimen hankinta ja DNS

Domain-nimien osto alun perin oli suunniteltu Online Solutions Oy:n kautta, mutta aikataulusyistä domain-nimi ostettiin joker.com-palvelun kautta. Domain-nimen ostamisen jälkeen aloitettiin DNS-konfiguraatioiden määrittely. DNS-nimelle on ensimmäisenä asetettu A-tietue, jossa määritetään domain-nimen kohde IP-osoite (ks. kuvio 27).

A Records

Domain/Host	Target	TTL	Status
portal.lsoweli.fi	217.112.252.89	86400	P

Kuvio 27. Domain nimen A-tietue

Yleensä A-tietue ohjattaisiin tiedoilla www.esimerkki.fi ja @.esimerkki.fi, mutta meidän palvelimella ainoa osoite johon halutaan käyttäjä ohjata on portal.lsoweli.fi, joten se täytyy jättää ainoaksi A-tietueeksi (Add an A record 2020).

4.5.3 Sähköpostin hankinta ja konfigurointi

Toimeksiantaja on antanut minulle oman tenantin Microsoft 365 -palveluun. Olen käynyt asettamassa itselleni yhden Exchange Online (Plan 1) -lisenssin käyttöön, joka mahdollistaa web-käyttöliittymällä varustetun sähköpostin käytön.

Sähköpostin asettaminen omalla domain-nimellä on Microsoftin hallinnassa helppoa. Ensimmäiseksi domain tuli vahvistaa TXT-tietueella, jonka Microsoft generoi, kun uutta domain-nimeä lisätään tenantiin (ks. kuvio 28).

TXT Records

Domain/Host	Target	TTL	Status
@.lsoweli.fi	MS=ms[redacted]	3600	P

Kuvio 28. Microsoftin vahvistuskoodi TXT-tietueena

Kun domain-nimi on vahvistettu, voidaan asettaa DNS-konfiguraatioihin Microsoftin postiasetukset. Microsoft tarjoaa nämä tiedot automaattisesti domain-nimeä konfiguroidessa. Postiasetuksiin kuuluvat lähtevän postin MX-tietue, SPF-tiedot TXT-tietueena sekä autodiscover CNAME-tietueena (ks. kuvio 29).

Exchange Online

Type	Priority	Host name	Points to address or value	TTL	Actions
MX	0	@	lsoweli-fi.mail.protection.outlook.com	1 Hour	
TXT	-	@	v=spf1 include:spf.protection.outlook.com -all	1 Hour	
CNAME	-	autodiscover	autodiscover.outlook.com	1 Hour	

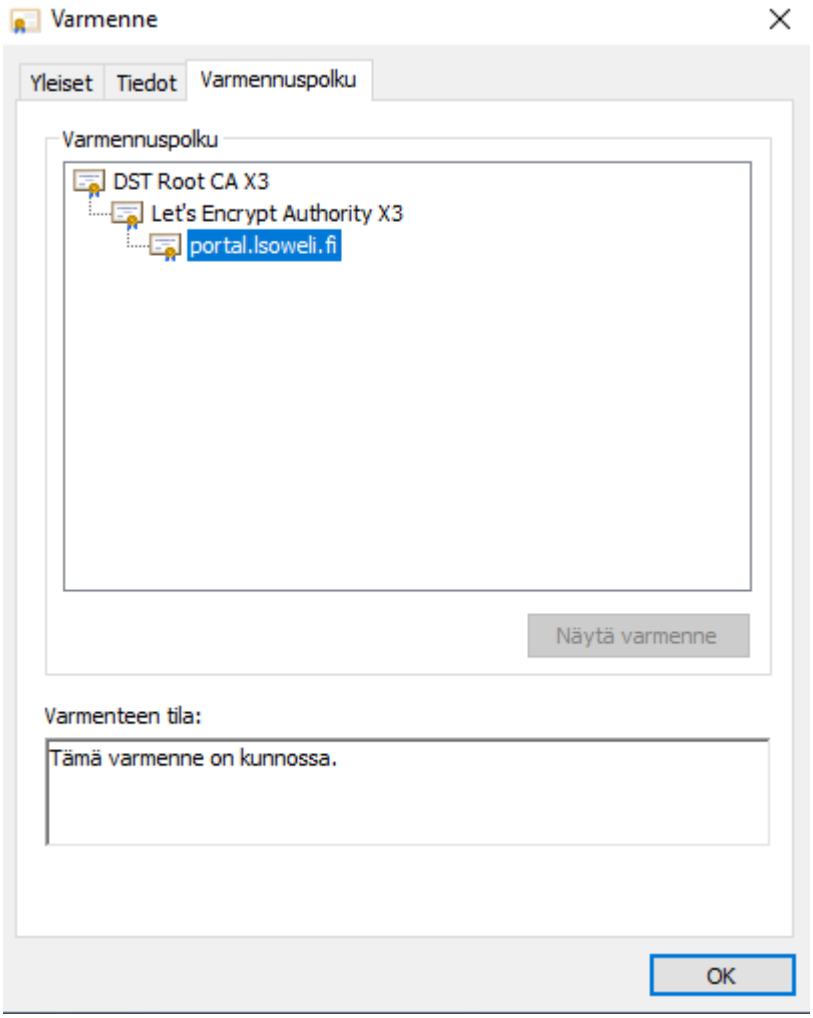
Kuvio 29. Postitietueet Microsoft 365 -hallinnassa

Tämän jälkeen olen luonut tenantiin käyttäjätilin ”tuki@lsoweli.fi” jolle olen asettanut sähköpostilisenssin. Käyttäjän sähköpostin lähetys on testattu ja se on todettu toimivaksi. SPF vahvistaa käyttäjän, joten yleisimmät sähköpostipalvelut eivät merkkaa viestiä epäilyttäväksi (How Microsoft 365 uses Sender Policy Framework (SPF) to prevent spoofing 2016).

4.5.4 Palvelun pystytys julkiverkkoon ja HTTPS

Kun kaikki palvelut on otettu käyttöön ja testattu toimivaksi, tietojenkalastelusivusto on käynnistetty Dockerilla. Kun palvelu on käynnistynyt, on palvelimella menty webkontin sisään ja ajettu certbotilla sivustolle HTTPS-sertifikaatti.

Certbotin käyttö on hyvin yksinkertaista, se ohjaa käyttäjän alusta loppuun sertifikaatin hankinnassa ja konfiguroi halutessaan myös Apachen toimimaan siten, että kaikki liikenne ohjataan suoraan HTTPS-sivustolle. Sertifikaatin asennuksen jälkeen sen toimivuus on todettu toimivaksi (ks. kuvio 30) (Certbot n.d).



Kuvio 30. Sivuston varmenne

Sivuston normaalin toiminnan jälkeen on siihen syötetty testidataa, vedetty se ulos kontista ja testattu kaikki vaiheet datan visualisoinnin suhteen. Testauksen jälkeen palvelin on jätetty ulkoverkkoon pystyyn ja sen toiminta on tarkastettu useita kertoja päivässä, jotta voidaan varmistua sen toimivuudesta.

Muutaman päivän testiajan jälkeen verkkosivulle on tehty Maltego-ohjelmalla skannaus ja tämän avulla saatu topologia kuva julkiverkon jalanjäljistä (ks. liite 17). Kuviosta on poistettu yksilöiviä puhelinnumeroita, joita Maltego on sinne löytänyt. Kuvasta löytyy myös viittaus ”dududu[.]net” nimiselle sivulle, joka osoittautui ilmeisesti kiinalaiseksi tietojenkeruusivustoksi. Myös kun on tarkasteltu CentOS 8 -palvelimen SSH-lokeja, on huomattu, että palvelimelle yritetään murtautua jatkuvalla syötöllä kiinalaisista IP-osoitteista. Täten on hyvin tärkeää koventaa palvelinta välittömästi tai mielellään ennen, kuin se kytketään julkiverkkoon.

5 Yhteenveto

5.1 Tulokset

Työn tekninen toteutus aloitettiin alkuvuodesta 2020 ja tehtiin hiljalleen muun työn ohessa pitkin kevättä. Kuitenkin loppukeväästä tekniseen toteutukseen otettiin nopeampi tahti, jotta palvelu saataisiin käyttöön, koska asiakkaat olivat kyselleet mahdollisuuksista tehdä tietojenkalastelutestausta.

Opinnäytetyön tavoitteita oli saada dataa koulutuksia varten sekä automatisoida palvelua mahdollisimman paljon. Myös opinnäytetyön kirjoittajan oma tavoite oli saada palvelusta mahdollisimman tietoturvallinen, jotta se tukee hänen ammatillista suuntausta. Tavoitteisiin päästiin opinnäytetyössä hyvin.

Datan visualisointiin olisi voitu käyttää monia muitakin tekniikoita, mutta tässä tapauksessa Python-ohjelmointikielen Pandas-kirjasto lisäosineen oli luonteva ratkaisu. Muilla tekniikoilla olisi voinut luoda visuaalisesti hienompia graafeja, mutta toteutuksen kannalta ne eivät ole oleellisia asioita. Myös esimerkiksi JavaScript-ohjelmointikielellä olisi voinut luoda reaaliaikaisen näkymän tuloksista toiselle verkkosivulle, joka olisi ollut salasanasuojauksen takana. Kuitenkin uusi verkkosivu olisi lisännyt hyökkäyspinta-alaa ja se ei olisi tarjonnut toteutukselle merkittävää lisäarvoa.

Automatisoinnin osalta työssä onnistuttiin myös suhteellisen hyvin. Palvelu saadaan käynnistettyä minuuteissa vakioasetuksilla ja valmiina olevat sähköpostiasetukset mahdollistavat kampanjan aloituksen välittömästi.

Lisäkustannuksia palveluun tulee, jos asiakas haluaa käyttää esimerkiksi itse valitsemaansa domain-nimeä testauksessa, tai haluaa muuttaa testauksen ulkoasua. Kuitenkin erilaiset muutosskenaariot Dockerin toiminnassa ovat helppoja hallita ja toteuttaa. Suurin aika itse palvelussa tulee todennäköisesti olemaan kampanjan suunnittelu asiakkaan kanssa, sekä lopullisen raportin koostaminen.

Henkilökohtaisesti järjestelmän rakentaminen ja kehittäminen antoi paljon: Todella paljon opittuja asioita verkkosivujen toiminnasta ja niiden kehityksestä, automatisoinnista Dockerin avulla sekä CentOS 8 -palvelinstruktuurista. Tärkeimpänä aiheena oli Dockerin syvällisempi tutkiminen ja hyödyntäminen, sillä sitä voi käyttää myös moneen muuhun projektiin tulevaisuudessa.

Olen itse tyytyväinen opinnäytetyön lopulliseen toteutukseen. Verkkosivu näyttää hyvältä yleisimmillä selaimilla ja toimii pääsääntöisesti kuten se on suunniteltu. Mikään järjestelmä ei kuitenkaan ole koskaan täydellinen, joten hiomista ja kehittämistä tulee varmasti olemaan tulevaisuudessa. Järjestelmä myös antaa itselle mahdollisuuden toteuttaa asiakkaille niitä töitä, joita henkilökohtaisesti pidän erittäin mielenkiintoisena.

5.2 Suurimmat haasteet

Opinnäytetyössä suurimpia haasteita olivat:

- HTML- sekä CSS-määrittelyt
- DNS-asetuksien asettaminen
- Dockerin konfigurointi

HTML- ja CSS-määrityksien luominen tyhjästä pitkän tauon jälkeen aiheutti välillä päänvaivaa. Kyseisiä asioita oli viimeksi käytetty ensimmäisen vuoden web-tekniikoiden peruskurssilla, eikä niitä sen jälkeen ole tarvinnut hyödyntää juurikaan projekteissa. Haasteita asetti myös eri selainten omat ongelmat. Esimerkiksi Microsoftin Edge selain ei tue heksadesimaalimuodossa annettuja värimäärityksiä, joten ne täytyi muuttaa RGB-muotoiseksi osassa määrityksiä. Myöhemmin kuitenkin kävi ilmi, että Edge ei tukenut 4- ja 8-merkkisiä heksadesimaalimäärityksiä, mutta tuki 3- ja 6-merkkisiä. Edgen uudessa versiossa tämä oli korjattu. Internetistä löytyy kuitenkin laajasti näistä tietoa ja ohjeita, joten suurimpia kompastuskiviä ei näiden osalta ilmestynyt.

DNS-asetuksien kanssa joutui lähtemään aika pohjatiedosta liikkeelle. DNS:n perustoiminnot olivat tuttuja, mutta esimerkiksi prosessi domain-nimen ostosta, sen liittämisestä sähköpostiin ja ohjaaminen verkkosivuille olivat vielä vieraita. Kuitenkin internet tarjosi tähän myös laajasti ohjeita ja Microsoft 365 -palvelu tarjosi sähköpostin asetuksiin hyvin selkeät ohjeet, miten se saadaan toimimaan nopeasti ja tehokkaasti.

Dockerin kanssa oli työssä ylivoimaisesti suurimmat haasteet. Palvelut saatiin toimimaan lokaaleina palveluina virtuaalikoneen päällä, mutta Dockerin puolella asiat täytyi purkaa osiin ja lähteä melkein alusta asti toteuttamaan. Jälkiviisaana olisi voinut alkaa testaamaan palvelua suoraan Dockerin päällä.

Dockerissa suurimpia haasteita olivat sen yhteensopimattomuus uudehkon CentOS 8 -palvelimen kanssa. Esimerkiksi CentOS 8 -palvelimen palomuuri Firewalld aiheutti paljon ongelmia Dockerin sisäisten verkkojen kanssa. Firewalld:n konfigurointiin löytyi internetistä ohjeita osittain, mutta moni keskustelu aiheesta johti siihen, että he ovat ottaneet koko palomuurin pois käytöstä. Toteutuksen virtuaalikoneessa on myös palomuuri pois päältä, mutta se ei vaaranna tietoturvaa merkittävästi, koska virtuaalipalvelinta suojaa julkiverkolta laitesalin oma WatchGuard -palomuuri, jossa palvelulle on määritetty oma VLAN-verkko. Firewalld esti esimerkiksi konttien keskustelun toistensa kanssa suljetussa verkossa sekä esti niiden pääsyn julkiverkkoon, vaikka oikeat konfiguraatiot omasta mielestä oli tehty.

Toinen suuri haaste oli DNS-palvelimen määritys. Dockerin kontit normaalisti hakevat DNS-palvelimen IP-osoitteen isäntäkoneelta, mutta jostain syystä ne eivät hakeneet niitä. Siksi Docker-composen YML tiedostossa on käsin määritelty kontit käyttämään Googlen DNS-palvelimia. Silti jostain syystä kontit näyttävät komennolla ”cat /etc/resolv.conf” localhostin osoitteen 127.0.0.1, mutta kontit osaavat silti selvittää DNS-nimet esimerkiksi päivityksiä hakemalla.

Kolmas suurin haaste Dockerin kanssa oli tietokantakontin tietokannan määrittely. Kuten luvussa 4.4.2 on todettu, ”docker-entrypoint-initdb.d” -kansioon lisäämällä suoritettavia tiedostoja ne suoritetaan kun kontti luodaan. Jostain syystä kuitenkaan tämä ei toiminut kertaakaan. Vianselvityksessä kävi ilmi, että kansiossa olevia tiedostoja ei ajeta, jos tietokannassa on jo jotain dataa. Yritin täten luoda docker-compose.yml-tiedostoon erinäisiä parametreja tyhjentää tietokannan dataa, mutta ne eivät onnistuneet. Täten kun kontti luodaan ensimmäistä kertaa, CreateTables.sql-tiedosto täytyy käydä kontin sisällä suorittamassa, kun se käynnistetään ensimmäistä kertaa.

5.3 Jatkokehitys

Kuten luvussa 5.1 on mainittu, mikään järjestelmä ei koskaan ole täysin valmis. Täten jo lyhyessä ajassa on tullut monta kehitysideaa, miten palvelua voitaisiin viedä eteenpäin ja viilata entistä paremmaksi.

Ensimmäisenä jatkokehityksen kohteena olisi olemassa olevien ongelmien korjaus ja automatisoinnin tehostaminen. Esimerkiksi tietokannan taulurakenteen luomisessa oleva ongelma olisi hyvä ratkaista tulevaisuudessa, jotta sen ominaisuus saataisiin tehokkaasti käyttöön.

Automatisoinnin puolesta myös palvelimelle voitaisiin tehdä erilaisia bash-skriptejä, joilla saataisiin viimeisetkin käsin syötetyt komennot poistettua palvelimelta. Toinen merkittävä uudistus olisi myös luoda usein käytetyille komennoille aliakset palvelimelle. Aliaksella voidaan määrittää, että esimerkiksi joku pitkä komento, kuten

”docker exec -it phish_mysql_1 bash”, jolla saadaan yhteys tietokantakonttiin, voitaisiin korvata komennolla ”cmysqlbash”, jolloin käsin kirjoittamisen virheet saadaan minimoitua.

Sivuston ulkoasua on myös hyvä hioa tulevaisuudessa paremmaksi ja varsinkin tilanteissa, jos Microsoft muuttaa oman kirjautumissivustonsa ulkoasua. Samalla myös, mikäli asiakas haluaa käyttää jotain muuta palvelua testauksessa, voidaan luoda yleisimmistä sivustoista myös kopioita. Myös testauksen edetessä tuli huomio, että suorat sähköpostiosoitteet sivuston kommenttiosioista tulee muuttaa muotoon siten, että automaattiset sähköpostiosoitteiden skannerit eivät niitä pääse keräämään tietokantoihinsa.

Tällä hetkellä palvelu on konfiguroitu siten, että sillä voidaan ajaa yhtä tietojenkalastelutestausta kerrallaan. Tulevaisuudessa, mikäli kyseisiä tilauksia alkaa tulla enemmän, palvelua voisi skaalata siten, että dockerin päällä pyörittää useampaa verkkosivua ja tietokantaa. Tämä onnistuu esimerkiksi sillä, että toisen palvelun liikenne ohjataan esimerkiksi porttiin 80 ja toisen 8080, jolloin verkkoportit eivät lopu kesken.

Koska verkkosivutyylinen huijaus ei ole ainoa muoto harjoittaa tietojenkalastelua, palvelua voisi laajentaa tulevaisuudessa käyttämällä haitallisia liitetiedostoja. Asiakkaille voisi luoda esimerkiksi Word-tiedoston, jonka sisällä on makro, joka esimerkiksi avattaessa ottaa yhteyden meidän palvelimelle ja kertoo mistä IP-osoitteelta ja tietokoneesta tiedosto on avattu.

Koska palvelu on luotu itsenäisesti, toimeksiantajalle täytyy luoda ohjeet sen käyttöön. Monella ei välttämättä ole kokemusta Dockerin käytöstä, joten ohjeisiin on hyvä lisätä sen yleisimmät ongelmakohdat ja kertoa, mitä niissä kannattaa kokeilla. Ohjeet pyritään luomaan siten, että kuka tahansa yrityksestä osaisi pystyttää tietojenkalastelutestauksen ja viemään sen loppuun onnistuneesti.

Samalla kun palvelimen käyttöä aletaan opastamaan muille työntekijöille, sille täytyy luoda omat käyttäjätunnukset ja avainparit muille käyttäjille. Myös palvelun kovennusta on hyvä jatkaa, esimerkiksi vaihtamalla SSH-portti vakioportista joksikin

muuksi, niin automaattiset verkkoskannerit eivät häiritse meidän aktiivista SSH-porttiamme.

Varmuuskopioiden otto tulee myös liittää palvelimelle. Palvelu ei häviä, kunhan konfiguraatiotiedostot ovat tallessa, mutta itse palvelimen pystytys alusta asti on itsessään pitkä työ.

Lähteet

Add an A record. 2020. GoDaddy-sivuston ohje A-tietueen asettamiseen DNS-asetuksiin. Viitattu 20.5.2020. https://fi.godaddy.com/help/add-an-a-record-19238

Aitchison, R. 2011. Pro DNS and BIND 10. Books24x7. Viitattu 12.5.2020. http://library.books24x7.com.ezproxy.jamk.fi:2048/toc.aspx?bookid=41403

APT28: A WINDOW INTO RUSSIA'S CYBER ESPIONAGE OPERATIONS?. 2014. FireEyen raportti APT28-ryhmittymästä. Viitattu 13.5.2020. https://www2.fireeye.com/rs/fireye/images/rpt-apt28.pdf

APT28: AT THE CENTER OF THE STORM. 2017. FireEyen raportti APT28-ryhmittymästä. Viitattu 14.5.2020. https://www2.fireeye.com/rs/848-DID-242/images/APT28-Center-of-Storm-2017.pdf?mkt_tok=eyJpI-joiWm1RMU1tTTVOMlkwTkdJMiIsInQiOiIyK0dNUFdxXC9sUWFKZTlzQmZ6U0x5SnluQTRCMm1SeTZHeDZkN0hxWW1Ozk9DumE2cFpFdm5vSE-ZYQUxJbnFYWXN0Z1NjenZkU2c2VGVYd1E5cW5ybVFOaVJkV2EyVWthY1BER-nUwMVhTckUxUW03UulxbWl5QjhCOG1kMnVUamsifQ%3D%3D

Certbot. Certbotin etusivu EFF-organisaation verkkosivuilla. Viitattu 19.5.2020. https://certbot.eff.org/

Change your email address to use your custom domain. 2020. Microsoftin tukisivuston artikkeli, kuinka saada oma domain-nimi sähköpostin käyttöön 7.5.2020. Viitattu 19.5.2020. https://docs.microsoft.com/en-us/microsoft-365/admin/email/change-email-address?view=o365-worldwide

docker-compose up. 2020. docker-compose up -komennon dokumentaatio Dockerin tukisivulla. Viitattu 20.5.2020. https://docs.docker.com/compose/reference/up/

How Microsoft 365 uses Sender Policy Framework (SPF) to prevent spoofing. 2016. Microsoftin tukisivuston artikkeli SPF:n toiminnasta 15.12.2016. Viitattu 20.5.2020. https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/how-office-365-uses-spf-to-prevent-spoofing?view=o365-worldwide

ICANN Org's Multifaceted Response to DNS Abuse. 2020. ICANN-organisaation tiedote heidän toimistaan DNS-väärinkäyttöön 20.4.2020. Viitattu 13.5.2020.https://www.icann.org/news/blog/icann-org-s-multifaceted-response-to-dns-abuse

Isoweli Oy. Isoweli Oy:n yrityskuvaus heidän verkkosivujen etusivulla. Viitattu 19.5.2020. https://www.isoweli.fi/

Kumar, P. 2019. How to Install Docker CE on CentOS 8 / RHEL 8 23.12.2019. Viitattu 20.5.2020. https://www.linuxtechi.com/install-docker-ce-centos-8-rhel-8/

M-TRENDS 2020. 2020. FireEyen raportti APT-hyökkäyksistä. Viitattu 14.5.2020. https://content.fireeye.com/m-trends/rpt-m-trends-2020

mysql. 2020. MySQL-kontin etusivu Docker Hub -sivustolla. Viitattu 20.5.2020. https://hub.docker.com/_/mysql

mysqli_close. 2020. mysqli_close funktion virallinen dokumentaatio PHP:n verkkosivuilla. Viitattu 20.5.2020. https://www.php.net/manual/en/mysqli.close.php

Office 365 -sähköpostin tietojenkalastelu ja tietomurrot erittäin yleisiä – havaitse, suojaudu, tiedota!. Päivitetty 14.10.2019. Kyberturvallisuuskeskuksen varoitus tietojenkalastelusta. Viitattu 13.5.2020. https://www.kyberturvallisuuskeskus.fi/fi/office-365-sahkopostin-tietojenkalastelu-ja-tietomurrot-erittain-yleisia-havaitse-suojaudu-tiedota

Overview of Docker Compose. 2020. Docker Compose -ominaisuuden virallinen dokumentaatio Dockerin ohjesivustolla. Viitattu 20.5.2020. https://docs.docker.com/compose/

password_hash. 2020. Password_hash funktion virallinen dokumentaatio PHP:n verkkosivuilla. Viitattu 13.5.2020. https://www.php.net/manual/en/function.password-hash.php

Spam Is Still the Choice of Online Criminals, 40 Years Later. 2018. F-Securen kirjoitus haitallisesta sähökpostiliikenteestä 31.7.2018. Viitattu 19.5.2020. https://press.f-secure.com/2018/07/31/spam-is-still-the-choice-of-online-criminals-40-years-later/

Suojautuminen Microsoft Office 365 -tunnusten kalastelulta ja tietomurroilta. 5.4.2019. Kyberturvallisuuskeskuksen laatima ohje tietojenkalastelulta suojautumiseen. Viitattu 19.5.2020. https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/Suojautuminen%20Microsoft%20Office%20365%20-tunnusten%20kalastelulta%20ja%20tietomurroilta%20web.pdf

Tagliaferri, L. 2017. How To Install Python 3 and Set Up a Local Programming Environment on CentOS 7. Viitattu 20.5.2020. https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-local-programming-environment-on-centos-7

Vase, T. 2015. ADVANTAGES OF DOCKER. Kandidaatintutkielma. Jyväskylän Yliopisto, tietojärjestelmätieteen kandidaattiohjelma. Viitattu 20.5.2020. https://jyx.jyu.fi/bitstream/handle/123456789/48029/1/URN%3ANBN%3Afi%3Ajyu-201512093942.pdf

Vatanen, P. 2017. Näin tietomurtautuja huijaa sinua – hyökkäys voi olla kuin agenttielokuvasta. Yle päivitetty 6.3.2017. Viitattu 13.5.2020. https://yle.fi/uutiset/3-9488768

Visualization. 2014. Pandas-kirjaston käyttöopas visualisointiin. Viitattu 20.5.2020. https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

What is a Container?. 2020. Dockerin virallisen verkkosivun informatiivinen sivu Docker-konttien toiminnasta. Viitattu 14.5.2020. https://www.docker.com/resources/what-container

What Is Phishing?. KnowBe4-yrityksen kuvaus tietojenkalastelusta phishing.org -verkkosivulla. Viitattu 13.5.2020. https://www.phishing.org/what-is-phishing

Liitteet

Liite 1. docker-compose.yml

```
#kaytetaan 3.7 compose-versiota
version: '3.7'

#maaritetaan palvelu "web", joka kayttaa web.dockerfilea luomiseen
services:
  web:
    build:
      context: ./
      dockerfile: web.dockerfile
    #maaritellaan portit ja DNS asetukset kontille
    ports:
      - 80:80
      - 443:443
    dns:
      - 8.8.8.8
    #maaritetaan kontti kayttamaan "network1" sisaista verkkoa, web-kontti aliaksella "web"
    networks:
      network1:
        aliases:
          - web
    #kopioi tiedostot lokaalilta hostilta konttiin
    volumes:
      - ./DocumentRoot:/var/www/html
  #luodaan mysql-kontti kayttaen mysql:5.7.25 imagea
  mysql:
    image: mysql:5.7.25
    #maaritellaan portit ja DNS asetukset kontille
    ports:
      - 3306:3306
    dns:
      - 8.8.8.8
    #maaritetaan kontti kayttamaan "network1" sisaista verkkoa, mysql-kontti aliaksella "db"
    networks:
      network1:
        aliases:
          - db
    #maaritetaan muuttujat tietokannalle
    environment:
      TZ: "Europe/Helsinki"
      MYSQL_ALLOW_EMPTY_PASSWORD: "no"
```

```
    MYSQL_ROOT_PASSWORD: “*DATABASEPASSU*”
    MYSQL_USER: ‘root’
    MYSQL_PASSWORD: ‘*DATABASEPASSU*’
    MYSQL_DATABASE: ‘phishingdb’
  #maaritetaan kontti kaynnistymaan jos se sammuu automaattisesti
  restart: always
  #maaritetaan general loki paalle mysql-palvelulle mahdollisia vikatilanteita varten
  command: mysqld —general-log=1 —general-log-file=/var/log/mysql/general-
log.log
  #luodaan users-table tietokantaan luonnin yhteydessa. scripti on ”sql_magic” kan-
siossa hostilla
  volumes:
   - ./sql_magic:/docker-entrypoint-initdb.d
#luodaan sisainen verkko konteille
networks:
 network1:
```

Liite 2. master1.css

```
/*taustan tyylit*/
body {
  margin: 0;
  padding: 0;
  background: url(bg.jpg) no-repeat center top;
  background-size: cover;
  font-family: sans-serif;
  height: 100vh;
  position: relative;
}
/*kirjautumislaatikon tyylit*/
.login-box {
  width: 450px;
  height: 400px;
  background: url("../logo.png") no-repeat top left RGB(255,255,255);
  background-position: 15px 15px;
  background-size: 150px 50px;
  color: #000;
  top: 50%;
  left: 50%;
  position: absolute;
  transform: translate(-50%, -50%);
  box-sizing: border-box;
  box-shadow: 0 30px 40px rgba(0,0,0,.1);
  overflow: hidden;
}
.loginboxcontent {
  width: 400px;
  height: 400px;
  display: block;
  position: relative;
  left: 1500px;
  padding: 90px 0px 0px 30px;
}

.logo {
  padding: 440px 430px;
  width: 150px;
  height: 50px;
  position: absolute;
  top: 20px;
  z-index: 5;
}

.login-box .avatar {
  width: 150px;
```

```
  height: 50px;
  position: absolute;
  top: 20px;
}

.login-box h1 {
  margin: 0;
  padding: 0 0 20px;
  text-align: left;
  font-size: 22px;
}

.login-box label {
  margin: 0;
  padding: 0;
  font-weight: bold;
  display: block;
}

.login-box input {
  width: 100%;
  margin-bottom: 20px;
}

.login-box input[type="email"], .login-box input[type="password"] {
  border: none;
  border-bottom: 1px solid #000;
  background: transparent;
  outline: none;
  height: 40px;
  color: #000;
  font-size: 16px;
}

.login-box input[type="submit"] {
  border: none;
  outline: none;
  height: 40px;
  width: 100px;
  background: #0067b8;
  color: #fff;
  font-size: 15px;
  top: 64%;
  left: 70%;
  position: absolute;
}

.login-box input[type="submit"]:hover {
  cursor: pointer;
```

```
  background: #0060A9;
  color: #fff;
}

.login-box a {
  text-decoration: none;
  font-size: 12px;
  line-height: 20px;
  color: #0067b8;
}

.login-box a:hover {
  color: #808080;
  text-decoration: underline;
}
}
.footer {
    position: fixed;
    bottom: 0;
    width: 100%;
    overflow: visible;
    z-index: 99;
    clear: both;
    background-color: rgba(0,0,0,0.6);
    text-align: end;
}
.footer a {
    margin-top: 5px;
    margin-bottom: 5px;
    color: #fff;
    font-size: 12px;
    line-height: 28px;
    white-space: nowrap;
    display: inline-block;
    margin-left: 8px;
    margin-right: 8px;
}
.footer a:hover {
    text-decoration: underline;
    color: aliceblue;
}
```

Liite 3. master2.css

```
/*taustan tyylit*/
body {
  margin: 0;
  padding: 0;
  background: url(bg.jpg) no-repeat center top;
  background-size: cover;
  font-family: sans-serif;
  height: 100vh;
  position: relative;
}
/*kirjautumislaatikon tyylit*/
.login-box {
  width: 450px;
  height: 400px;
  background: url("../logo.png") no-repeat top left RGB(255,255,255);
  background-position: 15px 15px;
  background-size: 150px 50px;
  color: #000;
  top: 50%;
  left: 50%;
  position: absolute;
  transform: translate(-50%, -50%);
  box-sizing: border-box;
  box-shadow: 0 30px 40px rgba(0,0,0,.1);
  overflow: hidden;
}
.loginboxcontent {
  width: 400px;
  height: 400px;
  display: block;
  position: relative;
  left: 1500px;
  padding: 90px 0px 0px 30px;
}

.logo {
  padding: 440px 430px;
  width: 150px;
  height: 50px;
  position: absolute;
  top: 20px;
  z-index: 5;
}

.login-box .avatar {
  width: 150px;
```

```
  height: 50px;
  position: absolute;
  top: 20px;
}

.login-box h1 {
  margin: 0;
  padding: 0 0 20px;
  text-align: left;
  font-size: 22px;
}

.login-box label {
  margin: 0;
  padding: 0;
  font-weight: bold;
  display: block;
}

.login-box input {
  width: 100%;
  margin-bottom: 20px;
}

.login-box input[type="email"], .login-box input[type="password"] {
  border: none;
  border-bottom: 1px solid #000;
  background: transparent;
  outline: none;
  height: 40px;
  color: #000;
  font-size: 16px;
}

#emailme2 {
  border: none;
  border-bottom: 1px solid #000;
  background: transparent;
  outline: none;
  height: 40px;
  color: #555;
  font-size: 16px;
}


.login-box input[type="submit"] {
  border: none;
  outline: none;
  height: 40px;
```

```
  width: 100px;
  background: #0067b8;
  color: #fff;
  font-size: 15px;
  top: 64%;
  left: 70%;
  position: absolute;
}

.login-box input[type="submit"]:hover {
  cursor: pointer;
  background: #0060A9;
  color: #fff;
}

.login-box a {
  text-decoration: none;
  font-size: 12px;
  line-height: 20px;
  color: #0067b8;
}

.login-box a:hover {
  color: #808080;
  text-decoration: underline;
}

.footer {
    position: fixed;
    bottom: 0;
    width: 100%;
    overflow: visible;
    z-index: 99;
    clear: both;
    background-color: rgba(0,0,0,0.6);
    text-align: end;
}
.footer a {
    margin-top: 5px;
    margin-bottom: 5px;
    color: #fff;
    font-size: 12px;
    line-height: 28px;
    white-space: nowrap;
    display: inline-block;
    margin-left: 8px;
    margin-right: 8px;
}
.footer a:hover {
```

```
    text-decoration: underline;
    color: aliceblue;
}
```

Liite 4. config.php

```
<?php
#maaritellaan errorit nakyviin mikali niita tulee, tuotannossa pois paalta
#error_reporting(E_ALL);
#ini_set('display_errors', 1);

#maaritellaan mysql-yhteyden parametrit
define('DB_HOST', 'db');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '*DATABASEPASSU*');
define('DB_NAME', 'phishingdb');

#Luodaan mysql-yhteys
$link = mysqli_connect(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_NAME);

#tarkistetaan mysql-yhteys
if($link === false){
   die("ERROR: Could not connect. " . mysqli_connect_error());
}

#kerataan sivun tiedot post-parametrille talteen
$username = $_POST['username'];
$password = password_hash($_POST['password'], PASSWORD_DEFAULT);

#valmistellaan tietojen syotto kantaan ja syotetaan
$sql = "INSERT INTO users (username,password) VALUES ('$username','$password')";
mysqli_query($link,$sql);

#suljetaan mysql-yhteys
mysqli_close($link);

#ohjataan kayttaja pois sivulta
header("location: https://portal.office.com/");
?>
```

Liite 5. index.html

```
<!DOCTYPE html>
<!—
 THIS IS AUTHORIZED PHISHINGTEST BY ISOWELI OY.
 PLEASE DON'T REPORT THIS SITE OR DOMAIN.
 FOR MORE INFORMATION, CONTACT pekka.sivusuo@isoweli.fi or tuki@isoweli.fi
→
<html>
 <head>
  <meta charset="utf-8">
  <title>Isoweli-365</title>
  <link rel="stylesheet" href="css/master1.css">
  <!—scripti ajaa animoinnin kirjautumislaatikolle →
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script>
   $(function(){
    $(".loginboxcontent").animate({left: '0px'});
   });
</script>
 </head>
 <body>
  <div class="login-box">
    <div class="loginboxcontent">
   <h1>Sign in</h1>
   <!—POST-muuttujalla syotetaan kayttajatunnus seuraavalle sivulle →
   <form action="/verify.php" method="post">
    <!—Syotetaan kayttajatunnus →
    <label for="username">Username</label>
    <input type="email" name="username2" id="username2" placeholder="Email,
phone, or Skype" required>
    <a href="http://172.31.253.101/test2.php/">
     <input type="submit" value="Next">
    </a>
                                <a href="https://www.microsoft.com/">Create account
<br><br></a>
    <a href="https://www.microsoft.com/">Can't access your account?</a>
   </form>
    </div>
  </div>
  <div class="footer">
    <a href="#">Terms of condition</a>
  </div>
 </body>
</html>
```

Liite 6. verify.php

```
<!DOCTYPE html>
<!—
 THIS IS AUTHORIZED PHISHINGTEST BY ISOWELI OY.
 PLEASE DON'T REPORT THIS SITE OR DOMAIN.
 FOR MORE INFORMATION, CONTACT pekka.sivusuo@isoweli.fi or tuki@isoweli.fi
→
<html>
 <head>
  <meta charset="utf-8">
  <title>Isoweli-365</title>
  <link rel="stylesheet" href="css/master2.css">
  <script src="https://ajax.goog-
leapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script>
   $(function(){
    $(".loginboxcontent").animate({left: '0px'});
   });
</script>
 </head>
 <body>
  <div class="login-box">
    <div class="loginboxcontent">
   <h1>Sign in</h1>
   <!—formi kayttaa config.php-tiedostoa ja keraa siihen parametrit→
   <form action="config.php" method="post">
    <label for="username">Username</label>
  <!—scripti estaa sahkopostikentan muuttamisen ja tuo sen edelliselta sivulta naky-
ville→
                    <?php
                    $email_address = $_POST['username2'];
                    echo '<input type="email" name="username"
id="username" value="' . $email_address . '" disabled />';
     echo '<input type="hidden" name="username" id="username" value="' .
$email_address . '" />';
                    ?>
     <label for="password">Password</label>
     <input type="password" placeholder="Enter Password" name="password" re-
quired>
     <input type="submit" value="Log In">
                    <a href="https://www.microsoft.com/">Forgot pass-
word?</a>
   </form>
    </div>
  </div>
   <div class="footer">
    <a href="#">Terms of condition</a>
```

```
    </div>
  </body>
</html>
```

Liite 7. sqltograph.py

```
#!/usr/bin/env python 3

#ladataan tarvittavat kirjastot
import pandas as pd
import sqlalchemy
import pymysql
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

#Luodaan yhteys tietokantaan ja tuodaan tietokannan data dataframeksi
engine = sqlalchemy.create_en-
gine('mysql+pymysql://root:*DATABASEPASSU*@localhost/phishingdb')
df = pd.read_sql_table('users', engine,)

#Siistitaan aikaleimoista kellonajat pois
df['created_at'] = df['created_at'].apply(lambda x: pd.Timestamp(x).strftime('%Y-
%m-%d'))
df['created_at'] = df['created_at'].apply(pd.to_datetime)
#asetetaan paivamaara dataframen indeksiksi
df.set_index('created_at', inplace=True)
#parsitaan pelkat domain-nimet nakyville
df['domain'] = df['username'].str.split('@').str.get(1)

#luodaan uusi dataframe df2 jossa lasketaan kirjautumismaarat per paiva
df2 = df.groupby(["created_at"]).count()
#maaritellaan pylvasdiagrammin tyylit ja selitteet
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(10,6))
ax.bar(df2['username'].index, df2['username'])
ax.set_title('Kirjautumisyritykset päivittäin')
ax.set_ylabel('Kirjautumiskerrat')
ax.set_xlabel('Päivämäärät')
ax.xaxis.set_major_formatter(mdates.DateFormatter('%m-%d'))
#tallennetaan pylvasdiagrammi PNG-muodossa
plt.savefig('/home/pekka/phish/pylvas.png', dpi=400)

#luodaan uusi dataframe df3 jossa erotellaan syotetyt domainit ja lasketaan
df3 = df.groupby(["domain"]).count()
#muotoillaan piirakkakuvio
plot = df3.plot.pie(y='username', figsize=(5,5))
fig2 = plt.gcf()
plt.xlabel('Syötetyt domainit')
plt.ylabel('')
#tallennetaan piirakkakuvio PNG-muodossa
plt.savefig('/home/pekka/phish/piirakka.png', dpi=400)
```

```
#Luodaan uusi dataframe df4 jossa tulostetaan sen sisalto xlsx-muotoon
df4 = pd.read_sql_table('users', engine,)
df4.to_excel('/home/pekka/phish/sqldata.xlsx')

#Kun skripti ajettu, tulostetaan komentoriville siita ilmoitus
print('Ready?')
```

Liite 8. web.dockerfile

```
#kaytetaan php:7.3-apache imagea
FROM php:7.3-apache
#asennetaan nano seka tarvittavat PHP-liitannaiset
RUN apt-get update && apt-get install -y \
                                    nano \
                && docker-php-ext-install pdo_mysql mysqli pdo
#asennetaan certbot valmiiksi palvelimelle
RUN apt-get install -y certbot python-certbot-apache
#siirretaan muokattu apachen konfiguraatiotiedosto palvelimelle hostilta
COPY 000-default.conf /etc/apache2/sites-available/000-default.conf
#maaritellaan avonaiset portit
EXPOSE 80
EXPOSE 443
```

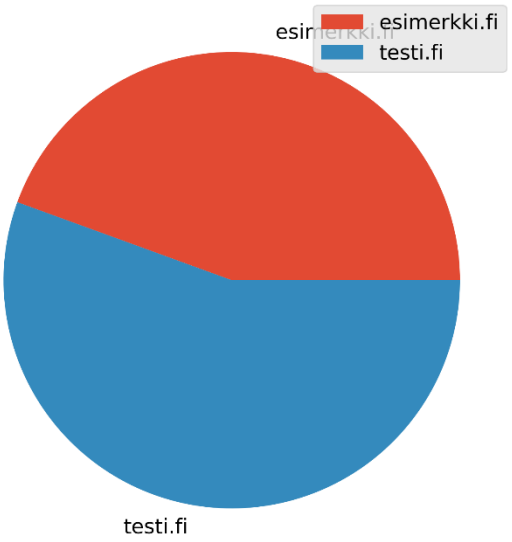
Liite 9. 000-default.conf

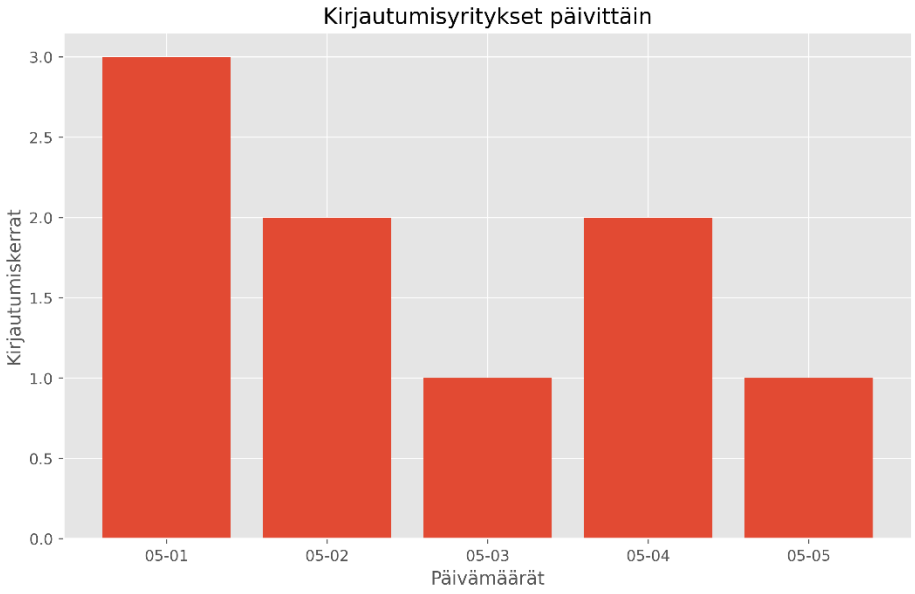
```
<VirtualHost *:80>
        ServerName portal.lsoweli.fi

        ServerAdmin pekka.sivusuo@isoweli.fi
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

                # JUST A TEST LINE FOR DOCKER-COMPOSE
</VirtualHost>
```

Liite 10. CreateTable.sql

```
CREATE TABLE users (
        id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        username VARCHAR(50) NOT NULL,
        password VARCHAR(255) NOT NULL,
        created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Liite 11. Graafit sekä Excel





	id	username	password	created_at
0	1	liisa@testi.fi	hashedpassword	2020-05-01 12:10:22
1	2	kalle@testi.fi	hashedpassword	2020-05-01 12:16:01
2	3	pekka@testi.fi	hashedpassword	2020-05-01 14:32:54
3	4	markku@testi.fi	hashedpassword	2020-05-02 09:13:09
4	5	saima@testi.fi	hashedpassword	2020-05-02 11:55:23
5	6	esim1@esimerkki.fi	hashedpassword	2020-05-03 16:12:59
6	7	esim2@esimerkki.fi	hashedpassword	2020-05-04 11:10:48
7	8	esim3@esimerkki.fi	hashedpassword	2020-05-04 12:57:34
8	9	esim4@esimerkki.fi	hashedpassword	2020-05-05 07:19:51

Liite 12. fail2ban-sovelluksen jail.local

```
[DEFAULT]
ignoreip = *toimeksiantajan toimiston ip-osoite*
#IP-osoitteen estoaika sekunneissa
bantime  = 86400
#monta kertaa tietyssa ajassa epaonnistuneita kirjautumisia ennen estoa.
findtime  = 300
maxretry = 3

#fail2ban kauttaa iptablesia muutoksiin ja kayttaa systemd:ta lokien monitorointiin
banaction = iptables-multiport
backend = systemd

[sshd]
enabled = true
```

Liite 13. lsoweli.fi DNS-konfiguraatio

DNS Configuration: lsoweli.fi

MX Records

Mail Server (from)	Target Mail Server (to)	TTL	Status
@.lsoweli.fi	lsoweli-fi.mail.protection.outlook.com	3600	P

TXT Records

Domain/Host	Target	TTL	Status
@.lsoweli.fi	MS=ms[illegible]	3600	P
@.lsoweli.fi	v=spf1 include:spf.protection.outlook.com -all	3600	P

CNAME Records

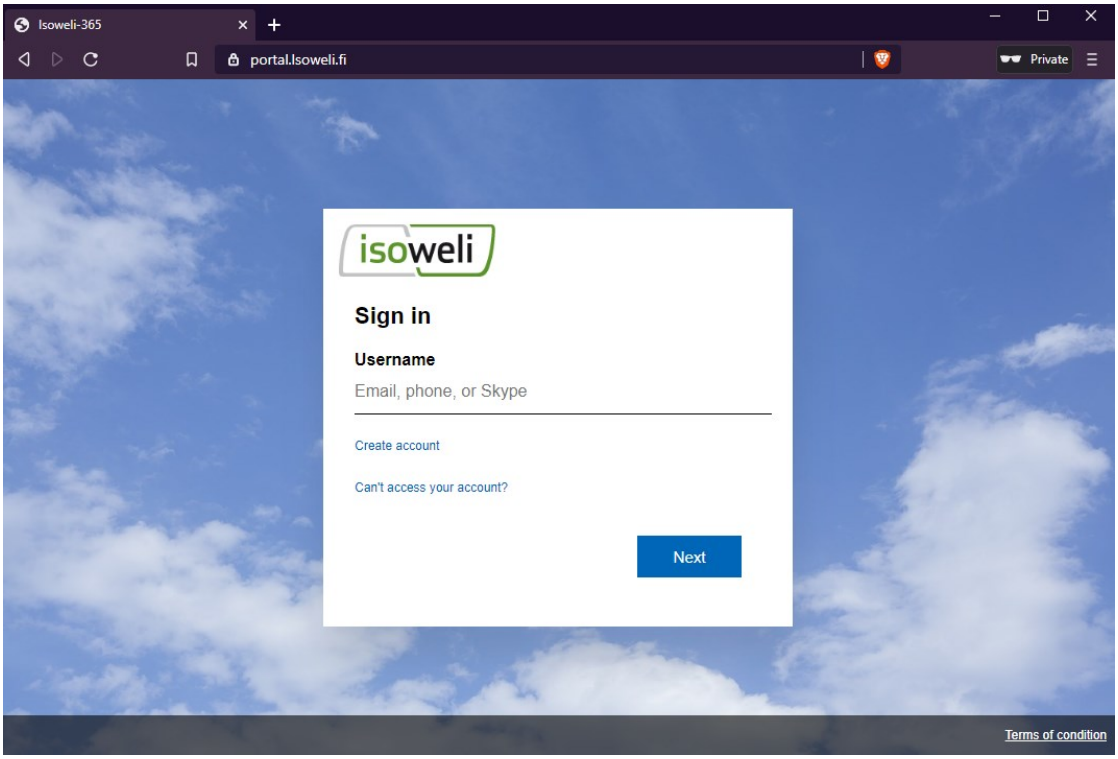
Domain/Host	Target	TTL	Status
autodiscover.lsoweli.fi	autodiscover.outlook.com	3600	P
enterpriseenrollment.lsoweli.fi	enterpriseenrollment.manage.microsoft.com	3600	P
enterpriseregistration.lsoweli.fi	enterpriseregistration.windows.net	3600	P

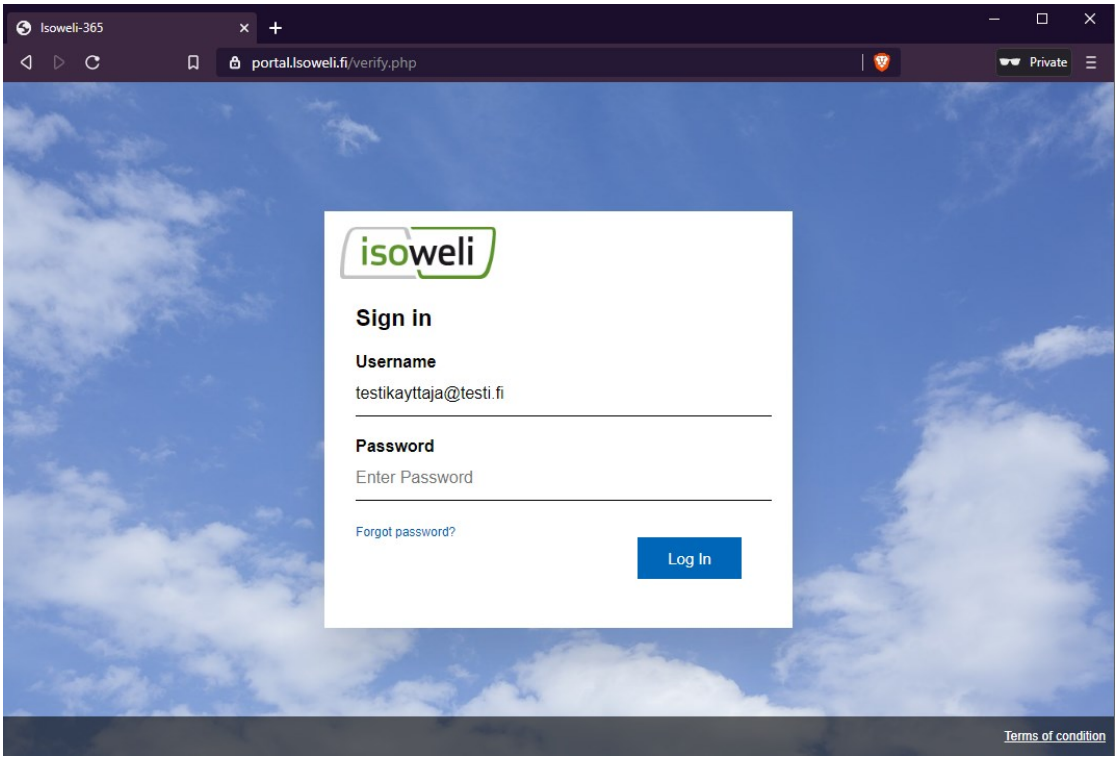
A Records

Domain/Host	Target	TTL	Status
portal.lsoweli.fi	217.112.252.89	86400	P

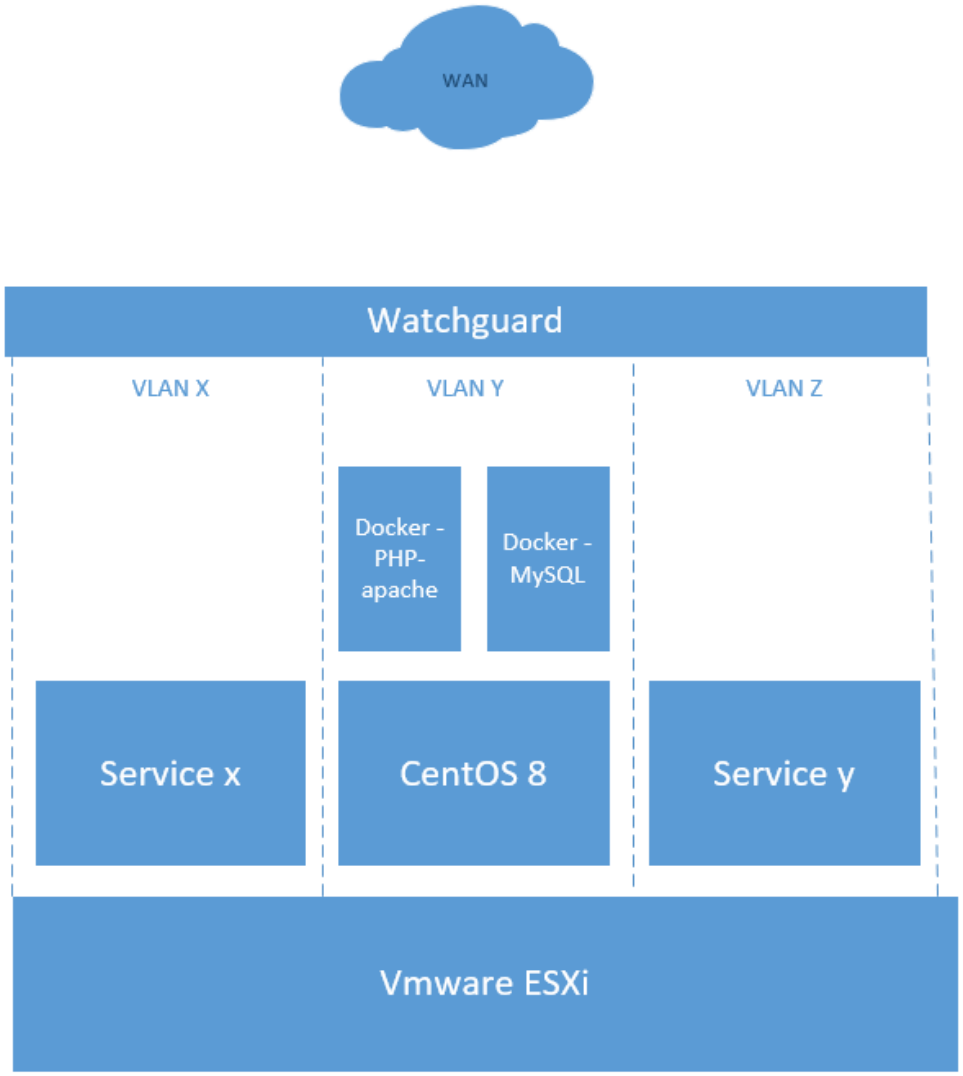
Status:
X: scheduled to be deleted,
C: scheduled to be created,
M: scheduled to be changed
Mail Forward Status:
A: active
O: not active
B: blocked
Valid From/To Status:
Y: temp. active,
S: scheduled
E: expired
P: permanently active

Liite 14. Tietojenkalastelusivun ulkoasu

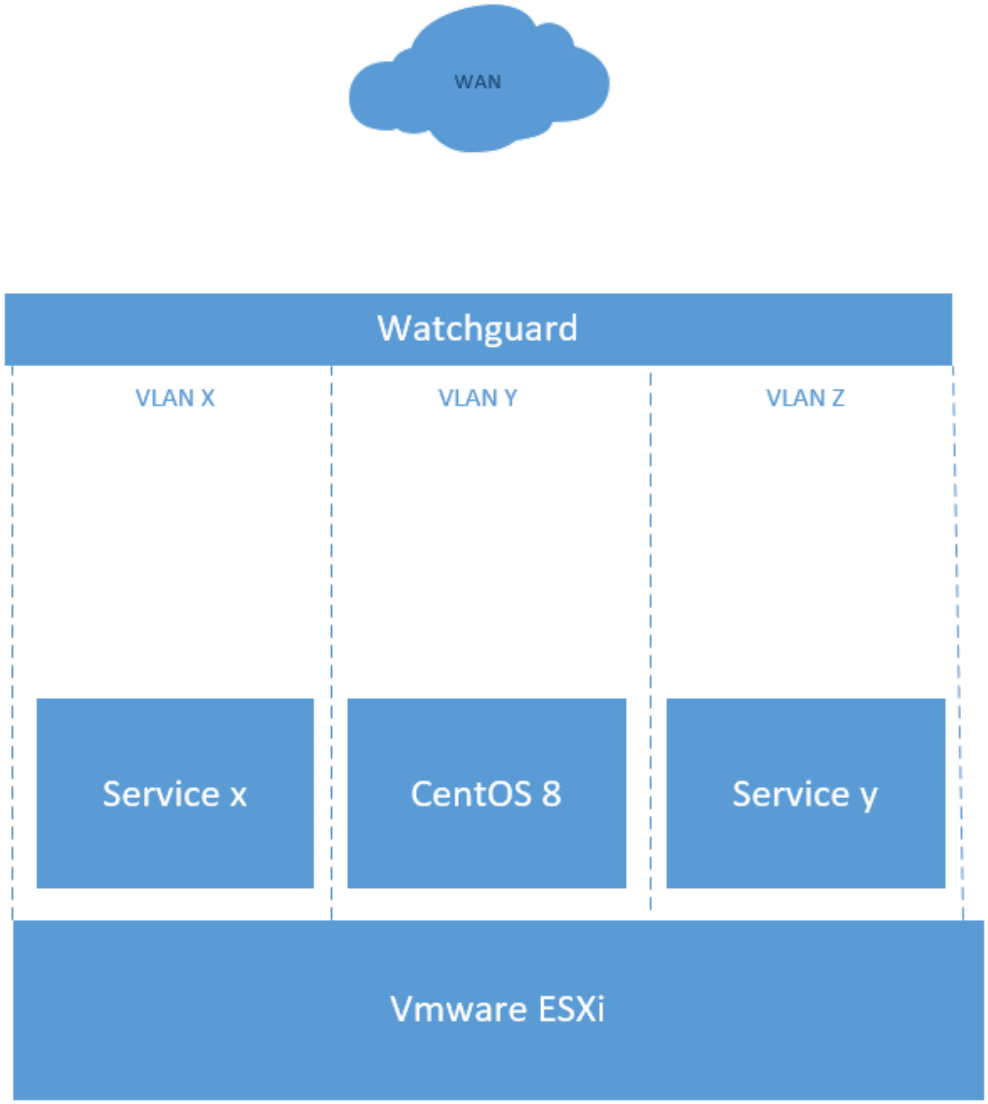




Liite 15. Tuotantoympäristön topologia



Liite 16. Testiympäristön topologia



Liite 17. Tuotantoympäristön Maltego-raportti