Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Jerry Träskelin

Tietoturvallinen etätyöpöytäympäristö

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

3.9.2019

Tekijä Otsikko Sivumäärä Aika	Jerry Träskelin Tietoturvallinen etätyöpöytäympäristö 28 sivua + 1 liite 3.9.2019
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikan tutkinto-ohjelma
Ammatillinen pääaine	IoT and Cloud Computing
Ohjaajat	Kehityspäällikkö Miikka Kallberg, CSC Lehtori Tapio Wikström, Metropolia AMK

Insinöörityön tavoitteena oli suunnitella ja toteuttaa turvallinen etätyöpöytäympäristö, jossa loppukäyttäjät voivat laatia raportteja ja käyttää tietokantayhteyttä käyttäen asiakkaan määrittelemiä varusohjelmia. Kirjautumisen järjestelmään tulisi tapahtua CSC:n keskitettyyn käyttäjänhallintapalveluun rekisteröityjen asiakastunnusten avulla. Tietoturvasyistä kirjautumisen tulisi myös vaatia kaksivaiheinen tunnistautuminen.

Ympäristön toteutuksessa käytettiin mahdollisimman paljon PowerShell-komentosarjoja, mikä helpotti suuresti dokumentointia. Lisäksi luotujen komentosarjojen avulla voidaan tarvittaessa toteuttaa ympäristö uudelleen tai luoda uusia vastaavanlaisia ympäristöjä. Toteutuksen aikana todettiin yhdessä asiakkaan kanssa, että ympäristöön tarvitaan joitakin lisäominaisuuksia. Tämän seurauksena suunnittelua tehtiin jonkin verran myös toteutuksen aikana, ja ylipäänsä oli tarpeen kehittää ympäristöä melko joustavasti.

Kokonaisuutena projekti onnistui hyvin, ja asiakas sai ympäristön, joka toteuttaa siltä vaaditut ominaisuudet. Ympäristöllä on nykyään noin 100 käyttäjää, ja se toimii CSC:n normaalien ylläpitorutiinien alaisuudessa.

Avainsanat	Windows Server, PowerShell, etätyöpöytä, etäkäyttö

Author Title Number of Pages Date	Jerry Träskelin Secure Remote Desktop Environment 28 pages + 1 appendix 3 September 2019
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Professional Major	IoT and Cloud Computing
Instructors	Miikka Kallberg, Development Manager, CSC Tapio Wikström, Senior Lecturer, Metropolia UAS

The goal of this thesis was to plan and implement a secure remote desktop environment where end users can produce reports and use a database connection with software defined by the customer. Logging in to the environment should happen work with accounts registered to CSC's centralized user account management service. Logging in should also require a two-factor authentication for security reasons.

The environment was implemented using PowerShell scripts whenever possible which made documentation significantly easier while also making it possible to rebuild the environment if needed and also implement similar new environments. During implementation, it was concluded together with the customer that some additional features are required. This resulted that some planning had also be done at the same time as the implementation was already in progress. Altogether, a flexible development model was required.

End result of the project was that the customer got an environment that fulfills their requirements. Today, the environment has some 100 users and is maintained to the usual CSC standards.

Keywords	Windows Server, PowerShell, remote desktop, remote use

Sisällys

Lyhenteet

1 Johdanto 1

2 Määrittely 1

3 Käytetyt teknologiat 2

3.1 Windows Server 2016 2
3.2 Aktiivihakemisto 3
3.3 Etätyöpöytäpalvelun tukipalvelut 4
3.4 Kaksivaiheinen tunnistautuminen 5

4 Suunnittelu 6

5 Toteutus 9

5.1 Aktiivihakemiston asennus ja konfigurointi 9
5.2 IdM-käyttäjänhallinta ja tutkijan käyttöliittymä 10
5.3 Tukipalveluiden asennus 12
5.4 MultiOTP:n asennus 14
5.5 Etätyöpöytäpalvelimen asennus 15

6 Testaus ja käyttöönotto 16

6.1 Projektiryhmän sisäinen testaus 16
6.2 Rajatun käyttäjäryhmän testaus 17
6.3 Laajamittainen testaus 17
6.4 Palautteeseen reagointi 18
6.5 Käyttöönotto 19

7 Ylläpito 20

7.1 Asiakkaan kanssa sovitut asiat 20
7.2 Palvelinten monitorointi 21
7.3 Huoltokatkot 23
7.4 Dokumentointi 24

8 Yhteenveto 25

Lähteet 27

Liitteet

Liite 1. Windowsin päivityspalvelun asennus PowerShellillä

Lyhenteet

CB	Connection Broker. Etätyöpöytäyhteyden välittäjä.
CSC	CSC – Tieteen tietotekniikan keskus Oy on Suomen valtion ja korkeakoulujen omistama osakeyhtiö.
DC	Domain Controller. Toimialueen ohjainpalvelin.
DNS	Domain Name System. Nimipalvelujärjestelmä.
HMAC	Hash-message authentication code. Salausalgoritmi.
HTML	Hypertext Markup Language. Ohjelmointikieli verkkosivuja varten.
HTML5	Hypertext Markup Language 5. HTML:n vuonna 2014 julkaistu versio.
HTTP	Hypertext Transfer Protocol. Protokolla selainten ja WWW-palvelinten väliseen tiedonsiirtoon.
HTTPS	Hypertext Transfer Protocol Secure. HTTP:n salattu versio.
IdM	Identity Management. CSC:llä käytössä oleva käyttäjän tunnistamis- ja hallinnointipalvelu.
LDAP	Lightweight Directory Access Protocol. Protokolla hakemistopalveluita varten.
LDAPS	Lightweight Directory Access Protocol Secure. LDAPin salattu versio.
ODBC	Open Database Connectivity. Tietokantarajapintastandardi.
OTP	One-time password. Kertakäyttöinen salasana, jota voidaan käyttää kaksivaiheisen tunnistautumisen toisena vaiheena.

PHP	PHP: Hypertext Preprocessor, web-kehityksessä käytettävä ohjelmointikieli.
QR	Quick Response. Kuviokoodi, joka sisältää informaatiota, kuten aktivointilinkin.
RD	Remote Desktop. Etätyöpöytä.
RDP	Remote Desktop Protocol. Microsoftin kehittämä protokolla graafisen etäyhteyden muodostamiseen.
SFTP	SSH File Transfer Protocol. Salattu tiedostonsiirtoprotokolla, joka käyttää liikenteen salaamiseen SSH:ta.
SSH	Secure Shell. Tietoliikenteen salaamiseen tarkoitettu protokolla, jota käytetään usein etäyhteyden salaamiseen.
SQL	Structured Query Language. Kyselykieli tietokantakyselyjen tekemiseen.
SUI	Scientist User Interface. Tutkijan käyttöliittymä, jonka avulla CSC:n asiakkaat voivat hallinnoida käyttäjätunnuksiaan ja käyttämiään palveluita.
TLS	Transport Layer Security. Tietoliikenteen salaamiseen käytettävä protokolla.
TOTP	Time-based one-time password. Kertakäyttöinen salasana, joka vaihtuu määritellyin väliajoin.
UPN	User Principal Name. Tunnuksesta ja toimialueesta koostuva käyttäjätunnus.
WSUS	Windows Server Update Services. Windows-palvelinten päivityspalvelu.

1 Johdanto

Insinöörityön tavoitteena oli suunnitella ja toteuttaa asiakkaalle ratkaisu, joka mahdollistaa turvallisen etätyöskentelyn asiakkaalle jo aiemmin tarjotun tietokantapalvelun parissa. Asiakkaalle tarjottiin Windows-etätyöpöytäympäristöä, joka koostuu etätyöpöytäpalvelimesta ja sen vaatimista tukipalvelimista, joiden tarjoamiin palveluihin kuuluu muun muassa aktiivihakemisto, etätyöpöydän yhdyskäytävä, lisensointipalvelu sekä muita etätyöpöytäympäristölle tarvittavia palveluita. Lisäksi palveluun kuului asiakkaan määrittelemiä varusohjelmia, joita tarvittiin tietokantapalvelun käyttämiseen.

Työ toteutettiin projektimuotoisena ja asiakas pidettiin aktiivisesti mukana suunnittelun ja toteutuksen eri vaiheissa. Palvelua testattiin asiakkaan useissa eri vaiheissa sekä asiakkaan kanssa että sisäisesti, ja testien esiin tuomioon muutostarpeisiin reagoitiin nopeasti suunnitelmaa muuttamalla. Esimerkiksi käyttäjämäärän kasvu ja lisätarpeet tiedostojen siirrossa ja tarjottavissa varusohjelmissa aiheuttivat tarpeen tehdä muutoksia suunniteltuun palveluun.

Työn tilaaja oli CSC – Tieteen tietotekniikan keskus. CSC on voittoa tavoittelematon osakeyhtiö, jonka omistajia ovat Suomen valtio ja korkeakoulut. CSC:n asiakkaita ovat esimerkiksi korkeakoulut ja tutkimuslaitokset, joille CSC tuottaa erilaisia palveluita. Tarve insinöörityössä kehitetylle palvelulle syntyi yhdessä CSC:n ja asiakkaan kanssa tarkoituksena luoda paremmat edellytykset toisen jo olemassa olevan palvelun käyttämiselle.

2 Määrittely

Yhdessä asiakkaan kanssa tehtiin keväällä 2018 etätyöpöytäympäristölle määrittely, jonka mukaan sen pitäisi mahdollistaa noin 20 käyttäjän turvallinen työskentely. Etätyöpöytäympäristö määriteltiin siten, että sen tulee täyttää seuraavat kriteerit mahdollisimman edullisesti.

- Etätyöpöydällä on Windows-käyttöjärjestelmä.
- Jokaisella käyttäjällä on henkilökohtainen käyttäjätunnus ja salasana.
- Tunnuksia hallinnoidaan CSC:n keskitetyllä käyttäjänhallinnalla.

- Kirjautuminen vaatii kaksivaiheisen tunnistautumisen.
- Etätyöpöytäyhteys on salattu.
- Tiedostojen siirtäminen ulos etätyöpöytäympäristöstä tapahtuu hallitusti ja ylläpitäjällä on mahdollisuus tarkastaa, mitä tiedostoja on siirretty ulos.
- Etätyöpöytäpalvelimelta ei ole pääsyä julkiseen verkkoon.
- Käyttäjien toimintaa etätyöpöydällä rajoitetaan mahdollisimman paljon, kuitenkaan haittaamatta työntekoa.
- Toteutusta ja ylläpitoa automatisoidaan niin pitkälle kuin mahdollista muun muassa PowerShell-komentosarjoilla.
- Etätyöpöytäympäristöä valvotaan CSC:n yleisillä valvontatyökaluilla.
- Etätyöpöydällä voidaan käyttää ODBC-yhteyttä Microsoftin Access- ja Oraclen SQL Developer -ohjelmilla.

3 Käytetyt teknologiat

Projektissa käytetyt teknologiat määräytyivät osittain asiakkaan tarpeiden mukaisesti, ja osittain sen perusteella, mistä teknologioista oli jo olemassa olevaa osaamista ja hyviä kokemuksia. Esimerkiksi asiakkaan käyttämien varusohjelmien vuoksi ainoa käyttöjärjestelmävaihtoehto oli Microsoftin Windows-käyttöjärjestelmä. Windowsin valinta etätyöpöydän käyttöjärjestelmäksi tarkoitti myös sitä, että ympäristön tukipalvelut, kuten identiteetinhallinta oli myös toteutettava Microsoftin ohjelmistoilla. Toisaalta Windows-pohjaisista etätyöpöytäympäristöistä löytyi CSC:ltä entuudestaan paljon osaamista, mistä oli suuri hyöty projektissa.

3.1 Windows Server 2016

Windows Server 2016 on vuonna 2016 julkaistu Microsoftin Windows-käyttöjärjestelmän palvelinversio. Etätyöpöytäyhteyttä käyttävän loppukäyttäjän näkökulmasta suurin ero edelliseen Windows Server -versioon on työpöydän ulkoasu, joka on samanlainen kuin Windows 10 -käyttöjärjestelmässä. Lisäksi Windows Server 2016:ssa on useita virtualisointiin ja pilviteknologioihin liittyviä uusia ominaisuuksia ja parannuksia [1]. Jotta Microsoftin uutta, HTML 5 -pohjaista etätyöpöytäsovellusta voi käyttää, täytyy etätyöpöydän yhdyskäytävän, välittäjäpalvelun ja web-käyttöliittymän olla asennettu Windows Server 2016- tai Windows Server 2019 -palvelimelle [2].

3.2 Aktiivihakemisto

Aktiivihakemisto on Microsoftin kehittämä identiteetinhallintapalvelu, jonka avulla voidaan keskittää Windows-ympäristöjen käyttäjätunnusten ja tietokoneiden hallinta. Aktiivihakemiston keskeisin osa on ohjainpalvelin, jolle tallennetaan kaikki aktiivihakemiston jäsenet, eli esimerkiksi tietokone- ja käyttäjätunnukset. Aktiivihakemisto perustuu avoimeen Lightweight Directory Access Protocol -standardiin. [3.] Kun käyttäjä kirjautuu aktiivihakemiston toimialueen jäsenenä olevalle tietokoneelle tunnuksillaan, tietokone ottaa yhteyden ohjainpalvelimeen käyttäen Kerberos-protokollaa [4].

Aktiivihakemiston looginen rakenne perustuu toimialueisiin, joiden ylimpänä auktoriteettina toimii ohjainpalvelin. Toimialueella voi olla hierarkiassa alempana sijaitsevia toimialueita, jolloin ne muodostavat toimialuepuun, jonka juuri on hierarkiassa ylempänä sijaitseva toimialue. Esimerkiksi toimialuetta yritys.com alempana hierarkiassa voi sijaita toimialueet helsinki.yritys.com, espoo.yritys.com ja vantaa.yritys.com. Tällöin yritys.com on toimialuepuun juuritoimialue. Mikäli hierarkiaan kuuluu vain yksi toimialue, muodostaa se yksinään toimialuepuun. Toimialuepuut voivat muodostaa keskenään luottosuhteita, jolloin eri toimialuepuissa sijaitsevat toimialueet voivat käyttää toistensa resursseja. Tällöin toimialuepuut muodostavat metsän. Myös metsät voivat luoda keskenään luottosuhteita. [5, s. 20–24.]

Aktiivihakemistossa dataa, kuten käyttäjätunnuksia käsitellään objekteina. Toimialueen sisäinen hierarkia koostuu organisaatioyksiköistä, jotka ovat käytännössä kansioita, joihin voidaan sijoittaa erilaisia objekteja. Toimialueen hierarkiaa on hyvä pohtia jo suunnitteluvaiheessa, sillä esimerkiksi ryhmäkäytäntöjä voidaan kohdentaa organisaatioyksikkötasolla niin, että esimerkiksi järjestelmänvalvojatunnuksia sisältävään organisaatioyksikköön sovelletaan eri ryhmäkäytäntöobjekteja kuin tavallisia käyttäjätunnuksia sisältävälle organisaatioyksikölle. Hyvin suunniteltu hierarkia helpottaa ympäristön rakentamista ja ylläpitoa. [5, s. 24–25.]

3.3 Etätyöpöytäpalvelun tukipalvelut

Etätyöpöytäpalvelua varten tarvittiin tukipalveluiksi etätyöpöytäyhteyden yhdyskäytävä, etätyöpöytäyhteyden välittäjäpalvelu, etätyöpöytäyhteyden web-käyttöliittymä, etätyöpöytäyhteyden web-sovellus ja etätyöpöydän lisensointipalvelu.

Etätyöpöytäyhteyden yhdyskäytävä on palvelu, joka koteloi Remote Desktop -protokollan mukaisen liikenteen TLS-salatuksi HTTPS-liikenteeksi ja suojaa siten yhteyden ulkopuolisilta tahoilta [6]. Etätyöpöytäyhteyden TLS-salaaminen on erityisten tärkeää internetin yli käytettävissä etätyöpöytäpalveluissa, sillä mikäli yhteys jätettäisiin täysin salaamatta, voisi ulkopuolinen taho salakuunnella etätyöpöytäyhteyttä. Yhdyskäytävän toinen suuri hyöty on se, että sen takana olevien isäntäpalvelinten ei tarvitse olla suoraan saavutettavissa ulkoverkosta. Isäntäpalvelimet voidaan siis tarvittaessa eristää täysin omaan verkkoonsa, ja ainoa sisään saapuva yhteys tulee yhdyskäytäväpalvelimen kautta, mikä lisää omalta osaltaan etätyöpöydän turvallisuutta.

Etätyöpöytäyhteyden välittäjäpalvelu ohjaa etätyöpöytäyhteyksiä isäntäpalvelimille. Välittäjäpalvelun avulla voidaan luoda useammasta isäntäpalvelimesta koostuvia etätyöpöytäklustereita ja tasata kuormaa isäntäpalvelinten välillä. [7.]

Etätyöpöytäyhteyden web-käyttöliittymä tarjoaa välittäjäpalvelulla luodut etätyöpöytäklusterit verkkosivuston välityksellä, jolloin käyttäjien ei tarvitse itse osata konfiguroida ja avata etätyöpöytäyhteyttä, vaan käyttäjä saa valmiin RDP-asetustiedoston verkkosivustolta, ja se aukeaa automaattisesti etätyöpöytäsovelluksella. [8.]

Etätyöpöytäyhteyden web-sovellus on vuonna 2018 julkaistu sovellus, jonka avulla etätyöpöytäyhteyttä voi käyttää HTML5-yhteensopivalla selaimella. Käytännössä jokaiselta tietokoneelta löytyy HTML5-yhteensopiva selain, joten web-sovelluksen käyttämiseen ei tarvita lainkaan ylimääräisiä ohjelmia, liitännäisiä tai lisäosia. [9.]

Etätyöpöydän lisenssipalvelu on lisenssienhallintapalvelu, jonka avulla voidaan asentaa ja hallinnoida etätyöpöytäpalvelun vaatimia lisenssejä. Lisenssit voivat olla joko laite- tai käyttäjäkohtaisia. Laitekohtaisia lisenssejä tarvitaan yksi per jokainen etätyöpöytäyhteyttä käyttävä laite, ja käyttäjäkohtaisia lisenssejä yksi per yhteyttä käyttävä käyttäjä.

Etätyöpöytäympäristö tarvitsee lisensoida ainoastaan jommallakummalla tavalla, eli lisenssejä tarvitaan joko käyttäjille tai laitteille, ei molemmille. [10.] Remote Desktop Web Clientiä käytettäessä on kuitenkin käytettävä käyttäjäkohtaisia lisenssejä [2].

3.4 Kaksivaiheinen tunnistautuminen

Ympäristön vaatimuksena oli, että kirjautuminen etätyöpöytäpalvelimelle vaatii kaksivaiheisen tunnistautumisen. Kaksivaiheinen tunnistautuminen tarkoittaa sitä, että käyttäjä todennetaan käyttäjätunnuksen ja salasanan lisäksi jollakin muulla tavalla, kuten puhelimeen saapuvalla kertakäyttösalasanalla, mobiilivarmenteella tai token-pohjaisella ratkaisulla [11].

Kaksivaiheisen tunnistautumisen teknologiaksi valittiin aikaan perustuva kertakäyttöinen salasana (time-based one-time password protocol, TOTP). TOTP on muunnos HMACiin perustuvasta kertakäyttöisestä salasanasta (HMAC-based one-time password algorithm, HOTP) [12]. TOTP:ta käytettäessä käyttäjälle luodaan ensin uniikki salainen avain, jonka avulla voidaan luoda kertakäyttöisiä salasanoja. Käyttäjä lisää avaimen omaan tunnistautumissovellukseensa, usein lukemalla QR-koodin. Sovellus lukee avaimesta tarvittavat määrittelyt sille, miten uusia salasanoja luodaan ja miten usein luominen tapahtuu. Salaisen avaimen myöntänyt taho käyttää samaa algoritmia salasanojen luomiseen, ja pystyy näin todentamaan, että käyttäjän syöttämä kertakäyttöinen salasana on oikea. [13.]

TOTP-tuotteeksi valittiin ilmainen, avoimen lähdekoodin multiOTP, jonka on kehittänyt sveitsiläinen SysCo. Saman kehittäjän multiOTP Credential Provider -sovellusta käytettiin varsinaisen etätyöpöytäpalvelimen suojaamiseen. MultiOTP on PHP-pohjainen sovellus, jonka avulla voidaan luoda ja hallinnoida TOTP-avaimia. Käyttäjätiedot voidaan tuoda suoraan aktiivihakemistosta, mikä helpottaa ympäristön automatisointia. [14.]

4 Suunnittelu

Tehdyn määrittelyn perusteella suunniteltiin ympäristön rakenne eli se, kuinka monta palvelinta tarvitaan ja millaisia rooleja eri palvelimilla on. Koska turvallisuuden ohella yhtenä tärkeänä kriteerinä oli toteuttaa ympäristö mahdollisimman edullisesti, päätettiin yhdistellä rooleja siten, että yhdellä palvelimella oli useita eri rooleja. Asiakkaalta saadun käyttäjämääräarvion sekä ohjelmistolistan perusteella arvioitiin myös palvelinten tarvitsemat resurssit.

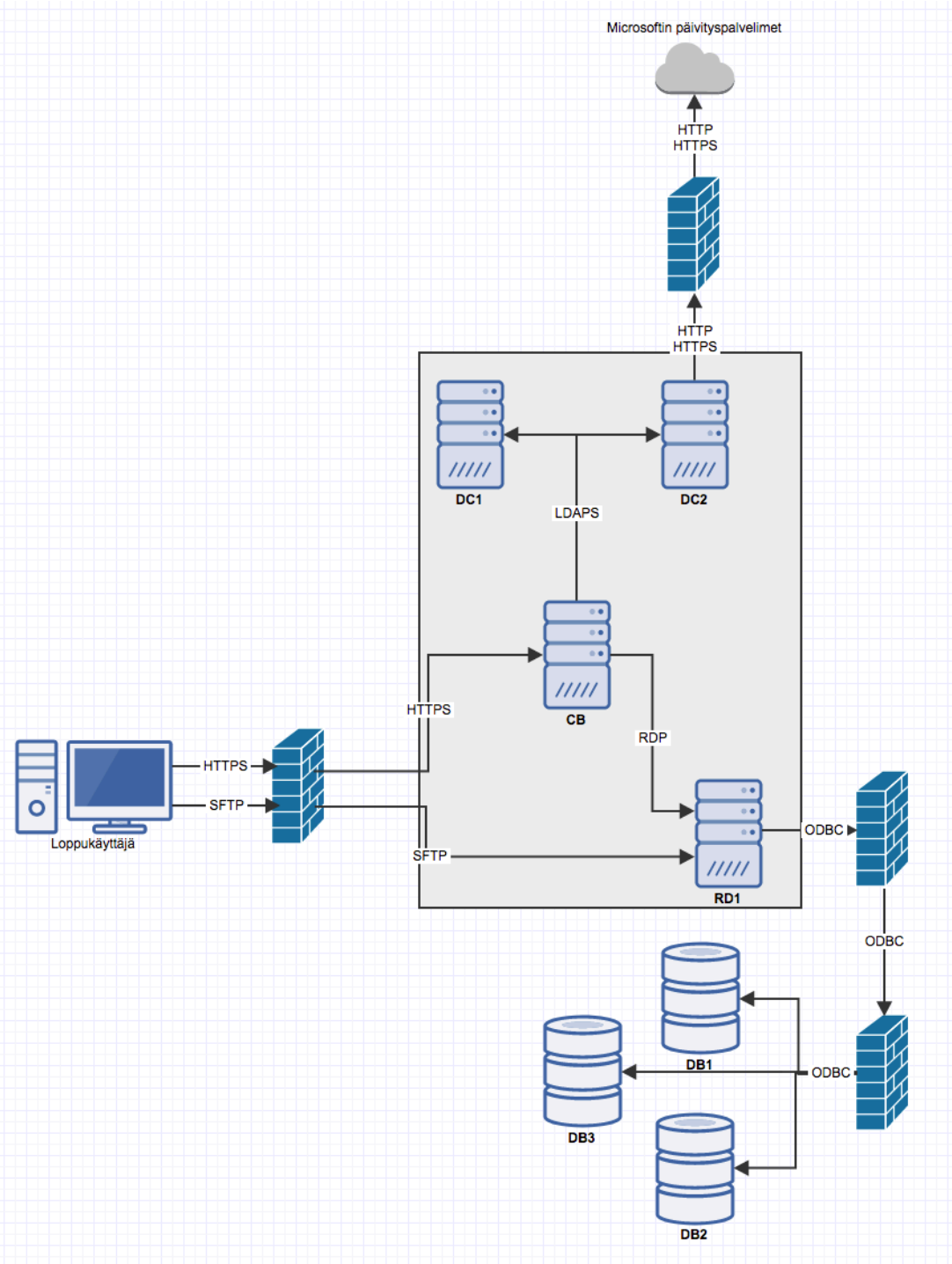
Suunnittelun lähtökohtana oli se, että ympäristöön tarvitaan vähintään neljä palvelinta: kaksi Domain Controller -palvelinta, yksi etätyöpöytäpalvelin sekä yksi palvelin, joka voi huolehtia kaikista muista rooleista. Palvelinten resurssit ja roolit on kuvattu taulukossa 1.

Taulukko 1. Palvelinten resurssit ja roolit.

Palvelimen nimi	**Virtuaaliprosessorien määrä (kappaletta)**	**Keskusmuistin määrä (gigatavua)**	**Tallennustilan määrä (gigatavua)**	**Roolit**
DC1	2	4	C: 100	Toimialueen ohjainpalvelin.
DC2	2	4	C: 100 D: 100	Toimialueen ohjainpalvelin, Windows päivityspalvelu
CB	2	4	C: 100	Etätyöpöytäyhteyden yhdyskäytävä, Etätyöpöytäyhteyden välittäjä, Etätyöpöydän web-käyttöliittymä, Etätyöpöydän lisensointi, kaksivaiheisen tunnistautumisen hallinoija.
RD1	2	4	C: 100	Etätyöpöytäistunnon isäntä.

CSC:llä suurin osa tietoverkkojen ylläpidosta on ulkoistettu erilliselle tietoverkkoryhmälle, joten projektissa ei tarvinnut juuri suunnitella tai rakentaa verkkoratkaisuja. Tarvittavien IP-osoitteiden määrä oli päätettävä etukäteen, ja kasvuvaran jättämiseksi päätettiin tilata verkko, jossa on käytettävissä 16 IP-osoitetta.

Osana suunnittelua laadittiin myös arkkitehtuurikuva, josta käy ilmi palvelinten tarvitsemat verkkoyhteydet suhteessa toisiinsa ja Internetiin. Arkkitehtuurikuva on esitetty kuvassa 1



Kuva 1. Ympäristön arkkitehtuuri.

Arkkitehtuurikuva on hieman pelkistetty. Esimerkiksi kaikkia aktiivihakemistoon liittyviä yhteyksiä ei ole esitetty, sillä kuvan ei ole tarkoitus olla täydellinen kuvaus verkkoympäristöstä, vaan lähinnä auttaa hahmottamaan sitä, miten järjestelmän pääasialliset komponentit toimivat. Kuvasta käy esimerkiksi ilmi se, että loppukäyttäjän työasemalta ympäristöön tulee ainoastaan HTTPS- ja SFTP-liikennettä, ja ainoa ulospäin sallittu liikenne kulkee DC2-palvelimelta Microsoftin palvelimille. Esimerkiksi tietokantapalvelimet on arkkitehtuurin mukaan eristetty Internetistä.

5 Toteutus

5.1 Aktiivihakemiston asennus ja konfigurointi

Ympäristön asennus aloitettiin konfiguroimalla kaksi palvelinta ohjainpalvelimiksi asentamalla niihin Active Directory Domain Services -palvelu ja sen hallintatyökalut siten, että ensin konfiguroitiin ensisijainen ohjainpalvelin DC1. Palvelu asennettiin PowerShell-komentosarjalla esimerkkikoodin 1 mukaisesti.

```
Install-WindowsFeature AD-Domain-Services -IncludeManagementTools
Import-Module ADDSDeployment
$ADDSForestParams = @{
      CreateDnsDelegation = $False
      DatabasePath = "C:\Windows\NTDS"
      DomainMode = "WinThreshold"
      DomainName = $DomainName
      DomainNetBiosName = $DomainBios
      ForestMode = "WinThreshold"
      InstallDns = $True
      LogPath = "C:\Windows\NTDS"
      NoRebootOnCompletion = $False
      SysvolPath = "C:\Windows\SYSVOL"
      Force = $True
      }
Install-ADDSForest @ADDSForestParams
```

Esimerkkikoodi 1. Active Directory Domain Servicesin asentaminen.

Koska palvelinten täytyi kyetä kommunikoimaan myös muissa verkoissa sijaitsevien palvelinten kanssa, asetettiin aktiivihakemiston nimipalvelu ohjaamaan nimikyselyt eteenpäin Funetin nimipalvelimille esimerkkikoodin 2 mukaisesti.

```
Add-DnsServerForwarder -IPAddress 193.166.4.24 -PassThru
Add-DnsServerForwarder -IPAddress 193.166.4.25 -PassThru
```

Esimerkkikoodi 2. Nimikyselyiden ohjaaminen Funetin nimipalvelimille.

Lopuksi asetettiin vielä vanhentuneiden nimitallenteiden automaattinen poistaminen, luotiin käänteinen nimialue ja sallittiin aluesiirtojen hyväksyntä ainoastaan ennalta määrätyiltä nimipalvelimilta. Tämä on kuvattu esimerkkikoodissa 3.

```
Set-DnsServerScavenging –ScavengingState $True –ScavengingInterval 7:00:00:00
-RefreshInterval 1.00:00:00 –ApplyOnAllZones Set-DnsServerZoneAging
linnea.local -Aging $True
Add-DnsServerPrimaryZone -DynamicUpdate Secure -NetworkId '$CIDR' -
ReplicationScope Domain
Set-DnsServerPrimaryZone -Name linnea.local -notify notify -SecondaryServers
$DC2IP -SecureSecondaries TransferToSecureServers
```

Esimerkkikoodi 3. Nimipalvelun lisäasetukset.

Esimerkkikoodin 4 mukaisesti liitettiin toinen ohjainpalvelin DC2 toimialueeseen ja ylennettiin ohjainpalvelimeksi:

```
Install-WindowsFeature AD-Domain-Services -IncludeManagementTools
$ADDSDomainControllerParams = @{
      NoGlobalCatalog = $False
      CreateDnsDelegation = $False
      Credential = $(Get-Credential)
      CriticalReplictationOnly = $False
      DatabasePath = "C:\Windows\NTDS"
      DomainName = $Domainname
      InstallDns = $True
      LogPath = "C:\Windows\NTDS"
      NoRebootOnCompletion = $False
      SiteName = "Default-First-Site-Name"
      SysvolPath = "C:\Windows\SYSVOL"
      Force = $True
      }
Install-ADDSDomainController @ADDSDomainControllerParams
```

Esimerkkikoodi 4. Toisen ohjainpalvelimen liittäminen toimialueeseen ja ylentäminen ohjainpalvelimeksi

Nimipalvelu määritettiin samoilla asetuksilla kuin DC1:lläkin. Windowsin päivityspalvelu WSUS asennettiin DC2-palvelimelle (liite 1).

5.2 IdM-käyttäjänhallinta ja tutkijan käyttöliittymä

Eräs projektimäärittelyn keskeisimpiä asioita oli käyttäjätunnusten hallinta. Kaksi tärkeää kriteeriä olivat se, että tunnuksen luominen ja salasanan asettaminen on käyttäjän kan-

nalta mahdollisimman vaivatonta, ja se, että tunnukset ovat yhdessä keskitetyssä paikassa, jotta käyttäjien mahdollisesti tekemiä tietopyyntöjä varten voidaan kerätä kaikki käyttäjätiedot mahdollisimman helposti.

CSC:llä on olemassa olevana palveluna Identity Management -palvelu (IdM), johon voidaan liittää Lightweight Directory Access Protocol (LDAP) -pohjaisia käyttäjänhallintapalveluita, kuten FreeIPA ja Active Directory. LDAP-pohjaisen käyttäjänhallintapalvelun rinnalle tarvitaan vain sopiva ajuri sekä palvelutunnus, jolla on riittävät oikeudet tunnusten luomiseen, minkä jälkeen käyttäjätunnuksia voidaan synkronoida suoraan CSC:n keskitetystä IdM-palvelusta. IdM:n ja LDAPin välinen yhteys ei sinänsä ota kantaa siihen, miten ja mistä tunnukset päätyvät alun perin IdM:ään. IdM:n käyttäminen ratkoi vaatimuksen käyttäjänhallinnan keskittämisestä, sillä tunnuksen lukitseminen, salasanan vaihtaminen tai poistaminen IdM:ssä johtaa automaattisesti samaan toimenpiteeseen myös kaikissa siihen liitetyissä palveluissa.

Käyttäjiä varten tarvittiin käyttöliittymä, jonka kautta voidaan rekisteröityä palveluun sekä asettaa käyttäjätunnuksen salasana. Tätä tarkoitusta varten päätettiin käyttää CSC:n tutkijan käyttöliittymää (Scientist's User Interface, SUI). Tutkijan käyttöliittymää käytetään muun muassa CSC:n supertietokoneiden ja Pouta-pilviympäristöjen käyttäjänhallintaan. Tutkijan käyttöliittymä toimintaperiaatteena on, että käyttäjät kuuluvat projekteihin, joiden perusteella he saavat käyttöoikeuksia ja resursseja palveluihin. Kullakin projektilla on vastuuhenkilö, joka voi lisätä ja poistaa projektin jäseniä. Tutkijan käyttöliittymään rekisteröityessään käyttäjälle luodaan tunnus IdM:ään, ja projektiin liittyessään hänen tunnukseensa liitetään asianmukainen palveluprofiili. Tässä projektissa luotiin LinneaRD-niminen projekti, jonka vastuuhenkilönä toimii palvelun päävastuullinen ylläpitäjä.

Koska projektissa käytettiin Active Directorya, tarvittiin IdM:ää varten Active Directory -ajuri. Active Directory -ajuri toimii Domain Controller -palvelimella agenttina kuunnellen ennalta määriteltyyn porttiin tulevaa Transport Layer Security (TLS) -salattua liikennettä. Ennen kuin ajuri voitiin asentaa, täytyi liikenne CSC:n IdM-palvelimelta sallia ympäristön palomuureilla sekä asettaa ympäristön ryhmäkäytännön salasanapolitiikka vastaamaan IdM:n salasanapolitiikkaa. Tämä oli erityisen tärkeää, sillä mikäli esimerkiksi Active Directoryn salasanapolitiikka määrittelisi salasanan vanhentumaan nopeammin kuin IdM, aiheuttaisi tämä tilanteen, jossa käyttäjän salasana vanhentuu, mutta hän ei saa siitä

minkäänlaista ilmoitusta. Lisäksi salasana toimisi edelleen tutkijan käyttöliittymässä, mutta ei etäyhteyttä käytettäessä. Viimeisenä valmistelutoimenpiteenä luotiin vielä tarvittavat organisaatioyksiköt Active Directoryyn sekä ajurin käyttämä palvelutunnus esimerkkikoodin 5 mukaisesti.

```
New-ADOrganizationalUnit -Name "Service_Accounts" -Path $BasePath

$Password = Read-Host -AsSecureString
$NewUserParams = @{
    GivenName = "$AccountName"
    Name = "$AccountName"
    Displayname = "$AccountName"
    Description = "IdM sync account"
    SamAccountName = "$AccountName"
    UserPrincipalName = "$AccountName@$Domainname"
    AccountPassword = $Password
    Path = $SAPath
    }

New-ADUser @NewUserParams
Enable-ADAccount -Identity IDMRL

New-ADOrganizationalUnit -Name "IDM-Users" -Path $BasePath
```

Esimerkkikoodi 5. Palvelutunnuksen luominen aktiivihakemistoon.

Lopuksi delegoitiin palvelutunnukselle riittävät oikeudet organisaatioyksikköön, johon IdM:stä synkronoitavat käyttäjätunnukset luodaan. Tämän jälkeen asennettiin komponentit, jotka muodostavat Active Directory -ajurin ja asetettiin tarvittavat salasanat ja varmenteet.

5.3 Tukipalveluiden asennus

Esimerkkikoodissa 6 on kuvattu, miten kaikki tukipalvelut asennettiin tukipalvelimelle CB.

```
Add-WindowsFeature Remote-Desktop-Services,RDS-Connection-Broker,RDS-
Licensing,RDS-RD-Server,RSAT-RDS-Tools,RDS-GATEWAY,RDS-Licensing-UI -
IncludeManagementTools
```

Esimerkkikoodi 6. Tukipalveluiden asentaminen.

Etätyöpöytäympäristön konfigurointi aloitettiin luomalla uusi Session Deployment, kuten esimerkkikoodissa 7 on esitetty.

```
Import-Module RemoteDesktop
New-RDSessionDeployment -ConnectionBroker $ConnectionBroker -WebAccessServer
$ConnectionBroker -SessionHost $SessionHost
```

Esimerkkikoodi 7. Uuden Session Deploymentin luominen.

Yhdyskäytäväpalvelu yhdistettiin välittäjäpalveluun ja määriteltiin verkko-osoite, jossa palvelua tarjotaan sekä turvallisuuteen liittyviä asetuksia esimerkkikoodin 8 mukaisesti.

```
Add-RDServer -Server $ConnectionBroker -Role RDS-GATEWAY -ConnectionBroker
$ConnectionBroker -GatewayExternalFqdn $GWFQDN
Set-RDDeploymentGatewayConfiguration -GatewayMode Custom -BypassLocal $False -
ConnectionBroker $ConnectionBroker -GatewayExternalFqdn $GWFQDN -LogonMethod
Password -UseCachedCredentials $True
```

Esimerkkikoodi 8. Session Deploymentin lisäasetukset

Kaikkia yhdyskäytävän asetuksia ei ollut mahdollista määritellä PowerShellillä, joten käyttöliittymän kautta määriteltiin etätyöpöytäyhteyden automaattinen aikakatkaisu ja estettiin laitteiden ja leikepöydän uudelleenohjaus.

Seuraavaksi luotiin uusi etätyöpöytäklusteri, määriteltiin yhdyskäytävän aikakatkaisu- ja uudelleenohjausasetuksia vastaavat asetukset ja lisättiin RD1-palvelin etätyöpöytäyhteyden isäntäpalvelimeksi. Etätyöpöytäklusterin luominen ja asetusten määrittäminen on kuvattu esimerkkikoodissa 9.

```
Set-RDWorkspace -Name $WorkspaceName
New-RDSessionCollection -CollectionName $CollectionName -SessionHost
$SessionHost -CollectionDescription $Description -ConnectionBroker
$ConnectionBroker
Set-RDSessionCollectionConfiguration -CollectionName $CollectionName -
ClientDeviceRedirectionOptions None -ClientPrinterRedirected $false -UserGroup
$UserGroup -ConnectionBroker $ConnectionBroker
```

Esimerkkikoodi 9. Etätyöpöytäklusterin luominen ja konfigurointi.

Web-sovellus asennettiin NuGet-paketinhallinnan avulla, kuten esimerkkikoodista 10 käy ilmi.

```
$PackageProviderParams = @{
      Name = "NuGet"
      MinimumVersion = 2.8.5.201
      Force = $True
      }
Install-PackageProvider @PackageProviderParams
Install-Module -Name PowerShellGet -Force
```

```
# Restart Powershell
Install-Module -Name RDWebClientManagement
Install-RDWebClientPackage
Import-RDWebClientBrokerCert $CertificatePath
Publish-RDWebClientPackage -Type Production -Latest
```

Esimerkkikoodi 10. Web-sovelluksen asentaminen NuGet-paketinhallinnan avulla.

Lopuksi asennettiin vielä lisenssipalvelu esimerkkikoodin 11 mukaisesti.

```
Add-RDServer -Server $ConnectionBroker -Role RDS-LICENSING
```

Esimerkkikoodi 11. Lisenssipalvelun asentaminen.

Alihankkijalta ostetut lisenssit asennettiin lisenssipalvelun käyttöliittymän kautta syöttämällä aktivointikoodi palvelulle.

5.4 MultiOTP:n asennus

MultiOTP asennettiin tukipalvelimelle CB purkamalla ZIP-paketti haluttuun hakemistoon ja suorittamalla asennuskomentosarja webservice_install.bat järjestelmänvalvojan oikeuksilla. Tämä asensi ja käynnisti web-palvelun, joka toimii sovelluksen hallintapaneelina. Ennen muiden asetusten tekemistä vaihdettiin pääkäyttäjän salasana hallintapaneelin kautta. Lopuksi asetettiin multiOTP hakemaan käyttäjätiedot aktiivihakemistosta, kuten esimerkkikoodissa 12 on kuvattu.

```
multiotp -config default-request-prefix-pin=0
multiotp -config default-request-ldap-pwd=1
multiotp -config ldap-server-type=1
multiotp -config ldap-cn-identifier="sAMAccountName"
multiotp -config ldap-group-cn-identifier="sAMAccountName"
multiotp -config ldap-group-attribute="memberOf"
multiotp -config ldap-ssl=1
multiotp -config ldap-port=636
multiotp -config ldap-domain-controllers=$DC1, $DC2
multiotp -config ldap-base-dn=$BaseDN
multiotp -config ldap-bind-dn=$BindDN
multiotp -config ldap-server-password=$Password
multiotp -config ldap-in-group="OTP-Users"
multiotp -config ldap-network-timeout=10
multiotp -config ldap-activated=1
multiotp -debug -display-log -ldap-users-sync
```

Esimerkkikoodi 12. MultiOTP:n konfigurointi.

Asetuksilla määriteltiin aktiivihakemiston tyyppi ja osoite, käyttäjätunnuksen sisältävä attribuutti, käyttäjäryhmä, jonka jäsenet haetaan, sekä tunnus, jolla käyttäjäkysely tehdään.

5.5 Etätyöpöytäpalvelimen asennus

Etätyöpöytäpalvelin RD1 liitettiin toimialueeseen PowerShell-komentosarjalla samalla tavalla kuin tukipalvelin, kuten esimerkkikoodista 13 käy ilmi.

```
Set-DnsClientServerAddress -InterfaceAlias $Interface -ServerAddresses
$ADDNSAddresses
Add-Computer -DomainName $Domainname -Credential $(get-credential)
Restart-Computer
```

Esimerkkikoodi 13. Palvelimen liittäminen toimialueeseen.

Lisäksi poistettiin ylimääräinen optisen aseman kirjain esimerkkikoodin 14 mukaisesti.

```
$drive = gwmi win32_volume -Filter "DriveLetter = 'd:'"
Set-WmiInstance -input $drive -Arguments
@{DriveLetter=$null;Label=$newReplayName}
```

Esimerkkikoodi 14. Optisen aseman kirjaimen poistaminen.

Kaksivaiheista tunnistautumista varten asennettiin multiOTP Credential Provider -sovellus, jonka asetuksiin määriteltiin multiOTP-palvelimen osoite, käyttäjätunnuksen lähettäminen UPN-muodossa sekä se, että ainoastaan etätyöpöytäyhteys suojataan kaksivaiheisella tunnistautumisella. Viimeksi mainittu oli erityisen tärkeää, jotta mahdollisessa multiOTP-palvelun ongelmatilanteessa ylläpitäjä voisi yhä kirjautua palvelimelle virtuaalikonsolin kautta. Vaikka tämän kirjautumistavan suojaamatta jättäminen tekee palvelimesta periaatteesta turvattomamman, arvioitiin riski kuitenkin hyvin pieneksi, koska virtuaalikonsoliin pääsy on erittäin rajattu, eivätkä esimerkiksi loppukäyttäjät kirjaudu palvelimelle sen kautta.

Palvelimelle asennettiin lisäksi asiakkaan toivomia sovelluksia, kuten Microsoftin Office-ohjelmistot ja erilaisia tietokantayhteyteen liittyviä ohjelmia. Tiedostojen siirtoa varten asennettiin SFTP-palvelu käyttäen OpenSSH-sovellusta. OpenSSH-asennettiin suoritta-

malla asennuskomentosarja install-sshd.ps1 järjestelmänvalvojan oikeuksilla. Windowsin palomuuriin lisättiin sääntö, joka sallii SFTP-yhteydet palvelimelle. Palomuurisäännön lisäys on kuvattu esimerkkikoodissa 15.

```
$FirewallRuleParams = @{
      Name = "sshd"
      DisplayName = "OpenSSH SSH Server"
      Enabled = $True
      Direction = "Inbound"
      Protocol = "TCP"
      Action = "Allow"
      LocalPort = 22
New-NetFirewallRule @FirewallRuleParams
```

Esimerkkikoodi 15. Uuden palomuurisäännön lisääminen.

OpenSSH:n asetuksiin määriteltiin, että käyttäjät voivat selata ainoastaan SFTP-hakemiston alla olevia tiedostoja. Lopuksi asetettiin OpenSSH-palvelu käynnistymään automaattisesti Windowsin palveluidenhallinnasta.

6 Testaus ja käyttöönotto

Etätyöpöydän testaaminen jaettiin kolmeen vaiheeseen, jotka olivat projektiryhmän sisäinen testaus, ennalta valitun, edistyneen käyttäjäryhmän testaus ja lopuksi laajamittainen testaus, johon otettiin mukaan käytännössä tuotantovaiheen loppukäyttäjät. Jokaiselle testauksen vaiheelle oli määritelty asiat, joihin kyseinen testaus keskittyy.

6.1 Projektiryhmän sisäinen testaus

Projektiryhmän sisäisessä testauksessa oli mukana etätyöpöytäjärjestelmän kehittäjä sekä kaksi henkilöä, jotka tulisivat myöhemmin ylläpitämään järjestelmää. Sisäisen testauksen pääasiallisena tavoitteena oli varmistaa, että etätyöpöydän perusosat toimivat odotetulla tavalla ja että erityistä teknistä osaamista vaativat loppukäyttäjän toimenpiteet saadaan minimoitua.

Sisäisessä testauksessa havaittiin muutamia kehityskohteita liittyen lähinnä etätyöpöytäpalvelimen kovettamiseen ryhmäkäytäntöä muokkaamalla. Itse etätyöpöydän toimivuus oli heti odotetulla tasolla, joten testauksen seuraavaan vaiheeseen voitiin siirtyä varsin nopeasti.

6.2 Rajatun käyttäjäryhmän testaus

Toisessa testivaiheessa testaajiksi kutsuttiin asiakkaan määrittelemä joukko edistyneempiä käyttäjiä, joilla oli paitsi kyky havaita mahdollisia puutteita etäpöydän käytettävyydessä myös vaadittava taito todeta tietokantayhteyden toimivan asianmukaisesti. Testiryhmää käytettiin myös apuna käyttöohjeistuksen tekemisessä siten, että aluksi heille annettiin varsin vähäinen ohjeistus etätyöpöydän käyttöön, ja saadun palautteen avulla täydennettiin ohjeistusta.

Testeissä havaittiin luonnollisesti paljon kehitettävää ohjeistuksen suhteen, mutta myös muutamia käytännön asioita. Etätyöpöydän kehittämisvaiheessa ja projektiryhmän sisäisessä testauksessa ei ollut onnistuttu määrittelemään kyllin tarkasti todellisten loppukäyttäjien työskentelytapoja liittyen esimerkiksi tiedostojen siirtämiseen etätyöpöydän ja loppukäyttäjän oman tietokoneen välillä sekä siinä, miten loppukäyttäjät kirjautuvat sisään ja ulos etätyöpöydältä. Testiryhmältä tuli myös toiveita alkuperäisen käyttötarkoituksen laajentamisesta erilaisten varus- ja apuohjelmien avulla sekä sallimalla vapaamman tiedostonsiirron etätyöpöydän ja loppukäyttäjän oman tietokoneen välillä.

6.3 Laajamittainen testaus

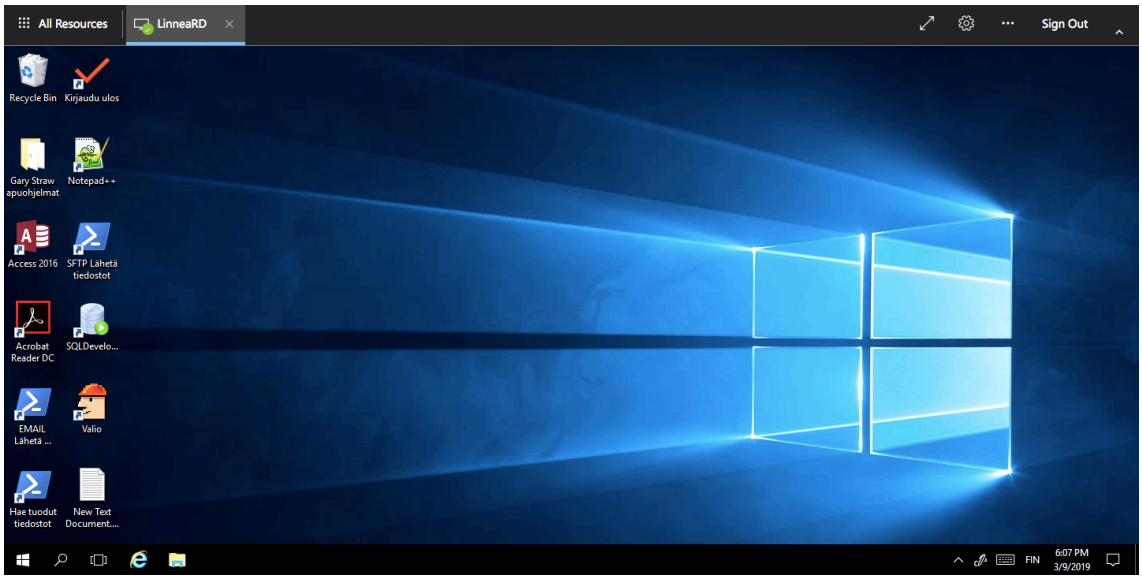
Testauksen viimeiseen vaiheeseen päästessä etätyöpöytään liittyvät tekniset asiat, ohjeistus sekä käyttötarkoituksen laajentamiseen liittyvä palaute oli pääosin ratkaistu. Viimeisen testivaiheen tarkoitukseksi jäi näin lähinnä varmistaa, että aiemmin arvioitu resurssien määrä vastasi todellista tarvetta. Uusia testikäyttäjiä kutsuttiin hallitusti noin kymmenen käyttäjän erissä seuraten samalla etätyöpöydän kuormitusta asennettujen monitorointityökalujen avulla. Koska käyttäjämäärä oli kasvanut huomattavasti siitä, mitä asiakkaan kanssa oli alun perin määritelty, ei määritelty resurssien määrä ollut enää riit-

tävä ja käyttäjiltä tulikin jonkin verran palautetta etätyöpöydän hidastelusta. Samoin saatiin muutamia yksittäisiä palautteita liittyen ohjeistukseen, tietoturvaan sekä ongelmiin tiettyjen erikoismerkkien syöttämisessä.

6.4 Palautteeseen reagointi

Testaajilta saatuun palautteeseen reagoitiin pääsääntöisesti nopeasti. Koska etätyöpöytä oli vielä testivaiheessa, pystyttiin myös muutoksia ja korjauksia tekemään lyhyelläkin varoitusajalla. Osa kehitysehdotuksista liittyi olennaisesti etätyöpöydän käyttötarkoitukseen ja tietoturvaan, joten niistä keskusteltiin asiakkaan kanssa ennen ryhtymistä toimenpiteisiin.

Yksinkertaisimmasta päästä oli esimerkiksi palaute siitä, että kirjautuminen etätyöpöydältä ulos oli vaikeaa. Monet käyttäjät etsivät uloskirjautumisvaihtoehtoa joko sivun oikeassa ylälaidassa sijaitsevasta uloskirjautumispainikkeesta (kuva 2), joka on osa etätyöpöytäyhteyden tarjoavaa web-sovellusta eikä itse etätyöpöytää. Kyseinen painike ei kirjaa käyttäjää ulos etätyöpöydältä, vaan ainoastaan web-sovelluksesta, jolloin etäistunto jää voimaan eivätkä käynnissä olevat ohjelmat sulkeudu.



Kuva 2. Web-sovelluksen uloskirjautumispainike näkyy kuvan oikeassa yläreunassa

Käyttäjiä ohjeistettiin kirjautumaan ulos Käynnistä-valikon kautta, minkä jälkeen monet yrittivät kirjautua ulos virtapainikkeen takaa löytyvän Disconnect-painikkeen kautta. Etätyöpöytäistunnon kannalta vaikutus oli sama kuin web-sovelluksen uloskirjautumisen kanssa, eli istuntoa ei kirjattu ulos ja ohjelmat jäivät käyntiin. Käyttäjiä ohjeistettiin vielä kirjautumaan käyttäjäkuvake-siluetin takaa löytyvän Sign Out -painikkeen kautta, mutta lopulta todettiin helpommaksi ratkaisuksi luoda jokaisen käyttäjän työpöydälle pikakuvake, jonka avulla käyttäjät onnistuivat kirjautumaan ulos. Pikakuvake luotiin esimerkkikoodin 16 mukaisesti.

```
$WshShell = New-Object -ComObject WScript.Shell
$Shortcut = $WshShell.CreateShortcut("C:\Users\Public\Desktop\Logoff.lnk")
$Shortcut.TargetPath = "C:\Windows\System32\logoff.exe"
$Shortcut.Description = "Kirjaudu ulos"
$Shortcut.IconLocation = "C:\Windows\System32\shell32.dll,144"
$Shortcut.Save()
```

Esimerkkikoodi 16. Pikakuvakkeen luominen.

Käyttötarkoituksen laajentamiseen liittyvät kehitysehdotukset olivat hieman vaikeampia, sillä vaikka palvelua vielä kehitettiin, oli kehittäminen tarkoitus tehdä aiemmin tehdyn vaatimusmäärittelyn ja suunnitelman pohjalta. Kehitysehdotuksista keskusteltiin asiakkaan kanssa, ja lopulta päädyttiin asentamaan tiettyjä varus- ja apuohjelmia, joita käyttäjät olivat pyytäneet. Koska käyttötarkoitus muuttui, syntyi myös tarve muuttaa tiedostojensiirron periaate täysin. Alkuperäisen määrittelyn mukaan etätyöpöydältä oli mahdollista siirtää kooltaan ja sisällöltään sopivia tiedostoja ulos sähköpostin välityksellä. Käyttötarkoituksen laajennuttua tarvittiin kuitenkin tietoturvallisempi tapa siirtää tiedostoja ulos, sekä mahdollisuus tuoda etätyöpöydälle omia tiedostoja. Tarpeeseen vastattiin asentamalla etätyöpöydälle SFTP-palvelu, jonka toteuttaminen on kuvailtu luvussa 4.6.

6.5 Käyttöönotto

Kun testeissä esiin nousseet kehitystarpeet oli täytetty, siirryttiin palvelun käyttöönottoon. Käyttöönottovaihe tarkoitti käytännössä sitä, että asiakas toimitti listan käyttäjistä, joille myönnetään pääsy palveluun. Palvelun ylläpitäjä puolestaan salli palomuurista käyttäjän ilmoittaman IP-osoitteen ja lähetti käyttäjälle kutsun, jonka avulla käyttäjän oli mahdollista luoda itselleen käyttäjätunnus tutkijan käyttöliittymässä ja liittää tunnuksensa

palveluun. Ylläpitäjä huolehti myös käyttäjien ohjeistamisesta. Prosessin selkeyttämiseksi laadittiin taulukossa 2 kuvattu työvuo, josta ilmenee myös kunkin toimenpiteen suorittava taho.

Taulukko 2. Kuvaus uuden käyttäjän pääsystä palveluun.

Asiakas	**Palvelun ylläpitäjä**	**Käyttäjä**
		Pyytää pääsyä palveluun Asiakkaalta.
Hyväksyy Käyttäjän ja lähettää tarvittavat tiedot Palvelun ylläpitäjälle.		
	Avaa pääsyn palomuurista.	
	Lähettää Käyttäjälle ohjeistuksen.	
	Kutsuu Käyttäjän palveluun Tutkijan käyttöliittymässä.	
		Hyväksyy kutsun ja palvelun käyttöehdot.
	Lähettää Käyttäjälle ohjeistuksen kaksivaiheista tunnistautumista varten.	
		Kirjautuu palveluun ja aloittaa käytön.

7 Ylläpito

Osana projektia suunniteltiin ja toteutettiin palvelun ylläpitoon tarvittavat menetelmät, työkalut ja sopimukset. Näihin lukeutuivat sopiminen palvelutasosta, tukipyyntökäytännöistä ja huoltokatkoista, palvelinten monitorointi, huoltokatkojen automatisointi sekä ylläpito-ohjeiden dokumentointi.

7.1 Asiakkaan kanssa sovitut asiat

Asiakkaan kanssa sovittiin palvelutasosta, tukipyyntöjen käsittelystä ja huoltokatkojen ajankohdasta. Palvelutaso koostuu palveluajasta ja reagointiajasta. Palveluaika tarkoittaa aikoja, joiden puitteissa palvelun ylläpito on tavoitettavissa. Reagointiaika tarkoittaa aikaa, jonka kuluessa palvelun ylläpito viimeistään aloittaa vian tai palvelupyynnön edellyttämät selvitys- tai korjaustoimenpiteet. Palveluajaksi määriteltiin arkipäivät kello 8:30 - 16:00. Reagointiajat määriteltiin kiireellisyyden perusteella taulukon 3 mukaisesti.

Taulukko 3. Palvelutason mukaiset reagointiajat.

Kiireellisyysluokitus	Tilannekuvaus	Reagointiaika
Korkein	Palvelunkäytön kokonaan estävä ongelma tai laajaa asiakaskuntaa merkittävästi haittaava ongelma	4 tuntia
Normaali	Muu asiakkaan palvelupyyntö tai loppukäyttäjälle näkymätön, ei-kriittinen ongelma	2 työpäivää
Alhainen	Loppukäyttäjälle näkymätön vähäinen ongelma tai muutostyö	5 työpäivää

Palvelutason osana ei määritelty ratkaisuaikaa, mikä tarkoittaa sitä, että ylläpito on niin sanotusti ”best effort”. Huoltokatkojen ajankohdaksi sovittiin jokaisen kuukauden viimeisen keskiviikon aamu. Käytännössä automatisoinnin ansiosta huoltokatkon manuaalisiksi toimenpiteiksi jää ainoastaan varmistaa palvelun toimivuus, jolloin huoltokatko kyettiin ajastamaan normaalien toimistoaikojen ulkopuolelle ja näin välttää loppukäyttäjille näkyviä häiriöitä palvelun saatavuudessa.

7.2 Palvelinten monitorointi

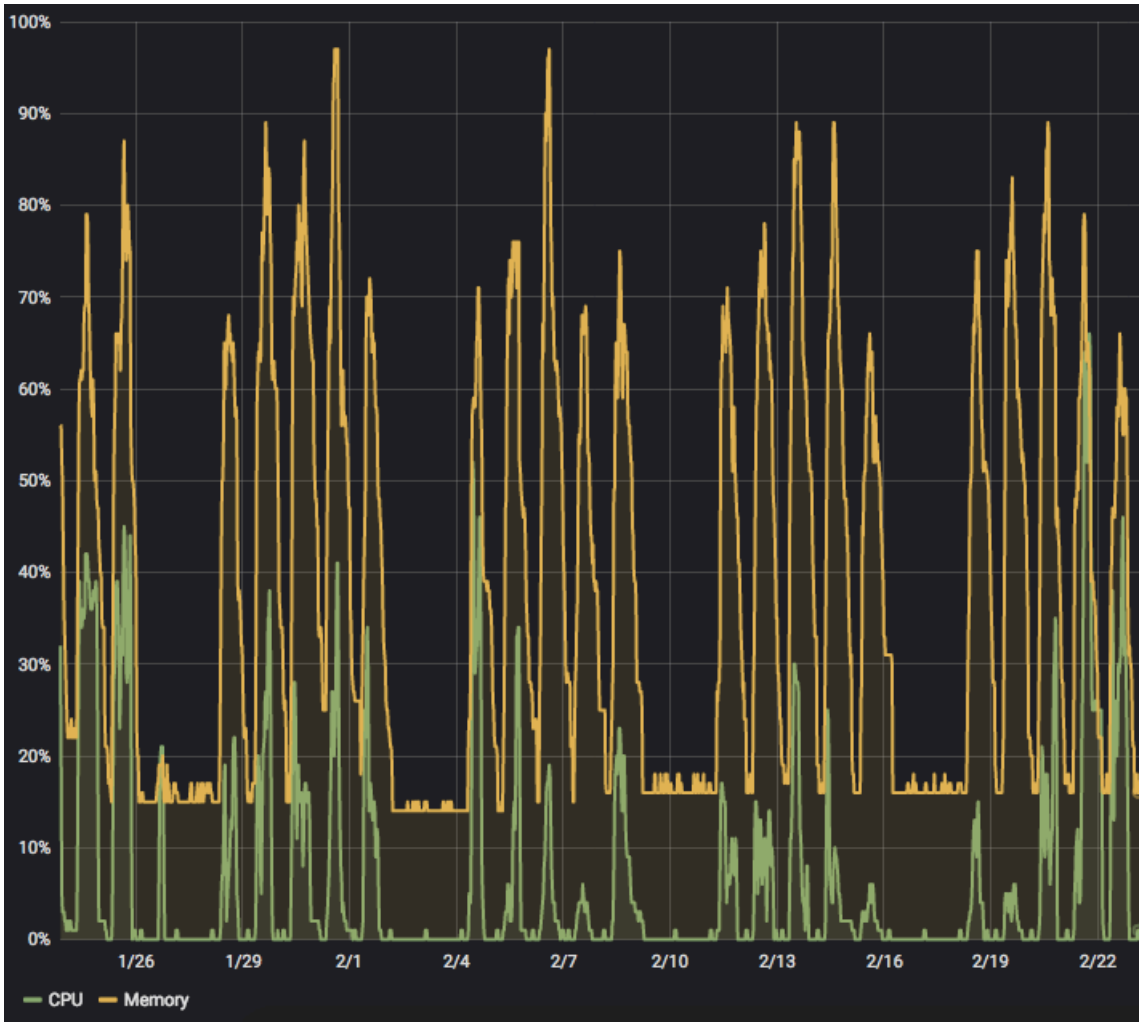
Palvelinten monitorointi on tärkeä osa ylläpitoa, sillä sen avulla saadaan tärkeää tietoa mahdollisista ongelmatilanteista jo etukäteen. Lisäksi monitorointidatan avulla voidaan tehdä päätelmiä esimerkiksi palvelun käyttöasteesta ja resurssien riittävyydestä, mikä puolestaan on tärkeää tietoa niin ylläpidolle kuin asiakkaallekin.

Palvelinten monitorointiin käytettiin CSC:n yleistä Nagiokseen perustuvaa monitorointipalvelua. Palvelimilla monitoroitiin seuraavia asioita

- saavutettavuus verkon yli
- prosessorin kuormitusaste
- monitorointiagentin tila
- sivutustiedoston käyttöaste
- keskusmuistin käyttöaste
- tallennustilan käyttöaste.

Lisäksi tukipalvelimella CB monitoroitiin HTTPS-palvelun saavutettavuutta ja TLS-varmenteen vanhentumista.

Monitoroinnin visualisointiin käytettiin Grafana-työkalua, jonka avulla voidaan piirtää monitorointidataan perustuvia kuvaajia. Nagios tuottaa dataa viiden minuutin välein ja tallentaa sen pitkäkestoisesti tietokantaansa, joten Grafanan avulla voidaan tutkia erilaisia trendejä pitkänkin ajanjakson yli. Kuvassa 3 on esimerkki datan visualisoinnista Grafanan avulla.



Kuva 3. Etätyöpöytäpalvelimen prosessorin ja muistin kuormitus kuukauden ajalta.

Visualisoituja suorituskykytietoja on käytetty hyväksi esimerkiksi ongelmatilanteiden selvittämisissä. Niitä on myös esitelty asiakkaan kanssa pidetyissä laatupalavereissa. Saadusta datasta on lisäksi tehty tulkintoja resurssien riittävyydestä, ja dataan perustuen on kerran toteutettu resurssien lisääminen etätyöpöytäpalvelimelle.

7.3 Huoltokatkot

CSC:llä palvelutuotannon peruskiviä on säännöllisten huoltokatkojen pitäminen. Huoltokatkojen avulla varmistetaan, että palvelu pysyy turvallisena ja käyttökelpoisena. Palvelun huoltokatkorutiini sisältää seuraavat toimenpiteet

- Monitoroinnin asettaminen huoltotilaan turhien hälytysten välttämiseksi.
- Käyttöjärjestelmän päivitysten asentaminen.
- Päivitysten viimeistely käynnistämällä palvelimet uudelleen.
- Mahdolliset muutokset järjestelmään, mikäli niitä ei voida tehdä ilman käyttöhäiriöitä.
- Palvelun toimivuuden varmistaminen muutosten, päivitysten ja uudelleenkäynnistysten jälkeen.

Koska palvelun vaatimuksiin ei sisältynyt korkea saatavuus, tarkoittaa huoltokatko samalla koko palvelua koskettavaa käyttökatkoa, jonka aikana käyttäjät eivät voi käyttää palvelua. Tästä syystä asiakkaan intressissä oli mahdollisimman nopea huoltokatko, mihin päästään parhaiten automatisoimalla huoltokatkoon sisältyviä toimenpiteitä.

Ympäristöön sisältyvän Windowsin päivityspalvelun ja ryhmäkäytännön avulla automatisoitiin päivitysten hyväksyminen, lataaminen, asentaminen ja palvelinten uudelleenkäynnistys. Windowsin päivityspalveluun asetettiin sääntö, joka hyväksyy automaattisesti Windows Server 2016 -käyttöjärjestelmän päivitykset, ja asetettiin sääntö epäaktiiviseksi. Säännön aktiiviseksi asettamista varten laadittiin PowerShell-komentosarja, jonka suorittaminen automatisoitiin Windowsin Tehtävien ajoitus -työkalun avulla tapahtumaan huoltokatkoa edeltävänä päivänä kello 21:00. Vastaavasti säännön epäaktiiviseksi asettava komentosarja ajastettiin suoritettavaksi huoltokatkopäivänä kello 9:00. Näillä toimenpiteillä automatisoitiin päivitysten hyväksyminen.

Päivitysten lataaminen, asentaminen ja palvelinten uudelleenkäynnistys automatisoitiin ryhmäkäytännön avulla. Palvelimille luotiin ryhmäkäytäntöobjekti, joka määritteli seuraavat asetukset

- Palvelin käynnistyy päivitysten asentamisen jälkeen automaattisesti 15 minuutin kulututtua.
- Uusia päivityksiä etsitään neljän tunnin välein.
- Palvelin lataa ja asentaa saatavilla olevat päivitykset joka keskiviikkoaamu (porrastetusti riippuen palvelimesta).
- Palvelin ei etsi päivityksiä internetissä sijaitseviin Windowsin päivityspalveluista.
- Palvelin rekisteröi itsensä ennalta määrättyyn ryhmään ympäristön päivityspalvelussa.
- Sisäverkon päivityspalvelimen osoitteeksi määriteltiin ympäristöön asennettu päivityspalvelin.

Vaikka palvelimet on asetettu etsimään, lataamaan ja asentamaan päivityksiä joka keskiviikko, ne eivät saa päivityksiä huoltokatkojen ulkopuolella, koska niiden käyttämä päivityspalvelu tarjoaa niille päivityksiä ainoastaan 12 tunnin ajan, alkaen hieman ennen huoltokatkoa. Tähän päivityspalvelun ja ryhmäkäytännön yhdistelmään päädyttiin, koska kumpikaan työkalu ei yksin tarjoa riittäviä ominaisuuksia päivitysten asentamisen ajastamiseen kerran kuukaudessa tapahtuvaksi.

7.4 Dokumentointi

Osana projektia toteutettiin seuraavat, lähinnä tekniset dokumentit.

- pelkistetty projektisuunnitelma ja askelmerkit
- verkkokontekstin tiedot
- lista palomuurisäännöistä
- palvelinluettelo
- yksityiskohtaiset asennusohjeet jokaiselle palvelimelle
- kuva, josta käy ilmi palvelimet ja niiden välinen verkkoliikenne
- ryhmäkäytäntöasetukset
- kokoelma komentosarjoja
- ylläpitäjän opas.

Kaikki dokumentaatio tallennettiin CSC:n sisäiselle wikisivustolle. Dokumentaatio syntyi projektin suunnittelun ja toteutuksen ohella samanaikaisesti siten, että esimerkiksi tehdyistä asennuksista tehtiin muistiinpanot, joista puhtaaksikirjoitettiin dokumentaatio. Dokumentaation tuottamista helpotti merkittävästi se, että suurin osa asennuksista tehtiin PowerShellin avulla, jolloin dokumentaatioksi riitti hyvin kommentoitu PowerShell-komentosarja.

Dokumentointi aloitettiin pelkistetystä projektisuunnitelmasta, johon kirjattiin luetelma palveluun tarvittavista komponenteista ja suuremmista toimenpidekokonaisuuksista. Käytännössä suunnitelma oli luetelma tehtäviä. Dokumentti oli myös elävä, eli tarvittaessa lisättiin toimenpiteitä, ja toimenpiteet merkittiin suoritetuiksi sitä mukaa, kun ne valmistuivat. Projektisuunnitelman tärkein tehtävä olikin varmistaa, että mikään toimenpide ei jää tekemättä ja samalla toimia indikaattorina sille, miten pitkällä projektissa oltiin.

Verkkokontekstin tietoihin kirjattiin virtuaalilähiverkon numero, verkon osoite ja aliverkon peite, oletusyhdyskäytävä sekä palvelimille käytettävissä olevat osoitteet. Palomuuridokumenttiin kirjattiin palvelun tarvitsemat palomuurisäännöt, mikä käsittää muun muassa loppukäyttäjien ilmoittamat IP-osoitteet, joista palvelua on mahdollista käyttää.

Projektin tärkeimpiä dokumentteja olivat palvelinten yksityiskohtaiset asennusohjeet. Asennusohjeissa kuvattiin kaikki palvelimille tehdyt toimenpiteet alkaen siitä hetkestä, kun palvelimet oli vastaanotettu päättyen siihen, kun palvelimen asennus oli valmis. Myös myöhemmin tehdyt muutokset kirjattiin asennusohjeisiin. Näiden dokumenttien keskeisin tarkoitus oli toimia ohjeistuksena sille, miten mahdollisesta ongelma- tai katastrofitilanteesta, kuten palvelimen ja sen varmuuskopioiden tuhoutuminen, voidaan palauttaa palvelun toiminnallisuus.

8 Yhteenveto

Insinöörityössä suunniteltiin ja toteutettiin turvallinen etäkäyttöympäristö. Samalla tuotettiin paljon dokumentaatiota ja komentosarjoja, joista on tulevaisuudessa yritykselle hyötyä muissa vastaavanlaisissa projekteissa. Insinöörityön ansiosta saatiin myös käyttöko-

kemuksia Microsoftin uudesta selainpohjaisesta etätyöpöytäsovelluksesta. Näistä kokemuksista oli suuri apu, kun kahdessa toisessa etätyöpöytäprojektissa harkittiin Microsoftin tuotteen ja avoimen lähdekoodin Guacamole-sovelluksen välillä.

Projekti käsitti muun muassa vaatimusmäärittelyn tekemisen, palvelinten resurssien määrittämisen, etätyöpöytäympäristön vaatimien komponenttien asentamisen ja konfiguroinnin, tarvittavien varusohjelmien asentamisen, testauksen, dokumentaation laatimisen sekä ylläpitorutiinien luomisen. Näistä varusohjelmien asentamista ei käsitelty tässä raportissa.

Kokonaisuutena projekti onnistui hyvin. Asiakas sai tarvitsemansa palvelun, ja alkuperäisen määrittelyn jälkeen ilmenneet lisätoiveet kyettiin toteuttamaan. Lisätoiveiden ilmeneminen vaikeutti hieman projektin toteuttamista, ja mikäli ne olisivat olleet tiedossa alusta saakka, olisi jotkin palvelun osat toteutettu hieman toisella tavalla. Esimerkiksi tiedostonsiirto-ominaisuutta varten olisi ollut parempi määritellä erillinen palvelin.

Käyttäjämäärät kasvoivat projektin edetessä merkittävästi alkuperäisistä odotuksista, ja nykyään palveluun on myönnetty pääsy noin sadalle käyttäjälle. Palvelun päivittäisiä, viikoittaisia tai kuukausittaisia käyttäjämääriä ei ole tilastoitu, vaan niiden sijaan on seurattu palvelinten kuormitusta. Kuormituksesta kerättyyn dataan perustuen etätyöpöytäpalvelimen resursseja lisättiin kerran.

Lähteet

1 Compare features in Windows Server versions. 2019. Verkkoaineisto. Microsoft Corporation. <https://www.microsoft.com/fi-fi/cloud-platform/windows-server-comparison>. Luettu 8.5.2019.

2 Lohr ym. 2018. Set up the Remote Desktop web client for your users. Verkkoaineisto. Microsoft Corporation. <https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/clients/remote-desktop-web-client-admin>. Luettu 22.9.2018.

3 Active Directory (AD). 2018. Verkkoaineisto. Secret Double Octopus. <https://doubleoctopus.com/security-wiki/authentication/active-directory/>. Luettu 22.9.2018.

4 Desmond, Brian. 2010. Kerberos in Active Directory. Verkkoaineisto. ITPro Today. <https://www.itprotoday.com/active-directory/kerberos-active-directory>. Luettu 8.5.2019.

5 Desmond ym. 2008. Active Directory. Yhdysvallat: O'Reilly Media.

6 Terminal Services Gateway (TS Gateway). 2012. Verkkoaineisto. Microsoft Corporation. <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc731264(v=ws.10)>. Luettu 22.9.2018.

7 Terminal Services Session Broker (TS Session Broker). 2012. Verkkoaineisto. Microsoft Corporation. <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc731045(v%3dws.10)>. Luettu 22.9.2018.

8 Terminal Services Web Access (TS Web Access). 2012. Verkkoaineisto. Microsoft Corporation. <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc771908(v%3dws.10)>. Luettu 22.9.2018.

9 Lohr ym. 2018. Access the Remote Desktop web client. Verkkoaineisto. Microsoft Corporation. < https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/clients/remote-desktop-web-client>. Luettu 22.9.2018.

10 Terminal Services Licensing (TS Licensing). 2012. Verkkoaineisto. Microsoft Corporation. <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc770371(v%3dws.10)>. Luettu 22.9.2018.

11 Kaksivaiheinen tunnistautuminen pelastaa paljolta - pelkkä salasana ei suojaa kaikilta uhkilta. 2017. Verkkoaineisto. Traficom. <https://legacy.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2017/08/ttn201708301327.html>. Luettu 30.9.2018.

12 M'Raihi ym. 2011. TOTP: Time-Based One-Time Password Algorithm. Verkkoaineisto. Internet Engineering Task Force (IETF). <https://tools.ietf.org/html/rfc6238>. Luettu 30.9.2018.

13 Sharma, Prakash. 2018. How Time-based One-Time Passwords work and why you should use them in your app. Verkkoaineisto. Free Code Camp. <https://medium.freecodecamp.org/how-time-based-one-time-passwords-work-and-why-you-should-use-them-in-your-app-fdd2b9ed43c3>. Luettu 30.9.2018.

14 MultiOTP Open Source. 2018. Verkkoaineisto. SysCo systèmes de communication sa. <https://github.com/multiOTP/multiotp/wiki>. Luettu 30.9.2018.

Windowsin päivityspalvelun asennus PowerShellillä

```
#INSTALL WSUS
Install-WindowsFeature -Name  UpdateServices  -IncludeManagementTools

New-Item -Path D: -Name WSUS -ItemType Directory
C:\'Program Files'\'Update Services'\Tools\wsusutil.exe postinstall
CONTENT_DIR=D:\WSUS
Set-WsusServerSynchronization –SyncFromMU

# Get WSUS Server Object
$wsus = Get-WSUSServer

# Connect to WSUS server configuration
$wsusConfig = $wsus.GetConfiguration()

# Set Update Languages to English and save configuration sett
$wsusConfig.AllUpdateLanguagesEnabled = $false
$wsusConfig.SetEnabledUpdateLanguages(“en”)
$wsusConfig.Save()

# Get WSUS Subscription and perform initial synchronization to get latest
categories
$subscription = $wsus.GetSubscription()
$subscription.StartSynchronizationForCategoryOnly()
write-host 'Beginning first WSUS Sync to get available Products etc' -
ForegroundColor Magenta
write-host 'Will take some time to complete'
While ($subscription.GetSynchronizationStatus() -ne 'NotProcessing') {
Write-Host "." -NoNewline
Start-Sleep -Seconds 5
}
write-host ' '
Write-Host "Sync is done." -ForegroundColor Green

# Configure the Platforms that we want WSUS to receive updates
# First disable all products
write-host 'Diabling products WSUS Products'
Get-WsusProduct | Set-WsusProduct -Disable

# Enable the ones you need
write-host 'Setting WSUS Products'
Get-WsusProduct | where-Object {
$_.Product.Title -in (
'Windows Server 2016',
'Windows Server Manager – Windows Server Update Services (WSUS) Dynamic
Installer')
} | Set-WsusProduct

# Configure Synchronizations
write-host 'Enabling WSUS Automatic Synchronisation'
$subscription.SynchronizeAutomatically=$true

# Set synchronization scheduled for midnight each night
$subscription.SynchronizeAutomaticallyTimeOfDay= (New-TimeSpan -Hours 0)
$subscription.NumberOfSynchronizationsPerDay=12
$subscription.Save()

# Kick off a synchronization
$subscription.StartSynchronization()
```

```
# Monitor Progress of Synchronisation

write-host 'Starting WSUS Sync, will take some time' -ForegroundColor Magenta
Start-Sleep -Seconds 60 # Wait for sync to start before monitoring
while ($subscription.GetSynchronizationProgress().ProcessedItems -ne
$subscription.GetSynchronizationProgress().TotalItems) {
Write-Progress -PercentComplete (
$subscription.GetSynchronizationProgress().ProcessedItems*100/($subscription.G
etSynchronizationProgress().TotalItems)
) -Activity "WSUS Sync Progress"
}
Write-Host "Sync is done." -ForegroundColor Green

# Configure Default Approval Rule

if ($DefaultApproval -eq $True)
{
write-host 'Configuring default automatic approval rule'
[void][reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Adm
inistration")
$rule = $wsus.GetInstallApprovalRules() | Where {
$_.Name -eq "Default Automatic Approval Rule"}
$class = $wsus.GetUpdateClassifications() | ? {$_.Title -In (
'Critical Updates',
'Definition Updates',
'Feature Packs',
'Security Updates',
'Service Packs',
'Update Rollups',
'Updates')}
$class_coll = New-Object  UpdateClassificationCollection
$class_coll.AddRange($class)
$rule.SetUpdateClassifications($class_coll)
$rule.Enabled = $True
$rule.Save()
}

# Configure Default Approval Rule
$wsus = Get-WSUSServer
$DefaultApproval = $True


if ($DefaultApproval -eq $True)
{
write-host 'Configuring default automatic approval rule'
[void][reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Adm
inistration")
$rule = $wsus.GetInstallApprovalRules() | Where {
$_.Name -eq "Default Automatic Approval Rule"}
$class = $wsus.GetUpdateClassifications() | ? {$_.Title -In (
'Critical Updates',
'Definition Updates',
'Feature Packs',
'Security Updates',
'Service Packs',
'Update Rollups',
'Updates')}
$class_coll = New-Object  UpdateClassificationCollection
$class_coll.AddRange($class)
$rule.SetUpdateClassifications($class_coll)
$rule.Enabled = $True
```

```
$rule.Save()
}
# Run Default Approval Rule
$RunDefaultRule = $True

if ($RunDefaultRule -eq $True)
{
write-host 'Running Default Approval Rule'
write-host ' >This step may timeout, but the rule will be applied and the
script will continue' -ForegroundColor Yellow
try {
$Apply = $rule.ApplyRule()
}
catch {
write-warning $_
}
Finally {

write-host 'WSUS log files can be found here: %ProgramFiles%\Update
Services\LogFiles'
write-host 'Done!' -foregroundcolor Green
}
}


MORE WSUS SETTINGS

#Add DNS alias for wsus in dns:
Add-DnsServerResourceRecordCName -Name "wsus" -HostNameAlias $DC2 -ZoneName
$DomainName

# Set "Use group policy or registry setting on computer"
$config = (get-wsusserver).GetConfiguration()
$config.TargetingMode = "Client"
$config.Save()

# Create a new client group
$wsusserver = “”
$portnumber = “”
#Load required assemblies
[void][reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Adm
inistration")
$wsus =
[Microsoft.UpdateServices.Administration.AdminProxy]::getUpdateServer($wsusser
ver,$False,$portnumber)
$wsus.IsValidComputerTargetGroupName($TargetGroup)
$wsus.CreateComputerTargetGroup($TargetGroup)
```