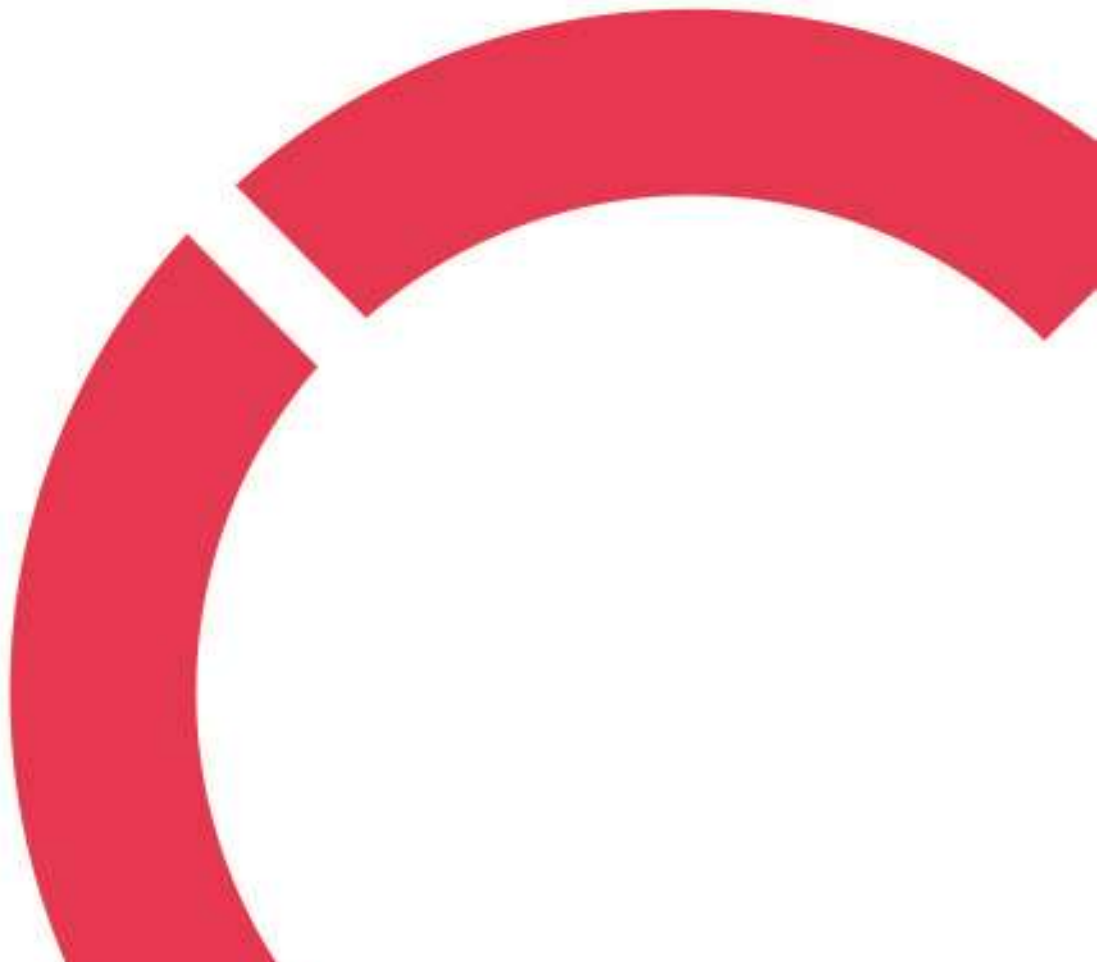


Nina Nykänen

**INTEGRAATIO KÄYTTÄJÄTIETOJÄRJESTELMÄN JA HENKI-
LÖSTÖHALLINTOJÄRJESTELMÄN VÄLILLÄ**

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tieto- ja viestintäteknikan koulutus
Huhtikuu 2023**



Centria-ammattikorkeakoulu	Aika Huhtikuu 2023	Tekijä/tekijät Nina Nykänen
Koulutus Insinööri (AMK), tieto- ja viestintätekniikka		<input checked="" type="checkbox"/> AMK <input type="checkbox"/> YAMK
Työn nimi INTEGRAATIO KÄYTTÄJÄTIETOJÄRJESTELMÄN JA HENKILÖSTÖHALLINTOJÄRJESTELMÄN VÄLILLÄ		
Työn ohjaaja Jari Isohanni		Sivumäärä 28+1
Työelämäohjaaja -		
<p>Opinnäytetyön tavoitteena oli tutkia ja toteuttaa integraatio Microsoftin tarjoaman käyttäjätietokannan ja ulkoisen palveluntarjoajan henkilöstöhallintojärjestelmän välillä. Päädyin aiheeseen yrityksen tarpeesta automatisoida uuden käyttäjän luonti, käyttäjätietojen muutos ja käyttäjätilin käytöstä poisto kahden järjestelmän välillä. Integraation katsottiin nopeuttavan tietojen lisäys-, muutos- ja poistoprosessia sekä varmistavan yhteneväiset tiedot molemmissa järjestelmissä.</p> <p>Teoriaosuus käsittelee yleisesti integraatioita, niiden tekniikkaa ja käyttökohteita. Teoriaosuus käsittelee myös PowerShell-komentotulkin toimintaa, FTP-tiedonsiirtoprotokollaa sekä lyhyesti muita tähän integraation tarvittavia komponentteja, kuten Komentokehotetta ja Tehtävien ajoitusta.</p> <p>Toiminnallisessa osuudessa kuvataan komponenttien toimintaa yhdessä. Suurimmassa roolissa on Microsoftin käyttäjätietokannan päivityksestä vastaava PowerShell-skripti, jonka toiminta kuvataan läpi pääpiirteittäin. Yksinkertaistettuna työssä siirretään tiedostoja yrityksen palvelimelta palveluntarjoajan palvelimelle ja palveluntarjoajan palvelimelta yrityksen palvelimelle. Jälkimmäisen siirron yhteydessä päivitetään palveluntarjoajalta saadut tiedot käyttäjistä käyttäjätietokantaan Komentokehotteen ja PowerShellin avulla.</p> <p>Lopputuloksen voidaan todeta olevan toimiva ja käytännöllinen tapa integroida tiedot kahden järjestelmän välillä. Uuden käyttäjän luominen Microsoftin käyttäjätietokantaan ei kuitenkaan tullut täysin automaattiseksi, vaan tietohallinnon tarvitsee manuaalisesti lisätä tarvittavat lisenssit ja oikeudet sekä lähettää ensimmäinen salasana uudelle käyttäjälle. Käyttäjätilin käytöstä poisto saatiin myös automaattiseksi, mutta käyttäjätilin poisto tulee vielä tehdä manuaalisesti yrityksen tietojensäilytyspolitiikan mukaisesti. Tietojen päivitykset ja muutokset olemassa oleville käyttäjille tulivat täysin automaattisiksi.</p> <p>Tässä toteutuksessa lisenssien lisäämistä ei vaihtelevien tarpeiden vuoksi voida automatisoida, mutta integraatiota on mahdollista jatkokehittää salasanan lähetyksen osalta. Tehtävien ajoituksella ja PowerShellillä voisi rakentaa toiminnallisuuden, joka lähettää käyttäjälle tekstiviestitse salasanan ensimmäisenä työpäivänä.</p>		

Asiasanat

CSV, Integraatio, Komentokehote, PowerShell, Tehtävien ajoitus

ABSTRACT

Centria University of Applied Sciences	Date April 2023	Author Nina Nykänen
Degree programme Bachelor of Engineering, Information and Communications Technology		
Name of thesis INTEGRATION BETWEEN DIRECTORY SERVICE AND PERSONNEL MANAGEMENT SYSTEM		
Centria supervisor Jari Isohanni	Pages 28+1	
Instructor representing commissioning institution or company -		
<p>The aim of the thesis was to investigate and implement the integration between the user database provided by Microsoft and an external personnel management system. I ended up on the subject of the company's need to automate the change of user data between two systems.</p> <p>The theoretical part was concerned with integrations in general, their technology and use cases. The theoretical part also covered the PowerShell command-line, file transfer protocol and briefly other components needed for this integration, such as Command prompt and Task Scheduler.</p> <p>The practical part consisted of the various components that were needed to implement the integration and the scheduling of file transfers to the right place at the right time. In the practical part, data is transferred from the company's server to the service provider's server and from the service provider's server to the company's server. After the latter transfer, the information about the users received from the service provider is updated to the user database using a command prompt and PowerShell.</p> <p>The outcome is a functional and practical way to integrate information between two systems. However, the creation of a new user in Microsoft's active directory did not become fully automatic, instead the IT department needs to manually add the necessary licenses and rights and send the first password to the new user. Disabling the user account also became automatic, but the deletion of the user account must still be done manually in accordance with the company's data retention policy. Data updates and changes for existing users became fully automatic.</p> <p>In this implementation, adding licenses cannot be automated due to varying requirements, but it is possible to further develop the integration. Password sending could be automated with Task Scheduler and PowerShell, by building a functionality that sends the password to the user via text message on the first day of work.</p>		

<p>Key words Command prompt, CSV, integration, PowerShell, Task Scheduler</p>
--

KÄSITTEIDEN MÄÄRITTELY

AD

Active Directory eli aktiivihakemisto on Microsoftin kehittämä ja ylläpitämä käyttäjätietokanta ja hakemistopalvelu, johon voidaan tallentaa tietoja käyttäjistä ja antaa järjestelmänvalvojille mahdollisuus hallita niitä.

Big data

Big data -termillä tarkoitetaan niin suuria tietokokonaisuuksia, ettei niitä voi perinteisillä tietojenkäsittelytyökaluilla hallita ja analysoida. Big data sisältää toisiinsa liittyvää jäsentelemätöntä, arkistoitua ja reaaliaikaista dataa, esimerkiksi sensoreilla kerättyä tietoa ilmastosta, sosiaalisen median alustoille julkaisuja viestejä, verkossa lähetettyjä kuvia ja videoita, verkko-ostoksista syntyneitä tapahtumatietoita ja matkapuhelimien GPS-signaaleja. (Ohlhorst 2012, 1–2.)

cmdlet

cmdlet on Powershell-ympäristössä käytettävä komento, ikään kuin kahden sanan mittainen skripti, jolle voidaan syöttää parametreina tietoja. cmdlet-komentoja on satoja erilaisia.

CSV

Comma-separated Values -tiedosto on tekstitiedosto, jossa jokainen rivi on yksi tietue, joka koostuu yhdestä tai useammasta pilkulla erotellusta kentästä. Tyypillisesti CSV-tiedostoihin tallennetaan taulukkomuotoista dataa tekstinä ja taulukoille ominaisesti jokaisella rivillä on yhtä monta kenttää. (Team 2022.)

IoT

Internet of Things kuvaa fyysisten objektien verkkoa, joka voidaan jakaa viiteen osaan: esineeseen upotettuun sensoriin, sensoritietoja mittaavaan laitteeseen, sensoritietoja eteenpäin lähettävään laitteeseen, viestintäverkkoon, jossa viestejä lähetetään, sekä ohjelmistoon, jolla analysoidaan ja tulkitaan dataa (Toivanen 2017). Laitteita on aina tavallisista kodin esineistä kehittyneisiin teollisuudessa käytettyihin työkaluihin.

Skripti

Sarja peräkkäin suoritettavia komentoja, jotka on etukäteen kirjoitettu ja tallennettu tiedostoon.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO	1
2 INTEGRAATIOT	2
2.1 Integraatio käytännössä.....	2
2.2 Hyötyjä ja esimerkkejä.....	3
2.3 Yleiset haasteet	4
3 KÄYTETYT TYÖKALUT	6
3.1 Tehtävien ajoitus ja PSFTP.....	6
3.2 Microsoft Active Directory.....	8
3.3 CSV-tiedostot.....	9
3.4 PowerShell	10
4 INTEGRAATIO AD:N JA HR-JÄRJESTELMÄN VÄLILLÄ	14
4.1 Eräajotiedostojen suoritus.....	16
4.2 AD:sta HR-järjestelmään.....	17
4.3 HR-järjestelmästä AD:hen	18
4.3.1 Käyttäjätietojen päivitys	20
4.3.2 Lokitiedoston kirjoitus.....	22
4.3.3 Uuden käyttäjän luonti.....	22
4.3.4 Virheviestin lähetys.....	24
5 LOPPUTULOS JA YHTEENVETO	25
LÄHTEET	27
LIITTEET	
KUVIOT	
KUVIO 1. Esimerkki organisaation yksiköistä ja ryhmistä.....	9
KUVIO 2. Järjestelmien yhteneväiset tiedot.....	14
KUVIO 3. Integroitiprosessi HR-yritykseltä yrityksen A suuntaan	15
KUVIO 4. Tiedostojen siirto palvelimien välillä.....	17
KUVAT	
KUVA 1. psftp-ohjelman ajo komentokehoteessa.....	7
KUVA 2. Eräajotiedosto, jossa kutsutaan psftp-asiakasohjelmaa	8
KUVA 3. CSV-tiedosto avattuna Notepad++-ohjelmalla	9
KUVA 4. CSV-tiedosto tuotuna Exceliin	10
KUVA 5. Kaikkien käyttäjien haku PowerShell cmdlet-komennolla tietyistä ryhmästä	11
KUVA 6. Tietystä ryhmässä olevien käyttäjien sähköpostiosoitteiden haku.....	12
KUVA 7. Ajoitetut tehtävät Tehtävien ajoitus -ohjelmassa	16
KUVA 8. Lokitiedosto AD:sta HR-yritykselle päin.....	18
KUVA 9. CSV-tiedoston prosessointi	19
KUVA 10. CSV-tiedoston sisältö PowerShell-tulosteessa	19
KUVA 11. PowerShell-skriptin osa, jossa käydään läpi CSV-tiedoston tietueet.....	20

KUVA 12. PowerShell-skripti käyttäjän tietojen päivittämiseen	21
KUVA 13. Lokitiedosto PowerShellin AD:hen tekemistä muutoksista	22
KUVA 14. Satunnaisen salasanan generointi	23
KUVA 15. Virheviestin lähetys	24

1 JOHDANTO

Työn tarkoituksena on perehtyä integraatioihin ja niiden yritykselle tuottamaan hyötyyn. Integraatioiden suurin hyöty lähtökohtaisesti on automatisoida erilaisia prosesseja ja näin vapauttaa työntekijöiden resursseja yrityksissä. Siinä missä automatisoitu prosessi vapauttaa työntekijän, automaatio myös vapauttaa prosessin kellonaikarajoitteista. Kun ihmistä ei tarvita prosessin suoritukseen, voi tietokone tehdä sen mihin kellonaikaan tahansa huolimatta siitä, onko yrityksessä ketään töissä siihen aikaan. Koska integraatiot vähentävät manuaalisyötä, ne myös nopeuttavat joitain manuaalisyönteä prosesseja ja poistavat inhimillisen virheen mahdollisuuden näissä osissa.

Opinnäytetyön aihe syntyi yrityksen tarpeesta automatisoida käyttäjätietojen muutos Microsoftin käyttäjätietojärjestelmän (AD) ja ulkoisen henkilöstöhallintojärjestelmän välillä. Työn tarkoituksena oli selvittää, miten integraatio olisi toteutettavissa PowerShellillä ja Tehtävien ajoituksella, ja miten tiedostojen siirto voidaan suorittaa tietoturvallisesti. Toteutuksen myötä oli myös tarkoitus selvittää, onko integraatiosta todella ajallista ja rahallista hyötyä yritykselle. Integraation rakentaminen ei ole lyhyt eikä yksinkertainen prosessi ja jo se vaatii aikaa ja rahaa, mutta pitkällä tähtäimellä kehitystyöhön käytetyn ajan voidaan ajatella olevan sijoitus tulevaisuuteen. Integraation toteutuksen onnistuessa se tulisi maksamaan itsensä takaisin vapautuneina resursseina.

Aikaisemmin vaadittiin kahden työntekijän työaikaa siihen, että henkilöstöhallinnon (HR) puolella työntekijä luo käyttäjän, poistaa käyttäjän tai päivittää käyttäjän tietoja ja tietohallinnon puolella työntekijän aikaa siihen, että samat muutokset tehtiin AD:hen. Tietojen päivytyksessä esiintyy näin myös vaihtelevaa viivettä ja inhimillisen virheen mahdollisuus kasvaa. Automatisoinnin ansiosta tiedetään, että tiedot siirtyvät ja päivittyvät tietyssä ennalta määritetyssä ajassa ja tiedot vastaavat toisiaan molemmissa järjestelmissä varmasti.

Työn teoriaosuudessa perehdytään yleisesti integraatioihin, niiden käyttökohteisiin ja niistä syntyviin hyötyihin. Erilaisten integraatioiden tekemiseen käytetään laajasti erilaisia järjestelmiä ja noudatetaan erilaisia protokollia. Teoriaosuudessa esitellään ne järjestelmät ja protokollat, joita tulee vastaan toiminnallisessa osuudessa. Toiminnallisessa osuudessa kuvataan tarkemmin läpi integraation toteutus ja toiminta. Kyseessä on toteutukseltaan verrattain yksinkertainen integraatio, jossa siirretään tiedostoja palvelimelta toiselle ja päivitetään AD:ta.

2 INTEGRAATIOT

Tietotekniikassa integraatio tarkoittaa tietojen integrointiprosessia, jossa yhdistetään tietoja useista eri lähteistä ja saadaan tieto liikkumaan esimerkiksi kahden eri palvelimen välillä. Integrointi on yksi tiedonhallinnan ydinelementeistä, ja sen ensisijaisena tavoitteena on tuottaa konsolidoituja tietokokonaisuuksia, jotka vastaavat organisaation eri loppukäyttäjien tarpeita. Erilaisiin käyttötarkoituksiin on luotu erilaisia integrointimenetelmiä, esimerkiksi tietyin väliajoin ajettavat eräintegraatiot ja reaaliaikainen, jatkuva integrointi. (Stedman & Wigmore.)

Keskivertoyrityksen ohjelmistoympäristö koostuu sadoista tai jopa tuhansista rakennetuista ja ostetuista, erilaisista ja muuttuvista tietoteknisistä järjestelmistä. Näistä erilaisista järjestelmistä saatu data on integroitava yhteensopivaksi, jos dataa on tarkoitus käyttää esimerkiksi raportointiin ja analysointiin. Kun vanhoja järjestelmiä korvataan uusilla, dataa voidaan joutua muuttamaan järjestelmämuodosta toiseen, kuitenkin muuttamatta varsinaisia tietoja. Eri järjestelmien ja palvelimien välillä kulkevan tiedon tehokas hallinta on suuri haaste ja huolenaihe, mutta myös liiketoiminnan kannalta avainasemassa jokaisessa tietotekniikkaorganisaatiossa. (Reeve 2013, 3.)

Usein yritysten käyttämät sovellukset ja operatiiviset työntekijät tarvitsevat pääsyn eri lähteiden tietoihin ja tiedostoihin tehdäkseen työnsä. Esimerkiksi yrityksen verkkokauppa vaatii tietoja asiakas-, varasto- ja logistiikkatietokannoista tilausten käsittelyyn ja asiakaspalvelijan on päästävä käsiksi näihin samoihin tietoihin asiakkaiden ongelmien ratkaisemiseksi. Lainavastaavien on tarkistettava tilitiedot ja luottotiedot, kiinteistöjen arvot ja muut tiedot ennen lainahakemuksen hyväksymistä. Sijoittajien on pidettävä silmällä sisäisistä järjestelmistä ja ulkoisista lähteistä tulevia markkinatietoja. Näissä ja muissa sovelluksissa tietojen integrointi kerää automaattisesti tarvittavat tiedot käyttäjille, joten heidän ei tarvitse manuaalisesti yhdistää niitä. (Stedman & Wigmore.)

2.1 Integraatio käytännössä

Teknisestä näkökulmasta katsottuna integrointiarkkitehdit ja ohjelmoijat kehittävät ohjelmistoja, jotka automatisoivat ja hallinnoivat integrointiprosesseja. Yksinkertaisimmillaan integrointi on tiedostojen kopiointia järjestelmästä toiseen. Usein kuitenkin eri tietokantamallit erillisissä lähdejärjestelmissä joudutaan harmonisoimaan osana integraatioprojektia. Yksi yleinen lähestymistapa on luoda välitetty

malli tai kaavio, joka yhdistää paikalliset lähdeskeemat globaaliksi ja kartoittaa tiedon niiden välisten erojen ja dataelementtien sovittamiseksi välitettyyn malliin. (Stedman & Wigmore.) Tietojen kartoitus on prosessi, jossa tietokenttä yhdistetään yhdestä lähteestä toisessa lähteessä olevaan tietokenttään. Tämä pienentää virheen mahdollisuutta, auttaa standardisomaan tietoja ja helpottaa tietojen ymmärtämistä.

Kehittäjät voivat koodata komentosarjojen eli skriptien muodossa käsin tietojen integrointitöitä, jotka on kirjoitettu Structured Query Language (SQL) -kielellä, joka on relaatiokannoissa käytettävä vakio-ohjelmointikieli. Käsin koodatut integraatiotyöt olivat useiden vuosien ajan yleisin lähestymistapa integraatioon, mutta eri IT-toimijoilta on nykyisin tullut saataville paketoituja tiedon integrointityökaluja, jotka automatisoivat, virtaviivaistavat ja dokumentoivat kehitysprosessin. (Stedman & Wigmore.)

2.2 Hyötyjä ja esimerkkejä

Asiakastietojen integrointi on yksi yleisimmistä tiedon integroinnin käyttökohteista. Se pitää sisällään asiakkaita koskevien tietojen yhdistämisen kaikista saatavilla olevista lähteistä, mukaan lukien yhteystiedot, tilitiedot, asiakkaan elinkaariarviot, tiedot asiakaspalvelupuheluista, verkkosivuvierailuista, suoramarkkinointiohjelmasta, kyselyistä, sosiaalisen median välityksellä käydyistä viestikeskusteluista sekä muista vuorovaikutuksista asiakkaan kanssa. Onnistunut integrointityö varmistaa, että yrityksen eri osastot saavat ajantasaisen ja kattavan käsityksen asiakkaista. Nämä tiedot auttavat yrityksiä kohdentamaan markkinointinsa paremmin ja tunnistamaan muun muassa lisämyynnin paikat asiakkaille. Hyvin integroidut asiakastiedot myös parantavat asiakaspalvelun laatua tarjoamalla puhelinasiakaspalvelun työntekijälle välittömästi kaikki hänen tarvitsemansa tiedot asiakkaasta, hänen asiakashistoriansa sekä nykyisistä tuotteista ja palveluista. (Stedman & Wigmore.)

Enenevässä määrin yritykset ovat siirtymässä integroimaan erilaisiin teollisuuslaitteisiin asennettujen sensoreiden keräämää dataa. Integroitua sensoritietoa voidaan käyttää toimintojen valvontaan ja enakoivien huoltotoimenpiteiden suunnitteluun ja toteutukseen, joiden tarkoituksena on tunnistaa ja korjata laiteviat ennen niiden ilmenemistä. Tämä auttaa vähentämään suunnittelemattomia katkoksia korjauksien ajaksi. (Stedman & Wigmore.)

Terveydenhuollon alalla integroidut potilastiedot eri terveys- ja lääkäriasemilta auttavat lääkäreitä diagnosoimaan sairauksia ja muuta terveydentilaa. Vakuutusyhtiöille tehokas tietojen integrointi sisäisesti sekä ulkoisesti terveydenhuollon tarjoajien kanssa helpottaa ja nopeuttaa korvausten käsittelyä ja varmistaa, että asiakkaiden nimet ja osoitteet ovat yhteneväiset. (Stedman & Wigmore.)

2.3 Yleiset haasteet

Tiedonhallinnassa suurin painopiste on datassa, joka on varastoitu tietokantoihin tai tiedostoihin ja huomattavasti pienempi painopiste on tietorakenteiden välillä ja ympärillä kulkevissa tiedoissa. Kuitenkin tietorajapintojen hallinta nousee koko ajan tärkeämmäksi organisaatioiden liiketoiminnassa ja tietotekniikassa. Kun organisaatio ottaa käyttöön uusia järjestelmiä, rajapintojen määrä ja monimutkaisuus kasvavat reilusti, mikä tekee hallinnoinnista entistä kriittisempää. Sovellusten ja järjestelmien välisten rajapintojen määrä voi kasvaa eksponentiaalisesti. Vaikka jokaisen järjestelmän välillä ei tarvitsekaan olla rajapintaa, kahden järjestelmän välillä voi olla useampi rajapinta erityyppisiä tietoja tai tarpeita varten. Sataa eri sovellusta hyödyntävän yrityksen käytössä voi olla viisisataa käyttöliittymää ja tuhannen sovelluksen yrityksessä jopa puoli miljoonaa käyttöliittymää hallinnoitavana. (Reeve 2013, 3.) Organisaatioiden luoman ja keräämän datan määrä aiheuttaa erityisen suuria haasteita integraatioissa. Datavolyymit jatkavat nopeaa kasvuaan ja kasvuvauhti vain kiihtyy big data -sovellusten laajentuessa, edullisten pilvitallennuspalvelujen käytön lisääntyessä ja IoT:n kehittyessä. Tietojen integrointi on elintärkeää kerätyn datan täyden potentiaalin realisoimiseksi, mutta onnistunut suunnittelu ja hallinnointi on monimutkainen prosessi. (Stedman & Wigmore.)

Tietojen integroinnissa esiintyvät järjestelmähaasteet ovat melko ilmeisiä ja tulevat esille varhaisessa vaiheessa, kun integrointisovellusta rakennetaan. Pohjimmiltaan haasteena on saada eri järjestelmät keskustelemaan saumattomasti toistensa kanssa. Esimerkiksi vaikka SQL on relaatiotietokantojen vakiokyselykieli, eri toimittajat käyttävät sitä eri tavoin ja nämä erot ovat sovitettava yhteen. Integraatioiden määrä ja kyselyjen käsittely hajautetuissa tietokannoissa, eli tietokannoissa, joissa tiedot on jaettu useisiin eri solmuihin, lisäävät monimutkaisuutta ja haasteita. Tietojen integroinnissa tulee vastaan jo olemassa olevia lähdekokoelmia ja kunkin tietolähteen kyselynkäsittelyehdot voivat olla ominaisuuksiltaan hyvin erilaisia. Esimerkiksi yksi lähde voi olla täysin SQL-pohjainen ja siksi hyväksyä hyvin monimutkaisia kyselyjä, kun taas toinen lähde voi olla esimerkiksi verkkolomake, joka hyväksyy vain pienen määrän kyselymalleja. (Doan, Halevy & Ives 2012, 6–7.)

Toinen tyypillinen haaste liittyy tietojen loogiseen järjestämiseen tietolähteissä. Suurin osa jäsenel-lyistä tietolähteistä on järjestelty skeeman mukaan; yleisessä relaatiotietokantatapauksessa skeema määrittää joukon taulukoita ja jokaiselle taulukolle joukon attribuutteja tietotyyppeineen. Toisenlai- sissa tietomalleissa skeema määritellään tunnisteilla, luokilla ja ominaisuuksilla. Ihmisten mielet ovat luonnostaan hyvin erilaisia, ja sen takia ihmisten suunnittelemat kaaviot voivat olla hyvinkin erilaisia, vaikka kaikille olisi annettu samat vaatimukset suunniteltavalle tietokantasovellukselle. Esimerkkinä voidaan käyttää kahden eri yrityksen työntekijätietokantaa, toinen yritys voi mallintaa osa-aikaisia ja kokoaikaisia työntekijöitä samaan taulukkoon, kun toinen yritys ylläpitää kahta erillistä taulukkoa. Toinen yritys yksilöi työntekijänsä henkilötunnuksella, kun taas toinen yritys erillisellä työntekijätun- nuksella. (Doan ym. 2012, 7–8.)

Tiedot voidaan myös esittää merkittävästi eri tavalla. Yritys voi kirjata työntekijöiden suku- ja etuni- men erillisiin kenttiin työntekijätietokannassa ja saman yrityksen toisessa tietokannassa, esimerkiksi työntekijöiden koulutustietokannassa, vain yhteen kenttään. Tietojen täsmäytys monimutkaistuu hie- man edellä kuvatussa tilanteessa, sillä järjestelmä ei tunnista yhteen kenttään kirjoitetusta nimestä kumpi on etunimi ja kumpi sukunimi. Järjestelmä ei siis osaa tunnista henkilöä samaksi, jos nimet on kirjoitettu eri päin. Kansainvälisissä tapauksissa myös mittayksiköt ja valuutta voivat vaihdella. (Doan ym. 2012, 8.) Suomessa tuotetietokannassa tuotteen voidaan kertoa olevan kymmenen senttimetriä pitkä, mutta integroitaessa tieto Yhdysvaltoihin ei mittayksikkö olekaan enää senttimetri, eikä luku- kaan ole enää kymmenen, kun senttimetrit muutetaan tuumiksi.

3 KÄYTETYT TYÖKALUT

Microsoftin käyttäjätietokannan ja HR-järjestelmän välisen integraation toteuttamiseen on käytetty eri ohjelmistoja, protokollia ja standardeja. Tehtävien ajoitus käynnistää koko prosessin, nimensä mukaisesti sen avulla on ajastettu prosessin aloitus. Ennalta määriteltynä ajankohtana Tehtävien ajoitus käynnistää Komentokehotteessa suoritettavan skriptin. Skripti luo ja kirjoittaa lokitiedostoa sekä kutsuu PowerShell-skriptiä. PowerShell-skripti taas luo ja kirjoittaa omaa lokitiedostoaan ja käynnistää psftp.exe -sovelluksen, joka vie ja noutaa HR-yrityksen palvelimelta käyttäjätietoja sisältävän csv-tiedoston. Tiedosto vietään attribuuttina PowerShell-skriptille, joka hakee AD:sta tiedoston HR-järjestelmälle vietäväksi ja joka päivittää AD:hen tiedot HR-järjestelmästä.

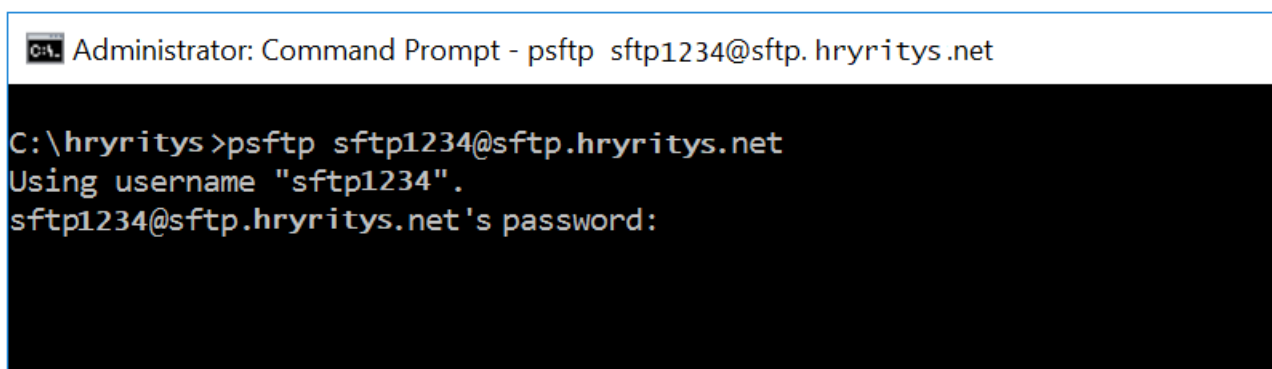
3.1 Tehtävien ajoitus ja PSFTP

Windows-käyttöjärjestelmän yksi tärkeistä hallinnollisista ohjelmista on Tehtävien ajoitus. Sen avulla voidaan ajoittaa erilaisia tehtäviä haluttuna päivänä määrättyinä kellonaikana, tai jonkun tietyn tapahtuman jälkeen. Windows-käyttöjärjestelmän mukana tulevalla komentokehotteella (cmd) on mahdollista luoda ajastettuja tehtäviä myös, mutta Tehtävien ajoitus tarjoaa siihen graafisen käyttöliittymän tehden ajastuksesta helpompaa. Tehtävien ajoituksella voi luoda ajastettuja tehtäviä, näyttää kaikki käynnissä olevat tehtävät, tuoda toisilta tietokoneilta vietyjä ajoitettuja tehtäviä ja yhdistää toiseen tietokoneeseen järjestelmänvalvojana ja hallita siellä ajoitettuja tehtäviä. (Carpenter 2012, 297–298.)

File Transfer Protocol (FTP) on yksi vanhimmista protokollista tiedostojen jakamiseen verkkojen kautta. FTP mallissa on kaksi osapuolta, asiakasohjelma ja palvelin. Asiakasohjelmaan syötetään käyttäjätunnus ja salasana ja palvelimen hyväksytyä kirjautumistiedot, asiakas pääsee etänä käsiksi palvelimella oleviin tietoihin. Asiakasohjelmaan syötetyt tunnukset vaikuttavat siihen, mitä palvelimella oleville tiedostoille voidaan tehdä. Mikäli tunnuksilla on täysi käyttöoikeus, voidaan tiedostoja luoda, muokata, poistaa ja ladata. FTP:n heikkous on tietoturva, kaikki asiakasohjelman ja palvelimen välillä liikkuu tekstimuotoisena, myös syötetty käyttäjätunnus ja salasana. (Negus 2015, 477.) Secure File Transfer Protocol (SFTP) taas toimii muuten samalla tavalla kuin FTP, mutta tähän protokollaan tietoturvaa on lisätty ja kaikki tieto liikkuu salattuna.

PuTTY on maksuton avoimen lähdekoodin ohjelma, jota voidaan käyttää tiedostojen siirtoihin verkon yli. Alun perin PuTTY kehitettiin Windows-käyttöjärjestelmille, mutta sitä on myöhemmin kehitetty käytettäväksi myös muissa käyttöjärjestelmissä. Se on monipuolinen työkalu, jonka avulla toisen tietokoneen etäkäyttö voidaan suorittaa turvallisesti ja siitä syystä se onkin todennäköisesti eniten käytetty Secure Shell -asiakasohjelma Windows-käyttöjärjestelmissä. Secure Shell (SSH) on verkkoprotokolla, jolla luodaan salattu kanava salaamattoman verkon yli. SSH tarjoaa järjestelmänvalvojille turvallisen tavan hallinnoida järjestelmiä etänä käyttäen vahvaa tunnistautumista ja turvallista, salattua tiedoston-siirtoa. (Tanner 2019, 14.)

PuTTY:n tarjoama tekstipohjainen psftp.exe on asiakasohjelma tiedostojen turvalliseen siirtoon palvelimelta toiselle SFTP:tä hyödyntäen. Tiedoston siirto voidaan tehdä komentokehoteella manuaalisesti komento kerrallaan, jolloin ohjelma osaa automaattisesti kysyä käyttäjältä tarvitsemiaan tietoja, esimerkiksi käyttäjätunnusta ja salasanaa (KUVA 1). Tiedoston siirto voidaan tehdä myös automaattisesti, jolloin ohjelma täytyy käynnistää jotain muuta kautta. Esimerkiksi Komentokehoteelle voidaan kirjoittaa komentojono ja tallentaa se bat-muotoiseksi eräajotiedostoksi. Kuvassa 2 on Notepad++-ohjelmalla avattu eräajotiedosto, jossa rivillä 16 kutsutaan psftp.exe-ohjelmaa ja syötetään parametreina käyttäjätunnus sekä salasana, joilla halutaan kirjautua. Yksinkertaisinta on kirjoittaa salasana näkyviin tiedostoon, mutta tietoturvasyistä se kannattaisi syöttää salattuna. Salaaminen on kuitenkin hieman monimutkaisempi prosessi.



```
Administrator: Command Prompt - psftp sftp1234@sftp.hryrityys.net

C:\hryrityys>psftp sftp1234@sftp.hryrityys.net
Using username "sftp1234".
sftp1234@sftp.hryrityys.net's password:
```

KUVA 1. psftp-ohjelman ajo komentokehoteessa

```

1 echo on
2
3 set LOGFILE=C:\HR-yritys\log\hryritys_to_ad.log
4
5 echo Starting HR-yritys to AD import... >> %LOGFILE%
6
7 set date = date /t
8
9 date /t >> %LOGFILE%
10 time /t >> %LOGFILE%
11
12 echo Copying file... >> %LOGFILE%
13
14 powershell C:\HR-yritys\movetoOldHRAD.ps1
15
16 "C:\hryritys\psftp.exe" sftp1234@sftp.hryritys.net -pw SALASANA -b C:\HR-yritys\gethryritys.txt < C:\HR-yritys\input.txt
17
18 echo Running Powershell script... >> %LOGFILE%
19
20 powershell C:\HR-yritys\updatead.ps1 -defaultpath "*"yritysa.com/Users"" -inputFile C:\HR-yritys\hryritys_to_ad.csv >> %LOGFILE%
21
22 date /t >> %LOGFILE%
23 time /t >> %LOGFILE%
24
25 echo Done. >> %LOGFILE%
26 echo ===== >> %LOGFILE%
27
28

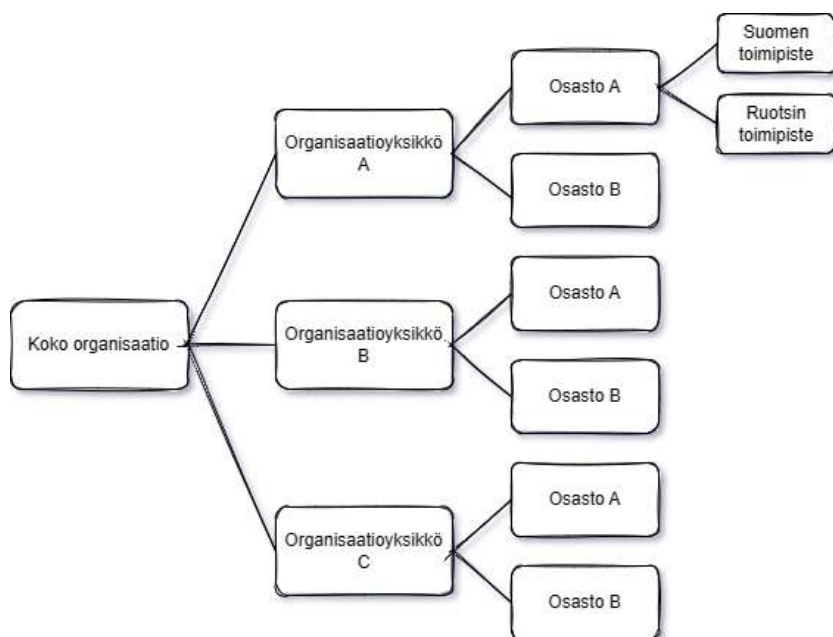
```

KUVA 2. Eräajotiedosto, jossa kutsutaan psftp-asiakasohjelmaa

3.2 Microsoft Active Directory

AD on käytössä useilla yrityksillä, jotta yrityksen tietohallinto voi hallinnoida käyttäjien käyttäjätietoja ja -tunnuksia. Tässä yhteydessä käyttäjällä ei tarkoiteta henkilöä itseään ja hänen henkilötietojansa, vaan käyttäjällä viitataan tiliin, jota hän käyttää kirjautuakseen työtietokoneelle ja jolla henkilö käyttää yrityksen tietoteknisiä palveluja. AD:hen voidaan tallentaa käyttäjästä muun muassa nimi, katuosoite jos yrityksellä on esimerkiksi useampia toimistoja, sähköpostiosoite, käyttäjätunnus, puhelinnumero, esihenkilötieto, työsuhteen loppumispäivämäärä ja monia muita käyttäjätilien hallinnan ja organisoimisen kannalta relevantteja tietoja.

Käyttäjätilien hallinta on merkittävä osa AD:n hallinnasta vastaaville henkilöille, vaikka organisaatiossa olisi vain yksi toimialue (domain). Mitä suurempi organisaatio on kyseessä, sitä todennäköisemmin AD:ssa on enemmän toimialueita, laajempia puurakenteita ja enemmän ryhmiä, jotta erilaisten käyttäjien organisoiminen olisi helpompaa. (Cyr & Hunter 2011, 246.) Organisaatioyksiköitä voi olla useampia ja organisaatioyksiköiden sisällä voi vielä olla spesifioitumpia ryhmiä, liittyen esimerkiksi maahan, paikkakuntaan tai osastoon, jossa henkilö työskentelee (KUVIO 1).



KUVIO 1. Esimerkki organisaation yksiköistä ja ryhmistä.

3.3 CSV-tiedostot

Comma-separated Values (CSV) on tiedostomuoto, joka on käytännössä tekstimuotoinen taulukko. Yhdellä rivillä on yksi tietue, jokaisella rivillä on yhtä monta kenttää ja kentät on oletuksena eroteltu nimensä mukaisesti pilkulla. Kuvassa 3 ensimmäisellä rivillä näkyvän otsikkorivin käyttö on valinasta, mutta oletuksena ensimmäisellä rivillä on aina kenttien otsikot. Otsikkorivin olemassaolon tai puuttumisen voi ilmoittaa erillisellä parametrilla. (Mitlohner, Neumaier, Umbrich & Polleres 2016, 72.) CSV-tiedosto voidaan avata eri sovelluksilla, esimerkiksi Notepad++, Microsoft Word, tai Microsoft Excel -ohjelmilla. Excelillä avattuna tiedostosta ei kuitenkaan automaattisesti muodostu taulukko, vaan koko tietue kenttineen on rivin ensimmäisessä solussa.



KUVA 3.CSV-tiedosto avattuna Notepad++-ohjelmalla

Riutta (2022) ohjeistaa videollaan, kuinka CSV-tiedoston data voidaan tuoda Exceliin niin, että tiedot voidaan nähdä valmiiksi taulukoituna (KUVA 4). Taulukon muotoilua voidaan muuttaa tuontivaiheessa ja päivitettyt tiedot voidaan viedä uuteen CSV-tiedostoon. Taulukkomuotoista dataa on ihmisen selkeämpi käsitellä ja taulukkoon tehdyt muutokset eivät muuta alkuperäisen CSV-tiedoston dataa. Jos alkuperäiseen CSV-tiedostoon tehdään muutoksia, tiedot tulevat Excel-taulukkoon, kun sitä päivitetään. Excel-taulukon data voidaan viedä uuteen CSV-tiedostoon, jolloin alkuperäinen tiedosto jää sellaiseksi kuin se oli ja uudessa tiedostossa on päivitettyt tiedot. Näin vältetään tahaton alkuperäisen datan tuhoaminen tai häviäminen. (Riutta 2022.)

AccountExpirationDate	company	department	displayName	number	givenName	mail	manager	sAMAccountName	sn	title
29.2.2023	Yritys A Oy	Kokkolan toimipiste	Majja Mehiläinen	2468	Majja	majja.mehilainen@yritysa.com	esimerkki.esihenkilö majja.mehilainen	Mehiläinen	CEO	
	Yritys A Oy	Riihimäen toimipiste	Kimmo Kuu		Kimmo	kimmo.kuu@yritysa.com	esimerkki.esihenkilö kimmo.kuu	Kuu	CEO	

Kuva 4. CSV-tiedosto tuotuna Exceliin

CSV-tiedostojen hyvinä puolina pidetään niiden yksinkertaisuutta ja helppoa siirrettävyyttä toisille alustoille, mutta standardeista huolimatta niistä on olemassa yhteensopimattomia muunnelmia. Vaikka oletuksena kenttien erottimena on pilkku, voidaan kentät erottaa toisistaan myös puolipisteillä, välilyönneillä, sarkainmerkeillä tai millä tahansa muilla merkeillä. Kentissä olevat tiedot voidaan myös laittaa lainausmerkkien tai heittomerkkien sisään, ettei kentässä olevat merkit sekoitu erottimena toimiviin merkkeihin, tai tiedot voivat olla heittomerkkien sisällä myös ilman mitään syytä. Standardeista poikkeavien muunnelmien olemassaolon vuoksi tiedostojen manuaalinen tarkistus on tarpeellista, ennen kuin sen sisältöä käytetään tietokoneen tekemään analysointiin. (van den Burg, Nazábal & Sutton 2019, 2.)

3.4 PowerShell

PowerShell on Windows-käyttöjärjestelmiin kehitetty komentotulkki, jonka avulla voidaan ajaa skriptejä eli merkkipohjaisia ohjelmia. Cyr ja Hunter (2011, 5) tuovat kirjassaan esille PowerShellin

ominaisuuksia, jotka hallinnollisen hyödyllisyyden kannalta tekevät siitä erinomaisen komentotulkin. PowerShell-komentotulkin voidaan ajatella olevan edistyneempi versio Windowsin Command Prompt (cmd) oletuskomentotulkista, edistyneemmän siitä tekee muun muassa jäsennellympi muoto tulostetuissa tiedoissa. Jäsennellympi muoto taas johtuu siitä, että PowerShell ei ole muiden komentotulkien tapaan tekstipohjainen, vaan PowerShell perustuu .NET Frameworkiin ja tekstin käyttämisen sijaan se ottaa vastaan .NET-objekteja syötteenä ja palauttaa .NET-objekteja tulosteena. (Cyr & Hunter 2011, 5–6.)

PowerShell-komentotulkissa on valmiina moduuli, joka sisältää cmdlet-komentoja AD käyttäjähallintaa varten. Kaikissa cmdlet-komennoissa on joukko parametreja, joilla voi viitata käyttäjä- ja ryhmäobjektien yleisiin attribuutteihin. (Cyr & Hunter 2011, 44.) Esimerkiksi Get-ADUser-komennolla voidaan hakea tiedot yhdestä tai useammasta käyttäjästä ja hakua voidaan rajata esimerkiksi niin, että haetaan kaikki käyttäjät tietyistä ryhmästä. (KUVA 2). Käyttäjiä voi hakea myös millä tahansa muulla tiedolla, joka AD:ssa on, kuten sähköpostilla tai puhelinnumerolla, tai voidaan hakea kaikki ne käyttäjät, joilla on vaikka sama sukunimi.

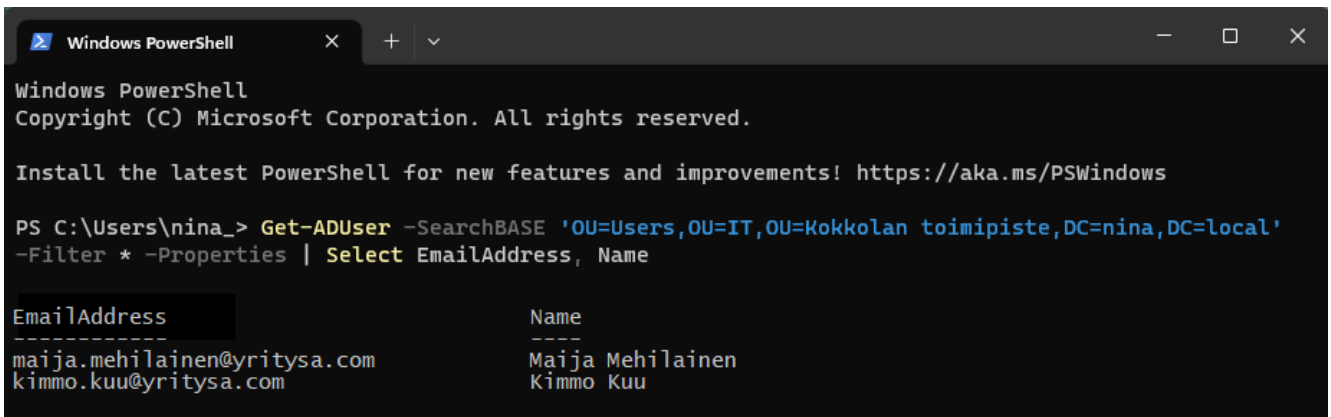
```
PS C:\Users\Administrator\Desktop> Get-ADUser -Filter * -SearchBase "OU=Users,OU=IT,OU=Kokkolan toimipiste,DC=nina,DC=local"

DistinguishedName : CN=Maija Mehiläinen,OU=Users,OU=IT,OU=Kokkolan toimipiste,DC=nina,DC=local
Enabled           : True
GivenName        : Maija
Name             : Maija Mehiläinen
ObjectClass      : user
ObjectGUID       : 99896a50-f280-4e84-9148-3c2ceaa5b843
SamAccountName   : maija.mehilainen
SID              : S-1-5-21-4278121371-3265755173-3508139252-1104
Surname          : Mehiläinen
UserPrincipalName : maija.mehilainen@nina.local
```

KUVA 5. Kaikkien käyttäjien haku PowerShell cmdlet-komennolla tietyistä ryhmästä

AD:n käyttäjähallinnassa paljon käytetty cmdlet-komento on Import-CSV, jolle syötetään parametriksi haluttu CSV-tiedosto ja jonka avulla saadaan PowerShell lukemaan CSV-tiedostoa oikein. PowerShell lukee CSV-tiedoston ensimmäiseltä riviltä otsikot tietueiden kentille ja seuraavilta riveiltä tiedot näille kentille. ForEach-Object komento taas käy läpi jokaisen rivin ja kenttiin voidaan viitata niiden otsikoilla, jotka määriteltiin ensimmäisellä rivillä. Esimerkiksi useiden uusien käyttäjien luominen kerralla on verrattain suoraviivaista Import-CSV ja ForEach-Object komentojen avulla. CSV-tiedostoon lisätään kaikkien uusien käyttäjien tiedot ja sitten tuodaan tiedosto PowerShell-skriptiin, joka luo käyttäjät. (Cyr & Hunter 2011, 250.)

Siinä missä Import-CSV-komennon avulla saadaan CSV-tiedostosta vietyä tietoa AD:hen, onnistuu toiminto myös toisinpäin. Get-ADUser cmdlet-komennolla haettujen käyttäjien tiedot voidaan viedä CSV-tiedostoon Export-CSV-komennolla. Mikäli halutaan käyttäjän kaikkien tietojen sijasta hakea vain tiettyjä tietoja, kuten käyttäjien sähköpostiosoitteita, voidaan Get-ADUser-komennolle syöttää attribuutteja ja niiden avulla rajata tulosta (KUVA 6). (Cyr & Hunter 2011, 292.)



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\nina_> Get-ADUser -SearchBASE 'OU=Users,OU=IT,OU=Kokkolan toimipiste,DC=nina,DC=local'
-Filter * -Properties | Select EmailAddress, Name

EmailAddress                                     Name
-----
maija.mehilainen@yritysa.com                    Maija Mehilainen
kimmo.kuu@yritysa.com                          Kimmo Kuu

```

KUVA 6. Tietystä ryhmässä olevien käyttäjien sähköpostiosoitteiden haku

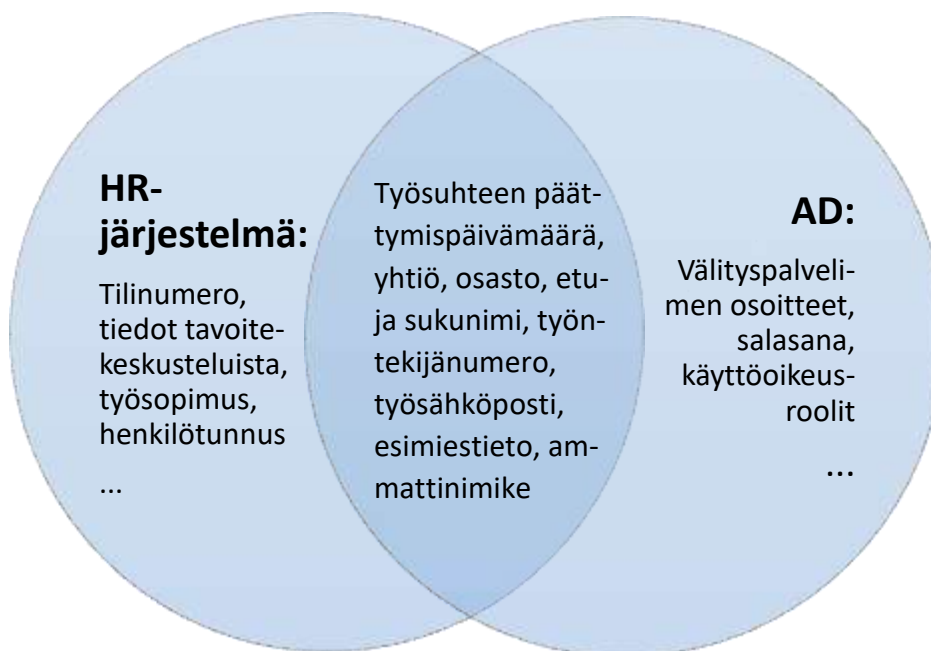
Mikäli AD:n tietoja haetaan toiselta palvelimelta, Import-CSV ja Export-CSV -komentojen sijasta voidaan käyttää Csvde-komentoa. Csvde-komennolle syötetään tarvittavat parametrit, kuten palvelimen nimi mistä tiedostot haetaan, onko kyseessä tiedostojen vienti vai tuonti ja tiedoston nimi, johon haetut tiedot halutaan viedä tai mistä ne halutaan tuoda. Csvde-komento käsittelee vain CSV-muotoisia tiedostoja, oli kyse tiedostoon viemisestä tai tiedostosta tuomisesta. (Microsoft 2016.)

PowerShell-komentotulkissa voidaan ajaa komentoja rivi kerrallaan, kuten kuvassa 6. Monissa tapauksissa on kuitenkin hyödyllisempää kirjoittaa skripti, jossa on kaikki tarvittavat komennot omilla riveillään siinä järjestyksessä, kun ne halutaan suorittaa. Komennot kirjoitetaan skriptiin samaan tapaan kuin ne kirjoitettaisiin suoraan komentotulkkiin ja skriptissä voidaan käyttää loogisia operaattoreita aivan kuten rivi kerrallaan suoritettavissa komennossakin. (Cyr & Hunter 2011, 19.) PowerShell ISE ja Notepad++ ovat tähän tarkoitukseen hyviä tekstieditoreja, sillä niihin kirjoitettaessa editori osaa tarjota olemassa olevia komentoja sitä mukaa, kun komentoa aletaan kirjoittaa. Nämä editorit myös tallentavat eräajotiedostot automaattisesti oikeassa .ps1-tiedostomuodossa PowerShellillä ajettavaksi. Cyr & Hunter (2011, 19–20) kertovat kirjassaan, kuinka skripti voidaan kuitenkin luoda millä tahansa tekstieditorilla ja tallentaa tiedostopäätteellä .ps1, jolloin siitä tulee suoritettava eräajotiedosto.

PowerShell-skripti voidaan suorittaa aina PowerShell-istunnon aikana, mutta skripti voidaan suorittaa myös avaamatta PowerShellia. Skripti voidaan ajaa avaamatta ohjelmaa esimerkiksi tietokoneelle tallennettua .ps1 -tiedostoa tuplaklikkaamalla PowerShell aukeaa automaattisesti, skripti ajetaan ja sen jälkeen PowerShell sulkeutuu. Tehtävien ajoituksella voidaan ajastaa PowerShell.exe aukeamaan ja suorittamaan haluttu eräajotiedosto. (Cyr & Hunter 201, 35.) Komentokehotteelle voidaan antaa PowerShell-käsky ja parametrinä polku, josta haluttu skripti löytyy (Batch File to run PowerShell ps1 script 2017).

4 INTEGRAATIO AD:N JA HR-JÄRJESTELMÄN VÄLILLÄ

Yrityksen A tietohallinto ylläpitää kaikkien työntekijöiden käyttäjätilejä Microsoftin käyttäjätietokannassa, AD:ssa. Yrityksen A HR-osastolla on käytössään oma järjestelmä, jossa puolestaan ylläpidetään työntekijöiden työsuhteen kannalta relevantteja henkilötietoja. Osa näistä tiedoista täytyy olla yhteneväisiä molemmissa järjestelmissä, osa tiedoista taas esiintyy vain toisessa järjestelmässä eikä niitä tarvita toisessa (KUVIO 2). Ilman integraatiota näiden kahden järjestelmän välillä tiedot täytyy manuaalisesti muuttaa molempiin. HR-osasto hallinnoi ja ylläpitää omaa järjestelmäänsä ja IT-osasto omaansa. Ennen integraation toteutusta yhden työntekijän tietojen muutos vaati siis työtä kahdelta osastolta; ensin toinen päivitti omaan järjestelmäänsä tiedot, sitten teki siitä kirjallisen ilmoituksen toiselle osastolle ja toisella osastolla joku taas päivitti omaa järjestelmäänsä.

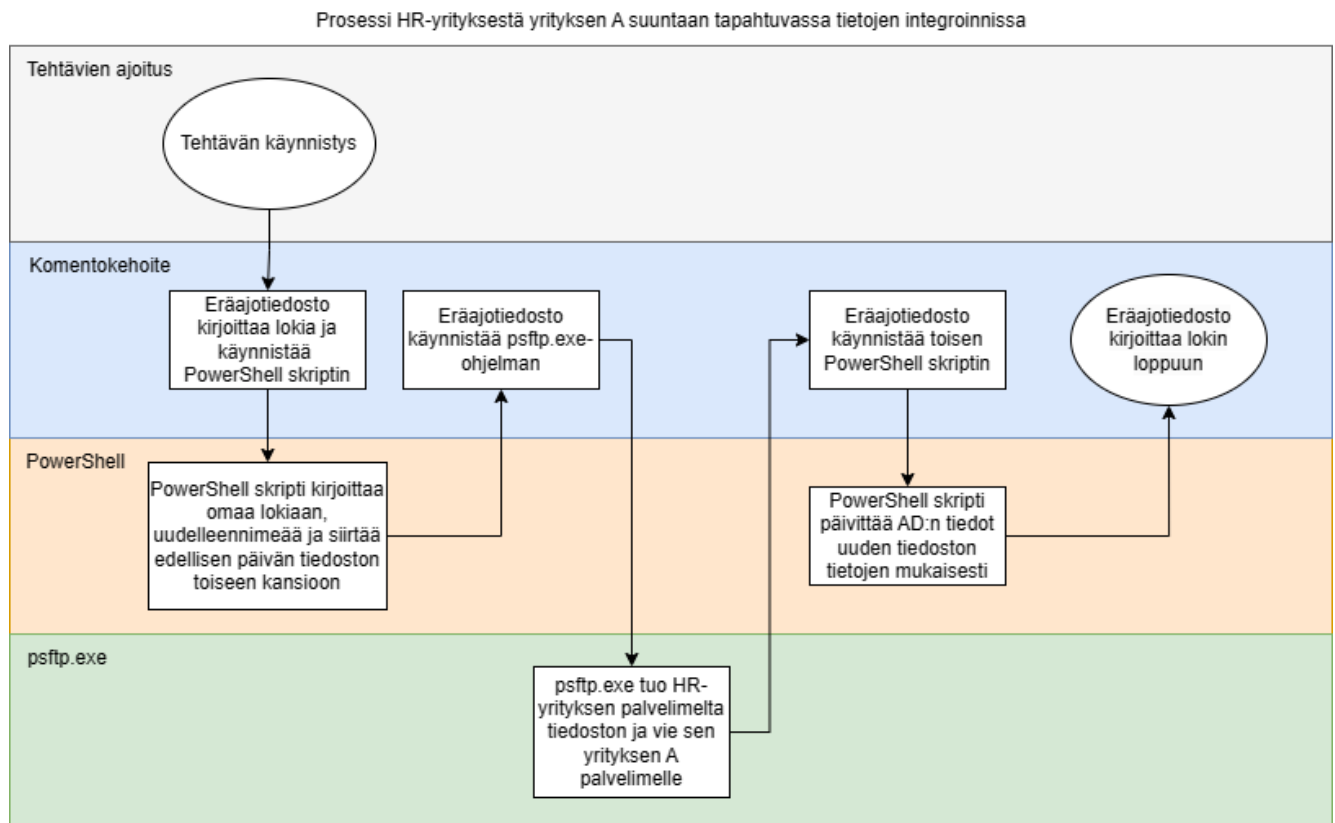


KUVIO 2. Järjestelmien yhteneväiset tiedot

Vapautuneiden resurssien lisäksi integraatio poistaa inhimillisen virheen mahdollisuuden ja varmistaa, että tiedot ovat yhteneväiset molemmissa järjestelmissä. Aikaisemmin tietojen päivytyksessä oli manuaaliryöstä johtuvaa vaihtelevaa viivettä, mutta integraation myötä tiedetään tarkasti, kuinka kauan aikaa tietojen päivittämiseen järjestelmästä toiseen kuluu. Tietoturvan kannalta tärkeimmäksi nousee työntekijän työsuhteen loppumispäivämäärän siirtyminen HR-järjestelmästä AD:hen. HR-osasto tietää

ensimmäisten joukossa, kun työntekijä irtisanoo itsensä. Kun he laittavat järjestelmään päättymispäivämäärän, siirtyy tili disabled-tilaan tuona päivämääränä ja käyttäjätunnus lakkaa automaattisesti toimimasta tuon päivän jälkeen eikä irtisanoutuneella työntekijällä sen jälkeen ole enää pääsyä yrityksen A järjestelmiin.

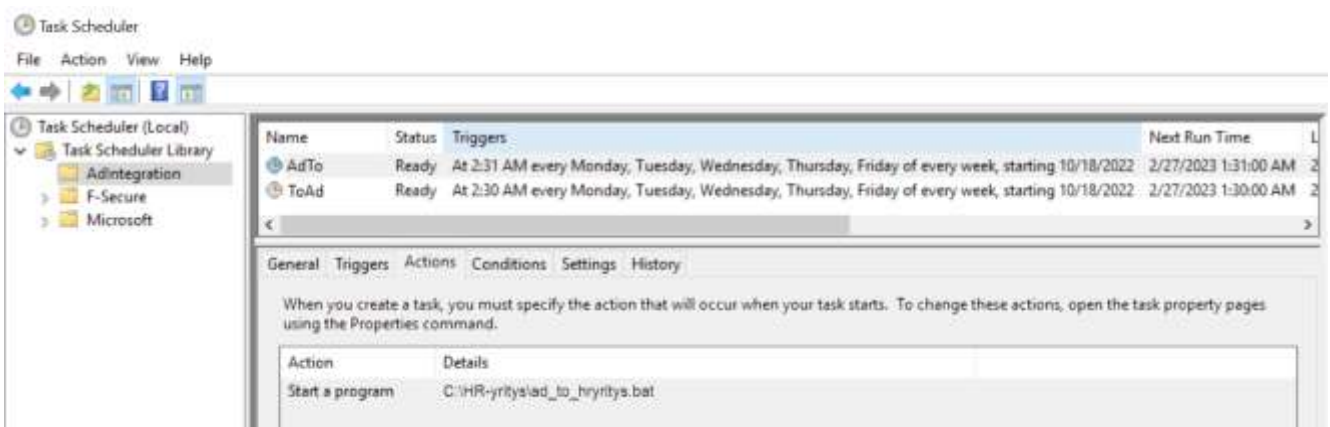
Tietojen integrointiprosessi on pääpiirteittäin samanlainen molempiin suuntiin. HR-yritykseltä yrityksen A suuntaan tapahtuvassa integraatiossa Tehtävien ajoitus käynnistää eräajotiedoston, joka siirtää edellisen päivän tiedoston arkistoon ja sen jälkeen käynnistää psftp.exe-ohjelman. Psftp.exe-ohjelma noutaa uuden tiedoston HR-yrityksen palvelimelta ja syöttää sen parametrina PowerShell-skriptille, joka puolestaan päivittää tiedot AD:hen (KUVIO 3). Yritys A:lta HR-yritykselle päin prosessi eroaa vain siltä osin, että yritys A:n ei tarvitse päivittää tietoja mihinkään. Tehtävä käynnistetään, edellisen päivän tiedosto siirretään arkistoon, AD:sta haetaan uudet tiedot ja psftp.exe-ohjelmalla viedään ne HR-yrityksen palvelimelle. HR-yritys vastaa tietojen päivytyksestä omaan järjestelmäänsä.



KUVIO 3. Integrointiprosessi HR-yritykseltä yrityksen A suuntaan

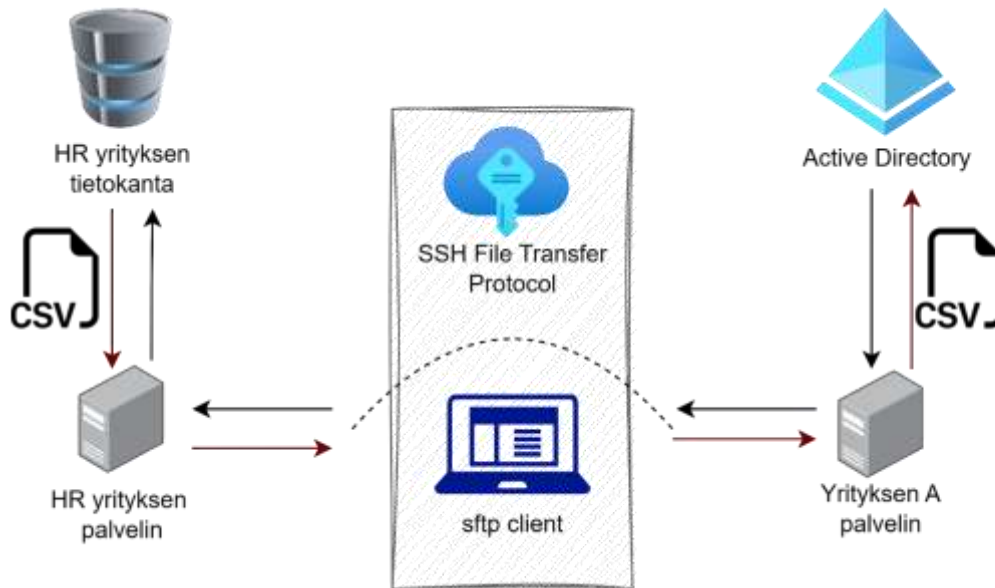
4.1 Eräajotiedostojen suoritus

Tietojen integrointi on ajastettu tapahtumaan kerran vuorokaudessa, eli jos jompaankumpaan järjestelmään tehdään muutoksia päivän aikana, siirtyvät päivitettyt ja uudet tiedot toiseen järjestelmään seuraavana yönä. Tietojen siirto ei tässä toteutuksessa ole tarpeellista viikonloppuisin, sillä riittää, että viikonloppuna tehdyt muutokset näkyvät seuraavana arkipäivänä. Tehtävien ajoitus on ajastettu suorittamaan Komentokehoteella kaksi eräajotiedostoa (KUVA 7); toinen johtaa tietojen viemiseen AD:sta HR-järjestelmään ja toinen tietojen tuomiseen HR-järjestelmästä AD:hen.



KUVA 7. Ajoitetut tehtävät Tehtävien ajoitus -ohjelmassa

Eräajotiedosto (KUVA 2) kirjoittaa yrityksen A palvelimelle lokitiedostoa, josta käyvät ilmi päivämäärä ja kellonaika, jolloin tiedoston suoritus on alkanut, mahdolliset ajon aikana tapahtuneet virheet sekä päivämäärä ja kellonaika, jolloin suoritus on päättynyt. Lokitiedoston avulla voidaan helposti jälkikäteen seurata ja selvittää, milloin tiedosto on ajettu, kuinka usein ja paljon ajoissa on virheitä ja kuinka kauan ajoon kuluu aikaa. Molemmat Tehtävien ajoituksella ajettavat tiedostot kirjoittavat omaa lokiaan, jotta on helpompi seurata kumpaankin suuntaan liikkuvien tietojen toimintaa erikseen. Tiedostoja siirretään molempiin suuntiin, sekä HR-yrityksen palvelimelta heidän järjestelmänsä haettuja tietoja sisältävä tiedosto yrityksen A palvelimelle että toisinpäin yrityksen A palvelimelta AD:sta haettuja tietoja sisältävä tiedosto HR-yrityksen palvelimelle (KUVIO 3).



KUVIO 4. Tiedostojen siirto palvelimien välillä

4.2 AD:sta HR-järjestelmään

Komentokehotteella suoritettava eräajotiedosto kirjoittaa lokitietoja joka päivä samaan tiedostoon (KUVA 8) edellisen päivän tietojen perään. Ajan myötä samassa tiedostossa on nähtävillä pidemmän aikavälin lokitiedot. AD:sta tiedot haetaan PowerShellillä Csvde-komentoa käyttäen. Csvde-komennolle syötetään parametreina tarvittavia tietoja, kuten tiedoston nimi, johon haetut tiedot viedään, ja tiedot, jotka sen halutaan hakevan AD:sta. Kun uusi, kaikkien käyttäjien tiedot sisältävä tiedosto on luotu yrityksen A palvelimelle, kutsuu eräajotiedosto psftp.exe-ohjelmaa, jolle syötetään parametreina käyttäjätunnus ja salasana, joilla ohjelma saa pääsyn HR-yrityksen palvelimelle. Parametrinä syötetään myös skripti, jolla se vie valitusta kansioista kyseisen tiedoston, tässä tapauksessa PowerShellin luoman CSV-tiedoston HR-yrityksen palvelimelle. HR-yritys hoitaa tiedostojen päivityksen heidän järjestelmänsä CSV-tiedostossa olevien tietojen mukaisesti.

```

ad_to_hryritys.log - Notepad
File Edit Format View Help
Starting AD to HR-yritys export...
Mon 02/27/2023
02:31 AM
Running CSVDE...
Connecting to "(null)"
Logging in as current user using SSPI
Exporting directory to file ad_to_hryritys.csv
Searching for entries...
Writing out entries
Export Completed. Post-processing in progress...
21 entries exported

The command has completed successfully

-----
Copying file...
Mon 02/27/2023
02:31 AM
Done.
-----

```

KUVA 8. Lokitiedosto AD:sta HR-yritykselle päin

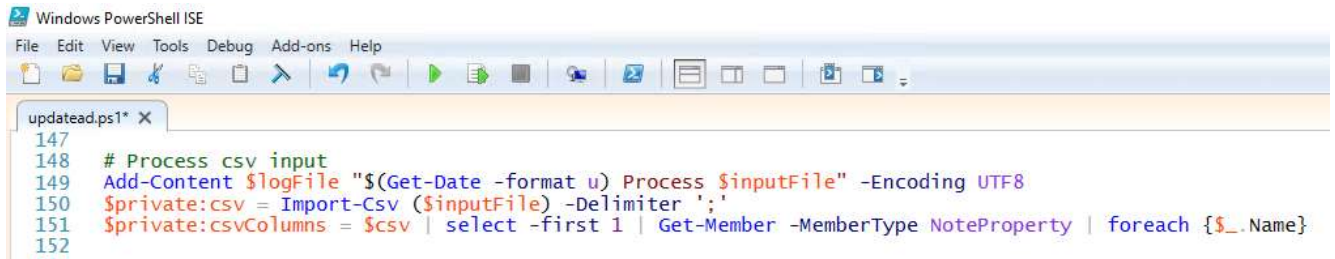
4.3 HR-järjestelmästä AD:hen

Lokitiedoston kirjoituksen lisäksi eräajotiedosto kutsuu PowerShell-skriptiä siirtääkseen edellisen päivän tiedoston arkistohakemistoon. HR-yrityksen palvelimelta haettavat tiedostot ovat joka päivä saman nimisiä, joten on tärkeää, että edellinen tiedosto siirretään toiseen hakemistoon ja muutetaan nimeä jollain tapaa. Mikäli nimeä ei muuteta, uusi tiedosto kirjottaa vanhan tiedoston päälle, eikä historia jää talteen. Tässä toteutuksessa tiedoston nimen perään lisätään päivämäärä, jolloin tiedosto on luotu. Näin muodostuu hakemisto, jossa on oma tiedosto jokaiselle päivälle, jolloin tiedostoja on siirretty.

Kun vanha tiedosto on siirretty talteen, kutsuu eräajotiedosto (KUVA 2) psftp.exe-ohjelmaa, joka noutaa uuden käyttäjätietoja sisältävän tiedoston HR-yrityksen palvelimelta. Rivillä 16 psftp.exe-ohjelmalle syötetään parametreina käyttäjätunnus ja salasana, joilla ohjelma saa pääsyn kohdepalvelimelle. Lisäksi yhtenä parametrina on skripti, jolla se noutaa tietystä kansioista tietyn tiedoston ja palauttaa sen haluttuun kansioon yrityksen A palvelimelle. Uuden tiedoston haun jälkeen eräajotiedosto kutsuu toista PowerShell-skriptiä. PowerShell-komennolle annetaan InputFile-parametriksi äsken haetun uusimman käyttäjätietoja sisältävän CSV-tiedoston koko hakemistopolku.

PowerShell-skriptissä määritellään Import-CSV-komennolla CSV-tiedosto, jota halutaan käsitellä (KUVA 9). Käsiteltävän tiedoston hakemistopolku on inputFile-muuttujassa, joka on annettu skriptille parametrina sen kutsumisvaiheessa. Delimiter-parametrilla kerrotaan, että CSV-tiedostossa käytetty kenttien erotin on oletuserottimesta poiketen puolipiste. CSV-tiedoston (KUVA 3) sisältö PowerShell-

muuttujassa on kuvan 10 mukainen. Jos tulostetta ajatellaan taulukkona, ensimmäisessä kolumnissa on kentän otsikko ja toisessa kolumnissa kentän sisältö. CSV-tiedostossa ollut yksi tietue näkyy siis PowerShellin tulosteessa useammalla rivillä ja ennen seuraavaan tietueeseen siirtymistä tulosteessa näkyy yksi tyhjä rivi. csvColumns-muuttujaan tallennetaan kolumnien nimet myöhempää käyttöä varten.

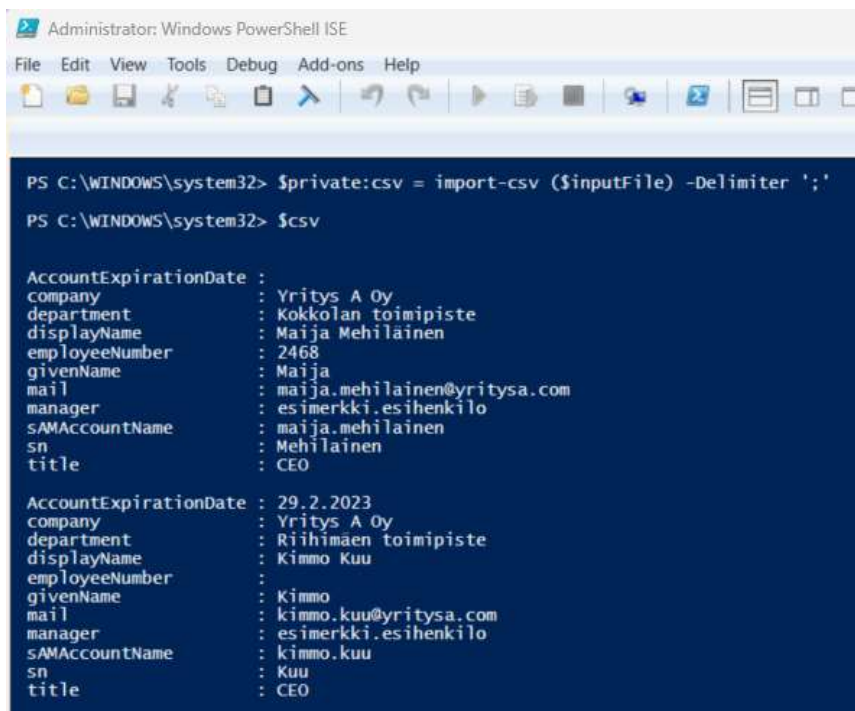


```

Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
updatead.ps1* X
147
148 # Process csv input
149 Add-Content $logfile "$($Get-Date -format u) Process $inputFile" -Encoding UTF8
150 $private:csv = Import-Csv ($inputFile) -Delimiter ';'
151 $private:csvColumns = $csv | select -first 1 | Get-Member -MemberType NoteProperty | foreach {$_.Name}
152

```

KUVA 9. CSV-tiedoston prosessointi



```

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
PS C:\WINDOWS\system32> $private:csv = import-csv ($inputFile) -Delimiter ';'
PS C:\WINDOWS\system32> $csv

AccountExpirationDate :
company                : Yritys A Oy
department             : Kokkolan toimipiste
displayName            : Maija Mehiläinen
employeeNumber        : 2468
givenName              : Maija
mail                   : maija.mehilainen@yritysa.com
manager               : esimerkki.esihenkilo
sAMAccountName        : maija.mehilainen
sn                    : Mehiläinen
title                 : CEO

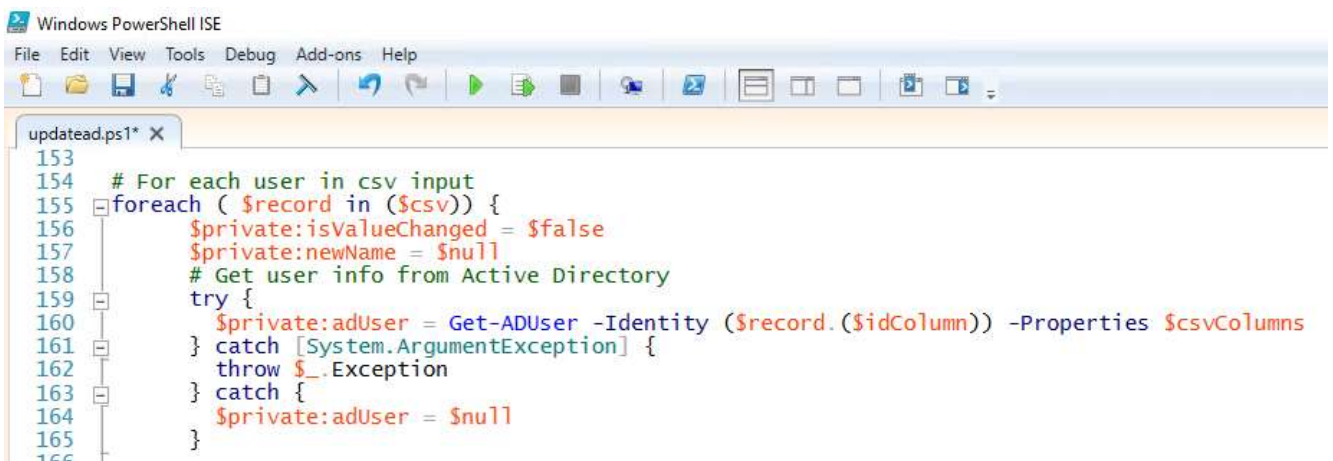
AccountExpirationDate : 29.2.2023
company                : Yritys A Oy
department             : Riihimaan toimipiste
displayName            : Kimmo Kuu
employeeNumber        :
givenName              : Kimmo
mail                   : kimmo.kuu@yritysa.com
manager               : esimerkki.esihenkilo
sAMAccountName        : kimmo.kuu
sn                    : Kuu
title                 : CEO

```

KUVA 10. CSV-tiedoston sisältö PowerShell-tulosteessa

4.3.1 Käyttäjätietojen päivitys

PowerShell ottaa käsittelyyn yhden tietueen kerrallaan ja käyttää sAMAccountName-attribuuttia yksilöllintietona. sAMAccountName toimii käyttäjien kirjautumisnimenä ja sen tulee olla yksilöllinen kaikilla käyttäjillä saman domainin, eli verkkotunnuksen sisällä. PowerShell käyttää Get-ADUser cmdlet-komentoa käyttäjän tietojen hakemiseen, identifioi haettavan käyttäjän sAMAccountName-attribuuttia hyödyntäen ja tallentaa tiedot adUser-muuttujaan (KUVA 11). AD:sta ei tarvitse hakea kyseisen käyttäjän kaikkia tietoja, vaan parametriksi syötetään aikaisemmin csvColumns-muuttujaan tallennetut kolumnien nimet, jolloin AD:sta haetaan vain kolumneja vastaavat tiedot. HR-yritys ei siis vie kaikkia käyttäjistä kaikkia tietoja CSV-tiedostoon, eikä AD:sta tuoda kaikkia käyttäjistä kaikkia tietoja, vaan tässä toteutuksessa käsitellään ainoastaan niitä tietoja, jotka ovat yhteneväiset molemmissa järjestelmissä (KUVIO 2). Ohjelma vertaa CSV-tiedostossa olevia tietoja yksi kerrallaan AD:sta haettuihin tietoihin; mikäli tiedot vastaavat toisiaan, siirtyy skripti seuraavaan tietueeseen ja kaikki tietueet läpikäytyään seuraavaan käyttäjään. Mikäli tiedot eivät vastaa toisiaan, päivittää PowerShell-skripti AD:ta.



```

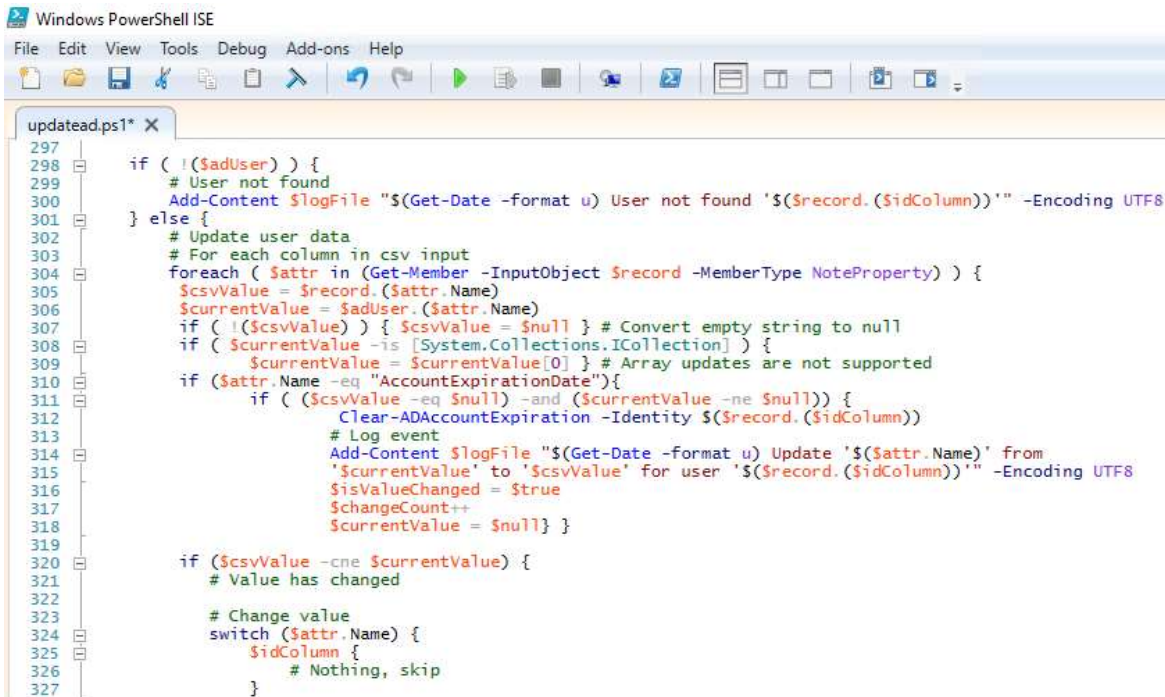
153
154 # For each user in csv input
155 foreach ( $record in ($csv)) {
156     $private:isValueChanged = $false
157     $private:newName = $null
158     # Get user info from Active Directory
159     try {
160         $private:adUser = Get-ADUser -Identity ($record.($idColumn)) -Properties $csvColumns
161     } catch [System.ArgumentException] {
162         throw $_.Exception
163     } catch {
164         $private:adUser = $null
165     }
166 }

```

KUVA 11. PowerShell-skriptin osa, jossa käydään läpi CSV-tiedoston tietueet

csvValue-muuttujaan tallennetaan CSV-tiedostosta löytynyt tämänhetkinen tieto ja currentValue-muuttujaan AD:sta löytynyt tieto (KUVA 12). Rivillä 320 PowerShell vertaa näitä kahta muuttujaa keskenään. Mikäli tiedot eivät vastaa toisiaan, päivittää PowerShell AD:hen sen tiedon, joka CSV-tiedostossa oli. Tiedon päivittäminen on poikkeuksia lukuun ottamatta varsin suoraviivaista. Jos esimerkiksi käyttäjän puhelinnumero on vaihtunut, voidaan uusi numero päivittää AD:hen ilman muita lisätoimenpiteitä tai varmistuksia. Jos taas käyttäjän esihenkilö on vaihtunut, täytyy ennen päivittämistä

Get-ADUser-komennolla etsiä uuden esihenkilön tiedot. HR-järjestelmässä esihenkilö merkitään sA-MAccountName-attribuutilla, mutta AD:hen se merkitään DistinguishedName-attribuutilla. Esihenkilö täytyy siis ensin hakea AD:sta sAMAccountName-tiedolla ja löytyneen käyttäjän DistinguishedName-tieto otetaan talteen ja päivitetään se käsittelyssä olevan henkilön esihenkilötiedoksi.



```

297
298     if ( !$adUser ) {
299         # User not found
300         Add-Content $LogFile "$(Get-Date -format u) User not found '$($record.($idColumn))'" -Encoding UTF8
301     } else {
302         # Update user data
303         # For each column in csv input
304         foreach ( $attr in (Get-Member -InputObject $record -MemberType NoteProperty) ) {
305             $csvValue = $record.($attr.Name)
306             $currentValue = $adUser.($attr.Name)
307             if ( !$csvValue ) { $csvValue = $null } # Convert empty string to null
308             if ( $currentValue -is [System.Collections.ICollection] ) {
309                 $currentValue = $currentValue[0] # Array updates are not supported
310             }
311             if ( $attr.Name -eq "AccountExpirationDate" ) {
312                 if ( ($csvValue -eq $null) -and ($currentValue -ne $null) ) {
313                     Clear-ADAccountExpiration -Identity $($record.($idColumn))
314                     # Log event
315                     Add-Content $LogFile "$(Get-Date -format u) Update '$($attr.Name)' from
316                         '$currentValue' to '$csvValue' for user '$($record.($idColumn))'" -Encoding UTF8
317                     $isValueChanged = $true
318                     $changeCount++
319                     $currentValue = $null }
320             }
321             if ( $csvValue -cne $currentValue ) {
322                 # Value has changed
323                 # Change value
324                 switch ( $attr.Name ) {
325                     $idColumn {
326                         # Nothing, skip
327                     }

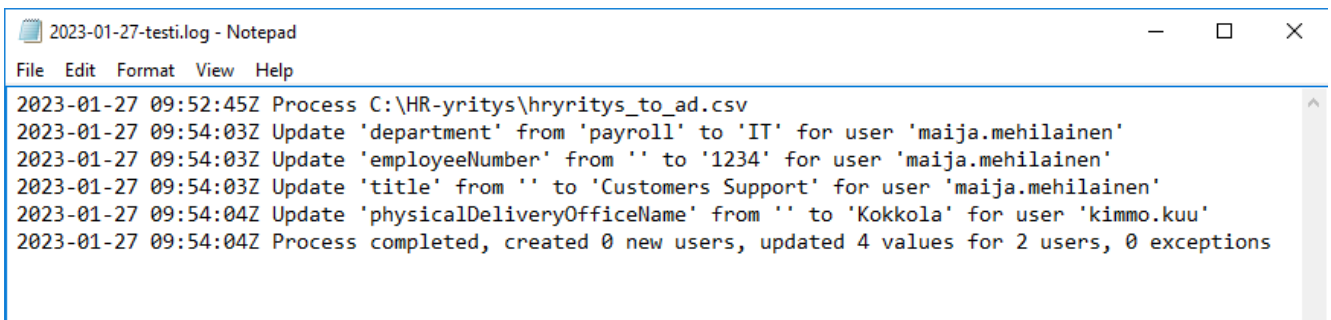
```

KUVA 12. PowerShell-skripti käyttäjän tietojen päivittämiseen

Työsuhteen päättymispäivämäärän päivitys poikkeaa myös hieman muiden tietojen päivityksestä (LIITE 1): Jos kummassakaan järjestelmässä ei ole dataa AccountExpirationDate-attribuutissa, muutoksia ei tehdä. Jos CSV-tiedostossa on päättymispäivä ja AD:ssa ei ole, päivitetään tiedoston mukainen päivämäärä AD:hen. Jos molemmissa on tulevaisuudessa oleva päivämäärä, päivitetään AD:hen CSV-tiedoston mukainen päivämäärä. Jos AD:ssa on historiassa oleva päivämäärä ja CSV-tiedostossa tulevaisuudessa oleva päivämäärä, päivittyy tieto kyllä AD:hen, mutta tili pysyy disabled-tilassa, eli poissa käytöstä siihen asti, että se manuaalisesti laitetaan enabled-tilaan, eli otetaan takaisin käyttöön. Jos taas AD:hen on aikaisemmin laitettu AccountExpirationDate, joka on sittemmin otettu HR-järjestelmästä pois, ei päivämäärän päivitys onnistu, koska päivämääräksi ei voida laittaa NULL-tietoa. Tästä syystä PowerShell tekee ensin tarkistuksen, jos HR-järjestelmässä ei ole päivämäärää ja AD:ssa on, poistetaan AD:sta tieto Clear-ADAccountExpiration cmdlet-komennolla.

4.3.2 Lokitiedoston kirjoitus

PowerShell kirjoittaa omaa lokitiedostoaan myös. Lokitiedoston avulla voidaan seurata tarkkaan, milloin muutokset on tehty ja kuinka kauan muutoksien tekemiseen on kulunut aikaa. Lokista selviää myös, minkä nimisen tiedoston sisältöä on käsitelty ja mitä muutoksia on tehty. PowerShell kirjoittaa lokia sitä mukaa kun muutoksia tehdään. Kuvassa 12 rivillä 315 hyödynnetään csvValue ja currentValue -muuttujia, jotta saadaan lokiin näkyviin AD:ssa ollut vanha AccountExpirationDate-tieto sekä CSV-tiedostossa ollut uusi, päivitetty tieto. Kyseinen skriptin osuus toistuu myöhemmin uudestaan myös muiden tietojen päivityksen kohdalla, jotta nähdään lokissa aina vanha tieto ja uusi tieto. Viimeisenä lokitiedostossa on yhteenveto muutoksista: onko uusia käyttäjiä luotu, montako yksittäistä tietoa on päivitetty, kuinka monen eri käyttäjän tietoja on päivitetty ja kuinka monta virhettä ajon aikana on tapahtunut (KUVA 13).



```

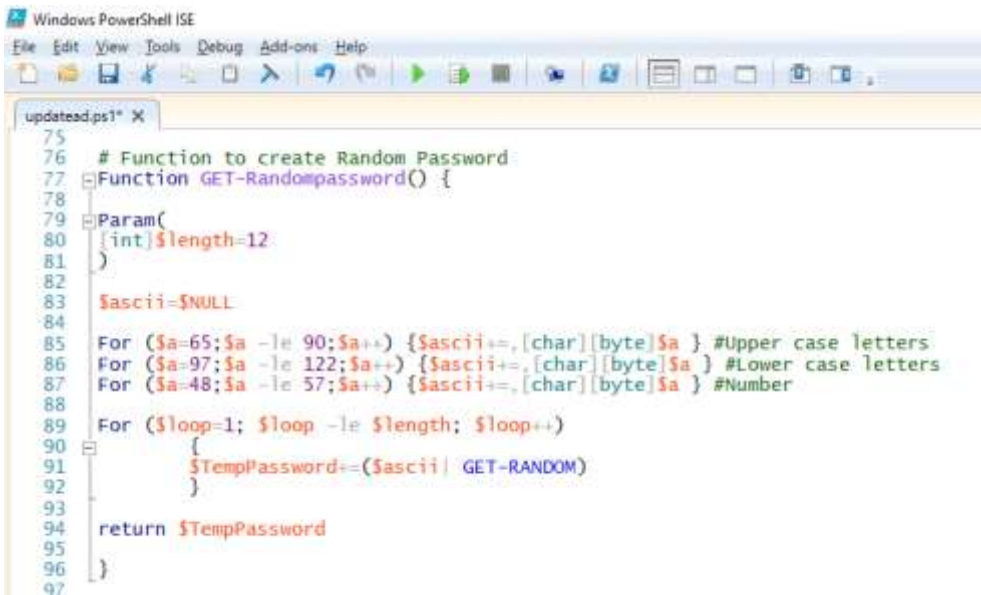
2023-01-27 09:52:45Z Process C:\HR-yritys\hryritys_to_ad.csv
2023-01-27 09:54:03Z Update 'department' from 'payroll' to 'IT' for user 'maija.mehilainen'
2023-01-27 09:54:03Z Update 'employeeNumber' from '' to '1234' for user 'maija.mehilainen'
2023-01-27 09:54:03Z Update 'title' from '' to 'Customers Support' for user 'maija.mehilainen'
2023-01-27 09:54:04Z Update 'physicalDeliveryOfficeName' from '' to 'Kokkola' for user 'kimmo.kuu'
2023-01-27 09:54:04Z Process completed, created 0 new users, updated 4 values for 2 users, 0 exceptions
  
```

KUVA 13. Lokitiedosto PowerShellin AD:hen tekemistä muutoksista

4.3.3 Uuden käyttäjän luonti

PowerShell käyttää yksilöintitietona sAMAccountName-attribuuttia ja etsii sillä käyttäjää AD:sta. Mikäli käyttäjää ei löydy, tallennetaan väliaikaisesti adUser-muuttujaan tiedoksi NULL, kuten kuvassa 11 rivillä 164. Uuden käyttäjän luonti on toteutettu niin, että PowerShell-skripti etsii uutta käyttäjää koskevasta tietueesta käyttäjän etu- ja sukunimen sekä userPrincipalName-attribuutin. Polku AD-ryhmään, johon käyttäjä luodaan, on annettu jo eräajotiedostossa PowerShellin kutsumisvaiheessa (KUVA 2). New-Mailbox cmdlet-komentoa käyttäen luodaan samalla kertaa sekä sähköpostilaatikko uudelle käyttäjälle, että käyttäjätili AD:hen.

Uuden käyttäjän luomisen yhteydessä on asetettava salasana, jolla käyttäjä kirjautuu sisään ensimmäisellä kerralla. Yrityksen A salasanakäytäntöjen mukaisesti käyttäjälle generoidaan 12 merkkiä pitkä salasana, joka sisältää isoja ja pieniä kirjaimia sekä numeroita (KUVA 14). New-Mailbox-komennolle tulee syöttää parametrina myös tieto siitä, mikäli käyttäjä halutaan pakottaa vaihtamaan salasana ensimmäisen kirjautumisen yhteydessä. Oletuksena sitä ei tarvitse vaihtaa, mutta tietoturvasyistä tässä toteutuksessa käyttäjälle annetaan niin sanotusti kertakäyttösalasana, joka täytyy vaihtaa ensimmäisen kirjautumisen yhteydessä.



```

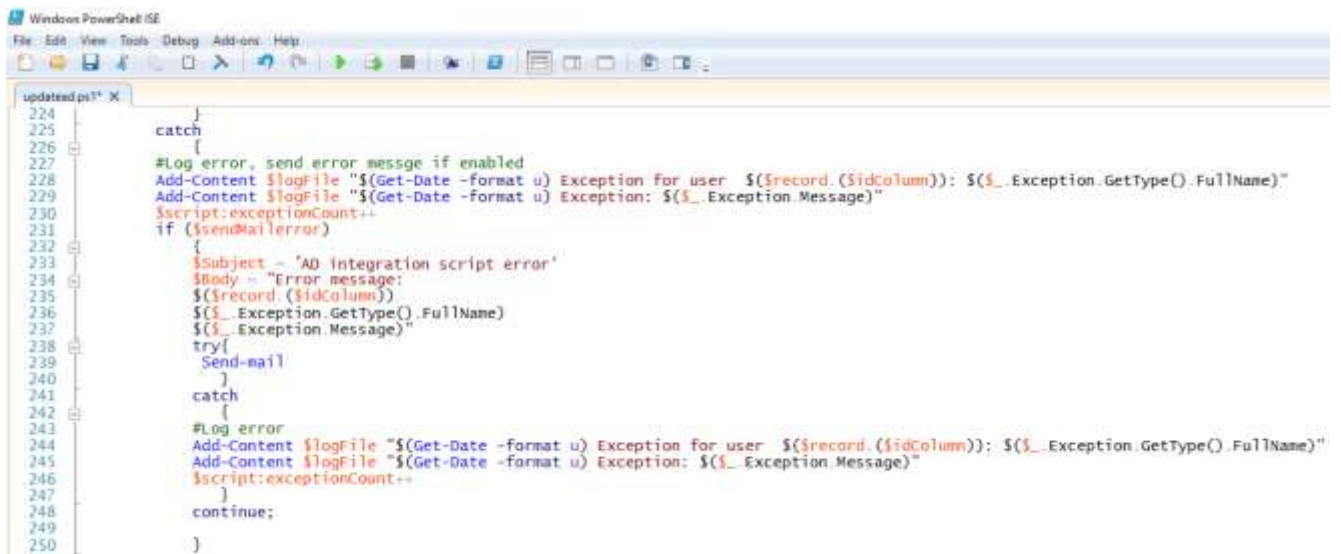
75
76 # Function to create Random Password
77 Function GET-Randompassword() {
78
79 Param(
80 [int]$length=12
81 )
82
83 $ascii=$NULL
84
85 For ($a=65;$a -le 90;$a++) {$ascii+=,[char][byte]$a } #Upper case letters
86 For ($a=97;$a -le 122;$a++) {$ascii+=,[char][byte]$a } #Lower case letters
87 For ($a=48;$a -le 57;$a++) {$ascii+=,[char][byte]$a } #Number
88
89 For ($loop=1; $loop -le $length; $loop++)
90 {
91     $TempPassword+=( $ascii | GET-RANDOM )
92 }
93
94 return $TempPassword
95
96 }
97
  
```

KUVA 14. Satunnaisen salasanan generointi

Uuden käyttäjän luonnin jälkeen skripti etsii uudestaan samaa käyttäjää sAMAccountName-yksilöintitiedolla. Koska käyttäjä on luotu, käyttäjä löytyy nyt tietokannasta ja sen jälkeen skripti päivittää käyttäjälle tiedot samaan tapaan kuin jo olemassa olevien käyttäjien kohdalla. Uuden käyttäjän luonnin yhteydessä skripti lähettää sähköpostin yrityksen A tietohallinnolle, jossa lukee uuden käyttäjän tiedot. Sähköposti tarvitaan, jotta IT-osasto tietää päivittää uudelle käyttäjälle tarvittavat käyttöoikeusroolit. Käyttäjän salasana on myös näkyvillä tässä sähköpostissa, ja IT-osaston tulee välittää se manuaalisesti käyttäjälle tämän ensimmäisenä työpäivänä.

4.3.4 Virheviestin lähetys

Mikäli tietojen päivityksen aikana tapahtuu virhe PowerShell-skriptin suorituksessa, lähetetään siitä yrityksen A tietohallinnolle sähköposti. Mikäli yhden ajon aikana on useita virheitä, lähtee jokaisesta virheestä oma viesti. Viestistä käy ilmi sen käyttäjän sAMAccountName, jonka kohdalla virhe tapahtui, sekä PowerShellin antama virheilmoitus. Sama viesti kirjoittuu myös lokitiedostoon, mutta erillisen sähköpostin lähetys katsottiin tarpeelliseksi, koska lokitiedostoa ei päivittäisellä tasolla lueta.



```

224 }
225 catch
226 {
227     #Log error, send error message if enabled
228     Add-Content $logFile "$(Get-Date -format u) Exception for user $($record.(SidColumn)): $($_.Exception.GetType().FullName)"
229     Add-Content $logFile "$(Get-Date -format u) Exception: $($_.Exception.Message)"
230     $script:exceptionCount++
231     if ($sendMailError)
232     {
233         $Subject = 'AD integration script error'
234         $Body = "Error message:
235         $($record.(SidColumn))
236         $($_.Exception.GetType().FullName)
237         $($_.Exception.Message)"
238         try{
239             Send-mail
240         }
241         catch
242         {
243             #Log error
244             Add-Content $logFile "$(Get-Date -format u) Exception for user $($record.(SidColumn)): $($_.Exception.GetType().FullName)"
245             Add-Content $logFile "$(Get-Date -format u) Exception: $($_.Exception.Message)"
246             $script:exceptionCount++
247         }
248     }
249     continue;
250 }
  
```

KUVA 15. Virheviestin lähetys

5 LOPPUTULOS JA YHTEENVETO

Opinnäytetyössä toteutetun integraation päätavoite oli automatisoida käyttäjien tietojen muutos kahden eri järjestelmän välillä. Tietoja voidaan muuttaa HR-osaston toimesta HR-järjestelmään tai IT-osaston toimesta AD:hen. Kumpaan tahansa järjestelmään tehdyistä muutoksista tuli aikaisemmin ilmoittaa kirjallisesti toiselle osastolle, jotta he voivat tehdä vastaavat muutokset hallinnassansa olevaan järjestelmään. Integraation toteutuksen ja käyttöönoton jälkeen erillisiä ilmoituksia ei enää tarvita, vaan tiedot päivittyvät automaattisesti seuraavana yönä. Olemassa olevien käyttäjien tietojen päivityksen lisäksi integraation myötä uusien käyttäjien luominen ja vanhojen käyttäjätilien käytöstä poisto on automatisoitu suurimmilta osin.

Uuden käyttäjän luonnin yhteydessä käyttäjälle täytyy lähettää salasana ja antaa käyttöoikeusroolit manuaalisesti. Käyttöoikeusroolien automaattinen antaminen ei kyseessä olevan yrityksen toteutuksessa ole mahdollista, sillä tarvittavia käyttöoikeuksia ei voida määrittellä tarkasti. Käyttöoikeudet riippuvat toimipisteestä ja työnkuvasta, mutta niitä on silti hajautettu niin, että vaikka kaksi henkilöä on samassa paikassa samassa roolissa, niin toisella voi olla eri oikeudet kuin toisella. Jotta käyttäjille annettaisiin vai ne käyttöoikeudet, jotka hän tarvitsee eikä mitään ylimääräistä, todettiin että työntekijän esihenkilön tulee pyytää tietyt roolit ja vain ne annetaan. Salasanan lähetys on toistaiseksi toteutettu manuaalisesti tekstiviestinvälityspalvelimen avulla, eli IT-osasto lähettää tiettyyn osoitteeseen sähköpostin ja sähköpostin sisältö toimitetaan tekstiviestinä uuden käyttäjän puhelinnumeroon.

Käyttäjätilin käytöstä poistosta tuli myös automaattista, mutta koko käyttäjätilin poistosta ei. Käyttäjätilin poiston voisi automatisoida niin, että tili ja kaikki sen tiedot poistuvat esimerkiksi kahden viikon kuluttua käytöstä poistosta. Näin ei kuitenkaan tässä toteutuksessa tehty, ettei väliaikaisesti käytöstä poistettuja tilejä poistettaisi kokonaan. Tili poistetaan väliaikaisesti käytöstä esimerkiksi perhevapaan tai opintovapaan ajaksi, mutta tiliä ja sen tietoja ei haluta kokonaan poistaa olettaen, että työntekijä palaa töihin vapaan jälkeen. Tilejä ei voida eikä haluta poistaa käytöstä määräajaksi, joten kaikki käytöstä poistot toteutetaan samalla tavalla. Tästä syystä ei voida määrittellä, mikä tili on sellaisen työntekijän, joka on tulossa takaisin töihin ja mikä sellaisen, joka on kokonaan irtisanonut itsensä. Käyttäjätili tietoineen tulee siis poistaa manuaalisesti, ja väliaikaisesti käytöstä poistettu tili tulee manuaalisesti ottaa takaisin käyttöön, kun sen aika on.

Yhteenvedona voidaan siis todeta, että tämä toteutus on hyvä ja toimiva tapa integroida jo olemassa olevien käyttäjätilien tietomuutokset. Erillisiä ilmoituksia ei enää tarvita, vaan tiedot päivittyvät öisin ja mahdollisista virhetilanteista tulee ilmoitus automaattisesti. Onnistuneen automatisoinnin ansiosta voidaan myös päätellä, että integraatiosta on ajallista hyötyä ja näin ollen myös rahallista hyötyä yritykselle. Jatkossa molemmissa järjestelmissä on varmasti samat tiedot, tiedot päivittyvät säännöllisesti eikä muutoksiin tarvita enää kahden työntekijän työaika. Uuden käyttäjän luonti automatisoitui myös suurimmilta osin ja integraation myötä myös tässä tapauksessa tiedot ovat varmasti samat molemmissa järjestelmissä, eikä samaa työtä tehdä kahteen kertaan kahden eri osaston toimesta. Tietoturvan kannalta kriittisin asia automatisoitui myös, käyttäjätilin automaattisen käytöstä poiston ansiosta.

Seuraava kehitysaskel voisi olla salasanan automaattinen lähettäminen käyttäjälle tämän ensimmäisenä työpäivänä. PowerShellillä on mahdollista luoda, muokata ja poistaa Tehtävien ajoituksella ajoitettuja tehtäviä, Tehtävien ajoituksella on mahdollista käynnistää PowerShell ja suorittaa skripti, jolla puolestaan on mahdollista lähettää sähköposti. Uuden käyttäjän luonnin yhteydessä voitaisiin luoda uusi, yhden kerran suoritettava tehtävä Tehtävien ajoitukseen ja ajastaa se suoriutumaan sen päivän aamuna, kun uusi työntekijä tulee taloon. Tämä tehtävä voisi hyödyntää PowerShellia sähköpostin lähetykseen ja sähköposti voisi käyttää tekstiviestinlähetysohjelmintaa lähettääkseen salasanan tekstiviestinä käyttäjän puhelinnumeroon. Tällaisessa toteutuksessa tarvittaisiin myös toiminto PowerShell-skriptiin, joka tarkistaa salasanan lähetykseen tarkoitetut tehtävät ja poistaisi sellaiset, jotka ovat suoriutuneet jo. Näin Tehtävien ajoitukseen ei kerääntyisi tehtäviä sitä mukaa kuin uusia työntekijöitä aloittaa työt yrityksessä.

LÄHTEET

- Batch File to run PowerShell ps1 script. 2017. Saatavissa: <https://www.youtube.com/watch?v=-IKSBNys5Jo>. Viitattu 1.3.2023.
- Carpenter, T. 2012. *Microsoft Windows Operating System Essentials*. Indianapolis: John Wiley & Sons, Inc. Saatavissa: <https://ebookcentral-proquest-com.ezproxy.centria.fi/lib/cop-ebooks/reader.action?docID=821866#>. Viitattu 28.1.2023.
- Cyr, K. S. & Hunter, L. E. 2011. *Automating Active Directory Administration with Windows PowerShell 2.0*. Indianapolis: John Wiley & Sons, Inc. Saatavissa: <https://ebookcentral-proquest-com.ezproxy.centria.fi/lib/cop-ebooks/reader.action?docID=693658#>. Viitattu 28.1.2023.
- Doan, A., Halevy, A. & Ives, Z. G. 2012. *Principles of data integration*. Waltham, Mass.: Morgan Kaufmann. Saatavissa: <https://doi.org/10.1016/C2011-0-06130-6>. Viitattu 30.11.2022.
- Microsoft. 2016. Csvde. Saatavissa: [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc732101\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc732101(v=ws.11)). Viitattu 1.3.2023.
- Mitlohner, J., Neumaier, S., Umbrich, J. & Polleres, A. 2016. *Characteristics of Open Data CSV Files. 2nd International Conference on Open and Big Data (OBD)*, Vienna, Austria, 2016, 72-79. Saatavissa: <https://doi.org/10.1109/OBD.2016.18>. Viitattu 23.2.2023.
- Negus, C. 2015. *Linux Bible. The comprehensive, tutorial resource*. Yhdeksäs painos. Indianapolis: John Wiley & Sons, Inc. Saatavissa: <https://ebookcentral-proquest-com.ezproxy.centria.fi/lib/cop-ebooks/reader.action?docID=1895205#>. Viitattu 28.1.2023.
- Ohlhorst, F. J. 2012. *Big Data Analytics: Turning Big Data into Big Money*. Indianapolis: John Wiley & Sons, Inc. Saatavissa: <https://doi.org/10.1002/9781119205005>. Viitattu 5.2.2023.
- Reeve, A. 2013. *Managing data in motion: Data integration best practice techniques and technologies*. 1st edition. Waltham, Mass.: Morgan Kaufmann. Saatavissa: <https://ebookcentral-proquest-com.ezproxy.centria.fi/lib/cop-ebooks/reader.action?docID=1152636>. Viitattu 31.1.2023.
- Riutta, E. 2022. *Näin tuot CSV-tiedoston Exceliin paremmin!*. Saatavissa: <https://www.youtube.com/watch?v=afmvpdXQfEw>. Viitattu 12.2.2023.
- Stedman, C. & Wigmore, I. *Data integration*. Saatavissa: <https://www.techtarget.com/searchdata-management/definition/data-integration>. Viitattu 4.11.2022.
- Tanner, N. H. 2019. *Cybersecurity Blue Team Toolkit*. Indianapolis: John Wiley & Sons, Inc. Saatavissa: <https://doi.org/10.1002/9781119552963>. Viitattu 29.1.2023.
- Team, F. 2022. *FAIRsharing record for: Comma-separated Values*. Saatavissa: <https://doi.org/10.25504/FAIRsharing.1943d4>. Viitattu 4.2.2023.
- Toivanen, L. & Centria-ammattikorkeakoulu. 2017. *IoT-integraatioalustat*. Centria-ammattikorkeakoulu. Saatavissa: <https://urn.fi/URN:ISBN:978-952-7173-21-3>. Viitattu 5.2.2023.

van den Burg, G. J. J., Nazábal, A. & Sutton, C. 2019. Wrangling messy CSV files by detecting row and type patterns. *Data mining and knowledge discovery*, 33(6), pp. 1799-1820. Saatavissa: <https://doi.org/10.1007/s10618-019-00646-y>. Viitattu 7.3.2023.

