Tampereen ammattikorkeakoulu

Autentikointi

OpenID Connect Autentikointiratkaisuna

Vilppu Kallonen

OPINNÄYTETYÖ
Toukokuu 2021

Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

KALLONEN VILPPU
Autentikointi
OpenID Connect Autentikointiratkaisuna

Opinnäytetyö 21 sivua, joista liitteitä 0 sivua
Toukokuu 2021

Opinnäytetyössä käytiin läpi OpenID Connect autentikointiprotokollan historiaa, perusominaisuuksia ja sitä vertailtiin muihin ratkaisuihin. Autentikointi ja protokolla -sanojen sisältö käytiin läpi ja autentikointiprotokollan yleinen merkitys ja tarkoitus selitettiin. Tarkemmin käytiin läpi salasanapohjainen sekä avainpohjainen autentikointi.

OpenID Connectin ominaisuuksia ja toimintaperiaatetta käytiin läpi tarkemmin, myös selittäen kuinka se perustuu OAuth 2.0 autorisointirakenteeseen. Tämän jälkeen käytiin yleisesti läpi mahdolliset käyttäjän, käyttöönoton sekä ylläpidon näkökulmat, ja mihin pitäisi keskittyä autentikointiratkaisun valinnassa näiden näkökulmien kannalta.

Kolmannessa osiossa vertailtiin eri OpenID Connect palveluntarjoajia ja ratkaisuita. Ratkaisuiden hyviä ja huonoja puolia käytiin läpi ja selitettiin minkälaiseen tarkoitukseen ne kävisivät parhaiten. Lopussa vielä pohdittiin OpenID Connectin ja muiden autentikoitiratkaisuiden ja -protokollien merkitystä alalle.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software Engineering

KALLONEN VILPPU
Authentication
OpenID Connect as an Authentication Solution

Bachelor's thesis 21 pages, appendices 0 pages
May 2021

In the thesis we went through the history and basic features of OpenID Connect authentication protocol and it was compared to other solutions. The meaning of the words authentication and protocol was looked over and authentication protocol was explained. Password and key based authentication were looked over with more detail.

OpenID Connect was the focus of the next chapter. It’s features and attributes were gone through with the explanation how OpenID Connect is based on OAuth 2.0 authorization standard. Then the focus was shifted to the point of view of the end user, deployment and the maintenance. Their different points of view are important for choosing the right authentication solution.

In the third chapter, the different service providers for OpenID Connect solutions were compared. Their good and bad features were gone through and their optimal places of use were thought over. In the end the importance of OpenID Connect and other authentication protocols and solutions to the industry was pondered.

SISÄLLYS

LYHENTEET JA TERMIT

PAP	Password Autentication Protocol
Hash	Datan yksisuuntainen obstruktointitekniikka, jossa se muunnetaan standardimaiseksi merkkijonoksi, josta alkuperäinen data on tunnistettamattomissa.
Salt	Satunnainen merkkijono, jota käytetään salasanojen salauksessa, Hashien vahvistamisessa.
OIDC	OpenID Connect
JWT	JSON Web Token
LDAP	Lightweight Directory Access Protocol
AD	Active Directory
ADFS	Active Directory Federation Service
AWS	Amazon Web Services

1 JOHDANTO

Tässä opinnäytetyössä tutustutaan OpenID Connect autentikointiprotokollaan ja vertaillaan sen palveluntarjoajia ja sen ominaisuuksia muihin autentikointiprotokolliin. Työssä käydään läpi myös autentikointiprotokollien ominaisuuksia, historiaa ja käyttötarkoituksia.

2 AUTENTIKOINTIPROTOKOLLA

Tässä luvussa käydään läpi autentikointiprotokollia ja niiden historiaa.

2.1 Mitä tarkoittavat autentikointi ja protokolla

Autentikoinnilla tarkoitetaan käyttäjän identiteetin varmistamista. Tämä ei aina tarkoita oikean nimen tai muun henkilökohtaisen tiedon antamista, vaan vain järjestelmän tarvitsevien tietojen varmistamista. Usein riittää vain se, että käyttäjällä on tarvittavat tunnukset.

Protokolla on yleisesti säännöstö, jonka mukaan toimitaan määrätyissä tilanteissa. Tämä ei eroa tietotekniikassa käytetystä protokolla termistä. Protokolla määrää, miten ohjelmiston pitäisi toimia tietyissä tilanteissa. Tämä helpottaa ohjelmistojen tekemistä, koska tiedetään, miten kaikkien osapuolten kuuluu tehdä osansa ja miten osat yhdistyvät toisiinsa.

2.2 Autentikointiprotokollien historiaa

Internetin kasvaessa ja internet-palveluiden lisääntyessä ja tullessa tärkeämmiksi, tarvittiin keinoja rajoittaa eri tahojen pääsyä käsiksi järjestelmiin. Tätä varten aloitettiin autentikoinnin käyttö ja erilaisten protokollien muodostaminen siihen tarkoitukseen oli tärkeää. Aluksi autentikointitapoja oli yhtä monta kuin sovelluksia, mutta kohta huomattiin, että yhteistyön tekeminen on tärkeää. Standardisoidun protokollan käyttö mahdollistaa yhteistyön ja helppokäyttöisyyden eri toimijoiden välillä.

2.3 Autentikointiprotokollien perustaa

Autentikointiprotokollien kehittyessä ja monimutkaistuessa, niiden perusteet ovat pysyneet kuitenkin kutakuinkin samoina. Kaikkien osapuolien, eli palveluntarjo-

ajan ja/tai ohjelmiston kehittäjän, pitää tuntea protokolla ja hallita sen käyttö. Autentikointiprotokollan pitää olla niin hyvin kuvailtu ja määritetty, että kaikki osapuolet pystyvät käyttämään sitä tarkasti. Protokollassa pitää olla määritetyt toiminnat kaikkiin tilanteisiin ja käyttötarkoituksiin, joissa sitä on suunniteltu käytettäväksi.

2.4 Autentikointiprotokollatyyppejä

Erilaisia laajasti käytettyjä autentikointiprotokollia on ajan kuluessa tehty paljon niin tietokoneiden väliseen, kuin tietokoneen ja ihmisen väliseen autentikointiin. Tietokoneiden välissä käytetyt autentikointiprotokollat ovat käyttölogiikaltaan hyvin samanlaisia, mutta niissä käytetyt salasanat ja avaimet (käsitellään kappaleessa 2.4.2) ovat kooltaan hyvin suuria. Näissä esimerkeissä keskitymme ihmisen ja tietokoneen välissä käytettyihin autentikointiprotokolliin.

2.4.1 PAP – Salasanapohjainen autentikointi

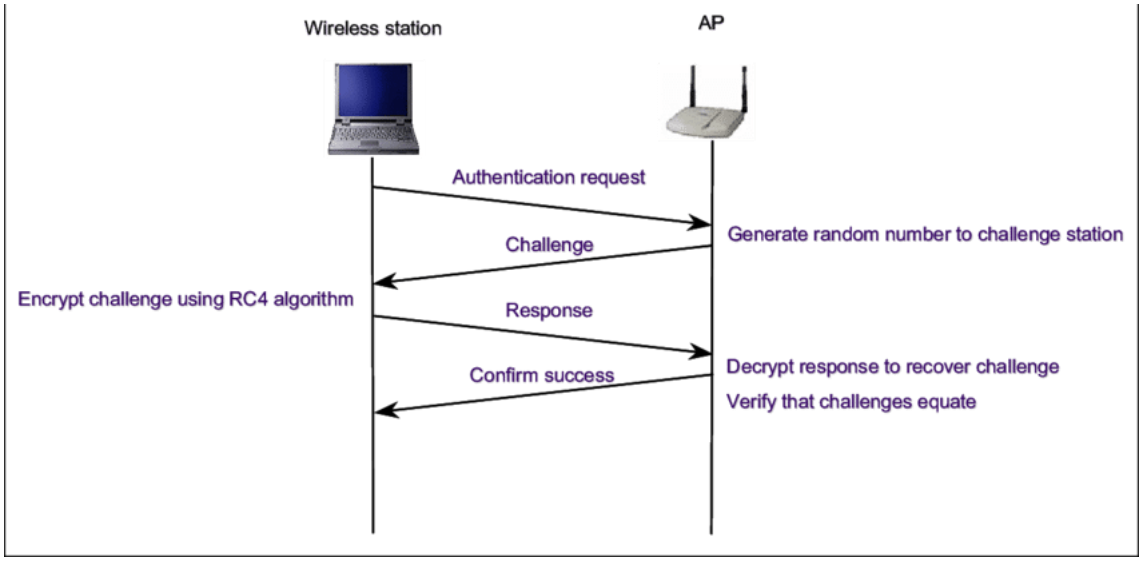
Salasanapohjainen autentikointi on yksi vanhemmista käytetyistä protokollista. Yksinkertaisimmillaan siinä käyttäjä lähettää salasanan ja palvelin tarkastaa sen oikeellisuuden, päästäen käyttäjän käsiksi resursseihin (Duncan, 2001). Pääasiallisessa käytössään OpenID Connect on myös salasanapohjainen autentikointiprotokolla, jonka toiminta ja tietoturva on paranneltu avaimien ja muiden tekniikoiden avulla, joita käydään läpi myöhemmissä luvuissa. Matalimmaksi tietoturvalliseksi tavaksi käsitellä salasanapohjaista autentikointia on salasanojen käsittely käyttäen Hash- sekä Salt -tekniikoita. Näillä tekniikoilla tarkoitetaan salasanan käsittelyä lisäten siihen tiedettyjä merkkejä, joka vaikeuttaa salasanan selville saamista.

2.4.2 Avainpohjainen autentikointi

Avainpohjaisessa autentikoinnissa käytetään avainpareja. Avainparin avaimet ovat toisiinsa korreloituvia, matemaattisesti generoituja merkkijonoja. Käyttäjän

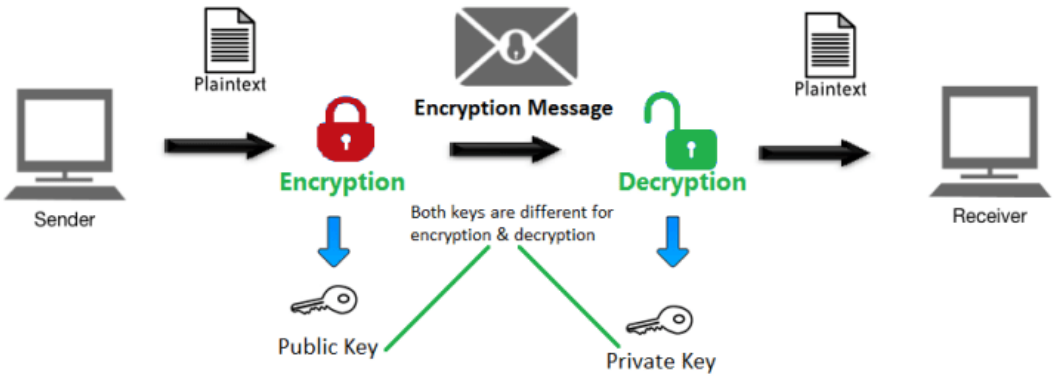
ei tarvitse muistaa avainpareja, vaan ne tallennetaan turvalliselle laitteelle, josta niitä voidaan käyttää. Vaikka käyttäjä ei joudukaan huomioimaan tai muistamaan avainparien käyttöä, lasketaan se silti ihmisen ja koneen väliseksi autentikaatioksi. Näitä autentikointiprotokollia voidaan käyttää koneiden väliseen autentikointiin lähes muuttumattomina. Avainpohjainen autentikointi on usein yhdistetty nykyaikaisten salasanapohjaisten autentikointiprotokollien taustalle, joka mahdollistaa sen, ettei salasanaa tarvitse kysyä jokaisella autentikointikerralla tai tallentaa lukumuistiin. Kaksi yleisintä avainpohjaista autentikointitekniikkaa ovat shared key ja public key. OpenID Connect käyttää public key -tekniikkaa. (Duncan, 2001).

Shared key -tekniikka on nykypäivänä vähemmän käytetty tekniikka siinä olevien tietoturvariskien takia. Tässä tekniikassa kaikilla osapuolilla on käytössään sama avain, jonka avulla dataa salataan ja puretaan. Koska avaimia on vain yksi, pystyy avaimen tietoonsa saanut taho esittämään sekä palvelinta, että käyttäjää. Alla olevassa kuvassa (KUVA 1) on shared keyn toimintaperiaate. Siinä käyttäjä lähettää palvelimelle autentikointipyynnön, johon palvelin vastaa satunnaisella merkkiyhdistelmällä, jonka käyttäjän tietokone salaa käyttäen annettua shared keytä. Tämä salattu merkkiyhdistelmä lähetetään takaisin palvelimelle, joka purkaa salauksen ja joko vahvistaa autentikoinnin onnistumisen tai sitten ei, jos purettu merkkijono on väärä.



KUVA 1. Shared Key toimintaperiaate (Frankel, Eydt, Owens & Scarfone, 2007)

Public key -tekniikassa käytetään kahta avainta. Toinen avaimista, private key, on vain palvelun tarjoajalla ja public key on julkisesti saatavilla kaikille. Tämä tekee public key -tekniikasta varmemman kuin shared key -tekniikasta, koska salausavain ei ole koskaan kenenkään muun kuin palveluntarjoajan hallinnassa. Alla olevassa kuvassa (KUVA 2) näytetään public key toiminta lähettäjän näkökulmasta. Siinä käyttäjä lähettää public keyllä salatun viestin, jonka viestin saaja purkaa private keyllään.



KUVA 2. Public Key lähettäjän näkökulmasta (Comodo SSL Store)

3 OPENID AUTENTIKOINTIPROTOKOLLA

Tässä luvussa esitellään OpenID-autentikointiprotokolla ja sen taustat. Lisäksi esitellään, mihin tarkoitukseen se on kehitetty, sen keskeiset ominaisuudet ja toimintaperiaate.

3.1 Taustaa; Mitä varten luotu

OpenID on avoin autentikointiprotokolla, jota hoitaa OpenID Foundation. Se mahdollistaa sisäänkirjautumisen moniin eri palveluihin ja verkkosivuille käyttäen samoja tunnuksia. Esimerkiksi jos kirjaudut kolmannen osapuolen palveluun käyttäen Facebook tai Google tunnuksiasi, käytät OpenID Connectia. Protokollan uusin versio on nimeltään OpenID Connect (OIDC), joka julkaistiin vuonna 2014.

OpenID:n tekeminen alkoi vuonna 2005 Brad Fitzpatrickin toimesta. Alkuperäisesti OpenID-projektin nimi oli Yadis (Yet another distributed identity system), mutta muuttui pian OpenID:ksi. Vähän yli vuoden sisällä projektiin oli liittynyt monia digitaalisen identiteettiin liittyviä tekijöitä sekä projekteja. Vuonna 2008 OpenID oli jo kasvanut isoksi authentikointiprotokollaksi ja saanut version 2.0. Suuret nimet kuten Microsoft, Google, IBM ja Yahoo! liittyivät OpenID:n hallituksen jäseniksi. Myöhemmin myös Facebook sekä Paypal liittyivät projektiin. Facebook on kuitenkin sittemmin lähtenyt projektista eikä enää mahdollista sisäänkirjautumista Google-tunnuksilla.

Vaikka OpenID:n kehittäminen alkoi jo vuonna 2005, on se edelleen laajasti käytetty protokolla ja OpenID käyttäjätunnuksia on yli miljardi ja niitä hyödyntäviä sivustoja yli miljoona. Myös Apple implementoi OpenID:n käyttäjätunnuksiinsa vuonna 2019 ja se sai nimekseen ’Sing in with Apple’. Muutamat suuret nimet, kuten Stack Overflow ja Blogger, ovat kuitenkin siirtyneet pois OpenID:stä sen vähäisen käytön takia sivustoillaan.

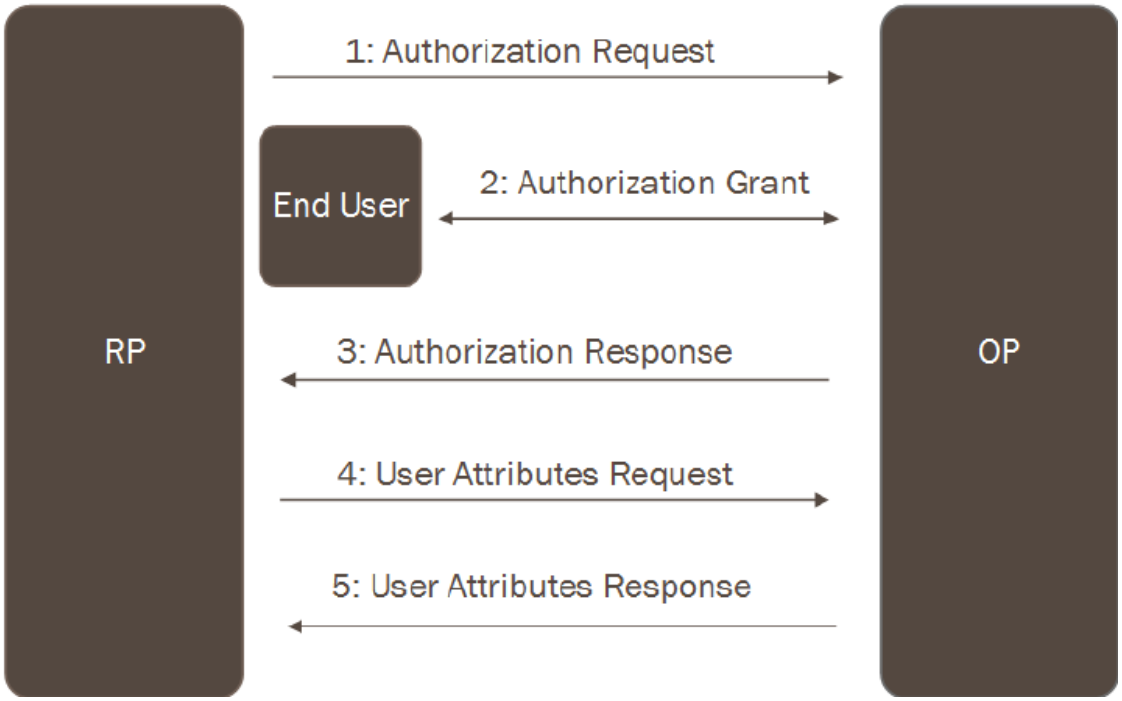
3.2 Pääominaisuudet

OpenID Connect (OIDC) on OAuth 2.0 autorisointirakenteen päälle rakennettu autentikointiprotokolla, joka standardoi sisäänkirjautumisen muiden palveluiden käyttäjätunnuksilla (Neal, 2019). OIDC standardoi myös dynaamisen sisäänkirjautumisen sekä sessionhallinnan. Dynaaminen sisäänkirjautuminen mahdollistaa kirjautumisen ilman käyttäjätunnuksien kysymistä, jos käyttäjä on jo sisäänkirjautuneena haluttuun palveluun. Sessionhallinta muistaa sisäänkirjautumiset, sekä hoitaa uloskirjautumisen.

Autorisointi (Eng. authorization) ja autentikointi (Eng. authentication) ovat omat käsitteensä ja tarkoittavat eri prosesseja. Autorisointi varmistaa käyttäjän tai sivuston pääsyn tiettyihin tietoihin tai palveluihin. Tätä hoitaa OIDC:n tilanteessa OAuth 2.0. Autentikointi puolestaan varmistaa sekä kertoo sivustolle käyttäjän identiteetin. Tällöin käyttäjä voi hyödyntää muissa palveluissa olevaa identiteettiä muilla sivustoilla. Esimerkiksi kirjautumalla Google-tunnuksilla toiselle sivustolle ja jättämällä sinne kommentin Google-identiteetillään. Tätä osaa hoitaa OIDC. (Neal, 2019).

3.3 Toimintaperiaate

Kuvassa (KUVA 3) nähdään OAuth 2.0 toimintaperiaate, johon OIDC perustuu. Kuvassa RP tarkoittaa termiä riippuvainen osapuoli (Eng. Relying Party), joka on nettisivu tai palvelu, johon käyttäjä haluaa kirjautua sisään käyttäen OpenID käyttäjätunnusta. OP tarkoittaa identiteetin tarjoajaa (Eng. OpenID Provider). Identiteetin tarjoaja jaetaan vielä sisäisesti autorisointipalvelimeen sekä resurssipalvelimeen. End User on käyttäjä.



KUVA 3. OIDC ja OAuth 2.0 protokollien toiminnan yleiskuvaus (Li & Mitchell, 2015)

Kuvassa 3 olevat vaiheet (1-5) ja niiden toimintaperiaatteet ovat (Li & Mitchell, 2015):

1. Autorisointipyyntö (authorization request) tapahtuu, kun käyttäjä haluaa kirjautua palveluun käyttäen sivulla mahdollistettuja OpenID tunnuksia. Palvelu lähettää autorisointipyynnön identiteetin tarjoajalle, eli autorisointipalvelimelle. Autorisointipyynnössä palvelu kertoo palvelimelle oman Client ID:nsä, uudelleenohjauslinkin sekä haluamansa tiedot (Eng. Scope). Palvelun käyttäessä OIDC:tä yksi scopen kohdista on *'openid'*, joka kertoo palvelimelle, että palvelu yrittää käyttää OIDC:ia käyttäjän identiteetin tunnistamiseen.

   Palvelun ottaessa yhteyttä autorisointipalvelimelle, palvelin myös luo palvelulle ID:n (client ID) sekä 'salaisuuden' (client secret), joka mahdollistaa tietojen turvallisen vaihdon.

2. Saadessaan autorisointipyynnön, palvelin pyytää käyttäjää kirjautumaan sisään, joka tapahtuu omalla sivullaan, jolloin palvelu ei saa tietoonsa käyttäjän kirjautumistietoja. Tällöin palvelin myös pyytää käyttäjältä lupaa

luovuttaa palvelun pyytämät tiedot. Käyttäjä voi hylätä pyynnön, jolloin palvelin ei lähetä mitään käyttäjäntilin tietoja palveluun ja kirjautuminen keskeytyy.

3. Jos käyttäjä hyväksyy tietojen lähettämisen palveluun, palvelin lähettää autorisointikoodin (Eng. authorization code) palvelulle kohdassa 2 annetun uudelleenohjauslinkin avulla. Tämän jälkeen palvelu ottaa itsenäisesti yhteyttä palvelimelle ja lähettää client ID:nsä, client secretin sekä autorisointikoodin. Palvelimen tarkistettua ne, se lähettää takaisin Access Tokenin sekä käytettäessä OIDC:tia, myös ID Tokenin, joka tulee JSON Web Token (JWT) -muodossa.

4. Kun palvelu on saanut access tokenin, se voi lähettää pyynnön resurssipalvelimelle saadakseen haluamansa tiedot käyttäjästä. Access token on vain kasa merkkejä ja siitä palvelu ei saa tietoonsa mitään, mutta resurssipalvelin tietää onko access token aito.

5. Jos access token on aito ja oikein, lähettää resurssipalvelin käyttäjätunnuksen tiedot, joihin käyttäjä antoi luvan kohdassa 2, palvelulle ja kirjautuminen on valmis.

3.4 Käyttö-, käyttöönotto- ja ylläpito-ominaisuudet

Jokaisella palvelulla käyttäjän, käyttöönottajan, sekä ylläpitäjän näkökulmat vaihtelevat suuresti. Pääasiallisesti käyttäjän kokemus on ensisijaista mutta käyttöönoton sekä ylläpidon helppous sekä hinta ovat myös suuressa osassa käyttöönotosta päätettäessä.

3.4.1 Käyttäjän näkökulma

Käyttäjä voi olla jokainen meistä. Käyttäjän ei tarvitse tietää tai ymmärtää miten palvelu toimii, mutta käyttäjän on hyvä olla tietoinen käyttämänsä palvelun luottamuksellisuudesta ja tietoturvatasosta. OIDC samanaikaisesti yhdistää palveluita sekä helpottaa niihin sisäänkirjautumista. Tämä kuitenkin tuo mukanaan kysymyksiä tietoturvasta. Uskaltaako käyttäjä kirjautua sisään palveluun toisen palvelun tunnuksilla? Osaako käyttäjä tunnistaa turvallisen sivun, jonne kirjoittaa tunnuksensa ja salasanansa. Tämä uhka tietenkin pienenee, jos palveluntarjoaja käyttää omaa OIDC-kirjautumista vain omissa palveluissaan, jotta käyttäjän ei tarvitse muistaa monia sisäänkirjautumistietoja.

3.4.2 Käyttöönoton näkökulma

OpenID Connectin voi julkaista monella eri tavalla. Tällä ei usein ole loppukäyttäjän kokemukselle suurta eroa, mutta käyttöönotto on vaikea vaihe usean eri seikan takia. Järjestelmissä, joita ollaan siirtämässä OpenID Connectin alle on jo todennäköisesti olemassa olevia käyttäjiä, joiden tunnukset täytyy saada toimimaan uudessa ympäristössä. Suurin osa OpenID-protokollista toimii LDAP protokollan kanssa, joka tekee migraatiosta helppoa. Järjestelmän toimintavarmuus on hyvä huomioida jo käyttöönottovaiheessa. Jos käyttöönotettavassa sovelluksessa on jo suuria käyttäjämääriä, on järjestelmä oltava skaalattavissa tai hyvin mitoitettu käyttäjämäärän mukaan. Jos palvelua käytetään laajasti ympäri maan, on palvelun saatavuus otettava huomioon. Käyttäjän ja palvelimen väliset suuret matkat aiheuttavat turhaa hitautta. Itse tehdyn palvelun toteuttaminen on paljon haastavampaa ja vie enemmän työaikaa, mutta jos palvelun sisäisen liikenteen pitää pysyä ainoastaan yrityksen sisällä, tai jos halutaan tietää tarkalleen mitä palvelussa tapahtuu, on itse toteutettu palvelu melkeinpä pakollinen. Jos näitä ei tarvita, on mahdollista myös ottaa käyttöön jo valmis palvelu tai sen luuranko, johon itse rakennetaan palvelu päälle.

3.4.3 Ylläpidon näkökulma

Tärkeimpänä asiana ylläpidon kannalta on palvelun saatavuus ja toimintavarmuus. Palvelusta ei ole mitään hyötyä, jos se on alhaalla tai tavoittamattomissa. Jotkin providereista voivat näyttää käyttäjäkunnalle toisia luotettavammilta (Google, Microsoft). Jotkin providereista käyttävät enemmän tilaa ja resursseja palvelimilta. Jotkut ovat saatavilla open-sourcena, jolloin muokattavuus on suurta, mutta virallista tukea ei välttämättä ole saatavilla.

4 AUTENTIKOINTIRATKAISUIDEN VERTAILU

Tärkein päätös autentikointiratkaisun valinnassa on, että onko se itseylläpidetty vaiko otetaanko sitä hoitamaan jokin palvelun tarjoaja.

4.1 Self-Hosted

Itseylläpidetyssä authentikointiratkaisuissa on joissain tapauksissa suuri muokattavuus ja vapaus, mutta se voi myös olla vain suljettu sovellus. Suurin hyöty itseylläpidettävässä ratkaisussa on se, että sen voi julkaista täysin suljettuun ympäristöön, mutta siinä voidaan silti käyttää OpenID Connect -protokollaa, jolloin sitä käyttävät sovellukset seuraavat silti tunnettuja standardeja ja niitä voidaan halutessa käyttää julkisten palveluntarjoajien kautta. Itseylläpidetyssä ympäristössä voit olla varmempi, ettei palveluntarjoajat ole tietoisia autentikointitapahtumista ja palvelun lokit ovat vain sinun hallussasi.

Itseylläpidetty järjestelmä voi olla valmiina ostettu tuote, kuten Microsoftin tarjoama ADFS. Tämä toimii suoraan olemassa olevan AD-ympäristön kanssa ja on täten erittäin helposti käyttöönotettavissa. Tässä tapauksessa pääsyä lähdekoodiin ei ole, joten muokattavuus on siltä osin rajoitettu. Tämä on myös Microsoftin rakentama tuote, joten siinä on sisäänrakennettuja parametreja. Näiden avulla saat tehtyä muokkauksia standardiin käyttökokemukseen.

Esimerkkejä open-source autentikointiratkaisuista ovat Hydra sekä Keycloak. Ollessaan open-source ratkaisuita, näiden muokattavuus on niin suurta kuin itse haluaa. Valmis pohja on kuitenkin jo annettu, joten käyttöönotto on suhteellisen helppoa. Hydralla on myös virallinen tuki, mutta Keycloakilla tätä ei ole. Täten ongelmien sattuessa niiden korjaaminen on täysin omilla harteilla. Open-source ratkaisuiden päivitettävyys laskee mitä enemmän niitä muokataan. Joidenkin tuotteiden lisenssi ei salli niiden muokkaamista yksityisesti. Joissakin korkean tietoturvaa tarvitsevissa ympäristöissä tämä saattaa olla este.

Jos haluaa täyden hallinnan kaikkeen, voi ratkaisun myös koodata itse. Tämä harvoin on järkevää, sillä sen päivittäminen, muokkaaminen ja kaikki muukin on

täysin omilla harteilla, joista tulee suuria kustannuksia. Tällöin kuitenkin saa täysin sellaisen ratkaisun kuin itse haluaa. Jossain tapauksissa open-sourcen ja itsetehdyn yhdistäminen on mahdollista ja se helpottaa hommaa huomattavasti. Näissä tapauksissa esimerkiksi autentikoinnin rajapinta voidaan ottaa valmiina toteutuksena ja oma toteutus voidaan rakentaa sen taustalle.

4.2 Third Party Provider

Authentikointiproviderit ovat kolmannen osapuolen ohjelmistoja tai palveluita, joiden tarkoitus on helpottaa OIDC:n käyttöönottoa ja ylläpitämistä. Käytettäessä ”valmista” ratkaisua, ei käyttöönottajan tarvitse rakentaa haluamaansa toiminnallisuutta alusta alkaen, vaan perustoiminta on jo valmiina. Muutoksia ja personalisointia tarvitsee kuitenkin aina vähän tehdä, että palvelu saadaan toimimaan halutulla tavalla halutussa ympäristössä. Jos palvelu on alhaalla, itse siihen ei pysty näissä ratkaisuissa vaikuttamaan. Luotto kolmannen osapuolen tietoturvaan pitää olla myös kunnossa. Jos ratkaisua haluaa käyttää pitkäaikaisesti, voi palveluntarjoaja vaihtaa protokollaansa ja muitakin ratkaisuitaan nopealla aikataululla ja näihin muutoksiin ei itsellä ole sananvaltaa.

Useimmiten on suositeltavaa käyttää täysin valmista ratkaisua, joka tarvitsee vain käyttöönoton. Tällaisia palveluita tarjoaa esimerkiksi Google, Microsoft, Facebook ja Auth0. Näiden ratkaisuiden käyttöönotto on erityisen helppoa, jos käyttäjillä ei ole olemassa olevia tunnuksia, joita tarvitsee siirtää. Monilla käyttäjillä on jo valmiina käyttäjätunnukset Googlen, Microsoftin ja Facebookin palveluihin. Tämä jättää myös käyttäjätunnusten hallinnasta johtuvat tietoturvaongelmat palveluntarjoajan huoleksi. Käyttöliittymät ovat myös valmiina näissä ratkaisuissa. Auth0:n tapauksessa käyttöliittymää pystyy muuttamaan oman yrityksen tyylin mukaiseksi. Monet näistä ratkaisuista on ilmaisia, kunnes käyttäjien määrä menee yli valitun määrän, jolloin ratkaisun hinta skaalautuu usein hyvin liikevaihdon kanssa. Laajennettavuus ei myöskään ole ongelma suurien palveluntarjoajien palveluissa.

Kolmannen osapuolen rakennettaviksi ratkaisuiksi voi laskea esimerkiksi Azuren sekä AWS. Näissä tapauksissa käyttöönottajalla on enemmän valinnanvaraa

käyttöliittymän ulkoasuun ja järjestelmillä on hyvä skaalautuvuus. Tällaisen järjestelmän käyttöönotto vaatii kuitenkin enemmän työtä ja usein maksaa enemmän, varsinkin jos käyttäjiä on vain vähän. Käyttäjien siirtäminen on myös taas ongelma näissä ratkaisuissa. Jos kaikki käyttäjät ovat uusia, ongelmaa ei tietenkään ole. Tämä antaa usein myös helposti pystytettävän, valmiin ympäristön, johtuen palasista, joista se pilvipalveluissa kootaan.

5 POHDINTA

Tässä työssä tehtiin selvitys autentikoinnista ja sen toteuttamisen eri vaihtoehdoista. Työssä keskityttiin OpenID Connectiin, johtuen sen tämänhetkisestä käyttölaajuudesta. Toteuttamistapoja autentikoinnin hoitamiseen palvelussa on monia, ja paras valinta ei ole niin yksinkertaista löytää kuin voisi luulla. Valintaan vaikuttaa moni asia, kuten budjetti, ylläpitokyky, sekä haluttu muokattavuus tai tieto ohjelmiston lähdekoodista. Autentikointijärjestelmän hidastuminen ja käyttökatkokset vaikuttavat aina käyttäjien käyttökokemukseen. Tämän takia autentikointi on kriittinen osa ohjelmistoa. Tämän vuoksi pohdinta autentikointipalvelun valinnasta ja käyttöönotosta on tärkeää ja tulee hoitaa huolellisesti.

Autentikointi on aiheena tärkeä kaikille ICT-alan osaajille. Melkein kaikkiin palveluihin kuuluu jonkintasoinen autentikointi. Nykyiset lakisäädöksen datan käsittelemisestä ja säilyttämisestä ovat tarkkoja, joka johtaa siihen, että yhä useampi palvelu joutuu miettimään, kannattaako dataa säilyttää omilla palvelimilla, vaiko hyödyntää ulkoisia palveluntarjoajia. Jotta tämä on mahdollista, yleisten standardien ja protokollien tunteminen on pakollista.

Mielestäni standardoitujen autentikointitapojen sekä protokollien yleistyminen on oikea suunta. Tämä tuo varmuutta sekä käyttäjille että palveluntuottajille. Kuitenkin käytön helppouden kasvaessa salasanavuotojen vakavuus nousee monen palvelun käyttäessä samaa autentikointipalvelua. Suurien palveluntarjoajien tietoturva sekä datan salaaminen on onneksi sillä tasolla, että datan vuotaminen ja sen purkaminen on erittäin epätodennäköistä.

LÄHTEET

AWS. Amazon Cognito. Luettu 19.9.2020. https://aws.amazon.com/cognito/?hp=tile&so-exp=below

Comodo SSL Store. Public Key and Private Key Pair: How it Works. Luettu 15.8.2020. https://comodosslstore.com/blog/public-key-and-private-key-pair-how-it-works.html

Duncan, R. (2001). An overview of different authentication methods and protocols. SANS Institute. Luettu 15.8.2020 https://www.sans.org/reading-room/whitepapers/authentication/overview-authentication-methods-protocols-118

Frankel, S. E., Eydt, B., Owens, L., & Scarfone, K. A. (2007). SP 800-97. establishing wireless robust security networks: A guide to IEEE 802.11 i.

Gupta, G. (2018). OpenID Connect in the Payara Platform 5.183. Luettu 3.12.2019. https://blog.payara.fish/openid-connect-in-the-payara-platform-5.183

Li, W. & Mitchell, C. (2015). Analysing the Security of Google's implementation of OpenID Connect. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 357-376). Springer, Cham.

Microsoft. AD FS Overview. Luettu 5.9.2020 https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/ad-fs-overview

Neal, D. (2019). An Illustrated Guide to OAuth and OpenID Connect. Luettu 2.12.2019. https://developer.okta.com/blog/2019/10/21/illustrated-guide-to-oauth-and-oidc

OpenID. Welcome to OpenID Connect. Luettu 3.12.2019 https://openid.net/connect/

OAuth 2.0. Luettu 19.9.2020. https://oauth.net/2/