Käyttöoikeuksien ja tietojen näkyvyyden hallinta verkkosovelluksissa

Turun yliopisto
Tietotekniikan laitos
LuK-tutkielma
Tietojenkäsittelytiede
Maaliskuu 2026
Petri Aarnio

TURUN YLIOPISTO
Tietotekniikan laitos

Petri Aarnio: Käyttöoikeuksien ja tietojen näkyvyyden hallinta verkkosovelluksissa

LuK-tutkielma, 24 s.
Tietojenkäsittelytiede
Maaliskuu 2026

Verkkosovellus on tietojärjestelmä, jota voidaan käyttää internet-verkkoyhteyden välityksellä käyttäen joko verkkoselainta tai erillistä asiakasohjelmaa. Monen käyttäjän verkkosovelluksissa tietoturva on olennainen osa järjestelmän suunnittelua. Tietoturva on laaja kokonaisuus, mutta tässä tutkielmassa keskitytään käyttäjähallinnan tutkimiseen, eli miten voidaan hallinnoida käyttäjien käyttöoikeuksia ja tietojen näkyvyyttä erilaisilla pääsynhallintamalleilla.
Tutkielmassa selvitetään kirjallisuuskatsauksen keinoin, millaisia yleisesti käytettyjä käyttöoikeuksien ja tietojen näkyvyyden hallintamalleja on olemassa, ja millaisten kriteerien perusteella voidaan sopiva käyttöoikeusmalli valita.
Lähdejulkaisuista löytyvät lukuisat pääsynhallintamallit viittaavat yleisesti neljään kantamalliin. Nämä perinteiset pääsynhallintamallit ovat harkinnnavarainen (DAC), pakollinen (MAC), roolipohjainen (RBAC) ja attribuuttipohjainen (ABAC) pääsynhallintamalli. Näiden mallien toimintaperiaatteiden selvityksestä saadaan myös lista kriteereistä, joita voidaan käyttää verkkosovelluksen pääsynhallintamallia sunnitellessa. Vaikka kaikkiin järjestelmiin sopivaa yleistä mallia ei ole olemassa, niin RBAC on edelleen eniten käytössä. Verkkosovellukselle sopivan pääsynhallintamallin valinnan pitää kuitenkin pohjautua tarkkaan analyysiin verkkosovelluksen toiminnallisista ja tietoturvavaatimuksista.

Asiasanat: verkkosovellus, tietoturva, pääsynhallinta, DAC, MAC, RBAC, ABAC

Sisällys

1 Johdanto **1**

2 Menetelmä **4**

3 Tietoturva **7**

3.1 Tietoturva 7

3.2 Todennus ja valtuutus 9

4 Pääsynhallintamallit ja soveltuvuus eri käyttötapauksiin **11**

4.1 Harkinnanvarainen pääsynhallinta DAC 12

4.2 Pakollinen pääsynhallinta MAC 13

4.3 Roolipohjainen pääsynhallinta RBAC 15

4.4 Attribuuttipohjainen pääsynhallinta ABAC 18

4.5 Pääsynhallintamallien vertailu 20

5 Yhteenveto **22**

Lähdeluettelo **25**

Kuvat

2.1 Lähdeaineiston haun vaiheet 4

3.1 CIA-malli 8
3.2 IAM:n konfigurointivaihe 9
3.3 IAM:n toimintavaihe 10

4.1 MAC-malli (Bell–LaPadula) 14
4.2 RBAC-malli 16
4.3 Perus-RBAC Chen ER -notaatiolla 16
4.4 Roolihierarkia 17
4.5 ABAC-mallin perustoiminnallisuus 19

Taulukot

4.1 Esimerkki pääsynhallintamatriisista 12
4.2 Pääsynhallintamallien vertailu 21

1 Johdanto

Verkkosovellus on tietojärjestelmä, joka tarjoaa tietoa ja palveluja käyttäjille sekä muille tietojärjestelmille internet-verkkoyhteyden välityksellä. Verkkosovellusta voi käyttää verkkoselaimella varustetulla laitteella tai erillisellä asiakasohjelmalla, joka kommunikoi sovelluksen API-rajapinnan (engl. Application Programming Interface) kanssa. Aiemmin ohjelmistoja ajettiin yritysten omilla palvelimilla mutta pilvipalveluiden yleistymisen myötä monet yritykset ovat siirtäneet ohjelmistojaan pilvipalvelutoimittajien alustoille. Näistä suurimmat ovat Amazon, Google ja Microsoft [1].

Pilvipalvelu tarjoaa internetin välityksellä pääsyn jaettuihin tietoteknisiin resursseihin, joita ovat esim. palvelimet, tietokannat, verkkolevyt ja sovellukset. Käyttäjä ei tyypillisesti tiedä näiden resurssien fyysistä sijaintia. Vuokraamalla kulloinkin tarvitsemansa resurssit pilvipalvelualustasta, käyttäjän ei tarvitse huolehtia oman laitteiston hankinnasta eikä sen ylläpidosta. [2]

Aloittava startup-yritys voi nopeasti ja kohtuullisin kustannuksin rakentaa oman ohjelmistotuotteensa SaaS-palveluna (engl. Software as a Service), jota ajetaan pilvipalvelussa. SaaS-palveluiden yleistyminen on johdonmukainen seuraus sen tarjoamista eduista kaikille osapuolille. Palvelun tarjoaja saa ohjelmistonsa helposti tarjolle suurelle käyttäjäjoukolle ilman fyysisen jakeluformaatin käyttöä, ja ohjelmisto on paremmin suojassa piratismilta. Vastaavasti käyttäjä vapautuu ohjelmiston erilliseltä asennukselta omalle koneelle ja sen ylläpidosta. [3]

Verkkosovellusten ohjelmiston toteuttamiseen on tarjolla paljon ilmaisia avoimeen lähdekoodiin perustuvia työkaluja ja sovelluskehyksiä, jotka tarjoavat valmiita suunnittelumalleja sovelluksen toiminnallisuuden rakentamiseen. Verkkosovellukset ovat usein monimutkaisia sekä rakenteeltaan että toiminnallisuudeltaan, joten ohjelmiston kehittäjän suurena haasteena on löytää käyttökohteeseen parhaiten soveltuva tietoturvaratkaisu [4].

Motivaationa tälle tutkielmalle on tarve jatkokehittää olemassa olevan verkkosovelluksen käyttäjähallintaa. Kyseinen verkkosovellus on pilvipalvelussa ajettava SaaS-palvelu, joka on toteutettu mikropalveluarkkitehtuurilla. Sovellus on jaettu pienempiin, vain oman rajatun vastuualueensa toteuttaviin palveluihin, jotka kommunikoivat keskenään rajapintojensa kautta. Asiakkaiden valtuutuksien (engl. authorisation) hallintaa hoitaa erillinen mikropalvelu. Asiakas voi olla käyttäjä tai toinen tietojärjestelmä.

Tämän tutkielman aiheena on kirjallisuuskatsauksen [5] keinoin selvittää, millaisilla yleisesti käytetyillä menetelmillä hallinnoidaan verkkosovelluksen käyttäjien käyttöoikeuksia ja tietojen näkyvyyttä. Työssä pyritään myös vertailemaan ja arvioimaan näiden menetelmien tehokkuutta ja käytännöllisyyttä eli millaisissa käyttötapauksissa tietty menetelmä on parempi kuin muut.

Tietoturva on laaja kokonaisuus, joka koskettaa informaatiojärjestelmän kaikkia kerroksia [6]. Tämä tutkielma keskittyy verkkosovellusarkkitehtuurin palvelinpuoleen, joka käsittää API-rajapinnan, sovelluslogiikan ja tietokannan. API-rajapintaan tulevien pyyntöjen käsittelyssä pitää todentaa lähettäjälle määritetyt valtuutukset käyttöoikeuksien ja tietojen näkyvyyden rajoittamiseksi. Viime vuosina voimakkaasti yleistyneet tekoälyn ja lohkoketjujen (engl. blockchain) käyttö sekä perinteisten pääsynhallintamenetelmien tehostaminen niiden avulla olisivat myös kiinnostavia tutkimuskohteita, mutta ne on kuitenkin rajattu tämän tutkielman ulkopuolelle samoin kuin muut tietoturvan perinteisemmät osa-alueet, kuten esimerkiksi käyttäjien

todentaminen (engl. authentication) ja alati paisuva kyberrikollisuuden torjunta.

Lisäksi tämän tutkielman ulkopuolelle on rajattu tietosuojaan (engl. data protection) liittyvä aineisto. Tietosuojan painopiste on henkilötietojen käsittelyyn liittyvät lainsäädännölliset ja eettiset näkökulmat, ja Euroopassa sitä säätelee Euroopan yleinen tietosuoja-asetus GDPR (General Data Protection Regulation). [7]

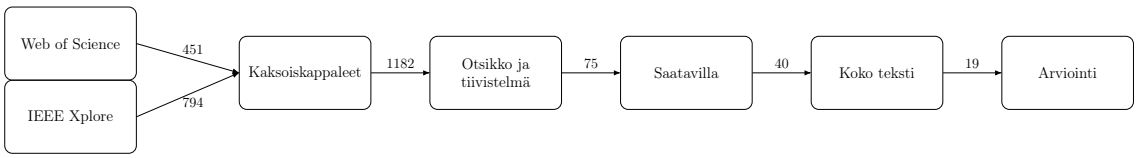
Tutkielman tavoitteena on löytää vastaukset seuraaviin tutkimuskysymyksiin:

- **TK1:** Millaisilla käyttöoikeuksien ja tietojen näkyvyyden hallintamalleilla voidaan ohjelmistopalvelun tietoturva toteuttaa?
- **TK2:** Millaisten kriteerien perusteella voidaan valita sopiva käyttöoikeusmalli tiettyyn käyttötapaukseen?

Tutkielman luvussa 2 dokumentoidaan käytetty menetelmä ja kirjallisuuskatsauksen vaiheet. Luvussa 3 tarkastellaan tietoturvan perusperiaatteita CIA-mallin avulla. Lisäksi selvitetään käyttäjien todennuksen ja valtuutuksen erot. Luvussa 4 esitetään käyttöoikeuksien hallintamallit ja pohditaan niiden soveltuvuutta erilaisiin käyttötapauksiin. Lopuksi luvussa 5 tiivistetään kirjallisuuskatsauksen tulokset ja pohditaan jatkotutkimuksen mahdollisuuksia.

2 Menetelmä

Työ toteutettiin kirjallisuuskatsauksena [5]. Lähdeaineiston haut suoritettiin 4.10. - 5.10.2025 välisenä aikana, ja tietoa haettiin pääsääntöisesti Web of Science ja IEEE Xplore -tietokannoista. Niistä löydetyn aineiston lukumäärä on esitetty kuvassa 2.1.



Kuva 2.1: Lähdeaineiston haun vaiheet

Hakutulosten karsintaan määritettiin kriteerit. Julkaisuista valittiin ne, jotka

- ovat englanninkielisiä
- ovat kirjoja, tieteellisiä artikkeleita, konferenssijulkaisuja tai standardeja
- ovat saatavilla ilmaiseksi verkossa tai Turun yliopiston kirjaston kautta
- käsittelevät tietoturvaan liittyviä hallintamalleja periaatteellisella tasolla sitoutumatta tiettyihin teknisiin toteutuksiin tai ympäristöihin.

Hakutulosten käsittelyn vaiheet ja niissä jäljelle jääneiden julkaisujen lukumäärät on esitetty kuvassa 2.1.

Vaihe 1: Lähdeaineiston haku

Riittävän määrän soveltuvien artikkeleiden löytämiseksi kokeiltiin useita hakusanoja, jotka johdettiin tutkimuskysymyksistä sekä muista aihealueeseen liittyvistä termeistä. Sopivaksi katsottu hakutulosten määrä saavutettiin hakulausekkeella (("authori?ation"OR "access control") AND "web application*"). Haku kohdistettiin otsikkoon ja tiivistelmään, ja hakutulosten määrää supistettiin edelleen valitsemalla vanhimmaksi julkaisuvuodeksi 1995. Tätä vanhemmissa julkaisuissa alkoi enenevässä määrin esiintymään nykyään uusissa järjestelmissä harvinaiset tekniikat kuten XML-merkintäkieltä viestien muotoiluun käyttävät SAML ja SOAP.

Vaihe 2: Kaksoiskappaleiden poisto

Kummankin tietokannan hakutulokset yhdistettiin ja kaksoiskappaleet poistettiin. Kaksoiskappaleista valittiin se, jolla oli täydellisemmät tiedot.

Vaihe 3. Poisto otsikon ja tiivistelmän perusteella

Poistettiin julkaisut, jotka eivät täyttäneet sisällyttämiskriteerejä. Tässä vaiheessa poistettiin tutkielman aihepiirin ulkopuolelle rajattuina viittaukset

- tekoälyyn ja lohkoketjuihin
- nykyään harvemmin käytettyihin vanhempiin tekniikoihin (esim. XML, AOP, SOAP, SAML)
- hallintamallien toteutuksiin erilaisilla ohjelmointikielillä ja ohjelmistokehyksillä
- tietosuojaan liittyviin termeihin (esim. GDPR)
- tietoturvauhkiin (esim. hyökkäyksen tunnistus, SQL injektio).

Vaihe 4. Poisto saatavuuden perusteella

Tässä vaiheessa osoittautui, että osasta viittauksista joko puuttui kokonaan linkki dokumenttiin tai siihen ei ollut pääsyä Turun yliopiston kirjaston kautta.

Vaihe 5. Poisto koko tekstin sisällön perusteella

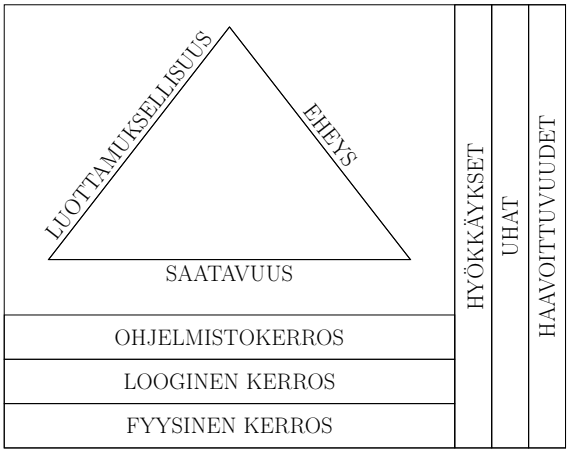
Lopuksi perehdyttiin tarkemmin julkaisujen teksteihin, joista valittiin ne, jotka käsittelivät tietoturvan toteuttamista erilaisilla pääsynhallintamenetelmillä ja tietojen näkyvyyden rajoittamista.

3 Tietoturva

3.1 Tietoturva

Tietoturva määritellään yleensä tiedon luottamuksellisuuden (engl. confidentiality), eheyden (engl. integrity) ja saatavuuden (engl. availability) suojaamisena sen tallennuksen, käsittelyn ja siirron aikana. Alan kirjallisuudessa näitä kolmea osa-aluetta kutsutaan CIA-kolmioksi (kuva 3.1). [6]

Kuvassa 3.1 CIA-kolmion alle on myös havainnollistettu tietoturvan toiminnalliset tasot, joilla minimoidaan haavoittuvuudet, uhat ja hyökkäykset fyysisellä, loogisella ja ohjelmiston tasolla. Näillä toiminnallisuuksilla turvataan tiedon luottamuksellisuuden, eheyden ja saatavuuden säilyminen ennallaan odottamattomista tapahtumista riippumatta. Tietoturvan tasot on summattu kolmeen kerrokseen: fyysinen, looginen ja ohjelmisto. Fyysinen kerros sisältää esim. laitteiston ja verkkorakenteen, looginen kerros sisältää liiketoimintaprosessit, lakisääteiset rajoitukset ja muut liiketoiminnan käytännöt. Ohjelmistokerros kuvaa ohjelmistoprosessien toiminnallisuuden. [6]



Kuva 3.1: CIA-malli

Luottamuksellisuus

Luottamuksellisuus käsittää tietojen näkyvyyden. Tietoihin pääsy rajoitetaan vain valtuutetuille käyttäjille. Tehokkaassa luottamuksellisuuden toteutuksessa valtuutuksien tasot määritellään tarkasti ja valtuutukset toteutetaan käyttämällä soveltuvia menetelmiä. [8]

Eheys

Eheys tarkoittaa kykyä varmistaa tietojen paikkansapitävyys sekä yhdenmukaisuus koko niiden elinkaaren ajan. Vain valtuutetut käyttäjät saavat muokata tietoa. Tietojen luvaton tuhoaminen tai muokkaaminen pitää pystyä havaitsemaan ja perumaan niin, että tieto palautetaan viimeisimpään varmasti eheään muotoon. Eheys voi sisältää myös kyvyn selvittää järjestelmän tapahtumien alkuperä ja liittää ne tiettyyn tahoon. [8]

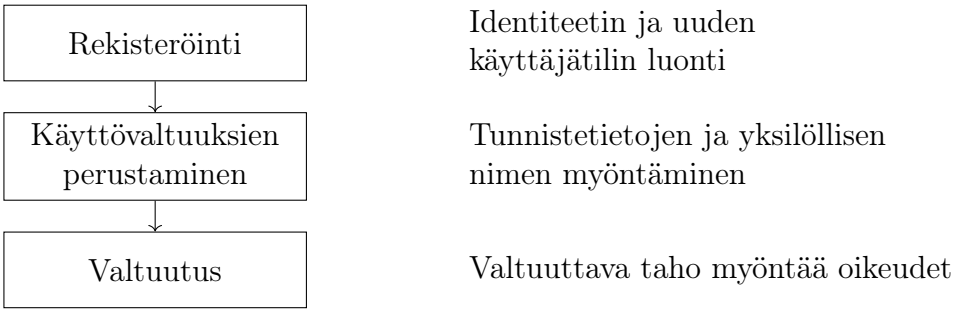
Saatavuus

Saatavuudella tarkoitetaan sitä, että tieto on käytettävissä silloin, kun sitä tarvitaan. Saatavuus edellyttää, että järjestelmät, verkot, tietokannat ja muut tallennusmeka-

nismit ovat valtuutettujen käyttäjien käytettävissä. Tieto ei ole saatavilla, jos se on kadonnut tai tuhoutunut, tai jos valtuutetulta käyttäjältä estyy tai viivästyy pääsy siihen. Saatavuuden varmistamiseksi tiedon käyttöön tarvittavien järjestelmien on toimittava luotettavasti ja tietoturvakäytäntöjen mukaisesti siten, että valtuutetuilla käyttäjillä on tarpeellinen pääsy tietoihin. [8]

3.2 Todennus ja valtuutus

Todennus ja valtuutus -termejä käytetään helposti ristiin, varsinkin silloin kun niihin viitataan englannin kielestä lainatuilla termeillä autentikointi ja auktorisointi, mutta ne toteuttavat erilaisen toiminnallisuuden. Niiden yhteisenä nimikkeenä käytetään myös lyhennettä IAM (engl. Identity and Access Management) [9]. Kuvassa 3.2 esitetään IAM:n konfigurointivaiheen toimenpiteet. Kuvassa 3.3 esitetään IAM:n operatiivisen vaiheen toimenpiteet [10].



Kuva 3.2: IAM:n konfigurointivaihe

Todennus on vaihe, jossa tunnistetaan käyttäjien identiteetti. Nykyään yleistyneenä käyttäjän todennusmenetelmänä käytetään kaksivaiheista tunnistautumista (2FA, engl. Two-Factor Authentication). Ensimmäisessä vaiheessa käyttäjä syöttää tunnistetietonsa, jotka ovat yleensä käyttäjänimi ja salasana. Toisessa vaiheessa käyttäjä varmentaa identiteettinsä esim. puhelimessa olevalla sovelluksella tai sormenjälkilukijalla. [11]

Kuva 3.3: IAM:n toimintavaihe

Pääsynhallinta (engl. access control) on todentamista seuraava vaihe. Onnistuneesti todennettu käyttäjä saa pääsyn vain niihin palveluihin ja resursseihin, joihin käyttäjällä on valtuudet. Valtuudet kattavat tyypillisesti vain osan kaikista järjestelmän palveluista ja resursseista. Järjestelmän pääsynhallinta tarkistaa käyttöpyynnöt valtuutuskäytännön mukaisesti ja panee päätöksen täytäntöön joko hyväksymällä (engl. access grant) tai hylkäämällä (engl. access reject) pyynnön. [10]

Todennuksen ja valtuutuksen näkökulmasta käyttäjät voivat olla joko henkilöitä tai muita järjestelmiä. Tätä tutkielmaa motivoineessa verkkosovelluksessa käyttäjinä ovat tietyn organisaation työntekijät, organisaation ulkopuoliset asiakkaat sekä verkkosovellukseen integroidut ulkopuoliset muut järjestelmät. Kaikki nämä käyttäjät pitää luotettavasti todentaa ja tarkasti määritellä, mihin tietoihin annetaan näkyvyys ja mitä toimintoja sallitaan käyttää.

4 Pääsynhallintamallit ja soveltuvuus eri käyttötapauksiin

Pääsynhallintajärjestelmän toteuttamiseen on käytettävissä useita erilaisia malleja ja menetelmiä. Tietoturvakäytännöt (engl. security policy) ovat joukko periaatteita ja sääntöjä, jotka määrittävät sekä käyttäjien valtuutukset eli pääsyoikeudet että valtuutuksien myöntämis- ja epäämisperusteet. Tietoturvamalli (engl. security model) määrittelee pääsynhallinnan tietoturvakäytäntöjen toteutuksen. Todellisissa käyttötilanteissa tietoturvaan liittyviä käytäntöjä ohjaavat monet monimutkaiset säännöt, jotka voivat olla peräisin esimerkiksi organisaation omista määräyksistä ja käytännöistä tai lainsäädännöstä. Tärkeintä on kuitenkin varmistaa, että tiedon luottamuksellisuus, eheys ja saatavuus säilyvät. [12]

Perinteiset pääsynhallintamallit ovat harkinnanvarainen pääsynhallinta (DAC, engl. Discretionary Access Control), pakollinen pääsynhallinta (MAC, engl. Mandatory Access Control), roolipohjainen pääsynhallinta (RBAC, engl. Role-Based Access Control) ja attribuuttipohjainen pääsynhallinta (ABAC, engl. Attribute-Based Access Control) [13]. Kullakin mallilla on omat vahvuutensa ja heikkoutensa, ja niitä tarkastellaan lähemmin seuraavissa aliluvuissa.

4.1 Harkinnanvarainen pääsynhallinta DAC

Harkinnanvaraisessa pääsynhallinnassa järjestelmäresurssin omistajalla eli resurssin luojalla on täysi valtuus määrittää muiden käyttäjien käyttöoikeudet (engl. permissions) kyseiseen resurssiin [12]. DAC toteutetaan tyypillisesti käyttöoikeusluettelojen (ACL, engl. Access Control List) avulla, mitä pidetään käypänä pääsynhallinnan ratkaisuna silloin, kun käyttäjien ja resurssien määrä on pieni. DAC on yleisin pääsynhallinnan ratkaisu Windowsissa ja UNIX-tyyppisissä käyttöjärjestelmissä. ACL voidaan yleistää taulukossa 4.1 esitetyksi pääsynhallintamatriisiksi, jossa sarakkeet ovat resurssien käyttöoikeuslistoja ja rivit käyttäjien käyttöoikeusprofiileja. [14]

Käyttäjä	**Tiedosto A**	**Tiedosto B**	**Tiedosto C**	**Ohjelma**
Eeva	lue/kirjoita	lue	lue	suorita
Tuomas	lue/kirjoita	lue/kirjoita	lue	suorita
Lassi	lue/kirjoita			suorita
Leevi	lue		lue	
Maaria			lue	suorita

Taulukko 4.1: Esimerkki pääsynhallintamatriisista

Vahvuudet

DAC-mallin rakenne on suoraviivainen, mikä tekee mallista helpon toteuttaa, ymmärtää ja käyttää. Käyttäjät voivat itse määrittää oman tietoturvakäytäntönsä ja laatia resursseilleen käyttöoikeuslistat. Tämän ansiosta järjestelmän ylläpitäjien ei tarvitse käsitellä jokaista käyttöoikeusmuutosta erikseen, joten heidän työmääränsä käyttöoikeuksien hallinnassa vähenee. [13]

Heikkoudet

DAC-malli ei mahdollista järjestelmän ylläpitäjille keskitettyä käyttöoikeuksien hallintaa [14]. Koko järjestelmän yhdenmukaisen tietoturvakäytännön ylläpitäminen

on hankalaa, ja käyttäjät voivat aiheuttaa tietoturvauhkia myöntämällä huolimattomasti tai virheellisesti tarpeettoman laajat käyttöoikeudet [12][14]. DAC-malli on myös altis haittaohjelmien hyökkäyksille: haittaohjelma voi käyttää hyväkseen toiselle käyttäjälle myönnettyjä käyttöoikeuksia ja saada niiden avulla luvattoman pääsyn resursseihin [12]. Merkittävä haittapuoli on se, ettei tiedon kulkua voida hallita. Tietoa voidaan kopioida resurssista toiseen, ja kopioille voidaan myöntää käyttöoikeus käyttäjälle, jolla ei ollut käyttöoikeutta alkuperäiseen resurssiin [14]. DAC-mallin heikkouksista johtuen se soveltuu huonosti kaupallisille ja hallinnollisille organisaatioille [13] sekä pilvipalvelupohjaisiin ympäristöihin [14].

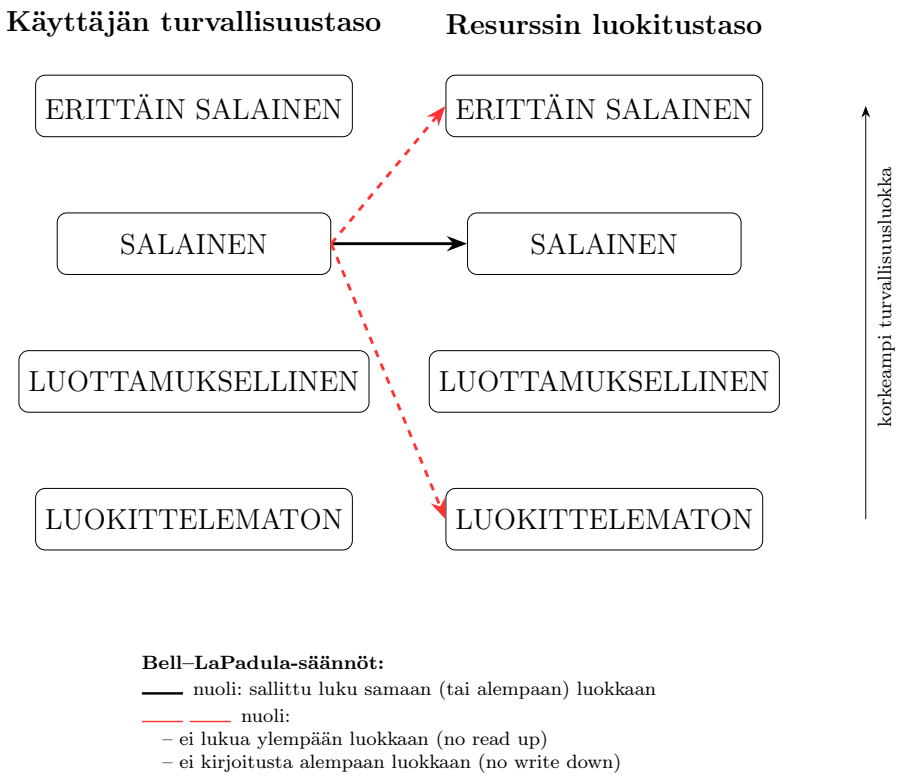
4.2 Pakollinen pääsynhallinta MAC

MAC-malli kehitettiin, koska DAC-malli ei tarjonnut riittävää turvallisuutta eikä pystynyt hallitsemaan kaikkea tietoa; MAC-mallia käytetäänkin tiukkojen turvallisuusvaatimusten ympäristöissä kuten hallinnon ja armeijan järjestelmissä [15]. MAC-mallissa käyttäjät eivät itse voi määrittää tai muuttaa resurssien käyttöoikeuksia, eikä resurssin omistajalla ole valtuutta jakaa käyttöoikeuksia muille käyttäjille. Ainoastaan järjestelmän ylläpitäjä hallitsee käyttöoikeuksia täysin keskitetysti organisaation tietoturvakäytäntäntöjä noudattaen. [13]

MAC-mallin toiminta perustuu tiedon ja käyttäjien luokitteluun eri turvallisuustasoille (engl. security level) niiden arkaluonteisuuden ja luotettavuuden mukaan. Perintönä luokituksen militaristisesta taustasta, tyypillisesti turvallisuustasot ovat seuraavat: erittäin salainen, salainen, luottamuksellinen ja luokittelematon. Käyttäjille määritetään turvallisuustaso (engl. clearance level), joka ilmaisee luottamuksen ja oikeuksien määrän. Resurssien luokitustaso (engl. classification level) ilmaisee arkaluonteisuuden tason. [12]

MAC-mallista on useita variaatioita, mutta mallin yleisen toimintaperiaatteen esittämiseen voidaan käyttää kuvassa 4.1 esitettävää Bell-LaPadula-mallia, joka pai-

nottaa luottamuksellisuutta. Sekä käyttäjille että resursseille annetaan turvallisuustunniste (engl. security label), joiden turvallisuustaso määrittelee pääsyoikeuden. Käyttäjä saa lukuoikeuden vain niihin resursseihin, joiden turvallisuusluokitus on sama tai alempi kuin käyttäjän turvallisuustaso. Kirjoitusoikeuteen käyttäjä tarvitsee saman tai ylemmän turvallisuusluokituksen verrattuna resurssin luokitustasoon. Kirjoitusoikeuden rajaus estää ylemmän tason tiedon vuotamisen alemmalle tasolle. [12]



Kuva 4.1: MAC-malli (Bell–LaPadula)

Vahvuudet

MAC-malli mahdollistaa erittäin korkean tietoturvatason järjestelmien toteuttamisen: resurssien pääsynhallintaa hallinnoidaan keskitetysti yhdenmukaisia tietoturvakäytäntöjä noudattaen, eikä käyttäjillä ole mahdollisuutta kiertää niitä. Tämä estää

tehokkaasti luvattoman tietoon pääsyn ja tiedon leviämisen. [15]

Heikkoudet

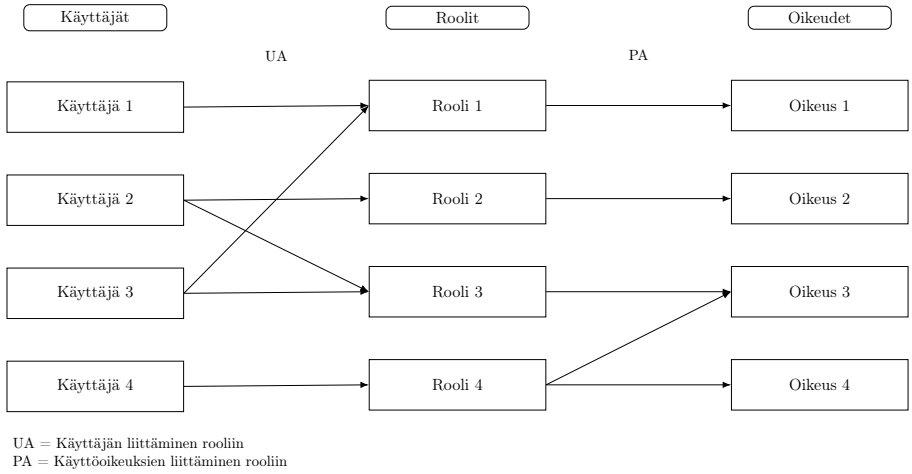
MAC-mallin keskitetty käyttöoikeuksien hallinta kuormittaa järjestelmän ylläpitoa merkittävästi. MAC-mallin tehokas käyttöönotto vaatii tarkkaa etukäteissuunnittelua. Käyttöönoton jälkeen käyttäjien ja resurssien muutokset aiheuttavat jatkuvaa ylläpitotarvetta, ja järjestelmän koon kasvun myötä MAC-malli monimutkaistuu. Edellä mainitut seikat aiheuttavat sen, että MAC-malli skaalautuu huonosti. MAC-malli ei myöskään pysty toteuttamaan hienojakoista pääsynhallintaa eikä tehtävien erottelua. [12][13][14]

4.3 Roolipohjainen pääsynhallinta RBAC

RBAC-malli kehittyi, kun käyttäjien tyytymättömyys 1970-luvulla käyttöön otettuja DAC- ja MAC-malleja kohtaan kasvoi. RBAC-mallia alettiin tutkia tieteellisesti, ja se standardoitiin 1990-luvun alussa. Nykyään RBAC on käytännössä vallitseva pääsynhallintamalli [16]. Permit.io raportoi vuonna 2025, että 86,6 % kyselyyn osallistuneista ohjelmistoinsinööreistä ilmoitti käyttävänsä RBAC-mallia järjestelmässään [17].

RBAC-mallissa käyttäjät liitetään eri rooleihin, jotka määrittävät käyttäjälle myönnetyt käyttöoikeudet. Rooleihin liitetään oikeudet rooli-oikeus-liitoksen (PA, engl. Permission Assignment) avulla, ja käyttäjien käyttöoikeudet määräytyvät sen mukaan, mihin rooleihin heidät on liitetty käyttäjä-rooli-liitoksen (UA, engl. User Assignment) kautta. Käyttäjä voidaan liittää useaan rooliin, ja samalla roolilla voi olla useita käyttäjiä. RBAC-mallin rakenne on esitetty kuvassa 4.2. [15]

RBAC-malli luokitellaan kolmeen päätasoon, jotka ovat perus-RBAC (engl. core RBAC), hierarkkinen RBAC (engl. hierarchical RBAC) ja rajoitteellinen RBAC (engl. constrained RBAC) [13]. Lisää tasoja voidaan määritellä yhdistämällä päätа-

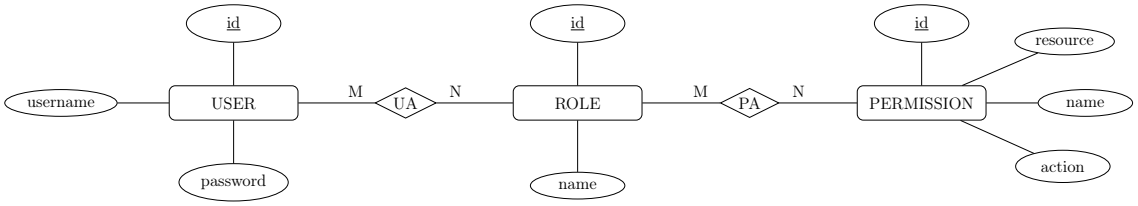


Kuva 4.2: RBAC-malli

soja tai eriyttämällä niistä jokin osa-alue. Esimerkiksi ajallinen RBAC (engl. temporal RBAC) saadaan siirtämällä rajoitteellisen RBAC-tason aikaperusteinen roolien aktivointi ja deaktivointi omaksi erilliseksi tasokseen [14].

Perus-RBAC

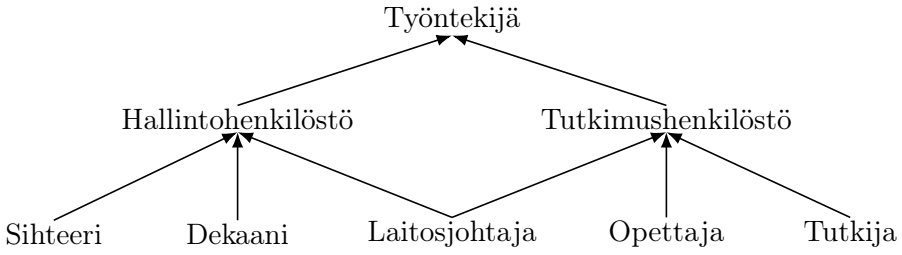
Perus-RBAC on RBAC-mallin olennainen ja perustavanlaatuinen osa, jossa keskeistä on käyttäjien ja käyttöoikeuksien monesta moneen -liittäminen rooleihin. Se on ensimmäinen vaihe RBAC-mallin käyttöönotossa. Perus-RBAC voidaan toteuttaa kuvan 4.3 mukaisella relaatiotietokannan rakenteella. [13]



Kuva 4.3: Perus-RBAC Chen ER -notaatiolla

Hierarkkinen RBAC

Hierarkkinen RBAC on seuraava taso, joka rakennetaan perus-RBAC-tason päälle. Roolit muodostavat hierarkian, joka perustuu yleistämisen ja erikoistamisen periaatteisiin. Kuvassa 4.4 on esimerkki roolihierarkiasta. Alemman tason roolin käyttöoikeudet muodostuvat sen omista käyttöoikeuksista sekä kaikista ylemmiltä tasoilta perityistä käyttöoikeuksista. [12]



Kuva 4.4: Roolihierarkia

Rajoitteellinen RBAC

Tehtävien erottamisessa (SoD, engl. Separation of Duty) tietyn kriittisen tehtävän suorittamiseen vaaditaan enemmän kuin yksi käyttäjä. Tehtävien erottaminen jaetaan rakenteelliseen kontrolliin (SSD, engl. Static Separation of Duty) ja operationaaliseen kontrolliin (DSD. engl. Dynamic Separation of Duty). Rakenteellisessa kontrollissa ristiriitaisia rooleja (esim. laskun tarkastaja ja laskun hyväksyjä) ei saa antaa samalle käyttäjälle ollenkaan, mutta operationaalisessa kontrollissa vastaava rajoite on tehtävä- tai istuntokohtainen. Tehtävien erottamisen lisäksi voidaan käyttää myös muita rajoitteita. Nämä rajoitteet voivat olla joko aikaan tai sijaintiin perustuvia, jolloin rooli ja siihen liitetyt käyttöoikeudet ovat voimassa vain, kun rajoitteeksi asetettu aika- tai sijaintiehto täyttyy. [13]

Vahvuudet

Roolipohjainen pääsynhallinta yksinkertaistaa valtuutuksien hallintaa määrittämällä käyttöoikeudet eri rooleille yksittäisten käyttäjien sijaan. Järjestelmän ylläpitäjä päättää työtehtävien suorittamiseen vaadittavat roolit, jolloin työntekijälle on helppo antaa roolit työtehtävien perusteella. Roolit myöntävät käyttäjälle tietyn tehtävän suorittamiseen tarvittavat pienimmät oikeudet vähimmän oikeuden periaatteen mukaisesti, mikä pienentää tahattomien virheiden aiheuttamia vahinkoja. Muita etuja ovat tehtävien erottaminen ja rajoitteiden täytäntöönpano, joita on jo käsitelty edellä. RBAC-malli soveltuu luontevasti organisaatioihin, joissa on selkeä roolijako. [12]

Heikkoudet

Roolien määrittäminen eri käyttötilanteissa ja ympäristöissä on haastavaa. Sekä käyttäjien roolien että roolien oikeuksien väliset suhteet määritellään erikseen, minkä seurauksena roolit on määriteltävä ennakolta eikä käyttöoikeuksien muuttaminen ole mahdollista ilman roolimäärittelyjen muuttamista. Seurauksena rooleja voi olla enemmän kuin käyttäjiä eli ”rooliräjähdys” (engl. role explosion). Lisäksi nykyjärjestelmät usein tarvitsevat RBAC-mallia hienojakoisemman pääsynhallintamallin. Vaikka rajoitteellinen RBAC-malli tuo mukanaan rajoitetun dynaamisuuden, hienojakoisuus perustuu ennalta määriteltyihin rooleihin eikä yleisiin attribuutteihin, minkä vuoksi RBAC-malli soveltuu huonommin dynaamisiin ja hajautettuihin ympäristöihin. [12][14]

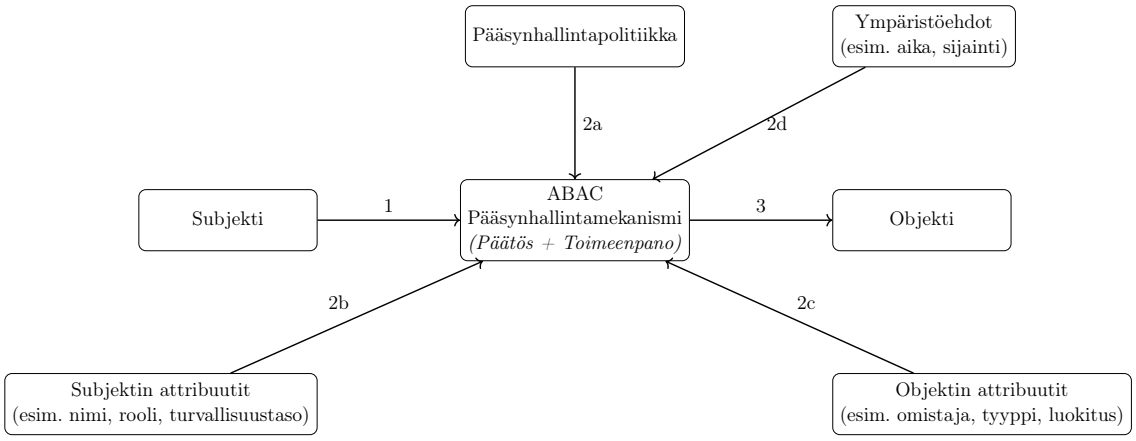
4.4 Attribuuttipohjainen pääsynhallinta ABAC

ABAC-malli on hienojakoinen ja dynaaminen käyttöoikeuksien hallintamalli, jossa pääsynhallintapolitiikka (engl. access control policy) eli arviointisäännöt toteute-

taan attribuuttijoukkoon kohdistuvalla Boolen logiikalla. Attribuutit ovat eri kohteiden ominaisuuksia. Näitä kohteita ovat subjektit (käyttäjät), objektit (resurssit), toiminto ja ympäristö. Käyttöoikeus myönnetään, kun attribuutit ja niiden arvot vastaavat toisiaan. [13]

Korkean tason ABAC-mallin toiminta on esitetty kuvassa 4.5, jossa pääsynhallinnan määrittäminen jakaantuu seuraaviin vaiheisiin [18]:

1. Subjekti pyytää käyttöoikeuden objektiin.
2. Pääsynhallintamekanismi arvioi
   a. pääsynhallintapolitiikan säännöt,
   b. subjektin attribuutit,
   c. objektin attribuutit sekä
   d. ympäristöehdot.
3. Subjektin käyttöoikeus sallitaan tai evätään.



Kuva 4.5: ABAC-mallin perustoiminnallisuus

Vahvuudet

ABAC-mallin attribuuttien yhdistelmillä voi toteuttaa hienojakoisen pääsynhallinnan. Lisäksi sama pääsynhallintapolitiikka voi kattaa laajan joukon käyttäjiä ja resursseja tarvitsematta määritellä erikseen jokaista käyttäjä-resurssisuhdetta kuten DAC-mallissa tehdään. ABAC-malli helpottaa pääsynhallintapolitiikkojen hallintaa suuressa yrityksessä tai useiden organisaatioiden välillä; eri osastot tai organisaatiot voivat yhteistyössä määrittää tarvitsemansa politiikat. ABAC-mallissa korostuu sovelluksen pääsynhallintapolitiikan ja liiketoimintalogiikan eriyttäminen, joten pääsynhallinnan sääntöjen muutokset eivät välttämättä aiheuta muutoksia sovelluskoodiin. ABAC-malli soveltuu hyvin hajautettuihin ja monimutkaisiin järjestelmiin. [19]

Heikkoudet

ABAC-malli voi olla monimutkainen, ja useiden pääsynhallintapolitiikkojen määrittely vie paljon aikaa, kun käyttäjät tarvitsevat erilaisia resurssikohtaisia attribuutteja. Monimutkaisen ABAC-mallin tietoturvan auditointi on hankalaa, koska ABAC-mallissa järjestelmän ylläpitäjällä ei ole suoraa näkyvyyttä käyttäjän käyttöoikeuksiin, vaan ne pitää selvittää erikseen jokaisen resurssin pääsynhallintapolitiikan perusteella. Pienissä järjestelmissä ABAC-mallin käyttöönotosta voi tulla turhan raskas. [14]

4.5 Pääsynhallintamallien vertailu

Edeltävissä aliluvuissa esiteltiin perinteiset pääsynhallintamallit DAC, MAC, RBAC ja ABAC. Seuraavaksi taulukossa 4.2 malleja vertaillaan systemaattisesti tutkimuskysymyksien ja tämän tutkielman motivaattorin eli verkkosovellusten näkökulmasta. Vertailukriteereinä käytetään lähdeaineistossa esille tulleita kriteereitä.

Hallinnan periaate tarkoittaa mallin keskeistä ohjausmekanismia.

Päätöksenteon perusta määrittää, mihin tietoihin käyttöoikeuspäätös perustuu.

Hienojakoisuus tarkoittaa käyttöoikeuksien tarkkuutta eri tasoilla.

Kontekstisidonnaisuus viittaa ympäristötekijöiden huomiointiin.

Dynaamisuus ilmaisee, voivatko käyttöoikeudet muuttua joustavasti ilman roolirakenteen tai pääsyrakenteen uudelleenmäärittelyä. Esimerkiksi ympäristötekijät voivat vaikuttaa myönnettyyn käyttöoikeuteen.

Skaalautuvuus ilmaisee mallin toimivuuden järjestelmän koon kasvaessa.

Hallinnollinen kuormitus kuvaa järjestelmän ylläpitäjän työmäärää.

Auditointi tarkoittaa oikeuksien todennettavuutta. Tässä yhteydessä auditoinnilla tarkoitetaan käyttäjille myönnettyjen valtuutusten todentamista.

Kriteeri	**DAC**	**MAC**	**RBAC**	**ABAC**
Hallinnan periaate	Omistajakeskeinen	Keskitetty luokittelu	Roolipohjainen	Attribuuttipohjainen
Päätöksenteon perusta	ACL / omistajan päätös	Turvallisuusluokat	Rooli-oikeusliitokset	Politiikat ja attribuutit
Hienojakoisuus	Matala / keskitaso	Matala	Keskitaso	Korkea
Kontekstisidonnaisuus	Ei tuettu	Rajoitettu	Rajoitettu (laajennuksin)	Luontainen osa mallia
Dynaamisuus	Matala	Matala	Rajattu	Korkea
Skaalautuvuus	Heikko	Heikko / kohtalainen	Hyvä	Hyvä / erittäin hyvä
Hallinnollinen kuormitus	Hajautettu	Korkea keskitetty	Kohtalainen	Aluksi korkea
Auditointi	Selkeä ACL-tasolla	Selkeä luokitusten kautta	Selkeä roolirakenteen kautta	Vaatii politiikka-analyysin

Taulukko 4.2: Pääsynhallintamallien vertailu

5 Yhteenveto

Monen käyttäjän verkkosovelluksissa tietoturva on olennainen osa järjestelmän suunnittelua. Tietoturva on laaja kokonaisuus, mutta tässä tutkielmassa keskityttiin käyttäjähallinnan tutkimiseen, eli miten voidaan hallinnoida käyttäjien käyttöoikeuksia ja tietojen näkyvyyttä erilaisilla pääsynhallintamalleilla. Tutkimuksen lähtökohtana oli löytää vastaukset kahteen tutkimuskysymykseen.

TK1: Millaisilla käyttöoikeuksien ja tietojen näkyvyyden hallintamalleilla voidaan ohjelmistopalvelun tietoturva toteuttaa? Lähdeaineistosta ilmeni, että pääsynhallinnan alalla on tehty paljon tutkimusta jo useiden vuosikymmenten ajan. Erilaisia pääsynhallintamalleja on kehitetty lukuisia, joista monet ovat aikaisemmin kehitettyjen mallien yhdistelmiä. Tämän tutkielman aikataulun ja koon rajaamiseksi tutkimus kohdistettiin yleisesti julkaisuissa viitattuihin perinteisiin malleihin, jotka ovat harkinnanvarainen pääsynhallinta DAC, pakollinen pääsynhallinta MAC, roolipohjainen pääsynhallinta RBAC ja attribuuttipohjainen pääsynhallinta ABAC. Lähdeaineistosta koostettiin niiden keskeiset toimintaperiaatteet sekä soveltuvuus erilaisiin järjestelmiin. Kaikilla neljällä mallilla voidaan toteuttaa käyttöoikeuksien ja tietojen näkyvyyden hallinta, mutta niiden soveltuvuus nykyaikaisiin verkkosovelluksiin vaihtelee merkittävästi.

TK2: Millaisten kriteerien perusteella voidaan valita sopiva käyttöoikeusmalli tiettyyn käyttötapaukseen? Lähdeaineistosta nousi esiin useita eri kriteerejä, jotka pitää ottaa huomioon käyttöoikeusmallia valittaessa:

- Hallinnan periaate
- Päätöksenteon perusta
- Hienojakoisuus
- Kontekstisidonnaisuus
- Dynaamisuus
- Skaalautuvuus
- Hallinnollinen kuormitus
- Auditointi

Kriteerejä voidaan hyödyntää soveltuvan pääsynhallintamallin valinnassa, johon vaikuttaa muun muassa järjestelmän koko, käyttäjätyypit, vaadittu hienojakoisuus, sääntelyvaatimukset ja hallinnolliset resurssit. Ei siis ole olemassa parasta kaikkiin järjestelmiin sopivaa mallia, vaan järjestelmän pääsynhallintamallin suunnittelijoiden pitää tehdä tarkka analyysi vaadittavista tietoturvakäytännöistä ja tietoturvamalleista.

Tätä tutkielmaa voisi jatkaa pro gradu -tutkielmalla tai diplomityöllä, joka keskittyisi tutkielman motivaattorina olevan verkkosovelluksen käyttäjähallinnan jatkokehittämiseen. Nyt saatuja tutkimustuloksia pääsynhallintamallien soveltuvuudesta ja valintakriteereistä voisi laajentaa uudempien mallien lisätutkimuksella, ja näitä tuloksia käyttäen tehdä analyysi käyttäjähallinnan nykytilanteesta sekä jatkokehitystarpeista. Myös varsinaisen kehitystyön voisi lisätä jatkotutkielmaan soveltuvalla tasolla.

Yleisesti tämän tutkielman tuloksena saatu kuvaus tietoturvan periaatteista ja perinteisistä pääsynhallintamalleista toimii hyvänä yleisenä johdantona tietoturvaan. Suoraksi jatkotutkimuksen aiheeksi sopisi esitettyjen mallien syvällisempi tutkiminen, miten niitä voidaan ohjelmistossa ja tietokantatasolla toteuttaa. Lisäksi

voitaisiin laajemmin tutkia muita perinteisten pääsynhallintamallien jälkeen kehitettyjä malleja. Muita kiinnostavia lisätutkimuksen aiheita ovat tietoturvan toiset osa-alueet ja tietosuoja, sekä tietoturvauhkien osalta tekoälyn hyödyntäminen tietomurtojen havainnointiin ja estämiseen.

Lähdeluettelo

[1] F. Richter. ”Infographic: AWS stays ahead as cloud market accelerates”, Statista Daily Data, viitattu 10. marraskuuta 2025. url: https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers.

[2] G. Kulkarni, ”Cloud computing-software as service”, *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, vol. 1, nro 1, s. 11–16, 29. tammikuuta 2012, ISSN: 2089-3337. DOI: 10.11591/closer.v1i1.218.

[3] A. S. Rumale ja D. N. Chaudhari, ”Cloud computing: Software as a service”, teoksessa *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, helmikuu 2017, s. 1–6. DOI: 10.1109/ICECCT.2017.8117817.

[4] M. D. Jacyntho, D. Schwabe ja G. Rossi, ”A software architecture for structuring complex web applications”, *Journal of Web Engineering*, s. 37–60, 26. syyskuuta 2002, ISSN: 1544-5976.

[5] M. J. Grant ja A. Booth, ”A typology of reviews: an analysis of 14 review types and associated methodologies”, *Health Information and Libraries Journal*, vol. 26, nro 2, s. 91–108, kesäkuu 2009, ISSN: 1471-1834. DOI: 10.1111/j.1471-1842.2009.00848.x.

[6] H. Hristov, S. Cheresharov, S. Chonkov ja K. Tsvetanov, ”Information Security in the Design of Web-Based Software Systems”, teoksessa *2020 International*

Conference Automatics and Informatics (ICAI), 1. lokakuuta 2020, s. 1–6. DOI: 10.1109/ICAI50593.2020.9311305.

[7] G. Chassang, "The impact of the EU general data protection regulation on scientific research", *ECANCERMEDICALSCIENCE*, vol. 11, 3. tammikuuta 2017, ISSN: 1754-6605. DOI: 10.3332/ecancer.2017.709.

[8] R. D. O. Albuquerque, L. J. Garcia Villalba, A. L. Sandoval Orozco, F. Buiati ja T.-H. Kim, "A Layered Trust Information Security Architecture", *Sensors*, vol. 14, nro 12, s. 22 754–22 772, joulukuu 2014, ISSN: 1424-8220. DOI: 10.3390/s141222754.

[9] I. Indu ja P. M. Rubesh Anand, "Hybrid authentication and authorization model for web based applications", teoksessa *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai: IEEE, maaliskuu 2016, s. 1187–1191, ISBN: 978-1-4673-9338-6. DOI: 10.1109/WiSPNET.2016.7566324.

[10] A. Jøsang, "A consistent definition of authorization", teoksessa *Security and Trust Management*, G. Livraga ja C. Mitchell, toim., Cham: Springer International Publishing, 2017, s. 134–144, ISBN: 978-3-319-68063-7. DOI: 10.1007/978-3-319-68063-7_9.

[11] J. Berrios, E. Mosher, S. Benzo, C. Grajeda ja I. Baggili, "Factorizing 2FA: Forensic analysis of two-factor authentication applications", *Forensic Science International: Digital Investigation*, vol. 45, heinäkuu 2023, ISSN: 26662817. DOI: 10.1016/j.fsidi.2023.301569.

[12] B. Jayant.D, U. Swapnaja A, A. Sulabha S ja M. Dattatray G, "Analysis of DAC MAC RBAC access control based models for security", *International Journal of Computer Applications*, vol. 104, nro 5, s. 6–13, 18. lokakuuta 2014, ISSN: 09758887. DOI: 10.5120/18196-9115.

[13] M. U. Aftab, A. Hamza, A. Oluwasanmi, X. Nie, M. S. Sarfraz, D. Shehzad, Z. Qin ja A. Rafiq, ”Traditional and hybrid access control models: A detailed survey”, *Security and Communication Networks*, vol. 2022, T. R. G, toim., s. 1–12, 7. helmikuuta 2022, ISSN: 1939-0122, 1939-0114. DOI: 10.1155/2022/1560885.

[14] L. Golightly, P. Modesti, R. Garcia ja V. Chang, ”Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN”, *Cyber Security and Applications*, vol. 1, s. 100 015, joulukuu 2023, ISSN: 27729184. DOI: 10.1016/j.csa.2023.100015.

[15] G. Karataş ja A. Akbulut, ”Survey on access control mechanisms in cloud computing”, *Journal of Cyber Security and Mobility*, vol. 7, nro 3, s. 1–36, 2018, ISSN: 2245-1439. DOI: 10.13052/jcsm2245-1439.731.

[16] V. C. Hu, D. Ferraiolo ja D. R. Kuhn, ”A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC”, teoksessa *Data and Applications Security and Privacy XXVI*, sarja Lecture Notes in Computer Science, N. Cuppens-Boulahia, F. Cuppens ja J. Garcia-Alfaro, toim., vol. 7371, Berlin, Heidelberg: Springer, 2012, s. 41–55. DOI: 10.1007/978-3-642-31540-4_4.

[17] ”The state of authorization - 2025”, viitattu 27. tammikuuta 2026. url: https://www.permit.io/blog/state-of-authorization-2025.

[18] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller ja K. Scarfone, ”Guide to attribute based access control (ABAC) definition and considerations”, National Institute of Standards ja Technology, NIST SP 800-162, tammikuu 2014, NIST SP 800–162. DOI: 10.6028/NIST.SP.800-162.

[19] D. Xu ja Y. Zhang, ”Specification and Analysis of Attribute-Based Access Control Policies: An Overview”, teoksessa *2014 IEEE Eighth International*

Conference on Software Security and Reliability-Companion, kesäkuu 2014, s. 41–49. DOI: 10.1109/SERE-C.2014.21.