Oskari Vänskä

Selaimen sormenjäljen vaikutus yksityisyyteen moderneissa verkkoselaimissa

Tietotekniikan kandidaatintutkielma

30. huhtikuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Oskari Vänskä

Yhteystiedot: osvekava@student.jyu.fi

Ohjaaja: Sanna Juutinen

Työn nimi: Selaimen sormenjäljen vaikutus yksityisyyteen moderneissa verkkoselaimissa

Title in English: The impact of Browser Fingerprinting on Privacy in Modern Browsers

Työ: Kandidaatintutkielma

Opintosuunta: Kandidaatintutkielma

Sivumäärä: 21+0

Tiivistelmä: Käyttäjän yksilöimistä varten kehitetyt sormenjälkiteknologiat selainympäristössä uhkaavat käyttäjän yksityisyyttä. Tutkimuksen tarkoitus on selvittää sormenjälkien luonne, vaikutus, suojakeinot ja mahdolliset ratkaisut.

Avainsanat: Selaimen sormenjälki, yksityisyys

Abstract: Fingerprinting technologies designed to identify users present a threat to users' privacy. The objective of this paper is to investigate the nature, effects, protective measures and possible solutions to fingerprinting.

Keywords: Browser fingerprint, privacy

Sisällys

1 Johdanto

Yksityisyys internetissä on ollut polttava aihe yhtä kauan kuin internet on ollut olemassa. Käyttäjä on aina pyritty tunnistamaan käytettävissä olevien tekniikoiden avulla ja niiden vanhentuessa on kehitetty uusia tekniikoita. Alkuaikoina pelkkä IP-osoite riitti tunnistukseen, kunnes dynaaminen IP-osoite ja osoitteenmuunnos tekivät siitä liian epäluotettavan keinon palveluntarjoajien tarkoituksiin (Boda ym. 2011). Myöhemmin evästeitä käytettiin käyttäjän tunnistamiseen, kunnes tekniikka menetti tehokkuutensa suojauskeinojen ja kasvaneen tietoisuuden myötä (Nikiforakis ym. 2013)[s.541]. Evästeiden ongelma oli se, että niiden poisto ja estäminen on modernien selainten ominaisuuksien avulla triviaalista.

Vanhojen tunnisteiden heikkous onkin ollut linkki käyttäjän selain- ja laiteympäristöön. Oikeilla työkaluilla on mahdollista hallita tunnisteita ja niiden jakamista palveluntarjoajille. Lähes kolmannes käyttäjistä poistaa kaikki evästeet kuukauden sisällä sivustolla käymisen jälkeen (Nikiforakis ym. 2013)[s.541]. Viimeisen vuosikymmenen aikana käyttäjän tunnistamisen kehitys on edennyt passiivisten tekniikoiden suuntaan. Bodan ym. (2011) määritelmän mukaan passiiviset tunnistetekniikat ovat luonteeltaan sellaisia, että ne poimivat käyttäjän tietoja erilaisten kyselyiden avulla ilman, että tunnistavia elementtejä tarvitsee säilyttää käyttäjän laitteistolla.

Passiivisten tunnistetekniikoiden käyttö selainympäristössä on luonut uuden suunnan käyttäjän tunnistamiseen. Selainten ja ohjelmistojen monimutkaistuessa myös tunnistetietojen monipuolisuus kasvaa, jolloin palveluntarjoajan mahdollisuudet tiedon hyödyntämiseen kasvavat. Tekniikkaa kutsutaan kokoavasti selaimen sormenjäljeksi (Eckersley 2010).

Tämä tutkielma käsittelee selaimen sormenjäljen määritelmää, yleisimpiä sormenjälkitekniikoita ja suojauskeinoja sekä sormenjäljen vaikutusta käyttäjän yksityisyyteen. Tutkielma pyrkii selvittämään, onko yksityisyyttä mahdollista saavuttaa sormenjälkiteknologian kanssa ollenkaan ja mitä sormenjäljeltä suojautuminen vaatii.

2 Selaimen sormenjälki

Selaimen sormenjälki on tunniste, joka on rakennettu käyttäjän laite-, ohjelmisto-, ja asetuskonfiguraatioista saaduista tiedoista eli attribuuteista (Laperdrix, Rudametkin ja Baudry 2015)[s.878]. Tietoja pyritään keräämään sen verran, että muodostettu sormenjälki esiintyy ainoastaan yhdellä käyttäjällä. Mitä enemmän tietoa konfiguraatioista saadaan selville, sitä todennäköisempää on, että sormenjälki on ainutlaatuinen. Ainutlaatuisen sormenjäljen avulla käyttäjän tunnistaminen sivustojen välillä on yksinkertaista, eikä tunnistukseen tarvita käyttäjän suostumusta (Fiore ym. 2014). Koska tunnistus tapahtuu palveluntarjoajan laitteistolla, käyttäjä ei voi myöskään tietää yksityiskohtia sormenjäljestään (Al-Fannah ja Mitchell 2020)[s.167].

Yli 80 prosenttia käyttäjistä selaa internetiä sellaisella konfiguraatiolla, että heistä pystytään muodostamaan välittömästi sivustolle saapuessaan ainutlaatuinen sormenjälki (Eckersley 2010). Tunnistusprosentti on pysynyt vuosikymmenenkin jälkeenkin samana muuttuneesta ympäristöstä huolimatta (Andriamilanto ym. 2021), mistä voidaan päätellä, ettei sormenjälkiteknologian kehitys ainakaan ole hidastumassa.

Sormenjäljen attribuutteina voidaan teoriassa käyttää mitä tahansa palveluntarjoajan keräämiä tietoja tai muuttujia, jotka saavat erilaisia arvoja käyttäjien välillä. Andriamilanto ym. (2021, s. 5) käyttivät tutkimuksessaan kahtasataa eri JavaScript- ja kuuttatoista HTTP header field-arvoa sormenjäljen muodostamiseen ja Unger ym. (2013)[s.257] identifioi HTML5:ssä 242 yksilöimiseen sopivaa attribuuttia. Vaikka suurempi attribuuttien määrä loogisesti johtaa voimakkaampaan sormenjälkeen, ainutlaatuinen sormenjälki voi muodostua jo vähistä attribuuteista, parhaimmillaan yhdestä, jos niiden arvot ovat harvinaisia käyttäjäkunnassa (Laperdrix, Rudametkin ja Baudry 2015)[s.881].

Attribuuttien muuttuessa sormenjäljetkin muuttuvat, mikä voi johtaa tunnisteen ainutlaatuisuuden häviämiseen. Ilmiötä kutsutaan attribuuttien entropiaksi ja se on merkittävä tekijä attribuuttien ainutlaatuisuuden määrityksessä (Andriamilanto ym. 2021; Boda ym. 2011). On kuitenkin mahdollista arvata yksinkertaisellakin heuristisella algoritmilla päivittyneen sormenjäljen alkuperä, koska vanhasta sormenjäljestä on usein jäljellä tarvittava määrä attri-

buutteja arvaukseen (Eckersley 2010). Tämä tarkoittaa sitä, että ainutlaatuisen sormenjäljen uudelleenmuodostus on ensimmäistä sormenjälkeä helpompi toteuttaa.

Selaimen sormenjälki voi koostua usean eri sormenjälkitekniikan attribuuteista, jotka ovat mielekästä luokitella kategorioihin alkuperän mukaan. Upathilake, Li ja Matrawy (2015, s. 1) jakavat sormenjälkitekniikat neljään pääkategoriaan: selainympäristön sormenjälkeen, piirtoalueen sormenjälkeen, JavaScript-moottorin sormenjälkeen ja selaimesta riippumattomaan sormenjälkeen. On tietenkin mahdollista, ja todennäköisesti myös järkevää, kehittää sormenjälkiä, jotka hyödyntävät useamman eri kategorian attribuutteja. Andriamilanto ym. (2021) totesivat monipuolisen attribuuttijoukon olevan huomattavasti tehokkaampi ainutlaatuisen sormenjäljen muodostukseen kuin suppea attribuuttijoukko, vaikka yksittäiset attribuutit olisivatkin molemmissa vahvoja.

2.1 Selainympäristön sormenjälki

Selainympäristön sormenjälki koostuu pääosin primitiivisistä attribuuteista, kuten liitännäisistä, käyttäjäagentista ja selaimen ikkunan koosta (Upathilake, Li ja Matrawy 2015). Eckersley (2010) toteutti ensimmäisen suuren luokan tutkimuksen selaimen sormenjäljestä käyttämällä selainympäristön attribuutteja. Tutkimuksessa todettiin liitännäisten toimivan parhaiten tarkkojen versionumeroiden takia. Nikiforakis ym. (2013) [s. 543] huomasi, että sormenjäljitystä toteuttavat yritykset tyypillisesti täsmentävät attribuuttilistaa kutakin selainta varten, sillä selainkohtaiset ominaisuudet toimivat erityisen vahvoina attribuutteina.

Selainympäristön attribuuttien heikkous tunnisteina on se, että niitä on melko helppo väärentää (engl. spoof) (Upathilake, Li ja Matrawy 2015). Suurin osa attribuuteista on käyttäjän hallinnassa ja luonteeltaan sellaisia, että ne voidaan muun muassa asettaa identtisiksi toisen laitteen kanssa (Boda ym. 2011). Muokattavuuden lisäksi selainympäristön attribuutit ovat usein epävakaita. Päivitykset ja muutokset selainympäristössä helposti muuttavat myös sormenjälkeä (Nikiforakis ym. 2013), jolloin palveluntarjoaja joutuu käyttämään heuristiikkaa alkuperäisen sormenjäljen arvaamiseen. Selainympäristön sormenjäljissä heuristisen algoritmit käyttö on kuitenkin varsin tehokasta (Eckersley 2010).

2.2 Selaimesta riippumaton sormenjälki

Primitiiviset attribuutit eivät ole pelkästään selainympäristön attribuutteja. Jos attribuutit saadaan selaimen ulkopuolelta, sormenjälki on pysyvä selaimesta tai selaimen ominaisuuksista riippumatta. Tällöin esimerkiksi sormenjälkeä vastustavat liitännäiset menettävät tehokkuutensa täysin (Boda ym. 2011).

Boda ym. (2011) määrittelevät tärkeimmiksi selaimesta riippumattomiksi tunnisteiksi verkon, sovelluskerroksen ja JavaScript-kyselyjen tiedot. Toisin kuin selainympäristön sormenjälki, tietojen kerääminen vaatii käytännössä aina JavaScriptiä (Upathilake, Li ja Matrawy 2015). Selaimesta riippumattoman sormenjäljen huomattiin olevan kestävä selaimen ja tietokoneen ohjelmistopäivityksiin sekä liitännäisten muutoksiin, mutta molemmat sormenjäljet jakavat vaikeuden samanlaisten konfiguraatioiden tunnistamisessa (Boda ym. 2011; Upathilake, Li ja Matrawy 2015).

Selaimesta riippumattomilla metodeilla on mahdollista tunnistaa myös joitain selainympäristön liitännäisten attribuutteja, arvioi Boda ym. (2011). Tekniikka lisäisi selainympäristön sormenjälkien kestävyyttä liitännäisten versionumeroiden ja kombinaatioiden kanssa. Tehokkaimpina liitännäisinä toimisivat Bodan ym. (2011) mukaan harvoin päivitetyt ja selaimen kehittäjien hallinnan ulkopuolella olevat liitännäiset, kuten Java Runtime Environment (JRE).

2.3 Piirtoalueen sormenjälki

HTML5 piirtoalue-elementillä voidaan muodostaa kuvioita, joiden Mowery ja Shacham (2012) huomasi toimivan erittäin voimakkaana sormenjälkenä. Muodostetut kuviot vaihtelevat hieman riippuen järjestelmän renderöintimottorista, pikseleiden sijainnista, grafiikkakortista, käyttöjärjestelmästä, anti aliasing -tekniikasta ja muista renderöintiin vaikuttavista tekijöistä (Mowery ja Shacham 2012, s. 6). Yllättävää Moweryn ja Shachamin (2012) tutkimuksessa onkin, että myös standardisoidut kuviot, kuten tekstin fontit, toimivat sormenjäljen muodostukseen. Tehokkuutensa takia piirtoalueen sormenjäljestä tuli nopeasti yleisin palveluntarjoajien käyttämä sormenjälki (Acar ym. 2014).

Mowery ja Shacham (2012, s. 8) huomasivat, että joissain tapauksissa jopa samanlaisella laitteistolla muodostetut kuviot erosivat toisistaan muutaman pikselin verran. Tutkimuksessa renderöinnin erojen kuvailtiin mahdollisesti olevan epädeterminististä, mikä toisi lisää epävarmuutta sormenjäljeltä suojautumiseen. Havainto oli kuitenkin harvinainen ja pääsääntöisesti samalla laitteisto- ja ohjelmistokonfiguroinnilla piirtoalueen sormenjälkikin on sama.

Piirtoalueen sormenjälki on mahdollista muodostaa jokaisella JavaScriptia ajavalla sivulla riippumatta oikeuksista päästä käsiksi järjestelmän resursseihin (Upathilake, Li ja Matrawy 2015, s.2). Upathilaken, Lin ja Matrawyn (2015) mukaan piirtoalueen sormenjäljen heikkous on se, että sormenjälki saattaa vaihdella selainten välillä, mutta Mowery ja Shacham (2012) ei näe tätä ongelmana, vaan mahdollisuutena selaintyyppien erotteluun.

2.4 JavaScript-moottorin sormenjälki

Selaimen version tunnistamiseen on perinteisesti käytetty käyttäjäagenttia (Eckersley 2010). Tämän tekniikan ongelma kuitenkin on se, että tunniste on käyttäjän muokattavissa ja siten epäluotettava yksilöintiin (Upathilake, Li ja Matrawy 2015). Mulazzani ym. (2013) ehdottivat selaimen version tunnistamiseen uutta JavaScript-moottoriin perustuvaa tekniikkaa, jota käyttäjä ei pysty kiertämään tietoa väärentämällä.

Sormenjälki perustuu erilaisten sisäänrakennettujen JavaScript-testien tuloksiin, jotka vaihtelevat selainten ja versioiden välillä. Yksinkertaisinta on tunnistaa testit, jotka eivät ole käytössä tai hajoavat kussakin selainversiossa. Koska JavaScript-moottorin sormenjälkeä ei ole mahdollista väärentää muokatulla versioilla selaimesta, sen avulla voidaan tarkastaa, onko esimerkiksi käyttäjäagenttia muutettu alkuperäisestä (Mulazzani ym. 2013, s. 1).

3 Sormenjäljeltä suojautuminen

Koska sormenjäljen muodostus tapahtuu palveluntarjoajan laitteistolla (Al-Fannah ja Mitchell 2020)[s.167], sen estäminen muodostuu hyvin hankalaksi. Ei ole olemassa yhtä keinoa tai työkalua, joka estäisi kaikkien sormenjälkien attribuuttien jakamisen ellei ole valmis konfiguroimaan selainta siten, että uhraa käytännössä kaikkien sivustojen ominaisuuksien oikeanlaisen toiminnan (Al-Fannah ja Mitchell 2020). On kuitenkin olemassa monta sormenjälkeä hämäävää ja häiritsevää tekniikkaa, joiden avulla ainutlaatuisen sormenjäljen muodostamista voidaan vaikeuttaa.

Gómez-Boix ym. (2019, s. 68-69) lajittelevat sormenjäljen puolustuskeinot viiteen kategoriaan: skriptien estoon, attribuuttien estoon, attribuuttien vaihtoon, attribuuttien sumentamiseen ja uudelleenasetteluun. Tutkimuksessa esiteltiin myös uudenlainen tekniikka, joka perustuu saman konfiguraation käyttöön suuren käyttäjäkunnan välillä, tosin muun muassa torselaimen toiminta jo perustuu tähän periaatteeseen (Saito ym. 2017).

Tekniikoista suosituimmat ovat skriptien ja attribuuttien esto, sillä ne tulevat mukana suosituissa mainosten estoon tarkoitetuissa selainlaajennuksissa (Nikiforakis ym. 2013)[s.551]. Kyseisiä sovelluksia kuitenkin harvoin on suunniteltu suoranaiseen sormenjäljeltä puolustautumiseen (Gómez-Boix ym. 2019, s.68). Laajennukset lisäksi menettävät tehoaan jos sormenjälkiattribuutteja saadaan selainympäristön ulkopuolelta (Boda ym. 2011). Attribuuttien estoon lasketaan myös tekniikat, joilla estetään pääsy ohjelmointirajapintoihin, esimerkiksi HTML5 piirtoalueeseen, jolloin piirtoalueen sormenjäljen muodostus laajalti estetään (Al-Fannah ja Mitchell 2020).

Attribuuttien vaihdolla tarkoitetaan väärän tiedon antamista palveluntarjoajalle (engl. spoofing) ja attribuuttien sumentaminen häiritsevän tiedon lisäämistä vaihdon yhteydessä (Gómez-Boix ym. 2019). Molempien tarkoitus on esittää käyttäjän konfiguraatio jonain toisena siten, että sormenjäljestä ei saa selville alkuperäistä konfiguraatiota. Uudelleenasettelussa taas palveluntarjoajalle esitetään uusi konfiguraatio joka session yhteydessä tai tietyin aikavälein (Gómez-Boix ym. 2019, s. 68-69).

3.1 Tor-selain

Eräs erittäin voimakas sormenjäljen suojakeino on saman sormenjäljen jakamiseen perustuva Tor-selain. Kyseessä on yksityisyyttä silmällä pitäen valmiiksi konfiguroitu selain, jonka asetukset ovat samat kaikilla käyttäjillä olettaen ettei käyttäjä niitä itse muuta tai lisää (Al-Fannah ja Mitchell 2020). Tor-selain lisäksi käyttää Tor-verkon kerrossysteemiä anonymisoidakseen lähtevän ja saapuvan tiedon, mutta sormenjäljen kannalta suurin merkitys on varsinaisella selaimella (Saito ym. 2017).

Tor-selaimen mukainen asetusten jakaminen kaikille käyttäjille on huomattavasti tehokkaampi tapa saavuttaa yhteinen sormenjälki kuin asetusten määrittäminen itse. Saito ym. (2017, s. 505) osoittivat, että vain noin 14 prosenttia Tor-selainta käyttäneistä sai ainutlaatuisen sormenjäljen tarkkailuajan jälkeen kun taas muilla selaimilla tunnistettiin lähes kaikki. Ainutlaatuinen sormenjälki on Tor-selaimen kanssa kuitenkin moninkertaisesti todennäköisempi saada, jos selaimeen on tehty minkäänlaisia muutoksia tai jos käytössä on vanhentunut versio (Saito ym. 2017, s.512).

Täytyy kuitenkin ottaa huomioon, että sormenjäljen piilotetun luonteen (Al-Fannah ja Mitchell 2020) takia Tor-selaimen asetukset ovat vain projektin kehittäjien arvio tarpeellisesta yksityisyyden ja käytettävyyden tasosta jolloin virhearvioiden ja käytettävyyden kannalta välttämättömien, mutta sormenjälkiä mahdollistavien, ominaisuuksien jääminen selaimeen on mahdollista.

3.2 JavaScriptin esto

Toistuva piirre sormenjäljissä on jonkinlainen JavaScriptin hyödyntäminen (Upathilake, Li ja Matrawy 2015). Upathilaken, Lin ja Matrawyn (2016) neljässä kategoriassa vain selainympäristön sormenjäljen attribuutteja voidaan saada selville ilman JavaScriptiä, jolloin jäljelle jää huomattavan rajoitettu joukko tunnistetietoja sormenjäljen muodostamista varten. Tämä ei kuitenkaan tarkoita, että JavaScriptin estäminen riittäisi suojakeinona (Laperdrix, Rudametkin ja Baudry 2015). Media-kyselyt, CSS-ominaisuuksien toimivuudet ja HTTP-ylätunniste toimivat yksinään vahvana sormenjälkenä, kuten Mostsevenko (2021) näytti demossaan. Joitain JavaScriptiä vaativia sormenjälkitunnisteita, kuten fonttien tiedot ja resoluutio, voidaan

saada CSS-moottorin antamista signaaleista (Mostsevenko 2021).

Laperdrix, Rudametkin ja Baudry (2015, s. 888) osoittivat, että ainutlaatuisten sormenjälkien määrä laski noin 90 prosentista jopa alle 30 prosenttiin, jos JavaScript on estetty. Tutkimuksessa todettiin, että yhdistettynä käyttäjäagentin onnistuneeseen väärentämiseen, ainutlaatuisten sormenjälkien määrä laski jopa seitsemään prosenttiin. JavaScriptin ollessa pois päältä myöskään JavaScript-moottorin kautta käyttäjäagentin todentaminen ei toimi (Mulazzani ym. 2013).

JavaScriptin laajamittainen esto ei kuitenkaan ole realistinen estokeino, sillä useimmat sivut tarvitsevat ainakin joitain JavaScript-ominaisuuksia toimiakseen normaalisti. Ilman JavaScriptiä sivustoista puuttuisi käytännössä kaikki dynaamiset elementit ja monet sivut eivät latautuisi ollenkaan (Laperdrix, Rudametkin ja Baudry 2015, s.888).

3.3 Sormenjäljen paradoksi

Sormenjäljen muodostamisen attribuutteja on olemassa niin paljon, että niilta kaikilta suojautuminen vaatisi valtavan määrän työkaluja ja konfiguraatioita (Al-Fannah ja Mitchell 2020). Ongelmaksi nousee se, että mitä enemmän suojautumisen työkaluja on käytössä, sitä suurempi on riski, että käyttäjä saa ainutlaatuisen sormenjäljen työkalujen tai niiden jättämien jälkien takia (Al-Fannah ja Mitchell 2020). Jokainen konfiguraation muutos, oli se liitännäinen tai tietoturvaa korostava selain, saattaa toimia uusien attribuuttien lähteenä. Ilmiötä kutsutaan sormenjäljen paradoksiksi, jonka noteerattiin jo ensimmäisessä selaimen sormenjälkeä käsittelevässä tutkimuksessa (Eckersley 2010, s. 14).

Yksinkertaisimmillaan palveluntarjoaja näkee mitä osia sivustosta tai pyynnöistä käyttäjä on estänyt ja millä keinoilla esto on toteutettu. Eckersley (2010, s. 14) huomasi että käyttäjistä, joilla oli käytössään yksityisyyttä parantava Browzar-selain tai Privoxy web-välityspalvelin, pystyttiin kerämään lähes sama määrä yksilöivää tietoa kuin muista käyttäjistä, koska työkalut itsessään toimivat yksilöivinä tekijöinä.

Palveluntarjoajalle annettavien tietojen väärentäminen saattaa myös johtaa tilanteeseen, jossa annettu konfiguraatio todetaan mahdottomaksi (Eckersley 2010). Väärennetyn tiedon tu-

lisi olla sisäisesti täysin yhtenäistä välttääkseen ainutlaatuisen sormenjäljen muodostumisen. Nikiforakis ym. (2013, s. 552) havaitsivat kaikkien tutkimuksessaan käytettävien tietoa väärentävien selainten laajennuksien unohtavan screen-objektin arvon muuttamisen, jolloin mobiililaitteita esittävät tietokoneet paljastuivat heti. Ottaen huomioon sormenjälkien attribuuttien määrän, yksityisyyden saavuttaminen väärennetyllä tiedolla ei ole suositeltavaa kuin väliaikaisena toimenpiteenä (Al-Fannah ja Mitchell 2020).

Suojautumiskeinot ovat toki kehittyneet vuosikymmenen aikana sormenjälkiteknologian rinnalla, mutta käyttäjän on vaikea arvioida yksilöiviä tietoja, joita työkalut tai työkalujen kombinaatio todellisuudessa jättävät jälkeensä. Toisinaan työkalut toimivat jopa tarkoitusta vastaan ja ongelma tyypillisesti pahenee työkalujen määrän kasvaessa, jolloin käyttäjä joutuu tarkoin harkitsemaan valitsemiaan suojakeinoja.

3.4 Käyttäjän toimenpiteet

Käyttäjällä on todellisuudessa kolme tapaa vaikuttaa sormenjäljeltä suojautumiseen, jotka eivät olennaisesti muuta verkkoselaamisen luonnetta: selaimen valinta, konfiguraatio ja laajennukset (Al-Fannah ja Mitchell 2020). Selaimen ulkopuoliset tekijät, kuten komponenttivalinnat, näytön resoluutio ja muut ohjelmistovalinnat tietty vaikuttavat joihinkin sormenjälkiin, mutta varsinainen tiedonsiirto tapahtuu aina selainympäristössä (Laperdrix, Rudametkin ja Baudry 2015). Komponenttivalintojen vaikutus sormenjälkeen sisältää lisäksi epävarmuustekijöitä ja jopa satunnaisuutta joidenkin sormenjälkien kanssa (Mowery ja Shacham 2012), minkä seurauksena tavallisen käyttäjän tuskin kannattaa vaihtaa komponenttejaan sormenjäljeltä suojautumisen toivossa.

Muun kuin Tor-selaimen käyttö ilman konfigurointia tai laajennuksia ei todennäköisesti vaikuta sormenjälkeen, sillä kaikki modernit selaimet pitävät oletuksena sisällään ainakin joidenkin sormenjälkien muodostusta varten tarpeelliset elementit ja attribuutit (Laperdrix, Rudametkin ja Baudry 2015). Suositun selaimen ja selainversion valinta kuitenkin tekee selainympäristön sormenjäljestä jonkin verran vaikeamman muodostaa, koska useampi käyttäjä jakaa silloin kyseiset attribuutit (Al-Fannah ja Mitchell 2020, s. 168).

Selaimen konfiguroinnilla tarkoitetaan sitä, että tiettyjä ominaisuuksia estetään suoraan se-

laimen konfigurointiasetuksista. Selaimet eivät yleensä tarjoa mahdollisuutta osittaiseen estoon, vaan ominaisuudet on otettava pois päältä kokonaan (Al-Fannah ja Mitchell 2020), jolloin konfigurointi tarkoittaa karkeaa tapaa suojautua sormenjäljiltä. Jotkut selaimet, kuten Firefox, tarjoavat sisäänrakennettuja ominaisuuksia, joiden tarkoitus on vastustaa sormenjäljitystä (Al-Fannah ja Mitchell 2020).

Laajennukset tarjoavat sormenjäljiltä suojautumiseen valtavan määrän vaihtoehtoja, vaikkakin niiden suhteellisesta tehokkuudesta ei ole olemassa yleisesti hyväksyttävää todistusaineistoa (Al-Fannah ja Mitchell 2020, s. 168). Sormenjäljen paradoksin takia laajennuksien suuri määrä lisää yksilöimisen riskiä ja yhteensopivuus voi olla kyseenalaista, sillä laajennusten välillä tapahtuu kilpailua. Laajennukset ovat lisäksi luonnostaan rajoitettu vain toimintaperiaatteisiin, jotka selaimen valmistajat ovat sallineet (Al-Fannah ja Mitchell 2020).

Eckersley (2010) tunnisti vain kolme selainryhmää, joilla on suhteellisen hyvä kyky vastustaa sormenjälkiä: JavaScriptin estävät selaimet, Tor-selain tai TorButton-liitännäistä käyttävät selaimet ja jotkin älypuhelimen selaimet. Eckersley (2010) kuitenkin myönsi, että on varmasti olemassa lukuisia muitakin vastaavanlaisia laajennuksia tai selaintyyppejä. Täytyy myös ottaa huomioon, että tutkimus ei ota käsitä kehittyneempiä sormenjälkitekniikoita, kuten piirtoalueen sormenjälkeä.

3.5 Käyttäjäkokemus

Tieto sormenjäljestä ja mahdollisista suojauskeinoista on vähintäänkin monimutkainen ongelma ratkaista tavanomaisen käyttäjän näkökulmasta. Sormenjälkiteknologia tulisi olla prioriteettina korkealla jos käyttäjä pitää verkossa toimimisen yksityisyyttä tärkeänä, mutta varmuutta yksityisyydestä tuskin voidaan saada (Al-Fannah ja Mitchell 2020). Sormenjäljistä tehdyt tutkimukset antavat pessimistisen arvion käyttäjän yksityisyyden toteutumisen kannalta (Mowery ja Shacham 2012) ja ongelmaa on kuvailtu jopa mahdottomaksi ratkaista (Al-Fannah ja Mitchell 2020). Mikäli käyttäjä haluaa maksimoida suojansa sormenjäljiltä, käyttäjäkokemuksen voidaan nähdä olevan parhaimmassa tapauksessa puutteellinen ja pahimmillaan kelvoton (Boda ym. 2011).

Käyttäjän näkökulmasta yksityisyys- ja tietoturva-asetusten säätäminen vie liikaa aikaa ja

säädön määrällä on negatiivinen vaikutus muun muassa itsensä ilmaisuun verkossa (Rauhala, Tyrväinen ja Zaidenberg 2019, s. 399). Sormenjälkiteknologian tehokkuuden, huomaamattomuuden ja suuren yksityisyyden uhan takia voidaan olettaa, että käyttäjäkokemus asetuksiin perehtyessä laskee Rauhalan, Tyrväisen ja Zaidenbergin (2019) havaintojen mukaisesti entistä enemmän.

4 Yleiset ratkaisut

Selaimen sormenjälkeen liittyvässä kirjallisuudessa on esitetty useita yleisiä teknisiä ratkaisuja ongelman lievittämiseksi. Ratkaisut perustuvat pääosin ominaisuuksien ja attribuuttien yhtenäistämiseen ja turhien sekä vanhentuneiden attribuuttien poistoon (Al-Fannah ja Mitchell 2020). Nikiforakis ym. (2013, s.553) ehdottavat kaikkien selainten ottavan käyttöön vain yhdenlaiset API-kutsut, jolloin JavaScriptin toiminta olisi yhtenäistä selainten välillä. Boda ym. (2011) lisäksi ehdottavat standardisoidun fonttisarjan kehittämistä fonttien renderöintiin liittyvien sormenjälkien ehkäisemiseksi ja laajennuksien versionumeroiden sekä käyttäjäagentin merkkijonon lyhentämistä.

Modernit selaimet sisältävät lisäksi useita ohjelmointirajapintoja, joita käytännössä mikään verkkosivu ei käytä (Snyder ym. 2016). Al-Fannahin ja Mitchellin (2020) mukaan tällaiset rajapinnat luovat vain ylimääräisiä mahdollisuuksia sormenjäljille, eikä niiden poistaminen monessa tapauksessa vaikuttaisi käyttäjäkokemukseen. Esimerkiksi laitteen akun rajapinta sisältää keinot tarkkailla käyttäjän akun kestoa jolloin alkuperäisen tarkoituksen mukaan sivun muuttaminen akun säästämiseksi olisi mahdollista, mutta rajapintaa käytetään lähes ainoastaan sormenjäljitykseen ja vuonna 2020 ainoastaan Chrome tuki kyseistä rajapintaa. (Diaz ym. 2015).

Piirtoalueen API mahdollistaa piirrettyjen kuvioiden pikselidatan poimimisen, jolloin palveluntarjoajat voivat vertailla kuvioiden eroja ja käyttää niitä hyödykseen piirtoalueen sormenjäljessä. Al-Fannah ja Mitchell (2020, s.171) ehdottavat kyseisen ominaisuuden poistamista kokonaan, koska se ei vaikuttaisi lainkaan piirtoalueen varsinaiseen käyttötarkoitukseen ja toimii ainoastaan sormenjäljityksen mahdollistavana ominaisuutena. Poisto tekisi piirtoalueen sormenjäljestä nykyisessä muodossaan käyttökelvottoman. Al-Fannah ja Mitchell lisäävät, että koska suurin osa sormenjäljityksestä tapahtuu kolmansien osapuolien kautta, ohjelmointirajapinnoista pitäisi estää muiden kuin alkuperäisen sivun pääsy ohjelmointirajapintoihin ellei käytölle ole selkeää syytä. Helpoimmin toteutettavaksi suojakeinoksi Mowery ja Shacham (2012) ehdottivat, että käyttäjältä vaadittaisiin suostumus aina kun sivu haluaa kerätä pikselidataa, tosin ratkaisu johtaisi käyttäjän näkökulmasta ylimääräisiin kyselyikkunoihin. Ominaisuus on jo toteutettu muun muassa HTML5 sijaintitietojen API:n kanssa.

Informoiminen ja varoittaminen mahdollisesta sormenjäljityksestä lisäisi käyttäjän mahdollisuuksia yksityisyytensä hallintaan (Al-Fannah ja Mitchell 2020), vaikka se olisikin teknisesti hankala toteuttaa. Al-Fannah ja Mitchell (2020, s.171) esittävät ratkaisuksi koneoppimistekniikoita, jotka arvioivat milloin sivustot keräävät palvelun tarkoitukseen ei-relevanttia tietoa. Esimerkiksi jos piirtoalueen rajapinnan käyttöä havaitaan silloin kun sille ei selvästi ole syytä, käyttäjä saisi ilmoituksen mahdollisesta sormenjäljityksestä. Al-Fannah ja Mitchell (2020) kuitenkin myöntävät, että ilmoitusten määrä todennäköisesti olisi liikaa ollakseen hyödyllinen, jos käyttäjä haluaa saada kaikista sormenjälkiepäilyistä tiedon.

5 Hyötykäytöt

Käyttäjän tunnistamiseen käytetään edelleen suurelta osin tunnuksen ja salasanan paria. Tunnistuksen vahvistamiseksi ja hyökkäysten estämiseksi käytetään lisäksi kasvavassa määrin ylimääräisiä tunnistuselementtejä, kuten kaksivaiheista tunnistusta ja turvakysymyksiä. Monimutkaisten tunnistusvaiheet heikentävät kuitenkin käytettävyyttä ja monimutkaistavat palvelun käyttöönottoa (Andriamilanto ym. 2021).

Selaimen sormenjäljen tuomaa tunnistusmahdollisuutta voidaan käyttää perinteisten tunnistuskeinojen kanssa vahvistavana tekijänä (Al-Fannah ja Mitchell 2020). Sormenjäljitys on hyvin voimakas ja käyttäjän näkökulmasta huomaamaton prosessi, eikä se vaadi käyttäjältä toimenpiteitä tai laitteelle muita ohjelmistoja. Muutamat yritykset käyttävät jo sormenjälkitekniikoita käyttäjien autentikoinnissa (Andriamilanto ym. 2021), mutta laajamittaista tutkimusta selaimen sormenjäljen soveltuvuudesta autentikointiin ei ole vielä toteutettu.

Sormenjälkeä voidaan myös käyttää istuntokaappausten estämiseen (Unger ym. 2013). Yhteyden autentikointi tapahtuu nykyään suurelta osin HTTPS-yhteyden kautta, mutta sen jälkeen sessio pysyy turvaamattomana. Sessionaikainen autentikointi on sormenjäljityksen avulla mahdollista, jolloin suuret muutokset käyttäjän sormenjäljessä voivat paljastaa kaappauksen. Unger ym. (2013) esittivät Session Hijacking Prevention Framework (SHPF) kehyksen, joka käyttäen selainympäristön ja JavaScript-moottorin sormenjälkeä tekee ajoittain testejä ja tarvittaessa keskeyttää istunnon hyökkäyksen havaittaessa. Mulazzani ym. (2013) kuvailevat JavaScript-moottorin sormenjälkeä erityisen tehokkaaksi istuntokaappauksen huomaamiseen, koska sormenjälki paljastaa väärennetyn käyttäjäagentin. Ungerin ym. (2013) kehystä on myös tutkimuksen mukaan yksinkertaista laajentaa muihin sormenjälkitekniikoihin. Sessionaikasta autentikointia käytetään nykyisin pääosin sivuilla, jotka sisältävät maksullisen tilauksen. Näistä suurin osa toimii aikuisviihteen alalla (Nikiforakis ym. 2013).

6 Yhteenveto

Selaimen sormenjälki on käyttäjän yksilöimisen tekniikoiden luonnollinen kehityssuunta evästeiden jälkeen. Kyseessä on laaja-alainen ja monimutkainen ilmiö, joka ei ole rajoittunut yksittäisiin teknologioihin tai tekniikoihin. Käyttäjän yksityisyyden kannalta selaimen sormenjälki on erittäin haitallinen tekijä, johon ratkaisua ei ainakaan vielä ole olemassa. Eckersley (2010) suositteli jo ensimmäisessä selaimen sormenjäljen tutkimuksessa lainsäädännön muuttamista sormenjälkien hillitsemiseksi.

Vaikka sormenjälkiteknologialla on olemassa joitakin hyötykäyttöjä käyttäjän tunnuksen ja session autentikoinnin kanssa, kiinnostus sormenjälkiin lähes aina perustuu käyttäjän seuraamiseen ja yksilöimiseen palvelun ominaisuuksien kohdistamisen tai mainonnan takia. Sormenjäljellä on myös todellinen väärinkäytön riski varsinkin kolmansien osapuolien kanssa.

Käyttäjällä on olemassa monia keinoja, joilla sormenjäljen tehokkuutta voidaan heikentää. Koska suojauskeinojen tehokkuudesta ei kuitenkaan voida olla varmoja ja sormenjäljen paradoksi monimutkaistaa prosessia entisestään, täydellisestä suojasta muodostuu käytännössä mahdoton tavoite. Käyttäjän näkökulmasta sormenjäljen ongelma on liian vaikea ratkaistavaksi ilman, että selaimen ja ohjelmistojen valmistajat tekevät muutoksia tiedonjako- ja käsittelyprosesseihin sormenjäjen estämiseksi.

Lähteet

Acar, Gunes, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan ja Claudia Diaz. 2014. “The web never forgets: Persistent tracking mechanisms in the wild”. Teoksessa *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security,* 674–689.

Andriamilanto, Nampoina, Tristan Allard, Gaëtan Le Guelvouit ja Alexandre Garel. 2021. “A Large-scale Empirical Analysis of Browser Fingerprints Properties for Web Authentication”. *ACM Transactions on the Web (TWEB)* 16 (1): 1–62.

Boda, Károly, Ádám Máté Földes, Gábor György Gulyás ja Sándor Imre. 2011. “User tracking on the web via cross-browser fingerprinting”. Teoksessa *Nordic conference on secure it systems,* 31–46. Springer.

Diaz, Claudia, Lukasz Olejnik, Gunes Acar ja Claude Casteluccia. 2015. “The leaking battery: a privacy analysis of the html5 battery status api”. *Lecture notes in computer science* 9481:254–263.

Eckersley, Peter. 2010. “How unique is your web browser?” Teoksessa *International Symposium on Privacy Enhancing Technologies Symposium,* 1–18. Springer.

Al-Fannah, Nasser Mohammed, ja Chris Mitchell. 2020. “Too little too late: can we control browser fingerprinting?” *Journal of Intellectual Capital.*

Fiore, Ugo, Aniello Castiglione, Alfredo De Santis ja Francesco Palmieri. 2014. “Countering browser fingerprinting techniques: Constructing a fake profile with google chrome”. Teoksessa *2014 17th International Conference on Network-Based Information Systems,* 355–360. IEEE.

Gómez-Boix, Alejandro, Davide Frey, Yérom-David Bromberg ja Benoit Baudry. 2019. “A collaborative strategy for mitigating tracking through browser fingerprinting”. Teoksessa *Proceedings of the 6th ACM Workshop on Moving Target Defense,* 67–78.

Laperdrix, Pierre, Walter Rudametkin ja Benoit Baudry. 2015. “Mitigating browser fingerprint tracking: multi-level reconfiguration and diversification”. Teoksessa *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems,* 98–108. IEEE.

Mostsevenko, Sergey. 2021. *Demo: Disabling JavaScript Won’t Save You from Fingerprinting,* lokakuu. https://fingerprintjs.com/blog/disabling-javascript-wont-stop-fingerprinting/.

Mowery, Keaton, ja Hovav Shacham. 2012. “Pixel perfect: Fingerprinting canvas in HTML5”. *Proceedings of W2SP* 2012.

Mulazzani, Martin, Philipp Reschl, Markus Huber, Manuel Leithner, Sebastian Schrittwieser, Edgar Weippl ja FC Wien. 2013. “Fast and reliable browser identification with javascript engine fingerprinting”. Teoksessa *Web 2.0 Workshop on Security and Privacy (W2SP),* 5:4. Citeseer.

Nikiforakis, Nick, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens ja Giovanni Vigna. 2013. “Cookieless monster: Exploring the ecosystem of web-based device fingerprinting”. Teoksessa *2013 IEEE Symposium on Security and Privacy,* 541–555. IEEE.

Rauhala, Juhani, Pasi Tyrväinen ja Nezer Zaidenberg. 2019. “Does Time Spent on Device Security and Privacy Inhibit Online Expression?” Teoksessa *ECCWS 2019 18th European Conference on Cyber Warfare and Security,* 394. Academic Conferences ja publishing limited.

Saito, Takamichi, Kazushi Takahashi, Koki Yasuda, Kazuhisa Tanabe, Masayuki Taneoka ja Ryohei Hosoya. 2017. “Tor fingerprinting: Tor browser can mitigate browser fingerprinting?” Teoksessa *International Conference on Network-Based Information Systems,* 504–517. Springer.

Snyder, Peter, Lara Ansari, Cynthia Taylor ja Chris Kanich. 2016. “Browser feature usage on the modern web”. Teoksessa *Proceedings of the 2016 Internet Measurement Conference,* 97–110.

Unger, Thomas, Martin Mulazzani, Dominik Frühwirt, Markus Huber, Sebastian Schrittwieser ja Edgar Weippl. 2013. “Shpf: Enhancing http (s) session security with browser fingerprinting”. Teoksessa *2013 International Conference on Availability, Reliability and Security,* 255–261. IEEE.

Upathilake, Randika, Yingkun Li ja Ashraf Matrawy. 2015. “A classification of web browser fingerprinting techniques”. Teoksessa *2015 7th International Conference on New Technologies, Mobility and Security (NTMS),* 1–5. IEEE.