Tuija Huusko

TIETOJEN KALASTELU SOSIAALISEN MEDIAN PALVELUISSA

JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2014

TIIVISTELMÄ

Huusko, Tuija
Tietojen kalastelu sosiaalisen median palveluissa
Jyväskylä: Jyväskylän yliopisto, 2014, 33 s.
Tietojärjestelmätiede, kandidaatintutkielma
Ohjaaja: Veijalainen, Jari

Tämän kandidaatintutkielman tarkoituksena on tutkia tietojen kalastelun keinoja sosiaalisen median palveluissa sekä menetelmiä, joilla voidaan suojautua tietojen kalastelulta. Sosiaalinen media on valtavan laaja kenttä, johon myös tietojen kalastelijat ovat suunnanneet mielenkiintonsa, ja siihen kohdistetaan samanlaisia hyökkäyksiä kuin muillekin sivustoille. Hyökkäykset aiheuttavat taloudellisia ja henkilökohtaisia vahinkoja riippumatta siitä, mihin sivustoon ne suunnataan. Sosiaalisen median hyödyntäminen on suuresta käyttäjämäärästä johtuen tehokasta, sillä suuressa joukossa hyökkäykset leviävät moninkertaisesti nopeammin kuin joukossa, jossa on vain muutama käyttäjä. Tutkimus perustuu aikaisemmin julkaistuun tieteelliseen kirjallisuuteen aiheesta.

Asiasanat: tietojen kalastelu, sosiaalinen media, suojautuminen, Facebook, Twitter

ABSTRACT

Huusko, Tuija
Phishing in the social media services
Jyväskylä: University of Jyväskylä, 2014, 33 p.
Information Systems, Bachelor's thesis
Supervisor: Veijalainen, Jari

The purpose of this Bachelor's thesis is to examine how the social media services are exploited for phishing. It also reviews the methods which are used against phishing. The number of the social media services is huge and the phishers and hackers exploit the services skillfully. The attacks towards the social media services cause a lot of damages because of the high number of the users. Compared to a little group of people the attacks spread faster inside a big group. Hackers and phishers use the similar phishing methods in the social media services as on any other website. The research approach in this thesis is literature review.

Keywords: phishing, social media, protecting, Facebook, Twitter

KUVIOT

SISÄLLYS

1 JOHDANTO

Sosiaalisen median palvelut ovat muutamassa vuodessa tulleet erittäin suosituiksi ja arkipäiväiseksi osaksi elämää (Patsakis, Asthenidis & Chatzidimitriou, 2009). Palveluista on tullut hallitseva osa Internetiä (Kontaxis, Polakis, Ioannidis & Markatos, 2011). Helsingin Sanomien artikkelin (26.1.2014) mukaan sosiaalisen median palveluihin rekisteröityneitä käyttäjiä on jo yli 1,7 miljardia. Sosiaalisen median palvelut tarjoavat käyttäjälle mahdollisuuden luoda profiili, virtuaalinen versio itsestä, ja verkostoitua muiden käyttäjien kanssa. De Paulan (2010) mukaan sosiaaliset verkostosuhteet käsittävät kaksi fundamentaalista eli perusluonteista ominaisuutta, jotka ovat 1) käyttäjät ja 2) käyttäjien väliset suhteet. Virtuaaliset suhteet ja interaktiot heijastavat fyysisen elämän vastaavia (de Paula, 2010). Virtuaalinen elämä tekee näkyväksi todellisen elämän suhteet, mutta toisaalta verkossa on helppo myös muodostaa suhde täysin tuntemattomaan henkilöön.

Sosiaalisen median palveluihin rekisteröityneet käyttäjät hyödyntävät palveluita kommunikoidakseen fyysisestä elämästä tuttujen henkilöiden kanssa tai tavatakseen uusia ihmisiä. Niin harrastukset ja yhteiset mielenkiinnonkohteet kuin työ yhdistävät ihmisiä myös Internetissä. (de Paula, 2010.). Sosiaalisen median palvelut tarjoavat myös mahdollisuuden luoda virtuaalisia ryhmiä, joihin käyttäjä voi liittyä. Näin yhteisöllisyys ilmenee myös muualla kuin fyysisessä elämässä. Ahmedin ja Abulaishin (2012) mukaan yhteisöllinen rakenne luo luottamuksen ja luotettavuuden verkoston.

Sosiaalinen media –käsitteeseen liittyy vahvasti kaksi käsitettä: Web 2.0 ja käyttäjien tuottama sisältö (Kaplan & Haenlein, 2010). Web 2.0 -käsite on lähtöisin vuodelta 2004, jolloin ohjelmistojen kehittäjät ja loppukäyttäjät alkoivat hyödyntää WWW:tä. Käyttäjät eivät enää tuottaneet sisältöä yksilöinä, vaan avainsanana oli yhteistyö. Henkilökohtaisuus jäi Web 1.0:n aikakaudelle, ja tilalle tulivat blogit ja wikit. (Kaplan & Haenlein, 2010.). Käyttäjien tuottama sisältö –käsite yhdistää tavat hyödyntää sosiaalista mediaa. Termi kuvaa sisältöä, joka on julkisesti saatavilla ja loppukäyttäjien itsensä luomaa. (Kaplan & Haenlein, 2010.).

Myös tietojen kalastelijat ovat huomanneet sosiaalisen median palveluiden suosion. Tietojen kalastelu tarkoittaa henkilökohtaisen ja arkaluontoisen tiedon urkkimista ja hyödyntämistä rikollisin keinoin, ja on arvioitu, että pelkästään vuonna 2011 tietojen kalastelun seurauksena kärsitty taloudellinen vahinko oli noin 520 miljoonaa Us-dollaria. (Aggarwal, Rajadesingan & Kumaraguru, 2012). Lisäksi myös tavallisen käyttäjän on yhä helpompi hyödyntää sosiaalisen median palveluiden heikkouksia haitallisessa tarkoituksessa (Chandramouli, 2011).

Tietojen kalastelun historia alkaa noin 1990-luvun puolivälistä. Ensimmäistä kertaa tietojen kalastelua tarkoittavaa sanaa ”phishing” käytettiin vuonna 1996 (Ollmann, 2007), jolloin American Onlinea (AOL) vastaan tehtiin hyökkäys. Hakkerit loivat huijaustilejä American Onlineen käyttämällä vääriä identiteettejä ja automaattisesti generoituja ja väärennettyjä luottokorttinumeroita. Algoritmilla luodut luottokorttinumerot eivät erityisen hyvin vastanneet oikeita luottokorttinumeroita, mutta ne läpäisivät kuitenkin American Onlinen seulan, ja näin ne teoriassa olivat oikeita ja päteviä numeroita. (Jakobsson & Myers, 2007.). Tilien avulla hyökkääjät kalastelivat muilta käyttäjiltä salasanoja, ja phishing-sana alun perin tuleekin hakkerien tavasta käyttää sähköpostia ”syöttinä”, jolla saatiin hyväuskoiset ihmiset paljastamaan luottokorttinumeronsa, salasanansa ja muut tärkeät tietonsa. (Ollmann, 2007.). Ajan kuluessa huijarit ovat siirtyneet sähköpostin kautta tapahtuvasta tietojen kalastelusta hyödyntämään myös muita tapoja. Tekniikan kehittyessä myös tietojen kalastelu kehittyy ja muuttuu.

Tutkielman tarkoituksena on tutkia kirjallisuuden perusteella, mitä sosiaalisen median palveluiden turvallisuutta uhkaavia menetelmiä ja tekijöitä on olemassa ja miten näitä vastaan voi suojautua. Tutkimuskysymyksiä on kaksi:

- Millä tavoilla tietoja voidaan kalastella sosiaalisen median palveluissa?
- Miten tietojen kalastelua vastaan voidaan suojautua?

Tutkielman toisessa luvussa käydään läpi erilaisia keinoja kalastella tietoa sosiaalisen median palveluissa. Olemassa olevien palveluiden kirjo on laaja, joten kaikkia palveluita ei ole mahdollista käydä läpi. Suurin osa aihetta käsittelevästä kirjallisuudesta kuvailee tietojen kalastelua Facebookissa ja Twitterissä, joten tutkielmassa käsitellään aihetta enimmäkseen näiden palveluiden näkökulmasta. Toisaalta useita tutkielmassa esiteltyjä tietojen kalastelukeinoja voidaan hyödyntää myös muissa palveluissa, sillä keinot eivät välttämättä ole palveluriippuvaisia. Tietoja voidaan kalastella niin teknisin keinoin kuin psykologiaan ja ihmisen luontaiseen käyttäytymiseen perustuvin keinoin.

Tutkielman kolmannessa luvussa käsitellään suojautumiskeinoja tietojen kalastelua vastaan. Tutkielmassa on pyritty tarjoamaan katsaus tietojen kalastelumenetelmää vastaavaan suojautumiskeinoon, mutta osa kalastelukeinoista on luonteeltaan sellaisia, ettei niille suoraan ole olemassa suojautumiskeinoa. Osa kirjallisuudessa esitellyistä suojautumiskeinoista on artikkelin kirjoittamisen

aikaan myös ollut vasta pilottivaiheessa, ja on hankala sanoa, voiko tavallinen käyttäjä hyödyntää suojautumiskeinoa toimivasti ja tehokkaasti.

Tutkielman viimeisessä luvussa eli Yhteenvedossa käydään tutkielman keskeiset tulokset ja päätelmät läpi sekä pohditaan tietojen kalastelun ja sosiaalisen median välistä ongelmaa. Lisäksi Yhteenvedossa esitellään jatkotutkimusaiheita, joihin liittyen tutkimusta voisi jatkaa.

2 TIETOJEN KALASTELUN KEINOT

Tässä luvussa käydään läpi eri keinoja kalastella tietoa sosiaalisen median palveluissa. Osa keinoista on teknisiä, kun taas toiset keinot perustuvat psykologiaan ja ihmisen luontaisen käyttäytymisen hyödyntämiseen.

2.1 Cross Site Scripting -haavoittuvuus

Cross Site Scripting on järjestelmän haavoittuvuus, joka mahdollistaa muutellun ja haitallisen koodin suorittamisen. Sille ei ole olemassa suomenkielistä vastinetta, mutta muun muassa Faghani ja Saidi (2009), Shar ja Tan (2012) sekä Sun ja He (2012) käyttävät lyhennettä XSS. Cross Site Scripting on yksi yleisimmistä sovelluskerrokseen kohdistuvista hyökkäyksistä (Faghani & Saidi, 2009).

Muun muassa Facebook, Twitter, ja MySpace ovat joutuneet XSS-haavoittuvuuteen perustuvan hyökkäyksen kohteeksi (de Paula, 2010). Myös palveluiden mobiilisovellukseen voidaan kohdistaa hyökkäyksiä. Esimerkiksi Facebookin mobiilisovelluksen ohjelmointirajapintaan eli API:iin (*engl. application programming interface*) kohdistuneessa hyökkäyksessä hyödynnettiin puutteellista JavaScript-koodin vahvistamista ohjelmointirajapinnassa. Hyökkäyksen tarkoituksena oli käynnistää itsestään levittyvän madon leviäminen. (Weir, Toolan & Smeed, 2011.).

Faghanin ja Saidin (2009) mukaan Cross Site Scripting –haavoittuvuutta hyödyntäviä hyökkäyksiä on olemassa kaksi erilaista, ja ne eroavat toisistaan koodin saamisen tyylissä. Tavat ovat pysyvä (*engl. persistent attack*) ja ei-pysyvä hyökkäys (*engl. non-persistent attack*). Pysyvä hyökkäys tunnetaan myös nimellä varastoitu hyökkäys (*engl. stored attack*) ja sitä käytetään erityisesti sosiaalisen median palveluissa. Pysyvän hyökkäyksen piirteisiin kuuluu, että muuteltu koodi on pysyvästi liitetty palvelimelle esimerkiksi html-tekstinä tai kommenttikenttänä (Faghani & Saidi, 2009). Sisältöön ujutetaan haitallista koodia ja se tallennetaan palvelimelle (Shar & Tan, 2012). Selain suorittaa koodin, ja tätä kautta käyttäjä saa haitallisen JavaScript-koodin. Ei-pysyvä hyökkäys, joka tun-

netaan myös nimellä heijastava hyökkäys (*engl. reflective attack*), tarkoittaa, että koodi lähetetään käyttäjälle takaisin esimerkiksi virheilmoituksena tai hakutuloksena. (Faghani & Saidi, 2009.).

Cross Site Scripting on yhdistelmä uutta ja vanhaa. Se yhdistää vanhan viruksen ja nykyaikaiset haavoittuvuudet web-sovelluksissa (Faghani & Saidi, 2009). Cross Site Scriptingiä hyödyntävien hyökkäysten laajuudelle on monta syytä. Hyökkäys ei vaadi monimutkaisia järjestelmiä, vaan se hyödyntää web-sovelluksia, jotka näyttävät syötteen ennen tarkistamista**.** Monissa web-sovellusten ohjelmointikielissä ei myöskään ole suodatusta epäluotettavien syötteiden lähettämistä koskien. (Faghani & Saidi, 2009.).

Faghanin ja Nguyenin (2013) mukaan kolme suurinta tekijää, jotka vaikuttavat Cross Site Scriptingiä hyödyntävän hyökkäyksen leviämiseen sosiaalisen median palveluissa, ovat 1) käyttäjien käyttäytyminen, 2) yhteisöjen ryhmittynyt rakenne ja 3) yhteisön koko. Käyttäjien käyttäytyminen liittyy suurempaan todennäköisyyteen vierailla käyttäjän ystävien profiileissa useammin kuin täysin vieraiden ihmisten profiileissa (Faghani & Nguyen, 2013). Artikkelin kirjoittajat olettavat tutkimuksessaan, että käyttäjän profiili on täysin näkyvä kaikille hänen ystävilleen. Lisäksi oletuksena on, ettei profiili välttämättä näy ystävälistan ulkopuolella oleville käyttäjille samanlaisena

Sosiaalisen median palveluissa käyttäjät voivat myös luoda ryhmiä ja yhteisöjä. Käyttäjät tietyssä ryhmässä vierailevat useammin toistensa profiileissa kuin niiden käyttäjien, jotka eivät kuulu ryhmään. Yhteisöjen ryhmittynyt rakenne hidastaa haitallisen koodin leviämistä, sillä aluksi koodi pyörii vain ryhmän tai yhteisön sisällä johtuen aikaisemmin mainitusta käyttäjien vierailutaipumuksista. (Faghani & Nguyen, 2013.).

Hyökkäyksen leviämisnopeuteen vaikuttaa myös yhteisön koko. Hyökkäys leviää hitaammin, jos yhteisöjä on paljon mutta niissä on vähän jäseniä verrattuna tilanteeseen, jossa on vain yksi iso yhteisö. Tämä perustuu siihen, että isossa ryhmässä vuorovaikutuksen seurauksena kaikki saastuvat, jos yksi saastuu, mutta jos samat käyttäjät ovat jakaantuneet pienempiin ryhmiin ja yksi saastuu, niin vain yhden ryhmän jäsenet saavat haitallisen koodin, jolloin saastuneiden käyttäjien määrä on pienempi. (Faghani & Nguyen, 2013.). Faghanin ja Nguyenin (2013) artikkeli selittää tämän skenaarioilla, joissa 3000 käyttäjää muodostaa yhden ryhmän ja 3000 käyttäjää on jakautunut sataan ryhmään. Oletuksena on, että käyttäjät vierailevat vain ryhmän jäsenten profiileissa. Yhden ryhmän kaikki 3000 käyttäjää saastuu, sillä he kuuluvat samaan ryhmään. Jos ryhmiä sen sijaan on monta, saastuu vain kymmenesosa, sillä oletuksen mukaan vain ryhmän jäsenten profiileissa vieraillaan. (Faghani & Nguyen, 2013.). Tosin tämä esimerkki ei huomioi tilannetta, jossa käyttäjät eivät vieraile toistensa profiileissa. Mikäli interaktioita ei tapahdu, myöskään hyökkäys ei pääse leviämään tehokkaasti eteenpäin.

Cross Site Scriptingiä voidaan hyödyntää myös käyttäjän manipulointiin (*engl. social engineering)* perustuvien hyökkäysten yhteydessä. Yhdistelmässä hyökkääjä käyttää saastunutta JavaScript -koodia sivuston taustalla, joka on heikoin osa, johon käyttäjä luottaa (Chaitanya, Ponnapalli, Herts & Pablo, 2012).

Esimerkiksi Facebookissa levisi Rihanna-niminen roskaposti, joka hyödynsi Cross Site Scriptingiä ja käyttäjän manipulointia. Hyökkäyksen tarkoituksena oli levittää saastunutta JavaScript-koodia, joka julkaisi käyttäjän puolesta hänen tietämättään kuvia ja linkkejä (Chaitanya ym. 2012). Linkit johtivat hyökkääjän Facebook-sivuille ja ruudulle ilmestyi video, joka hetken kuluttua muuttui teksteiksi ”turvallisuus tarkastettu” ja ”jatka”. Klikkaamalla jatka-painiketta ruudulle ilmestyi ohjeita ja pyyntö painaa tiettyjä näppäimiä, ja mikäli käyttäjä jatkoi ohjeiden noudattamista, saastui hänen Facebook-sivunsa hyökkääjän JavaScript-koodilla. Käyttäjän omalle sivulle ilmestyi julkaisu kuvasta ja linkistä, jotka skripti sijoitti käyttäjän ystävien seinille (Chaitanya ym., 2012.). Roskaposti levisi eteenpäin, kun käyttäjän ystävät klikkasivat julkaistua linkkiä ja saivat edelleen saman JavaScript-koodin, joka julkaisi taas heidän ystäviensä profiileissa kuvia ja linkkejä.

Hyökkäyksessä hyökkääjä hyödynsi IFrame-kehyselementtiä, joka osoitti käyttäjälle näkyvästä sivusta hyökkääjän omalle sivulle. Hyökkääjän sivulla oli flash-tiedosto tyyppiä .swf, jonka sisältämä ActionScript-koodi asetti saastuneen JavaScript-koodin käyttäjän leikepöydälle. (Chaitanya ym. 2012.). Käyttäjä ei siis itse missään vaiheessa kopioinut koodia, vaan koodi tuli automaattisena hänen leikepöydälleen, josta se kopioitui tilapäivityksenä käyttäjän Facebook-sivulle.

2.2 Klikkaushuijaus

Klikkaushuijaus (*engl. clickjacking)* on yleinen tietojen kalastelukeino sosiaalisen median palveluissa. Se tunnetaan myös nimellä käyttäjärajapinnan korvaus – hyökkäys (*engl. UI-redressing*), ja nimensä mukaisesti klikkaushuijauksessa tarkoituksena on saada käyttäjä klikkaamaan linkkiä, joka vie ei-toivotulle sivustolle. (Chaitanya ym., 2012.). Klikkaushuijauksen tehokkuus perustuu siihen, ettei käyttäjä tiedä klikkaavansa huijaussivustolle vievää linkkiä, vaan hän luulee linkin olevan aito ja alkuperäinen (Chaitanya ym., 2013; Lundeen & Alves-Foss, 2012). Klikkaushuijaus ei hyödynnä sivuston ohjelmointivirheitä tai virheellistä konfiguraatiota, vaan se käyttää näkymätöntä IFrame-rakennetta ja sen sisältöä, esimerkiksi hyperlinkkejä ja painikkeita, vahingon aiheuttamiseen (Callegati & Ramilli, 2009).

Callegatin ja Ramillin (2009) mukaan onnistuneesta klikkaushuijauksesta on hyökkääjälle paljon hyötyä. Sen avulla hyökkääjä voi kalastella arkaluontoista ja henkilökohtaista tietoa. Käyttäjä näkee vain oikean sivun ja kirjoittaa siihen tietonsa. Todellisuudessa päällä on kuitenkin näkymätön huijaussivu. Hyökkääjä voi asettaa hyökkäyksen toimimaan niin, että näkymättömälle sivulle kirjoitetut tiedot siirtyvät suoraan hänelle sen sijaan, että ne lähetettäisiin oikealle vastaanottajalle (Callegati & Ramilli, 2009). Tämän takia sivustoilla, joilla käyttäjän täytyy kirjoittaa esimerkiksi salasanansa tai luottokorttinumeronsa, klikkaushuijaus on vaarallinen. Hyökkääjä voi myös levittää saastunutta Ja-

vaScript-koodia käyttäjän koneelle tai levittää entistä vaarallisempia ja monimutkaisempia turvallisuusuhkia käyttäjän kautta. (Callegati & Ramilli, 2009.).

Klikkaushuijauksessa hyökkääjä asettaa alkuperäisen painikkeen päälle tai alle oman näkymättömän painikkeensa, joka viittaa hyökkääjän sivuun. Kun käyttäjä klikkaa luotetun sivuston painiketta, klikkaa hän todellisuudessa näkymätöntä hyökkääjän painiketta. (Callegati & Ramilli, 2009.).

Callegatin ja Ramillin (2009) mukaan hyökkääjä luo saastuneen sivun, jonka klikattavat kohteet vastaavat turvallisen sivun painikkeita. Tämän jälkeen hyökkääjä asettaa sivut päällekkäin. Jos käyttäjä klikkaa sivun painikkeita, luulee hän klikkaavansa turvallista ja näkyvää sivua, vaikka todellisuudessa hän klikkaa näkymätöntä huijaussivua. (Callegati & Ramilli, 2009.).

Sivustot asetetaan päällekkäin käyttämällä esimerkiksi HTML IFrame -elementtiä, jonka avulla voidaan web-sivusto jakaa erilaisiin kehyksiin. IFrame mahdollistaa sen, että kehykset ja sivustot voivat olla päällekkäin. (Callegati & Ramilli, 2009.). Huijaussivu saadaan näkymättömäksi käyttämällä CSS:ää (Cascading Style Sheets), jonka avulla määritellään verkkojulkaisujen ulkoasu. Hyökkääjän painikkeet ja linkit ovat näkymättömiä, koska niiden sameus on asetettu nollaksi. Arvo 0 tarkoittaa, että painike on näkymätön. (Callegati & Ramilli, 2009.). Monet kehittäjät käyttävät HTML IFrame –elementtiä, joten teknisesti klikkaushuijauksen toteuttaminen ei ole vaikeaa. Muun muassa Facebookin tykkää- ja jaa-painikkeet sekä Twitterin tviittaa–painike ovat helposti käytettävissä klikkaushuijaukseen, koska niissä käytetään HTML IFrame -elementtiä. (Chaitanya ym. 2012; Callegati & Ramilli, 2009.).

Eräs klikkaushuijauksen muoto on tykkäyshuijaus (*engl. likejacking*), jota käytetään erityisesti Facebookissa. Sen toimintaperiaatteet ja tekninen toteutus ovat samat kuin perinteisessä klikkaushuijauksessa, mutta hyökkäyksen kohteena ovat Facebookin tykkää-painikkeet. Facebookissa on mahdollista nähdä, mitä materiaalia käyttäjän ystävät ovat jakaneet ja kommentoineet. Monet sivustot ovat myös liittäneet Facebookin ominaisuuden sivuihinsa (Weir ym., 2011). Jos käyttäjä jakaa, kommentoi tai tykkää sivustosta, jossa on Facebookin ominaisuus käytössä, tulee hänen aikajanalleen ilmoitus, joka näkyy kaikille hänen ystävilleen. Tykkäys-painike tietyllä sivustolla kertoo käyttäjälle mahdollisuudesta tykätä. Tykkääminen tapahtuu painiketta klikkaamalla ja koska painike yleensä on näkyvä, käyttäjä tekee tietoisen valinnan, haluaako tykätä sivustosta. Tykkäyshuijauksessa nimensä mukaisesti käyttäjä huijataan tykkäämään jostain linkistä tai sivustosta tietämättään. Tämä onnistuu niin, että klikattavan kohteen päälle on asetettu näkymätön Facebookin tykkää-painike (Rehman, Khan, Saqib & Kaleem, 2013). Eli kun käyttäjä klikkaa kohdetta (esimerkiksi linkki, kuva, tekstikenttä), klikkaa hän todellisuudessa näkymätöntä tykkää-painiketta. Facebook-tilin aikajanalle tulee ilmoitus, että käyttäjä on tykännyt linkistä tai sivustosta, mutta käyttäjä itse ei tätä ilmoitusta näe (Weir ym., 2011). Linkki sisältää usein roskapostia tai muuten haitallista materiaalia, mutta ulospäin linkki näyttää harmittomalta ja mielenkiintoiselta. Tykkäyshuijauksessa hyödynnetään käyttäjän hänen ystäviensä välistä luottamussuhdetta, sillä käyttäjät luottavat toisiinsa, ja uskovat linkin olevan harmiton. (Weir ym., 2011.).

Avatessaan linkin he joutuvat samalla tavalla tykkäyshuijauksen kohteeksi ja levittävät linkkiä eteenpäin.

2.3 Ystävä välissä –hyökkäys

Ystävä välissä –hyökkäys (*engl. Friend-In-The-Middle –attack)* on aktiivinen salakuunteluhyökkäys sosiaalisen median palveluita vastaan (Huber, Mulazzani, Kitzler, Goluch & Weippl, 2011). Koska sosiaalisen median palveluissa on valtava määrä henkilökohtaista tietoa, hyökkäyksillä on suuri vaikutus. Ystävä välissä -hyökkäys perustuu kommunikaatiolinkin puuttuvaan suojaukseen käyttäjän ja sosiaalisen median palvelun tarjoajan välillä (Cashion & Bassiouni, 2011). Useimmilla sosiaalisen median palveluiden sivuilla istunnon evästeet on suojattu paikallisesti käyttäjäpuolella. Evästeiden avulla voidaan muun muassa todentaa käyttäjä oikeaksi käyttäjäksi. Kuitenkin yhteys käyttäjän ja palvelun tarjoajan välillä on helposti kaapattavissa johtuen evästeiden salaamattomasta lähetyksestä. Aikaisemmin kaikki sosiaalisen median palveluiden tarjoajat eivät myöskään tukeneet HTTPS:ää, joka on suojattuun tiedonsiirtoon käytettävä protokolla, vaan ne käyttivät suojaamatonta HTTP-protokollaa. Kaappaamalla käyttäjän istunnon hyökkääjä pystyi esiintymään käyttäjänä ja soluttautumaan tämän sosiaaliseen verkostoon. (Cashion & Bassiouni, 2011.).

Ystävä välissä -hyökkäyksiä voidaan hyödyntää yhdessä sosiaalisen tietojen kalastelun kanssa. Käyttäjä voi esimerkiksi saada viestin, joka näyttää tulevan hänen ystävältään. Ystävä välissä -hyökkäyksen avulla tehty tiedonlouhinta sosiaalisen median palveluissa tuo hyödyllistä tietoa hyökkääjälle, mikä parantaa hyökkäyksen tehokkuutta ja onnistumista. (Huber ym., 2011.). Hyökkääjä voi muun muassa hyödyntää käyttäjän ja hänen ystäviensä kuvia, viestejä ja julkaisuja ja saada näin käyttäjän uskomaan viestin oikeellisuuteen ja aitouteen.

Eräs esimerkki varsinkin Facebookissa ja MySpacessa leviävästä viruksesta, joka hyödyntää luotettavuutta käyttäjän ja tämän ystävien välillä, on Koobface-virus. Se kerää käyttäjistä arkaluontoista tietoa, kuten luottokorttinumeroita. (de Paula, 2010.). Virus on levinnyt laajalle onnistuneesti, sillä se hyödyntää osittain psykologiaa. Ihminen luottaa ystäviinsä ja tuttuihinsa, koska se on lajille luonnollista. Tämän takia käyttäjällä ei ole syytä epäillä esimerkiksi Facebook-ystäviensä jakaman materiaalin luotettavuutta. Robertsonin, Panin ja Yuanin (2010) mukaan virus leviää hyperlinkkien kautta. Ulkoisesti näyttää, että käyttäjän ystävä on linkannut hauskan ja mielenkiintoisen videon, ja uteliaisuudesta käyttäjä avaa linkin päästäkseen katsomaan sen. Kun käyttäjä avaa linkin, ruudulle ilmestyy ponnahdusikkuna, joka vaatii käyttäjää päivittämään Adobe Flash playerin. Mikäli käyttäjä hyväksyy päivityksen ja klikkaa päivityksen latauspainiketta, latautuu käyttäjän koneelle Adobe Flash playerin sijasta troijalainen. Troijalainen asentaa web-välityspalvelimen ja takaportin käyttäjän järjestelmään ja se alkaa esiintyä käyttäjänä ja monistaa itseään hyödyntäen käyttäjän sosiaalista verkostoa. (Robertson ym. 2010.).

Sosiaalisessa mediassa käyttäjä ei voi yksin vaikuttaa profiilinsa ja itsensä turvallisuuteen, vaan turvallisuus syntyy kaikkien toiminnasta ja vaikutuksesta. Ystävä välissä –menetelmää voi hyödyntää myös yhdessä troijalaisen viruksen kanssa (Erlandsson, Boldt & Johnson, 2012). Kirjoittajien mukaan tietojen kalastelun epäsuora uhka aiheutuu, kun käyttäjän ystävä lataa troijalaisen viruksen sisältävän sovelluksen. Tällöin vaarantuvat troijalaisen sovelluksen ladanneen käyttäjän lisäksi hänen ystäviensä tilit, sillä sovellukset vuotavat tietoa kolmansille osapuolille (Erlandsson ym., 2012). Näin ollen on mahdollista, että sovellus kerää tietoa myös käyttäjän ystävistä.

2.4 Profiilin deaktivoiminen

Mahmood ja Desmedt (2012) esittelevät artikkelissaan hyökkäysmenetelmän, jossa hyödynnetään profiilien kytkemistä pois päältä ja aktivoimista taas uudelleen (*engl. deactivated friend attack*). Menetelmässä hyökkääjän täytyy ensin saada käyttäjä Facebook-ystäväkseen. Hyökkääjä voi esimerkiksi esiintyä jonakin tunnettuna henkilönä tai väittää olevansa todellisessa elämässä käyttäjän oikea ystävä tai tuttu.

Kun käyttäjä ja hyökkääjä ovat Facebook-ystäviä, hyökkääjä deaktivoi eli kytkee oman profiilinsa pois päältä. Deaktivoinnin seurauksena hyökkääjän profiili muuttuu näkymättömäksi. (Mahmood & Desmedt, 2012.). Hän on siis edelleen olemassa Facebookissa, mutta muut käyttäjät eivät voi nähdä häntä. Deaktivointi-aktivointi –menetelmää hyödynnetään tiedon louhinnassa. Aktivoidessaan profiilinsa hyökkääjä voi toimia normaalisti Facebookissa kalastellen tietoja kohteensa profiilista. Kun tietojen etsintä loppuu, muuttaa hyökkääjä taas profiilinsa näkymättömäksi deaktivoimalla sen. (Mahmood & Desmedt, 2012.).

Hyökkäyksen vakavuuteen on monia syitä, ja sitä vastaan on vaikea suojautua. Deaktivointikertojen lukumäärää ei välttämättä ole rajoitettu mitenkään, joten hyökkääjä voi käytännössä kerätä tietoa loputtomasti ja päivittää tietoja sitä mukaa, kun käyttäjästä ilmenee uutta tietoa (Mahmood & Desmedt, 2012). Käyttäjä ei voi esimerkiksi päivittää tietojaan tai julkaista mitään hyökkääjän tietämättä ja on näin jatkuvan valvonnan alla.

Facebookissa ystävät voi jakaa eri ryhmiin, ja julkaisut sekä kuvat voi valita näkymään eri ryhmille. Mikäli hyökkääjän profiili on deaktivoitu, sitä ei voi siirtää ryhmästä toiseen (Mahmood & Desmedt, 2012), ja siten käyttäjän kyseiselle ryhmälle näkyväksi asetetut julkaisut näkyvät myös hyökkääjälle. Deaktivoitua profiilia ei voi myöskään poistaa ystävälistalta, mikä antaa etulyöntiaseman hyökkääjälle. Mahmoodin ja Desmedt'n (2012) mukaan hyökkääjän voi poistaa ystävälistaltaan ainoastaan silloin, kun profiili on aktivoitu. Koska hyökkääjä pitää profiiliaan aktiivisena vain hetken aikaa ja epäsäännöllisesti, on pieni todennäköisyys, että käyttäjä on profiilin aktivoimisen aikaan kirjautuneena sisään, huomaa aktiivisen hyökkääjän profiilin ja ehtii poistaa sen ystävälistaltaan ennen kuin hyökkääjä muuttaa profiilinsa taas deaktiiviseksi.

Valvomalla muutamaa käyttäjää hyökkääjä voi saada laajan kuvan käyttäjien välisistä vuorovaikutussuhteista. Facebookissa on ominaisuus, jonka avulla voi nähdä käyttäjän ystävien väliset interaktiot, mikäli he ovat keskenään ystäviä Facebookissa. Tätä kautta hyökkääjä voi saada tietoa muun muassa kohteidensa välisestä suhteesta, sillä ominaisuus paljastaa muun muassa Facebook-ystävyyden alkamisajankohdan, tapahtumat, joihin molemmat ovat osallistuneet, yhteiset ystävät, sivut, joista he molemmat ovat tykänneet ja toistensa aikajanalle kirjoittamat viestit. (Mahmood & Desmedt, 2012.). Tietoja yhdistelemällä hyökkääjä saa laajan ja monipuolisen kuvan käyttäjien henkilökohtaisesta elämästä, ja näitä tietoja voi hyödyntää tietojen kalastelussa ja hyökkäyksissä.

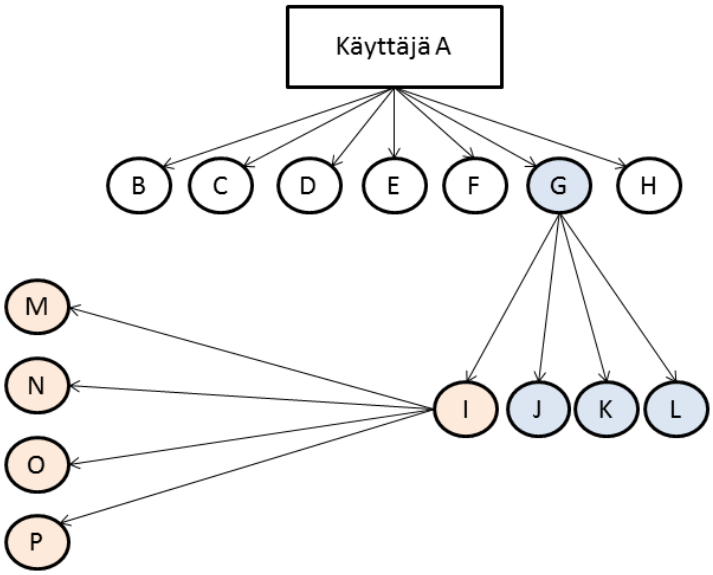
2.5 Tviittien hyödyntäminen

Twitterissä käyttäjä voi kirjoittaa 140 merkin pituisia päivityksiä eli tviittejä (*engl. tweet*). Hän voi myös tviitata uudelleen jonkun toisen kirjoittaman päivityksen (*engl. retweet*). Päivityksen voi osoittaa tietylle käyttäjälle kirjoittamalla tviittiin @käyttäjänimi tai sen voi luokitella koskemaan tiettyä aihetta kirjoittamalla #aihe. (Sanzgiri, Hughes & Ubadhyaya, 2013.). Näitä päivityksiä voidaan hyödyntää levitettäessä roskapostia ja muuta haitallista materiaalia. Ongelma ei niinkään ole tviittaus-toiminnossa itsessään, vaan tviittien sisällössä. Tviitteihin voi sisällyttää tavallisen tekstin lisäksi myös linkkejä. Koska tviittien pituus on rajoitettu, erilaiset URL-osoitteiden lyhennyspalvelut ovat tulleet suosituksi, sillä lyhyet osoitteet säästävät tilaa. Tällaisia palveluita ovat muun muassa bit.ly ja tiny URL. Lyhytosoitteet ovat normaaleita URL-osoitteita, jotka on muutettu lyhennyspalvelun avulla lyhyemmiksi. (Sanzgiri ym., 2013.).

Osa lyhennyspalveluista ei muuta samaa alkuperäistä URL-osoitetta samanlaiseksi lyhytosoitteeksi kaikille käyttäjille (Sanzgiri ym., 2013). Tämä tarkoittaa sitä, että yhdellä haitallista materiaalia sisältävällä alkuperäisellä URL-osoitteella voi olla monta erilaista lyhytosoitetta. Näin käyttäjän on hankala valvoa haitallisia osoitteita, sillä mikään ei takaa sitä, että näennäisesti eri lyhytosoitteet eivät vie samalle haitalliselle sivustolle. Käyttäjä ei myöskään etukäteen tiedä, mikä lyhytosoitteen alkuperäinen URL-osoite on, sillä hiirellä osoitteen osoittaminen ei paljasta sitä (Sanzgiri ym., 2013).

Twitterissä käyttäjä voi seurata (*engl. follow*) muita käyttäjiä. Toisin kuin muissa sosiaalisen median palveluissa, käyttäjien väliset suhteet eivät välttämättä ole kahdensuuntaisia (Sanzgiri ym., 2013). Tämä tarkoittaa sitä, että vaikka käyttäjä B seuraa käyttäjää A, käyttäjän A ei ole pakko seurata käyttäjää B. Twitterissä käyttäjän ja seuraajan välillä on puumainen rakenne. Kuvio 1 havainnollistaa, miten tviitit näkyvät käyttäjän seuraajille ja heidän seuraajilleen. Kuvio kuvaa tilannetta, jossa käyttäjän tviitit näkyvät oletuksena kaikille hänen seuraajilleen. Nuoli kuvaa tviitin näkyvyyden suuntaa. Käyttäjän A kirjoittama tviitti näkyy käyttäjille B-H, sillä he seuraavat käyttäjää A. Jos käyttäjä G tviittaa päivityksen eteenpäin, niin tviitin näkyvyys leviää käyttäjää G seuraaville käyt-

täjille I-L, vaikka he eivät seuraakaan käyttäjää A tai A heitä. Jos käyttäjä G:n seuraama käyttää tviittaa sen eteenpäin, laajenee tviitin näkyvyys edelleen.



KUVIO 1 Tviittien näkyvyys Twitterissä (Sanzgiri ym., 2013)

Hyökkääjä voi levittää haitallisia lyhytosoitteita osoittamalla päivityksen @-merkin avulla toisille käyttäjille. Sanzgirin ym. (2013) mukaan tämä ei vielä takaa huijauksen onnistumista, sillä käyttäjä ei välttämättä avaa päivityksen sisältämää linkkiä. Hyökkääjän ja kohteiden väliset aikaisemmat interaktiot vaikuttavat todennäköisyyteen avata linkki (Sanzgiri ym., 2013). Linkin avaamisen todennäköisyys lisääntyy, jos käyttäjä seuraa hyökkääjää. Jos päivityksen tehnyt käyttäjä on täysin tuntematon käyttäjä, on epätodennäköistä, että päivityksessä mainitut käyttäjät luottavat häneen. (Sanzgiri ym., 2013.). Lisäksi Twitter myös seuraa käyttäjien tekemiä @käyttäjänimi-päivityksiä ja seurattavien käyttäjien lukumäärää. Mikäli käyttäjä seuraa huomattavan monia käyttäjiä ja kirjoittaa paljon muille käyttäjille osoitettuja päivityksiä, voidaan käyttäjä tulkita roskapostin lähettäjäksi. (Sanzgiri ym., 2013.).

Seuraajien tulee tviitata uudelleen haitallinen linkki, jotta se levittyisi tehokkaasti. Puumainen rakenne on eduksi levittämisessä, mutta luottamalla pieneen todennäköisyyteen, että käyttäjä tviittaa tuntemattoman käyttäjän tviitin eteenpäin, hyökkäys voi epäonnistua. Hyökkääjä voi parantaa hyökkäyksen tehokkuutta ja onnistumista hyödyntämällä klikkaushuijausta (Sanzgiri ym., 2013). Jos linkkiin on upotettu klikkaushuijausansa, aina käyttäjän klikatessa linkkiä tviittaa hän sen tahtomattaan ja tietämättään eteenpäin.

2.6 Henkilökohtaisen tiedon kerääminen ja hyödyntäminen

Teknisten menetelmien, muun muassa virusten, lisäksi sosiaalisen median palveluita voidaan hyödyntää myös muuten esimerkiksi keräämällä tietoja käyttäjistä suoraan heidän käyttäjäprofiileistaan. Monet jakavat materiaalia, esimerkiksi kuvia, videoita ja blogi-kirjoituksia, Internetissä. Käyttäjäprofiilin tietoihin on mahdollista lisätä itsestä henkilökohtaista tietoa, muun muassa syntymäpäivä, osoite ja puhelinnumero. (de Paula, 2010.). Yksityisyyden suojaamiseen on erityisen tärkeää kiinnittää huomiota, sillä sosiaalisen median palveluissa on mahdollista etsiä muita käyttäjiä ja ryhmiä, ja saada näin jopa erittäin henkilökohtaista tietoa muista käyttäjistä ja heidän toiminnastaan (de Paula, 2010).

Sosiaalisen median suhteet heijastavat todellisen elämän suhteita, kun on kyse luottamuksesta. Sekä todellisen elämässä että virtuaalielämässä on eritasoista luottamusta riippuen henkilöiden välisestä suhteesta (de Paula, 2010). Myös virtuaalielämässä pitäisi olla mahdollisuus määrittää, kuka kuuluu henkilöihin, joihin voi luottaa ja kuka on vähemmän tuttu ja turvallinen. Sosiaalisen median palveluissa on usein mahdollista määrittää, näkyvätkö käyttäjäprofiilin tiedot vain itselle vai koko maailmalle vai joillekin siltä väliltä. De Paulan (2010) mukaan kuitenkin useat käyttäjät määrittävät profiilinsa näkyvyyden julkiseksi. Nykyisin monissa sosiaalisen median palveluissa on mahdollista ilmoittaa, missä paikassa käyttäjä tällä hetkellä on, ja käyttäjän tekemien päivitysten avulla hänestä voidaan muodostaa tarkkakin kuva. Koska profiilit voivat sisältää arkaluontoista ja henkilökohtaista tietoa käyttäjästä, saattavat tietojen kalastajat ja hakkerit hyödyntää tätä omiin tarkoituksiinsa.

Aina ei kuitenkaan edes yksityisyyden määrittäminen ole tehokas suoja. Facebookissa on mahdollista valita muun muassa ystävälistan näkyvyys kokonaan näkymättömäksi, mutta silti hyökkääjä voi selvittää käyttäjän ystävät tarkastelemalla muun muassa käyttäjän julkaisuja, kuvia, joihin käyttäjän ystävät ovat kommentoineet tai joista he ovat tykänneet (Mahmood, 2012). Pitkällä aikavälillä hyökkääjä saa todennäköisesti tietää suurimman osan käyttäjän suhteista seuraamalla käyttäjän julkaisujen ja kuvien tykkäyksiä ja kommentteja. Facebookin yksityisyysasetuksista ei myöskään ole mahdollista määrittää yhteisten ystävien näkyvyyttä (Mahmood, 2012). Tämä tarkoittaa sitä, että jos käyttäjällä A on yhteisiä ystäviä käyttäjän B kanssa, yhteiset ystävät näkyvät molemmille riippumatta siitä, miten näkyväksi he ovat määrittäneet ystävälistansa näkyvän.

Sosiaalisen median palveluissa käyttäjän olemassaoloa ei voida luotettavasti varmentaa, sillä kaikki mitä profiilin ja käyttäjätunnuksen tekemiseen tarvitaan, on toimiva sähköpostiosoite. Hakkereille ja muille huijareille tämä tarjoaa helpon mahdollisuuden tietojen kalasteluun. He voivat luoda huijausprofiileja matkiakseen oikeita henkilöitä ja yrityksiä. Julkisten profiilien ja sieltä paljastuvien tietojen avulla rikolliset ja terroristit voivat tehdä ”rikollista tiedon louhintaa” (de Paula, 2010).

Sähköpostiosoitteet ovat myös monelle markkinoijalle ja tietojen kalastelijalle hyödyllistä materiaalia. Tietojen kalasteluviestit ovat lähtökohtaisesti tehokkaampia, kun ne on kohdennettu henkilölle hänen omalla nimellään. (Mahmood, 2012.). Sosiaalisen median palveluihin ei ainoastaan kohdenneta hyökkäyksiä, vaan niitä myös hyödynnetään apuvälineenä suunnitellessa ja tehtäessä hyökkäyksiä muualla Internetissä. Hyökkääjä voi hyödyntää muun muassa Facebookia etsiessään sähköposteja vastaavia nimiä (Mahmood, 2012). Tällaista sähköpostiosoitteen ja nimen yhteensovittamista kutsutaan kartoittamiseksi (*engl. mapping*). Kartoittamisessa hyökkääjä kirjoittaa Facebookin etusivulla olevaan kirjautumisikkunaan haluamansa sähköpostiosoitteen ja klikkaa Kirjaudu sisään –painiketta. Mikäli tietokannasta löytyy käyttäjä kyseisellä sähköpostiosoitteella, tulee sivulle ilmoitus, jossa näkyy käyttäjän nimi ja hänen profiilikuvansa. Mahmood, 2012.). Näin ollen yksityisyysasetukset eivät vaikuta siihen, onko henkilön nimi löydettävissä kartoittamisen avulla.

De Paulan (2010) mukaan sosiaalisen median palveluihin liittyviin riskeihin sisältyy muun muassa käyttäjän tietosuojaoikeuksien vahingoittaminen, identiteettihuijaukset ja -varkaudet, kunnianloukkaus, vakoilu ja kiusaaminen. Identiteettivarkaus ei ole uusi huijausmuoto, vaan se on ollut olemassa eri muodoissa aikaisemminkin (de Paula, 2010). Hyökkääjät voivat helposti siirtyä uuteen käyttäjätiliin, mikä tekee identiteettivarkaudesta entistä helpompaa. Hyökkääjät yrittävät saada käyttäjän tuntemaan olonsa turvalliseksi ja luottamaan heihin. Luottamussuhde toisensa tuntevien käyttäjien välillä parantaa hyökkääjän mahdollisuuksia, sillä saavutettuaan esimerkiksi yhden ryhmän jäsenen luottamuksen, kasvaa riski joutua hyökkäyksen kohteeksi kaikilla niillä, joilla on suhde tähän ensimmäiseen käyttäjään. (de Paula, 2010.).

2.7 Kolmannet osapuolet

Monissa sosiaalisen median palveluissa on mahdollisuus käyttää sovelluksia ja pelata pelejä, jotka eivät ole palvelun omia, vaan kolmannen osapuolen tarjoamia. Nämä kolmannet osapuolet muodostavat tietojen kalastelun ja turvallisuuteen liittyen uhan, sillä sovelluksen tai pelin pelaamisen vastineena ne keräävät käyttäjästä tietoa, joka hänestä on julkisesti saatavilla.

Kolmansien osapuolien sovellukset käyttävät omia palvelimiaan. Kun esimerkiksi Facebook vastaanottaa käyttäjän pyynnön kommunikoida kolmannen osapuolen sovelluksen kanssa, siirtyy pyyntö edelleen sovelluksen tarjoavalle palvelimelle. Sovellus lähettää takaisin vaatimuksen nähdä ja käyttää tietoa käyttäjästä. (Ahmadinejad, Anwar & Fong, 2011.).

Facebook oli ensimmäisiä palveluita, joka salli kolmansien osapuolien sovellukset. Sen jälkeen myös monet muut palvelut ovat ottaneet sovellukset sallivan alustan käyttöön. (Ahmadinejad ym., 2011.). Sovellusten sallimiseen liittyy kuitenkin merkittäviä turvallisuusuhkia, sillä sovellusten tarjoajat ovat oma itsenäinen osansa, eivätkä kuulu esimerkiksi facebook.com–verkkotunnuksen

alle. Täten sovellusten luotettavuutta ei voida ehdottoman varmasti varmentaa. (Ahmadinejad ym., 2011.).

Kolmannet osapuolet eivät vain hyödynnä käyttäjän profiilissa saatavilla olevia tietoja, vaan ne keräävät tietoa myös käyttäjän liikkeistä muualla Internetissä, sillä monilla sivuilla käyttäjän on mahdollista kommentoida tai tykätä oman Facebook-tilinsä nimissä. Käyttäjän profiilin yksityisyysasetukset eivät vaikuta siihen, miten näkyvää hänen toimintansa muualla Internetissä on. Kolmansien osapuolten keräämien tietojen avulla käyttäjästä voidaan muodostaa kuva, joka kuvaa hänen mielenkiinnonkohteitaan ja toimintaansa. Ohjelmilla, joilla voidaan muodostaa sosiaalista toimintaa kuvaavia kuvioita, käyttäjästä voi paljastua paljon henkilökohtaista tietoa, jota voidaan hyödyntää. (Erlandsson ym., 2012.).

3 SUOJAUTUMINEN TIETOJEN KALASTELUA VASTAAN

Tässä luvussa käydään läpi, miten tietojen kalastelua vastaan voidaan suojautua. Osa keinoista toimii useampaa kuin yhtä hyökkäystyyppiä vastaan. Jotkut hyökkäysmenetelmät ovat myös sellaisia, ettei niihin ole olemassa teknistä suojauskeinoa, vaan suojaus vaatii käyttäjän toimintaa ja halukkuutta suojata yksityisyytensä.

3.1 Suojaus Cross Site Scripting –haavoittuvuutta vastaan

Turvallisuusasiantuntijoiden mukaan XSS-haavoittuvuus on yksi vakavimmista ja yleisimmistä web-sovelluksiin kohdistuvista uhista (Sun & He, 2012; Shar & Tan, 2012). Käyttäjiä voidaan suojata estämällä heidän pääsynsä haitalliseksi epäillylle sivustolle. (Sun & He, 2012.). Arulsujun (2011) mukaan tehokas keino suojautua XSS-haavoittuvuuksia vastaan on estää komentosarjakielien tukeminen selaimessa. Ongelmana kuitenkin on, että useimmat käyttäjät eivät halua estää tätä. Suojaustyökalut myös vaativat käyttäjää pitämään huolta selaimensa päivitysten ajantasaisuudesta, mikä tuo käyttäjälle ylimääräistä työtä. (Arulsuju, 2011.).

Sharin ja Tanin (2012) mukaan XSS-haavoittuvuuksia vastaan kehitetyt puolustusmenetelmät voidaan jakaa neljään luokkaan. Ne ovat puolustava ohjelmointi (*engl. defensive coding*), XSS-testaus (*engl. XSS testing*), haavoittuvuuksien havaitseminen (*engl. vulnerability detection*) ja ajonaikaisen hyökkäyksen estäminen (*engl. runtime attack prevention*).

XSS-haavoittuvuuksia hyödyntävät hyökkäykset perustuvat syötteiden heikkoon käsittelyyn, joten puolustavalla ohjelmoinnilla voidaan vahvistaa, että käyttäjän syöte vastaa vaadittua formaattia (Shar & Tan, 2012). Korvaavat ja poistavat menetelmät (*engl. replacement and removal methods*) puhdistavat koodia etsimällä haitallisia merkkejä (*engl. character*) mustien listojen (*engl. black lists*) perusteella. Korvaavat menetelmät lisäävät haitallisten merkkien tilalle vaarat-

tomia merkkejä ja poistavat menetelmät nimensä mukaisesti poistavat koodista haitallisiksi epäillyt merkit. (Shar & Tan, 2012.). Mustiin listoihin perustuva haitallisten merkkien etsintä kuitenkin epäonnistuu usein, sillä on hankalaa pitää yllä ajantasaista listaa kaikkien hyökkäysten piirteistä ja niiden muunnelmista (Shar & Tan, 2012).

Syötteen vahvistamisen testauksella (*engl. input validation testing*) voidaan havaita XSS-haavoittuvuuksia. Sharin ja Tanin (2012) mukaan määritysperusteiset (*engl. specification-based*) syötteen vahvistustestit luovat testejä, joiden päämääränä on tehdä yhdistelmiä hyväksyttyjen ja kelpaamattomien syötteiden ehdoista. Koodiperusteinen (*engl. code-based*) syötteen vahvistustestaus etsii hyväksyttyjen ja kelpaamattomien syötteiden ehtoja palvelinpuolen komentosarjoista (Shar & Tan, 2012).

Haavoittuvuuksia voidaan myös tunnistaa palvelinpuolen komentosarjoissa. Staattinen analyysi (*engl. static analysis*) havaitsee saastuneet syötteet ulkoisesta datalähteestä, ja jäljittää saastuneen datan. Analyysiin perustuvat menetelmät havaitsevat nopeasti XSS-haavoittuvuudet lähdekoodissa, mutta niiden heikkoutena on oletus, että käsittelemättömät ja tuntemattomat toiminnot palauttavat haitallista dataa. Staattisella merkkijonoanalyysilla (*engl. static string analysis*) voidaan analysoida merkkijono-operaatioiden vaikutuksia syötteeseen. Wassermannin ja Sun tekniikka hyödyntää kontekstiriippumatonta kielioppia (*engl. context-free grammar*) mallintaakseen arvoja, joita merkkijonomuuttuja voi pitää sisällään. (Shar & Tan, 2012.).

Ajonaikaiset hyökkäykset voivat keskittyä joko palvelin- tai käyttäjäpuolelle. Suojauskeinot voivat muun muassa valvoa syötteiden datan virtaa ajon aikana ja varmistaa, etteivät syötteet sisällä vaarallista sisältöä. (Shar & Tan, 2012.). Toisaalta voidaan myös valvoa luotettavaa sisältöä ja merkkijonoja, ja tämän perusteella suojautua palvelinpuolen XSS-haavoittuvuuksia vastaan. Osa palvelinpuolen suojauskeinoista toimii yhdessä selaimen kanssa. Esimerkiksi BEEP (Browser-Enforced Embedded Policies) on suojaustyökalu, joka muuntaa selainta niin, ettei se voi suorittaa haitallisia komentosarjoja. (Shar & Tan, 2012.).

Noxes on käyttäjäpuolelle tehty työkalu, joka toimii eräänlaisena palomuurina (Shar & Tan, 2012). Sen toiminta perustuu URL-osoitteista laadittuihin valkoisiin ja mustiin listoihin, joiden perusteella se joko sallii tai estää yhteyden sivustolle. Selaimen lähettäessä HTTP-pyynnön tuntemattomalle sivustolle, Noxes varoittaa käyttäjää. (Shar & Tan, 2012.). Noxes ei kuitenkaan voi estää sivustolle pääsemistä, sillä käyttäjällä on edelleen mahdollisuus varoituksesta huolimatta avata sivusto.

3.2 Klikkaushuijaukselta suojautuminen

Ensiaskel klikkaushuijausta vastaan suojautumisessa on estää omien sivujen kehystäminen. Kehystämisellä tarkoitetaan huijaussivun asettamista päällekkäin oikean sivun kanssa (Lundeen & Alves-Foss, 2011), mikä on klikkaushuijauksen ominaispiirre. JavaScriptillä koodiin voidaan upottaa eräänlainen ke-

hyksen paljastaja (*engl. frame-busting*). JavaScriptissä on useita keinoja kirjoittaa paljasta koodiin ja alapuolella on esitelty Lundeen ja Alves-Fossin artikkelissa ilmennyt koodi:

```
if (top != self) {

        top.location = self.location;

}
```

Kehyksen paljastaja on koodi, joka pyytää selainta tarkastamaan, onko www-sivu ylimmän tason sivu. Mikäli sivu ei ole ylimmällä tasolla, tulee ilmoitus esimerkiksi virheestä. Tämä tarkoittaa, että sivua on mahdollisesti käytetty klikkaushuijauksessa. (Lundeen & Alves-Foss, 2011.).

Useiden selainten uusimmat versiot, muun muassa Safari, FireFox, Internet Explorer ja Chrome tukevat nykyään ainakin osittain X-Frame-Optionsia, jota käytetään todennusta vaativilla sivuilla (Lundeen & Alves-Foss, 2011). X-Frame-Options on HTTP-vastaus, jonka avulla voidaan estää sivun kehystäminen joko kokonaan tai osittain tai kehystäminen voidaan sallia vain tietyille sivuille. Kehystämisen estäminen kokonaan estää kaiken kehystämisen, mutta osittaisen kehystämisen mahdollistava ratkaisu sallii kehystämisen, mikäli kehystämistä yrittävä sivu on samaa alkuperää. (Lundeen & Alves-Foss, 2011.). Tällöin siis estetään ulkoisten sivujen mahdollisuus kehystämiseen sivulla. Esimerkiksi www.esimerkki.fi –sivusto ei voi kehystää www.esim.fi –sivustoa, mutta www.esim.fi/esimerkki -sivu voi.

Vaikka X-Frame-Options on Lundeenin ja Alves-Fossin (2011) mukaan luotettava ratkaisu estää kehystämistä ja siten klikkaushuijausta, sisältyy siihen myös negatiivisia puolia. Vaikka useat selaimet tukevatkin sitä, se ei ole käytössä kaikissa selaimissa täydellisenä. Osa selaimista ei nimittäin tue sama alkuperä –ominaisuutta. Tällä hetkellä täysin X-Frame-Optionsia tukevat Internet Explorer versiosta 8 alkaen ja FireFox alkaen versiosta 18. (http://erlend.oftedal.no/blog/tools/xframeoptions/; https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options#Browser_compatibility.).

Kehystämisen kieltämiseen liittyy myös ongelma, sillä joillain sivuilla ominaisuuksien rakentamisessa on käytetty kehyselementtiä (Lundeen & Alves-foss, 2011). Tähän perustuu esimerkiksi Facebookin tykkää-painike. Koska sen toteutuksessa on käytetty kehystä, kehystämisen estäminen estäisi samalla painikkeen toimimisen.

Käyttäjä voi saada viitteitä klikkaushuijatusta sivustosta URL-osoitteen kautta (Callegati & Ramilli, 2009). Koska oikean sivun painikkeet on aseteltu hyökkääjän sivun painikkeita ja siten linkkejä vastaaviksi, voidaan hyödyntää useimmissa selaimissa olevaa ominaisuutta, joka näyttää URL-osoitteen taustalla, jos linkkiä osoitetaan hiirellä. Hyökkääjän linkki voi näyttää erehdyttävästi samanlaiselta kuin luotetun sivuston URL-osoite. Erona voi olla muun muassa erikoiset merkit tai sanat, jotka voivat paljastaa hyökkäyksen (Callegati & Ramilli, 2009). Esimerkiksi luotetun ja aidon sivun URL-osoite on

www.organisaatio.fi, mutta jos organisaation sivulla osoittaa kohtaa, jonka URL-osoite on www.organi€$saatio$£@.com, voi olettaa sivulle johtavan linkin olevan epäluotettava. Käyttäjä voi siten omalla toiminnallaan ehkäistä klikkaushuijaukseen lankeamista, mutta ongelmana on menetelmän hitaus. Sivuilla on usein paljon klikattavia kohtia, ja kaikkien läpikäyminen vie aikaa. Lisäksi useimmat ihmiset eivät ole tietoisia siitä, mitä URL-osoite paljastaa sivusta (Callegati & Ramilli, 2009).

Callegati ja Ramilli (2009) ehdottavat ei-graafisten selainten käyttöä silloin, kun käyttäjä selaa sivuja, jotka eivät ole täysin luotettavia. Ei-graafiset selaimet eivät tue sivujen graafisia tasoja, mikä suojaa klikkaushuijaukselta. Mikäli kuitenkin käytetään tavallista graafista selainta, voi erilaisilla selainten lisäosilla valvoa ja kontrolloida komentosarjojen eli skriptien toimintaa. Mozilla Firefox –selaimeen on saatavilla NoScript –lisäosa, joka estää upotetun sisällön näyttämisen (Callegati & Ramilli, 2009). Lisäosan haittapuolena on, että se estää samalla kaikki skriptit riippumatta niiden mahdollisesta vaarasta. Tämä tekee sivujen selaamisesta vaikeaa ja epämiellyttävää. (Callegti & Ramilli, 2009.).

Klikkaushuijauksen ehkäisemiseen on olemassa myös sovelluksia. Websense on yhdistetyn web-, data- ja sähköpostisisällön turvallisuuteen keskittynyt yritys, jonka kanssa Facebook tekee yhteistyötä ehkäistäkseen ja suojatakseen käyttäjiä klikkaushuijaukselta (Jagnere, 2012). Kun käyttäjä klikkaa Facebookissa julkaistua URL-osoitetta, lähetetään linkki Websenselle. Websense luokittelee osoitteen turvallisuuden ja lähettää sen ThreatSeeker Cloud -alustalle, joka on kehitetty tunnistamaan, luokittelemaan ja analysoimaan haittaohjelmia. (Jagnere, 2012.). Mikäli URL-osoitteen epäillään olevan epäluotettava, tulee käyttäjälle ilmoitus tästä. Websense ei pysty estämään haitallisten linkkien julkaisemista Facebookissa, vaan se voi ainoastaan varoittaa haitallisesta ja turvattomasta sivusta. (Jagnere, 2012). Käyttäjälle jää edelleen vastuu siitä, haluaako hän jatkaa sivustolle Websensen varoituksesta huolimatta.

3.3 Suojaus ystävä välissä –hyökkäystä vastaan

Huberin ym. (2011) mukaan tehokas suoja ystävä välissä –hyökkäystä vastaan vaatii HTTPS-protokollan käytön kaikessa tiedonsiirrossa käyttäjän ja palvelun tarjoajan välillä. Aikaisemmin oli ongelmana se, että vain harvat sosiaalisen median palvelujen tarjoajat käyttivät HTTPS-protokollaa, ja suurin osa palvelun tarjoajista käytti salaamattomaan tiedonsiirtoon tarkoitettua HTTP-protokollaa. Selaimiin oli olemassa lisäosia, joiden avulla tieto pakotettiin siirtymään HTTPS-protokollan yli, vaikka normaalisti olisi käytetty suojaamatonta HTTP-protokollaa. (Huber ym., 2011.). Esimerkiksi Facebook ja Twitter tarjosivat HTTPS-protokollan käyttömahdollisuuden vasta vuonna 2011. Nykyisin kuitenkin salatun tiedonsiirtoprotokollan käyttö on enemmänkin sääntö kuin poikkeus.

Selaimiin on olemassa myös lisäosia ja sovelluksia, jotka vähentävät hyökkääjän mahdollisuuksia louhia tietoa ystävä välissä -hyökkäystä varten käyttä-

jän profiilista rajoittamalla profiilissa olevan tiedon määrää. Face-Cloakin avulla käyttäjän henkilökohtaiset ja arkaluontoiset tiedot siirretään salattuna erilliselle palvelimelle ja alkuperäiset tiedot korvataan väärillä tiedoilla (Huber ym., 2011). Mikäli hyökkääjä yrittää hyödyntää käyttäjän ystävän profiilin tietoja esiintyäkseen uskottavasti, saa hän tietoonsa ainoastaan väärät tiedot ja paljastuu näin huijariksi.

Sovellusten avulla voidaan myös suojata käyttäjien keskinäinen kommunikointi sosiaalisen median palvelussa. FlyByNight-sovellus on tarkoitettu käytettäväksi Facebookissa salaamaan käyttäjien väliset viestit (Huber ym. 2011). Mikäli hyökkääjä saa viestin haltuunsa, hän ei salauksen takia kykene avaamaan sitä.

Vaikka ystävä välissä –hyökkäystä vastaan on olemassa suojausmenetelmiä, ongelmana on Huberin ym. (2011) mukaan se, että edellä mainitut lisäosat ja sovellukset suojaavat ainoastaan tekstisisältöä, kuten viestejä ja profiilitietoja. Suojauksen ulottumattomiin jää muun muassa kuva- ja videomateriaali, jota voidaan niin ikään hyödyntää hyökkäyksissä. Huber ym. (2011) esitettävät ongelmaan ratkaisuna sosiaalisen median palvelun, joka lähtökohtaisesti on suunniteltu yksityisyyden ja turvallisuuden kannalta. Turvallisuuslähtöinen suunnittelu edistää käyttäjien yksityisyyttä ja vähentää hyökkäysten mahdollisuuksia, sillä uhat on huomioitu jo suunnitteluvaiheessa.

3.4 Deaktivoitu profiili ja Markov Cluster –näkökulma

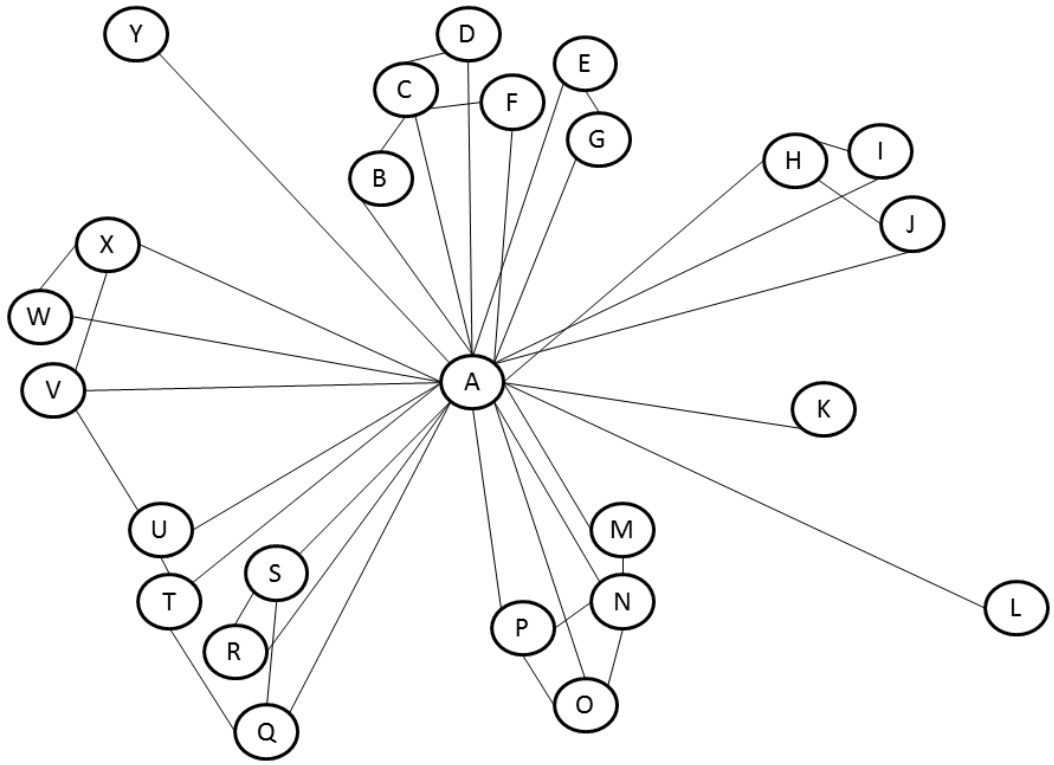
Deaktivoidun profiilin avulla tehtyä tietojen kalastelua vastaan ei ole olemassa varsinaisesti teknistä suojausmenetelmää. Sen sijaan käyttäjä voi omalla toiminnallaan yrittää vaikuttaa siihen, ettei hänen ystävälistallaan ole käyttäjiä, jotka hyödyntävät deaktivoitua profiilia. Oman profiilin ystävälistan säännöllisellä tarkastamisella voi huomata, jos joku vaihtelee jatkuvasti aktivoitu- ja deaktivoitu-tilojen välillä (Mahmood & Desmedt, 2012).

Riskiä joutua deaktivointihyökkäyksen kohteeksi voi vähentää pitämällä ystävälistansa mahdollisimman siistinä ja poistamalla epäilyttävien tai tuntemattomien käyttäjien profiilit. Mahmoodin ja Desmedt'n (2012) artikkelin perusteella voi kuitenkin päätellä, etteivät ihmiset juurikaan ole kiinnostuneita siitä, ovatko heidän ystävälistallaan olevat käyttäjät tuttuja vai ei. Tämä edesauttaa hyökkääjän mahdollisuuksia kalastella tietoa kohteistaan.

Muut sosiaalisen median palvelut, muun muassa Twitter ja Google+, eivät tarjoa mahdollisuutta aktivoida profiilia edestakaisin (Mahmood & Desmedt, 2012). Jos käyttäjä deaktivoi profiilin, hän ei pysty aktivoimaan sitä uudelleen. Näissä palveluissa deaktivointihyökkäystä ei siten pysty hyödyntämään. Facebookissa käyttäjä voi ilmoittaa oudoista ja epäilyttävistä profiileista. Facebook valvoo ilmoitusten perusteella näitä profiileita ja voi tarpeen tullen estää profiilin käyttäjän pääsyn Facebookiin (Mahmood & Desmedt, 2012).

Ahmedin ja Abulaishin (2012) artikkeli tarjoaa toisenlaisen näkökulman epäilyttäviä profiileita vastaan. Markov Cluster eli MCL on algoritmi, jonka

avulla voidaan tunnistaa roskapostiprofiilit (*engl. spam profile*) aitojen profiilien joukosta (Ahmed & Abulaish, 2012). Se ei siis itsessään ole suojauskeino tietojen kalasteluun, eikä se kykene torjumaan deaktivointihyökkäystä, mutta se voi antaa viitteitä mahdollisesta yrityksestä. Käyttäjän sosiaalisista interaktioista, joita ovat muun muassa käyttäjän ystävät, tykätyt ja suositellut sivustot, voidaan laatia painotettu graafi, jossa pallot kuvaavat käyttäjiä ja kaaret pallojen välillä ovat interaktioita (Ahmed & Abulaish, 2012). Kuviossa 2 havainnollistetaan Ahmedin ja Abulaishin (2012) artikkelin kuvausta graafista. Profiililla A on interaktioita jokaisen profiilin B-Y kanssa, mutta muut profiilit eivät välttämättä ole keskenään vuorovaikutuksessa.



KUVIO 2 Sosiaaliset interaktiot

Painotetusta graafista saadaan MCL:n avulla laskettua ja muodostettua eri kategorioittain lajiteltua ryhmiä, klustereita. Ryhmät lajitellaan sen perusteella, ovatko ryhmän profiilit roskapostiprofiileja vai aitoja profiileja vai sisältääkö ryhmä molempia. (Ahmed & Abulaish, 2012.).

Normaalit käyttäjät ovat tavallisesti vuorovaikutuksessa vain pienen joukon kanssa ja usein näillä käyttäjillä on samoja kiinnostuksen kohteita. Huijausja roskapostiprofiileiden käyttäjillä sen sijaan on paljon ystäviä, mutta vuorovaikutus heidän kanssaan on useimmiten yksisuuntaista. (Ahmed & Abulaish.). Tietojen kalastelijat saattavat käyttää useita olemattomilla identiteeteillä luotuja huijausprofiileita tehostaakseen huijausta. Myös nämä profiilit voivat olla vuorovaikutuksessa keskenään joko suoraan siten, että huijausprofiili A on hui-

jausprofiili B:n ystävälistalla tai epäsuorasti siten, että käyttäjä C on molempien ystävälistalla, mutta huijausprofiilit eivät ole toistensa ystävälistalla. (Ahmed & Abulaish, 2012.).

Ahmedin ja Abulaishin (2012) mukaan huijausprofiilin voi tunnistaa myös jaettujen linkkien määrän perusteella. Huijausprofiilin ominaispiirre on, että se jakaa esimerkiksi mainoksia, henkilökohtaisia blogeja ja muiden www-sivujen osoitteita useasti yhdellä sivustolla. Normaali käyttäjä jakaa monipuolisemmin materiaalia, muun muassa linkkejä Youtube-videoihin ja uutisiin, eikä hän jaa samaa URL-osoitetta yhdellä sivulla toistuvasti. (Ahmed & Abulaish, 2012.).

3.5 Tviittien analysointi

Aggarwalin ym. (2012) mukaan suuri osa saastuneista URL-osoitteista lyhennetään lähinnä todellisen osoitteen alkuperän peittämiseksi, eikä esimerkiksi tilan säästämiseksi. Tilastot myös osoittavat, että jopa 8 % kaikista tviiteistä sisältää roskapostia ja muuta haitallista materiaalia (Aggarwal ym., 2012). Haitallista materiaalia sisältävien tviittien määrä on valtava. Vuonna 2012 julkaistun Aggarwalin ym. artikkelin mukaan käyttäjät julkaisevat noin 200 miljoonaa tviittiä joka päivä. Nykyisestä tviittien julkaisumäärästä on hankala saada tieteellistä tietoa, mutta DMR:n (Digital Marketing Rambling) (5.2.2014) mukaan joka päivä julkaistaan keskimäärin puoli miljardia tviittiä.

Twitter varoittaa etukäteen, jos se epäilee tviitin sisältämän linkin olevan vahingollinen. Mikäli käyttäjä yrittää klikata haitalliseksi epäiltyä linkkiä, Twitter näyttää varoitussivun. Aggarwalin ym. (2012) mukaan Twitterin puolustautumismekanismi ei ole nopein mahdollinen, eikä se kykene reaaliaikaisesti huomaamaan kaikkia haitallisia tviittejä.

Haitalliset tviitit on muotoiltu usein niin, ettei käyttäjä epäile mitään, vaan avaa tviitin sisältämän linkin. Haitallinen tviitti lupaa tavallisesti tarjouksia ja etuja tai se herättää muuten huomiota. Yleisimpiä haitallisissa tviiteissä käytettyjä sanoja ovat muun muassa "product" (*suom. tuote*), "allow" (*suom. sallia*), "story" (*suom. tarina*) ja "friends" (*suom. ystävät*). (Aggarwal ym., 2012.). Haitallisen tviitin voi tunnistaa tviitin aiheesta ja sanoista, sillä tavalliset käyttäjät tviittaavat usein yleisistä aiheista ja sanasto on monipuolista. Sen sijaan haitallisissa tviiteissä toistuvat samat sanat eikä niiden tekstisisältö ole yhtä monipuolinen kuin tavallisissa tviiteissä. (Aggarwal ym., 2012.).

PhishAri on automaattinen ja reaaliaikainen suojautumiskeino, joka suojaa haitallisilta roskapostitviiteiltä. Menetelmä perustuu muun muassa tviittien sisältöön ja epäilyttäviin URL-osoitteisiin, ja se kerää tietoa myös niistä verkkotunnuksista, jotka ovat osoittautuneet tietojen kalastelusivustoiksi (Aggarwal ym., 2012). URL-osoitteissa epäilyttäviä piirteitä ovat esimerkiksi osoitteen huomattava pituus ja sen sisältämät useat pisteet. Lisäksi tarkastetaan verkkotunnuksen tarjoaja, sekä aika joka kuluu verkkotunnuksen ja Twitter-tilin luomisen välillä. Muita olennaisia piirteitä ovat käyttäjän seuraamien ja häntä seu-

raavien käyttäjien lukumäärä, tilin ikä ja tviiteissä olevien #- ja @-merkkien määrä. (Aggarwal ym., 2012.).

PhishArista on kehitetty Googlen Chrome-selaimessa toimiva lisäosa. Tviittien ja lisäosan välillä on PhishAri API, joka toimii rajapintana. Lisäosa lähettää rajapinnalle tviitin ID:n, ja API analysoi sen kautta kulkevat tviitit edellä mainittujen piirteiden pohjalta ja luokittelee ne joko turvallisiksi tai tietojen kalasteluksi. (Aggarwal ym., 2012.).

3.6 Henkilökohtainen tieto ja kolmannet osapuolet

Facebook-käyttäjä ei voi joiltakin osin vaikuttaa tietojen kalasteluun johtuen Facebookin ominaisuuksista ja suunnittelusta. Nimen ja sähköpostiosoitteen yhdistämisen eli kartoittamisen estämiseksi Facebookin ei tulisi tarjota mahdollisuutta nähdä sähköpostiosoitetta vastaavaa käyttäjän nimeä (Mahmood, 2012). Koska on erittäin epätodennäköistä, että käyttäjä unohtaa oman nimensä, Mahmood (2012) ehdottaa ratkaisuna, ettei Facebook näytäkään sähköpostia vastaavaa nimeä. Sen sijaan käyttäjän tulee itse kirjoittaa nimi, joka vastaa sähköpostiosoitetta. Tämä menetelmä ehkäisee nimien kartoittamista roskapostiviestejä varten. (Mahmood, 2012.).

Myös käyttäjien yhteisten ystävien näkyvyys aiheuttaa ongelman, sillä näkyvyyttä ei voi yksityisyysasetuksista huolimatta estää. Tähän ratkaisuna olisi se, että jos käyttäjä on rajoittanut ystävälistan näkymisen ainoastaan itselleen, niin edes yhteisiä ystäviä ei voisi nähdä (Mahmood, 2012). Koska käyttäjän ystävät voidaan pitkällä aikavälillä saada selville myös julkaisuista, kommenteista ja kuvista, tulisi niiden lähettäjät näyttää anonyymina kaikille niille, jotka eivät ole esimerkiksi kommentin kirjoittaneen käyttäjän ystäviä. Toisaalta tämä ratkaisu vaikeuttaa muun muassa kommenttiketjun seuraamista, mutta vastapainona käyttäjä voisi sallia luotetuille ystävilleen kommentit ja kuvat julkaisseiden käyttäjien näkymisen. (Mahmood, 2012.). Myös Nagy ja Pecho (2009) ehdottavat, että käyttäjä voisi jakaa ystävänsä eri luottamustasoille. Oletuksena jokainen ystävä on alimmalla tasolla ja heidät voi siirtää eri tasoille sen perusteella, mitä käyttäjä haluaa heidän näkevän (Nagy & Pecho, 2009).

Nykyisten sosiaalisen median palveluiden ongelmana on, että huijarit ja tietojen kalastelijat ovat jo sisällä palveluissa. Ratkaisuna voisi olla uusi sosiaalisen median palvelu, johon ei olisi mahdollista rekisteröityä vapaasti, vaan rekisteröityminen vaatisi kutsun käyttäjältä, joka jo käyttää palvelua (Nagy & Pecho, 2009). Jo rekisteröityneet käyttäjät lähettäisivät kutsuja vain niille, jotka he tuntevat ja jotka ovat luotettavia. Toisaalta tämä malli syrjii niitä henkilöitä, joilla ei ole yhtään ystävää rekisteröityneenä palveluun. (Nagy & Pecho, 2009.). Ongelmana on lisäksi se, ettei menetelmä täysin varmasti estä epäluotettavien henkilöiden pääsyä palveluun. Jo yksikin kutsun saanut hyökkääjä riittää siihen, ettei palvelu ole enää täysin turvallinen.

Sosiaalisen median palveluissa on olennaista varmistaa toisen käyttäjän henkilöllisyys, ennen kuin hänet lisää ystäväkseen tai hyväksyy hänen lähettä-

mänsä ystäväpyynnön. Lisäksi tulee välttää liian henkilökohtaisen ja arkaluontoisen tiedon paljastamista. (Tsai, Chang, Chung & Li, 2010.).

Huber, Mulazzani ja Weippl (2010) ehdottavat suojauskeinoksi henkilökohtaisen tiedon pitämistä kolmannen osapuolen palvelimella ja käyttäjän profiilissa näkyisi vain väärennettyä tietoa. Mikäli hyökkääjä yrittää kalastella käyttäjän henkilökohtaista tietoa, saa hän tietoonsa vain arvotonta informaatiota. Kolmansien osapuolten käyttö tiedon säilytysvarastona on kuitenkin ristiriitaista, sillä niiden luotettavuudesta ei voi olla täysin varma.

Kolmansia osapuolia vastaan voi käyttää osittain samoja menetelmiä kuin ystävä välissä –hyökkäystä vastaan. Sovellusten avulla voi salata käyttäjien väliset viestit. FlyByNight-sovellus käyttää käyttäjäpuolen salausta ja julkaisee salatut viestit omalla palvelimellaan. Viestin vastaanottaja hakee viestit ja purkaa salauksen lokaalisti. (Huber ym., 2010.). Näin ollen ulkopuolinen taho ei pysty purkamaan salausta. Tähän liittyy edelleen ongelma kolmannen osapuolen luotettavuudesta. Vaikka viesti pysyy vain käyttäjän ja FlyByNight-sovelluksen välisenä, on se silti käyttäjän näkökulmasta ulkopuolinen.

4 YHTEENVETO

Sosiaalisen median palveluita on hyödynnetty ja tullaan edelleen hyödyntämään laajasti tietojen kalastelussa. Kalastelukeinoja on useita, ja ne toimivat aivan yhtä hyvin myös muilla kuin vain sosiaalisen median palveluiden sivustoilla. Toisaalta on myös menetelmiä, jotka periaatteessa on kehitetty vain tiettyä palvelua vastaan. Osa keinoista vaatii teknistä osaamista, mutta on myös olemassa keinoja, jotka perustuvat manipulointiin ja ihmisen luontaisen käyttäytymisen hyödyntämiseen. Teknisiä tietojen kalastelukeinoja ovat muun muassa Cross Site Scripting- eli XSS-haavoittuvuus, joka mahdollistaa haitallisen koodin syöttämisen, ja klikkaushuijaus, jossa hyödynnetään päällekkäin asetettuja sivuja. Tietoa voidaan kalastella myös manipuloimalla käyttäjää ja uskottelemalla, että käyttäjä tuntee hyökkääjän. Kuitenkin monissa hyökkäysmenetelmissä on nähtävissä yhteisiä piirteitä ja yksi hyökkäyskeino voi yhdistää monen eri hyökkäystyypin ominaisuuksia. Siten voi olla hankala määrittää tietyn hyökkäyksen kuuluvan vain yhteen kategoriaan.

Ennen tietoja kalasteltiin hyödyntäen sähköposteja. Niin sanotut nigerialaiskirjeet ovat varmasti monelle tuttuja. Kirjeiden ideana on, että huijarit yrittävät uskotella, että käyttäjä on voittanut huomattavan summan rahaa tai tarjoavat korkeatuottoista sijoitusta, mutta saadakseen rahaa käyttäjän tulee maksaa erilaisia käsittelykuluja ja paljastaa henkilökohtaisia tietoja. Lupauksista huolimatta käyttäjä ei kuitenkaan koskaan saa rahoja. Sama ajatus on nähtävissä myös nykyisessä tietojen kalastelussa, vaikka ympäristö ja menetelmät ovat muuttuneet. Taustalla on edelleen tavoite saada tietoa, jota voidaan käyttää rikollisin keinoin taloudellisten päämäärien saavuttamiseksi tai vahingon aiheuttamiseksi.

Tietojen kalastelua vastaan on kehitetty myös suojauskeinoja. Osa artikkeleissa esitellyistä suojausmenetelmistä on kehitetty varta vasten tiettyyn sosiaalisen median palveluun, ja osa toimii yleisesti riippumatta siitä, kohdistuuko hyökkäys sosiaalisen median palveluun vai ei. PhishAri on haitallisilta tviiteiltä suojaava menetelmä ja on kehitetty Twitterissä ilmenevää tietojen kalastelua vastaan. Profiilin deaktivointi –hyökkäys tehoaa etenkin Facebookissa, joten tätä vastaan tulee kehittää oma suojausmenetelmänsä. Osa artikkeleissa esiin-

tyvistä suojauskeinoista on kuitenkin ”teoreettisia”, ja tavallisen käyttäjän voi olla vaikeaa hyödyntää näitä. Tällainen on muun muassa roskapostiprofiileita paljastava MCL-algoritmi, joka vaatii analysointia.

Sosiaalisen median palveluissa ihmiset eivät välttämättä ajattele yksityisyyttä ja tietojen kalastelun mahdollisuutta yhtä paljon kuin fyysisessä elämässä. Jos tuntematon henkilö tulisi kadulla kysymään puhelinnumeroa, osoitetta, koko nimeä tai muuta henkilökohtaista tietoa, saattaisimme olla epäileväisiä ja olla paljastamatta tietoja. Sosiaalisen median palveluissa sen sijaan käyttäjä paljastaa paljon tietoa itsestään ja suhteistaan. Yksityisyysasetuksista riippuen nämä tiedot voivat näkyä jopa koko maailmalle, ja yhdistelemällä näitä tietoja tietojen kalastelija voi saada hyvinkin tarkan kuvan käyttäjästä ja tämän elämästä. Internetissä ja etenkin sosiaalisessa mediassa ihminen ei ehkä osaa ajatella yksityisyyttä niin tärkeänä. Jotkut käyttäjät hyväksyvät ystäväkseen jopa täysin tuntemattomia käyttäjiä. Osaltaan kyse voi olla statuksen kohottamisesta: ”Olen suosittu, kun minulla on näin paljon Facebook-ystäviä”. Käyttäjä ei tule ajatelleeksi, että joku hänen ystävistään voikin olla vihollinen.

Käyttäjät voivat omalla toiminnallaan vaikuttaa siihen, mitä tietoja he paljastavat sosiaalisessa mediassa ja kenelle. Kukaan ei luultavasti pidä siitä, että fyysisessä elämässä joku vakoilee ja etsii tietoa, joten miksi vakoilulle, tietojen kalastelulle ja rikollisuudelle annetaan mahdollisuus Internetissä. Sosiaalisessa mediassa jos missä tulee olla varovainen. Kerran julkaistua tietoa ei välttämättä koskaan saa pois.

Tämä kandidaatintutkielma käsittelee tietojen kalastelua suurimmaksi osaksi Facebookissa ja Twitterissä johtuen lähteinä käytettyjen artikkeleiden näkökulmista. Osa tietojen kalastelukeinoista ja suojausmenetelmistä toimii myös muualla kuin sosiaalisen median palveluissa. Toisaalta tässä tutkielmassa ei ole käyty läpi jokaista hyökkäyskeinoa ja sen suojausta, joita palveluissa esiintyy, vaan tarkasteltaviksi kohteiksi on valittu yleisimmät ja mielenkiintoisimmat tyypit ja joista löytyy kattavimmin tieteellistä materiaalia. Tutkimusta voisi jatkaa esimerkiksi viemällä aihetta enemmän psykologisempaan suuntaan ja tarkastelemalla, mikä saa käyttäjän paljastamaan itsestään tietoja. Jatkotutkimuksen aiheena voisi myös olla sosiaalisen median palveluiden tietoturvallisuus palvelujen tarjoajien näkökulmasta.

On mielenkiintoista nähdä, miten tulevaisuudessa tietojen kalastelu ja sosiaalinen media muuttuvat. Uusia hyökkäystyyppejä kehitetään jatkuvasti ja tämä tulee ottaa huomioon tietoturvallisuudessa ja palveluiden kehittämisessä. On kuitenkin lähes mahdotonta kuvitella täysin tietoturvallinen maailma. Turvallisuuden menettämiseen voi riittää pienikin aukko, jonka joku ennen pitkää löytää. Ryhmä on vain niin vahva kuin sen heikoin lenkki. Tämä koskee myös virtuaalista maailmaa ja turvallisuutta.

LÄHTEET

Aggarwal, A., Rajadesingan, A. & Kumaraguru, P. (2012). PhishAri: Automatic realtime phishing detection on Twitter. Teoksessa *eCrime Researchers Summit (eCrime)* (s. 1-12). Las Croabas, October, 23-24, 2012.

Ahmadinejad, S. H., Anwar, M. & Fong, P. W. L. (2011). Inference attacks by third-party extension to social network systems. Teoksessa *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (s. 282-287). Seattle, WA, March 21-25, 2011.

Ahmed, F. & Abulaish, M. (2012). An MCL-based approach for spam profile detection in online social networks. Teoksessa *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (s. 602-608). Liverpool, June, 25-27, 2012.

Arulsuju, D. (2011). Hunting malicious attacks in social networks. Teoksessa *Third International Conference on Adcanced Computing (IcoAc)* (s. 13-17). Chennai, December, 14-16, 2011.

Callegati, F. & Ramilli, M. (2009). Frightened by links. *IEEE Security & Privacy, 7(6)*, 72-76.

Cashion, J. & Bassiouni, M. (2011). Protocol for mitigating the risk of hijacking social networking sites. Teoksessa *2011 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing(CollaborateCom)* (s. 324-331).Orlando, FL, October, 15-18, 2011.

Chaitanya, K., Ponnapalli, H., Herts, D. & Pablo, J. (2012). Analysis and detection of modern spam techniques on social networking sites. Teoksessa *Third International Conference on Services in Emerging Markets* (s. 147-152). Mysore, December 12-15, 2012.

Chandramouli, R. (2011). Emerging social media threats: technology and policy perspectives. Teoksessa *Second Worldwide Cybersecurity Summit (WCS)* (s. 1-4). London, June, 1-2, 2011.

De Paula, A. M. G. (2010). Security aspects and future trends of social networks. Teoksessa *Proceedings of the Fifth International Conference of Forensic Computer Science* (s. 60-79). Brazilia, September, 15-17, 2010.

Erlandsson, F., Boldt, M. & Johnson, H. (2012). Privacy threats related to user profiling in online social networks. Teoksissa *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT), Social Computing (SocialCom)* (s. 838-842). Amsterdam, September, 3-5, 2012.

Faghani, M. R. & Nguyen, U. T. (2013). A study of XSS worm propagation and detection mechanisms in online social networks. *IEEE Transactions on Information Forensics and Security, 8(11)*, 1815-1826.

Faghani, M. R. & Saidi, H. (2009). Social networks' XSS worms. Teoksessa *2009 International Conference on Computational Science and Engineering* (s. 1137-1141). Vancouver, August, 29-31, 2009.

Huber, M., Mulazzani, M., Kitzler, G., Goluch, S. & Weippl, E. (2011). Friend-in-the-middle attacks: Exploiting social networking sites for spam. *IEEE Internet Computing, 15(3)*, 28-34.

Huber, M., Mulazzani, M. & Weippl, E. (2010). Social networking sites security : Quo vadis. Teoksessa *IEEE Second International Conference on Social Computing (SocialCom)* (s. 1117-1122). Minneapolis, MN, August, 20-22, 2010.

Jagnere, P. (2012). Vulnerabilities in social networking sites. Teoksessa *Second IEEE International Conference on Parallel Distributed amd Grid Computing (PDGC)* (s. 463-468). Solan, December, 6-8, 2012.

Jakobsson, M. & Myers, S. (2007). Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft. USA: John Wiley & Sons, Inc.

Kaplan, A.M. & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of social media. *Business Horizons, 53,* 59-68.

Kontaxis, G., Polakis, I., Ioannidis, S. & Markatos, E.P. (2011). Detecting social network profile cloning. Teoksessa 2011 *IEEE Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (s. 295-300). Seattle, WA, March, 21-25, 2011.

Lundeen, B. & Alves-Foss, J. (2012). A practical clickjacking with BeFF. Teoksessa *IEEE Conference on Technologies for Homeland Security (HST)* (s. 614-619). Waltham, MA, November, 13-15, 2012.

Mahmood, S. (2012). New privacy threats for Facebook and Twitter users. Teoksessa *2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)* (s. 164-169). Victoria, BC, November, 12-14, 2012.

Mahmood, S. & Desmedt, Y. (2012). Your Facebook activated friend or cloaked spy. Teoksessa *2012 IEEE Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (s. 367-373). Lugano, March, 19-23, 2012.

Nagy, J. & Pecho, P. (2009). Social networks security. Teoksessa *Third International Conference on Emerging Security Information, Systems and Technologies* (321-325). Athens, June, 18-23, 2009.

Ollmann, G. (2007). The Phishing Guide Understanding and Preventing Phishing Attacks. IBM Internet Security Systems.

Patsakis, C., Asthenidis, A. & Chatzidimitriou, A. (2009). Social networks as a attack platform: Facebook case study. Teoksessa *Eighth International Conference on Networks* (s. 245-247). Gosier, March, 1-6, 2009.

Rehman, U.U., Khan, W.A., Saqib, N.A. & Kaleem, M. (2013). On detection and prevention of clickjacking attack for OSNs. Teoksessa *11th International Conference on Frontiers of Information Technology (FIT)* (s. 160-165). Islamabad, December, 16-18, 2013.

Robertson, M., Pan, Y. & Yuan, B. (2010). A social approach to security: Using social networks to help detect maliciousweb content. Teoksessa *2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE)* (s. 436-441). Hangzhou, November, 15-16, 2010.

Sanzgiri, A., Hughes, A. & Upadhyaya, S. (2013). Analysis of malware propagation in Twitter. Teoksessa *IEEE 32nd International Symposium on Reliable Distributed Systems (SRDS)* (s. 195-204). Braga, September, 30-October, 3, 2013.

Shar, L.K. & Tan, H.B.K. (2012). Defending against Cross-Site Scripting attacks. *Computer, 45(3)*, 55-62.

Simola, S. & Hietaneva, P. (2014). Kaksi näkemystä Facebookin tulevaisuudesta. Helsingin sanomat. Luettu 9.2.2014. Saatavilla osoitteessa http://www.hs.fi/tekniikka/Kaksi+n%C3%A4kemyst%C3%A4+Facebookin+tulevaisuudesta/a1390607169082 .

Smith, C. (2014). (February 2014) by the numbers : 116 amazing Twitter Statistics. Luettu 9.2.2014. Saatavilla osoitteessa http://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/#.Uve4-fl_t1Y .

Sun, Y. & He, D. (2012). Model checking for the defence against Cross-site Scripting attacks. Teoksessa *International Conference on Computer Science and Service System (CSSS)* (s. 2161-2164). Nanjing, August, 11-13, 2012.

The X-Frame-Options response header. Luettu 5.3.2014. Saatavilla osoitteessa https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options#Browser_compatibility .

Tsai, D-R., Chang, A.Y., Chung, S-C. & Li, Y. S. (2010). A proxy-based real-time protection mechanism for social networking sites. Teoksessa *IEEE International Carnahan Conference on Security Technology (ICCST)* (s. 30-34). San Jose, CA, October, 5-8, 2010.

Weir, G. R. S., Toolan, F. & Smeed, D. (2011). The threats of social networking: Old wine in new bottles?. *Information Security Technical Report, 16(2)*, 38-43.

X-Frame-Options Compatibility Test. Luettu 5.3.2014. Saatavilla osoitteessa http://erlend.oftedal.no/blog/tools/xframeoptions/.