Matias Ruonala, Miro Hintikka, Ville schulz, Joel Simola

Tietoturva web-sovelluksissa

Metropolia Ammattikorkeakoulu

Alaantutustumistehtävä

Raportti

21.9.2023

Sisällys

Sisällysluettelo

[1 Johdanto 1](#_Toc146531589)

[2 Web sovellus ympäristönä 1](#_Toc146531590)

[2.1 Mikä on web-sovellus? 1](#_Toc146531591)

[2.2 Web-sovelluksen hyödyt ja haitat 1](#_Toc146531592)

[3 Yleisimmät web-sovellusten tietoturvauhat 3](#_Toc146531593)

[3.1 Kolme suurinta tietoturvauhkaa 3](#_Toc146531594)

[3.1.1 Broken access control 4](#_Toc146531595)

[3.1.2 Cryptographic failures 4](#_Toc146531596)

[3.1.3 Injection 4](#_Toc146531597)

[3.2 Tietoturva kehittäjän näkökulmasta 5](#_Toc146531598)

[Lähteet 7](#_Toc146531599)

Liitteet

Liite 1: Kirjoita manuaalisesti

Liite 2: Liitteen nimi

# Johdanto

Tässä raportissa käsittelemme web-sovellusten tietoturvaa. Toinen kappale esittelee mikä web-sovellus on, miten se on pääpiirteissään rakennettu, ja mitä hyötyjä ja haittoja web-sovelluksella on natiiviin mobiilisovellukseen tai työpöytäsovellukseen verrattuna. Toinen luku kuvailee yleisimpiä web-sovelluksiin kohdistuvia tietoturvauhkia ja miten niitä tulee huomioida sovellusta kehittäessä. Kolmannessa luvussa …. LUKU KESKEN, MIHIN TULEE TIETOTURVA KÄYTTÄJÄN NÄKÖKULMASTA

# Web sovellus ympäristönä

## Mikä on web-sovellus?

Web-sovellus on verkkopalvelimella suoritettava ohjelmisto, jota käyttäjä ohjaa verkkoselaimellaan [1]. Sovelluksen käyttäjälle näkyvä osa, ns. frontend, on siis verkkosivu, joka on yleensä kirjoitettu HTML5:llä, Javascriptillä ja CSS:llä. Frontend vastaanottaa käyttäjän syötteen, ja näyttää käyttäjälle halutun sisällön. Palvelinpuoli eli backend huolehtii asiakaspyyntöjen käsittelystä, laskennasta sekä tiedonhallinnasta ja tallentamisesta. Backendin toimintalogiikka voidaan kirjoittaa esimerkiksi Pythonilla, Javalla tai Rubylla [2]. Kuva 1 esittelee web-sovelluksen rakenteen karkeasti.

## Web-sovelluksen hyödyt ja haitat

Koska web-sovellusta voi käyttää internet-selaimella, web-sovellus ei ole laitteistoriippuvainen. Toisin sanoen riittää, että ohjelmistokehittäjä suunnittelee yhden ohjelmiston ja käyttöliittymän, mitä voi käyttää millä tahansa laitteella ja käyttöjärjestelmällä jossa on asennettuna moderni verkkoselain [3]. Web-sovelluksen tuotantokustannukset ovat matalammat, kuin esimerkiksi natiivisovellusten, jotka räätälöidään joka alustalle erikseen [4].

Kuva . Web-sovelluksen rakenne

Työpöytäsovellukseen verrattuna web-sovelluksen suorituskyky on heikompi eikä sen käyttöliittymän ulkoasu ei ole yhtä hienostunut. Tämä johtuu paitsi siitä, että tietyt elementit voivat näkyä eri selaimissa eri tavoin, web-käyttöliittymän elementtien sijoittelua ei myöskään voi määritellä absoluuttisesti. Toisaalta web-käyttöliittymä voi olla selainympäristöön tottuneelle käyttäjälle intuitiivisempaa kuin työpöytäsovelluksen käyttäminen. Websovellusten päivittäminen on myös huomattavasti helpompaa, kuin työpöytäsovellusten [5].

# Yleisimmät web-sovellusten tietoturvauhat

Web-sovelluksiin kohdistuu nykypäivänä monia erilaisia uhkia, joita vastaan yritykset joutuvat taistelemaan päivittäin. Tätä voisi ajatella eräänlaisena kissa ja hiiri -leikkinä. Kun uusi haavoittuvuus löydetään, pitää se paikata mahdollisimman nopeasti ilman, että haavoittuvuudesta aiheutuu suurta vahinkoa yritykselle tai yrityksen asiakkaille.

## Kolme suurinta tietoturvauhkaa

”Yleisin tietoturvauhka on käyttäjä itse” [6], eli käyttäjä voi omilla teoillaan vaikuttaa käytöstä aiheutuviin riskeihin. The Open Worldwide Application Security Project, eli lyhyemmin OWASP, on koonnut vuoden 2021 yleisimpiä haavoittuvuuksia listaksi. [7]

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, viiva

Kuvaus luotu automaattisesti

OWASP top-10 tietoturva haavoittuvuudet 2017 ja 2021 [7]

### Broken access control

Access control, suomeksi valtuuksien tarkistus, valvoo, että web-sovelluksen käyttäjä ei pysty toimimaan yli hänelle annettujen oikeuksien sovelluksen sisällä. Viallinen valtuuksien tarkistus johtaa yleensä luvattomien tietojen paljastumisen, muokkaamisen tai tuhoamisen mahdollisuuteen käyttäjälle. Tältä voi välttyä oikeanlaisella suunnittelulla projektin alkuvaiheessa ja myöntää käyttäjälle vain ne oikeudet, jotka ovat välttämättömiä.

### Cryptographic failures

Tiedon salaamista on käytetty jo muinaisen Egyptin ja Rooman ajoista asti. Hallitsijoiden lähetit ovat kuljettaneet salattuja viestejä, jottei viestin sisältö ole ymmärrettävää väärissä käsissä. Sivilisaation kehittyessä ja teknologian kehittyessä myös salausmenetelmät kehittyivät. Tietokoneiden aikana salakirjoitusmenetelmät ja niiden tutkimus on kehittynyt huomattavasti ja salakirjoitusmenetelmää käytetään useissa sovelluksissa esimerkiksi pankkisovellukset ja pikaviestimet [8]. Salaustekniikan epäonnistuminen koskee arkaluotoisen tiedon, esimerkiksi salasanojen, luottokortin tietojen tai henkilökohtaisten tietojen huonoa salausta esimerkiksi vanhentunut salausalgoritmi tai salauksen puuttumista. Tähän kehittäjä voi vaikuttaa muistamalla salata arkaluontoisen tiedon säilyttäessään sekä siirtäessään sitä ja pitämällä huolen, ettei evästeet tallenna arkaluontoista tietoa ja olla käyttämättä vanhentunutta teknologiaa tiedonsiirroissa.

### Injection

Injektiohyökkäyksistä yleisimmät ovat SQL-injektio, NoSQL-injektio ja object-relational mapping, lyhyemmin ORM [9]. Structured query language, lyhennettynä SQL, on kyselykieli, jota käytetään ohjelmoinnissa tietokantoihin tallennettuihin tietoihin pääsemiseksi. SQL-injektio kuuluu yhteen vanhimmista verkkosovellushaavoittuvuuksista. SQL-injektio on hyökkäys, jossa virheellistä SQL-koodia syöttämällä verkkosovellukseen voidaan manipuloida, lukea tai jopa poistaa arkaluontoisia tietoja tietokannasta. Mitä keskitetympää tieto on, sitä pahempaa jälkeä injektiohyökkäykset saavat aikaan. Näitä hyökkäyksiä vastaan voidaan suojautua esimerkiksi käsittelemällä käyttäjän syöttämää tietoa ennen kuin sillä haetaan tietoja verkkosivuston käyttämästä tietokannasta [10, Kuva 11].

SQL injektio havainnollistettuna [11].

Kuva, joka sisältää kohteen teksti, animaatio, kuvakaappaus, kuvitus

Kuvaus luotu automaattisesti

## Tietoturva kehittäjän näkökulmasta

Verkkosovellukset kehittyvät koko ajan, uusia ominaisuuksia halutaan lisätä vanhoihin sivuihin, sillä yritykset vaativat niiltä jatkuvasti enemmän. Kehittäjät ovat toistaiseksi vielä ihmisiä ja ihmisaivot eivät ole kehittyneet viimeiseen sataan vuoteen samaa vauhtia kuin teknologia on. Ei ole epänormaalia, että uuden projektin kirjoitettuun koodiin jäisi pieniä virheitä, jotka saattaisivat vaarantaa sovelluksen tietoturvaa. Yleensä kehittäjä ei pääse nauttimaan uuden projektin aloittamisesta vaan ylläpitämään jonkun aikaisemman kehittäjän kirjoittamaa sovellusta. Tämä tarkoittaa koodin ymmärtämistä ja usein korjaamista välittämättä siitä kuka sen on kirjoittanut, missä se on kirjoitettu, kuinka lahjakas edellinen kehittäjä on ollut tai mikä tietoturvan tärkeys, sillä hetkellä on ollut. Tämän takia tietoturvallista sovellusta voi olla vaikea toteuttaa ilman, että rikkoo olemassa olevia toimintoja ja yritykselle kriittisiä laitteita [12].

Lähteet

1. <https://www.britannica.com/topic/Web-application> selattu 22.9.2023
2. <https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app> selattu 22.9.2023
3. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-web-app-development/> selattu 22.9.2023
4. <https://aws.amazon.com/compare/the-difference-between-web-apps-native-apps-and-hybrid-apps/> selattu 22.9.2023
5. Tehetena Masresha, Changing Desktop application to Web application, 10.8.2018, sivu 5
6. Suomen asiakastieto Oy https://www.asiakastieto.fi/omatieto/fi/artikkelit/20221012160000 21.09.2023
7. OWASP https://owasp.org/www-project-top-ten 21.03.2023
8. Veikko Siivola Salakirjoituksen historia https://www.cs.helsinki.fi/u/kerola/tkhist/k2000/alustukset/salakirjoitus/crypto.html 24.09.2023
9. OWASP https://owasp.org/Top10/A03\_2021-Injection/#description 21.09.2023
10. NordVPN https://nordvpn.com/fi/blog/sql-injektio/ 21.09.2023
11. https://techterms.com/definition/sql\_injection 24.09.2023
12. Secarma blogi kirjoitus https://secarma.com/software-security-a-developers-point-of-view-part-one/ 22.09.2023