Henna Korhonen

Web-sovelluksen penetraatiotestaus

Tieto- ja viestintätekniikka
Insinööri (AMK)
Syksy 2021

KAMK • University of Applied Sciences

Tiivistelmä

Tekijä(t): Korhonen Henna

Työn nimi: Web-sovelluksen penetraatiotestaus

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: tietoturva, tietoturvatestaus, penetraatiotestaus, OWASP, Burp Suite

Erilaiset web-sovellukset ovat arkipäiväistyneet ja 2020-luvulla melkein jokainen yritys tai julkinen toimija hyödyntää web-sovelluksia palveluntarjonnassaan. Tämän myötä myös erilaiset tietoturvauhat ovat kasvaneet samassa suhteessa uusien web-teknologioiden rinnalla ja tietoturvatestaamisesta on tullut entistä tärkeämpi osa tuotekehitysprosessia. Yksi tehokas tapa web-sovelluksien tietoturvahaavoittuvuuksien löytämiseksi on penetraatiotestaus.

Tässä opinnäytetyössä tutustutaan penetraatiotestaukseen ja sen vaiheisiin sekä suoritetaan penetraatiotestaus toimeksiantajan määräämälle kohteelle. Penetraatiotestauksessa jäljitellään tapoja, joita pahaaikeiset hyökkääjät käyttävät murtautuessaan web-sovelluksiin. Penetraatiotestauksen tarkoituksena on löytää ja tukkia nämä tietoturva-aukot, ennen kuin hyökkääjät ennättävät ne löytää. Penetraatiotestaus toteutettiin käyttämällä Burp Suite Professionalia ja sen tarjoamia lisäosia. Testaustuloksista raportoitiin toimeksiantajana toimineelle yritykselle kattavalla raportilla, jossa ilmenivät löydetyt haavoittuvuudet sekä niiden korjausehdotukset.

Työn tietoperustassa tarkastellaan laajemmin myös tietoturvallisuutta, tietoturvatestausta yleisellä tasolla sekä perehdytään OWASP Top 10 -raporttiin ja erilaisiin penetraatiotestauksissa käytettäviin standardeihin, joihin penetraatiotestaus perustui.

Työn lopussa pohdittiin, olisiko penetraatiotestaus mahdollista automatisoida toimeksiantajan tarpeisiin. Tämän työn toimeksiantajana toimii suomalainen ohjelmistoalan yritys.

Abstract

Author(s): Korhonen Henna

Title of the Publication: Penetration Testing of Web Application

Degree Title: Bachelor of Engineering, Information and Communication Technologies

Keywords: security testing, penetration testing, OWASP, Burp Suite

Various web applications have become commonplace, and in the 2020s almost every company or public actor will take advantage of web applications in its service offering. As a result, various security threats have increased in proportion to the new web technologies, and security testing has become an increasingly important part of the product development process. One effective way to find security vulnerabilities in web applications is through penetration testing.

In this thesis, the penetration testing, and its stages are introduced, and the penetration testing is performed on an object specified by the client. Penetration testing mimics the ways in which malicious attackers break into web applications. The purpose of penetration testing is to find and block these security vulnerabilities before attackers anticipate finding them. Penetration testing was performed using Burp Suite Professional and its add-ons. The test results were reported to the client company in a comprehensive report, which revealed the vulnerabilities found and suggestions for correcting them.

The knowledge base of the thesis also examines information security, security testing in general, and the OWASP Top 10 report and the various standards used in penetration testing, on which the penetration testing was based.

At the end of the work, it was considered whether it would be possible to automate penetration testing to meet the needs of the client. This work is commissioned by a Finnish software company.

Sisällys

Termiluettelo

ASP.NET	Microsoftin kehittämä ja markkinoima web-ohjelmistokehys.
Haavoittuvuus	Mikä tahansa heikkous, joka mahdollistaa vahingon toteutumisen tai jota voidaan käyttää vahingon aiheuttamisessa. Haavoittuvuuksia voi olla tietojärjestelmissä, prosesseissa ja ihmisen toiminnassa.
Hakkeri	Henkilö, joka tunkeutuu tietoverkkoon tai tietojärjestelmään tai käyttää luvatta tietoa, ohjelmaa tai palvelua.
Hakkerointi	Toimintaa, jossa tunkeudutaan tai vaikutetaan tietoverkkoon, tietojärjestelmään tai niiden sisältämään tietoon ja käytetään ohjelmaa, palvelua tai muuta resurssia.
HTML	Hypertekstin rakenteen ja ulkoasun kuvaukseen käytettävä merkintäkieli.
HTTP	Protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon.
IDS-järjestelmä	Tekninen järjestelmä, jonka on tarkoitus havaita tunkeutumiset ja niiden yritykset.
IP-osoite	Internetiin kytketyn tietojenkäsittely- tai tiedonsiirtolaitteen tai verkkoliittymän yksilöivä numeerinen tunnus.
IPS-järjestelmä	Tekninen järjestelmä, jonka on tarkoitus estää tunkeutumiset.
Java	Sun Microsystemsin kehittämä laitteisto- ja ohjelmistoympäristöstä riippumattomaksi tarkoitettu oliokieli.
Ohjelmistoviitekehys	Ohjelmistotuotteen runko.
OWASP	Järjestö, jonka tavoitteena on edesauttaa luotettavien sovellusten kehittämistä, valmistamista ja ylläpitämistä.
Palomuuri	Tekninen järjestely, jonka on tarkoitus estää asiaton pääsy verkosta toiseen.

SQL	Yleisesti relaatiotietokannan käsittelyssä käytettävä kysely- ja määritys-kieli.
Tietomurto	Luvaton tietojärjestelmään, palveluun tai laitteeseen tunkeutuminen tai sovelluksen, kuten esimerkiksi sähköpostitilin luvaton käyttö haltuun saatujen tunnusten avulla.
Tietoturva	Järjestelyt, joilla pyritään varmistamaan tiedon luottamuksellisuus, eheys ja saatavuus.
Tietoturva-aukko	Tietojärjestelmän tai sen osan tietoturvajärjestelyjen heikkous, joka vaarantaa tietoturvan.
Välityspalvelin	Palvelin, joka hakee internetistä tietoa työasemien puolesta ja säilyttää työasemien toistuvasti käyttämiä tietoja niiden saannin nopeuttamisek-si.
Web-sovellus	Tietoverkossa selaimen tai erityisen ohjelman avulla käytettävä sovel-lus.

1 Johdanto

Tieto- ja viestintätekniikoista on tullut viime vuosikymmeninä olennainen osa suomalaista yhteiskuntaa ja ihmisten jokapäiväistä elämää. Tekniikkaa sovelletaan jatkuvasti uusiin käyttökohteisiin, ja tietoa kerätään ja käsitellään tietoverkoissa valtavia määriä. Tietoturvalla on tärkeä rooli kriittisten sosiaalisten järjestelmien ja tavallisten kansalaisten jokapäiväisen elämän suojelemisessa. [1.]

Vaikka yhteys tietoturvan ja luottamuksen välillä käyttökokemukseen on lisääntynyt, käyttäjien kyky hallita ja ymmärtää tietoturvaa ei ole lisääntynyt samassa suhteessa [2]. Siksi tarvitaan menetelmiä ja asiantuntemusta käyttäjien suojelemiseksi ja luottamuksen säilyttämiseksi digitaalisiin palveluihin, mikä on digitaalisen yhteiskunnan toiminnan edellytys.

Erilaiset web-sovellukset ovat organisaatioiden yleinen tapa tarjota sähköisiä palveluja kuluttajille, mutta ne sisältävät usein vakavia tietoturva-aukkoja, jotka voivat paljastaa arkaluontoisia tietoja käyttäjille tai vahingoittaa organisaatiota. Vaikka ihmiset saattavat ajatella, että web-sovellukset ovat nykypäivänä turvallisempia, totuus on vähemmän optimistinen. Web-sovelluksien käyttämät teknologiat kehittyvät jatkuvasti ja uudet tekniikat tuovat myös uusia turvallisuushaasteita. Yhtenä ratkaisuna on tietoturvatestauksen integroiminen kiinteäksi osaksi ohjelmistokehitystä.

Tietoturvatestaus on yksi tärkeä osa digitaalisten sovellusten ja palveluiden tietoturvaa. Useat organisaatiot eivät testaa ohjelmistojaan, ennen kuin ne ovat jo elinkaarensa käyttöönottovaiheessa. Tehokas tapa parantaa web-sovellusten tietoturvallisuutta on puuttua mahdollisiin ongelmiin jo ohjelmistotuotteiden ja -palveluiden kehitysvaiheen aikana. Tällöin tietoturvariskit kyetään huomaamaan mahdollisimman nopeasti ja riskeihin voidaan tarpeen vaatiessa puuttua. Penetraatiotestaus on tietoturvatestausmenetelmä, joka simuloi luvatonta hyökkäystä web-sovelluksiin. Testauksen tarkoituksena on löytää haavoittuvuudet web-sovelluksista ja parantaa niiden suojaustasoa. [3.]

Tässä opinnäytetyössä käsitellään web-sovelluksien tietoturvallisuutta ja toteutetaan ohjelmistoalan yrityksen web-sovellukselle penetraatiotestaus. Työn toimeksiantajana toimii suomalainen ohjelmistoalan yritys ja penetraatiotestaus toteutetaan yrityksen tuotteelle, joka on .Net-alustalla toimiva web-sovellus.

2 Työn tavoitteet ja toteutus

Tämän opinnäytetyön tavoitteena oli tutkia penetraatiotestaukseen liittyviä vaiheita ja toimintatapoja, selvittää penetraatiotestauksen avulla tarkastettavan kohteen haavoittuvuudet ja informoida tuloksista työn toimeksiantajana toimineelle yritykselle. Toimeksiantajan toiveena oli myös selvittää, olisiko penetraatiotestausta mahdollista automatisoida yrityksen tuotteelle sopivaksi automaattiseksi testausprosessiksi.

Opinnäytetyö toteutettiin kolmessa vaiheessa. Ensimmäiseen vaiheeseen kuului tietoturvallisuuden ja tietoturvatestauksen perusteoriaan perehtyminen, mitkä ovat tietoturvatestauksen tavoitteet ja millaisia tuloksia sillä on mahdollista saada. Penetraatiotestaajan pohjatietojen on oltava kattavat, jotta testaustuloksia pystytään analysoimaan oikein ja todentamaan manuaalisesti.

Työn toisessa vaiheessa tutustuttiin perusteellisesti itse penetraatiotestaukseen, OWASP Top 10 -raporttiin ja sen listaamiin yleisimpiin web-ohjelmistojen haavoittuvuuksiin, OWASP Testing Guideen sekä Penetration Testing Execution Standardiin, joiden tietoja ja opastuksia hyödynnettiin koko penetraatiotestauksen ajan.

Kolmannessa vaiheessa keskityttiin itse testaukseen. Testaus suoritettiin käyttämällä Burp Suite Professional -työkalua ja sen eri lisäosia ja moduuleita. Opinnäytetyöhön kirjattiin testauksesta ja sen vaiheista niin kattavasti kuin vain oli mahdollista ilman, että toimeksiantajana toimineen yrityksen identiteettiä tai kohteen tietoja pystyisi tunnistamaan. Testauksesta laadittiin yritykselle kattava testausraportti, jossa lukee testauksen tulokset ja havaitut haavoittuvuudet ja riskit korjausehdotuksineen.

Työn lopuksi pohditaan, olisiko penetraatiotestauksen automatisointi mahdollista toimeksiantajan tuotteelle ja yhteenveto.

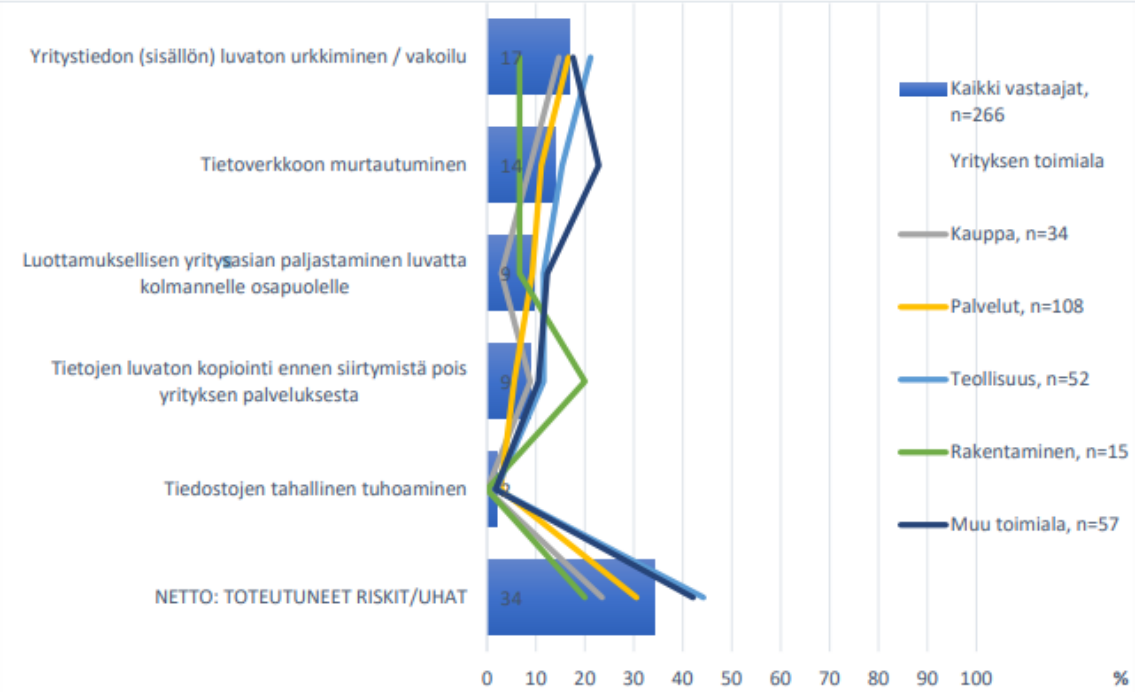
3 Tietoturvallisuus

Valtionhallinnon tietoturvallisuuden johtoryhmä (VAHTI) (3/3007, 13) määrittelee tietoturvallisuuden seuraavasti, *”Tietoturvallisuudella tarkoitetaan tietojen ja palvelujen, järjestelmien ja tietoliikenteen suojaamista ja varmistamista niihin kohdistuvien riskien hallitsemiseksi sekä normaali- että poikkeusoloissa hallinnollisilla, teknisillä ja muilla toimenpiteillä. Tietoturvallisuuden tavoitteena on tietojen luottamuksellisuuden, eheyden ja saatavuuden turvaaminen laitteisto- ja ohjelmistovikojen, luonnontapahtumien sekä tahallisten, tuottamuksellisten tai tapaturmaisten tekojen aiheuttamilta uhilta ja vahingoilta.”*

Perinteisessä jaottelussa tietoturva koostuu siis kolmesta tavoitteesta: tiedot ovat vain niiden käytössä, joilla on niihin oikeus (luottamuksellisuus), silloin kun he niitä tarvitsevat (saatavuus) ja että tieto on oikeassa muodossa eikä se ole matkan varrella muuttunut (eheys).

3.1 Tietoturva organisaatiossa

Yhä useammat yritykset kohtaavat koosta riippumatta turvallisuusriskejä. 34 prosenttia suomalaisista yrityksistä ilmoitti joutuneensa tietoturvahyökkäyksen kohteeksi viimeisten kolmen vuoden aikana. Suurin yksittäinen tietoturvariski oli ollut yrityksen tietojen luvaton vakoilu (kuva 1). Yrityksistä yli puolet arvioi, että tietoturvallisuusriskit ovat lisääntyneet paljon tai jonkin verran. Tiedot perustuvat vuonna 2020 Helsingin seudun kauppakamarin ja Taloustutkimuksen teettämään kyselyyn, jossa haastateltiin 266 yritystä. [4.] Trendi on hyvin selvä ja osoittaa tarpeen parantaa yritysten tietoturvaa.



Kuva 1. Yritysten tietoriskit viimeisen kolmen vuoden aikana.

Tietomurrot aiheuttavat taloudellisia- ja mainetappioita kohteena olevalle organisaatiolle. Lisäksi ympäristön korjaamisen tai uudelleenasennuksen vuoksi normaalit organisaation toiminnot voivat keskeytyä pitkäksi aikaa [5]. Olennaista on ymmärtää ja tunnistaa, mikä on yritykselle keskeinen tieto ja sen jälkeen kyetä ennakoimaan, havaitsemaan, torjumaan ja rajoittamaan uhkaavia tietoturvariskejä. Kyse onkin tavallaan eräänlaisesta tietoturvavakuutuksesta. Varautuminen merkitsee kustannuksia, mutta riskin realisoituessa vahingot voivat moninkertaistua.

Tietoturvallisuus ja varautuminen voivat olla parhaimmillaan yritykselle kilpailuetu. Asiakkaat saattavat vaatia yrityksen tuotteilta ja palveluilta turvallisuutta ja korkeaa laatua, vaikka he eivät pysty itse todentamaan näitä toimintoja. Yrityksen maine ja imago hyötyvät, kun he pystyvät viestimään kuluttajille ja asiakkaille ”meillä asiakkaiden tiedot pysyvät tallessa”.

Tietoturvallisuus vaikuttaa jokaiseen organisaation jäseneen eikä siitä huolehtiminen ole enää yksin tietoturva-ammattilaisten vastuulla. Yrityksiä uhkaavat esimerkiksi huijaussähköpostit, Internetin saastuttamat haittaohjelmat, identiteettipetokset, kriittiset yritystiedostot työntekijöiden kotitietokoneilla ja suojaamattomat langattomat verkot. On tärkeää, että organisaatio panostaa tarpeeksi henkilöstön tietoturvaosaamiseen esimerkiksi koulutuksilla, sillä työntekijöiden huolimattomuus on edelleen suurin syy tietomurtoihin [6]. Paras lopputulos olisi, että koko organisaation henkilöstö toimisi samojen käytäntöjen mukaisesti ja yhtenäisesti pienentäen tietoturvariskejä merkittävästi.

3.2 Tietoturvallisuuden standardit

Tietoturvallisuuteen liittyviä vaatimusstandardeja on oikeastaan vain yksi. ISO/IEC 27001 on kansainvälinen vaatimusstandardi, jota vasten organisaatio voi sertifioida oman toimintansa. Siinä esitetään yksityiskohtaiset vaatimukset tietoturvan perustamiselle, toteuttamiselle, ylläpidolle ja jatkuvalle parantamiselle [7]. Standardia hyödynnetään sekä julkisella että yksityisellä sektorilla maailmanlaajuisesti. ISO/IEC 27001:n vaatimuksiin voidaan päästä hyödyntämällä ISO/IEC 27002 -soveltamisstandardia [8].

Vuonna 2019 ISO-standardoimisjärjestö julkaisi standardilaajennoksen ISO/IEC 27701 ja liittää ISO 27001:een mukaan myös tietosuojan. Laajennos kattaa vuonna 2018 julkaistun EU:n tietosuoja-asetuksen (GDPR) tuomat vaatimukset ja sitä suositellaan otettavaksi käyttöön etenkin sellaisille yrityksille, joiden liiketoimintaan kuuluu henkilötietojen käsittely. [9.]

Tietoturvallisuuden kokonaisuuden hallitsemiseen käytetään myös Suomessa valtiovarainministeriön julkishallinnon digitaalisen turvallisuuden ohjausryhmä VAHTI:n antamaa ohjekokonaisuutta. Aiemmin VAHTI-ohjeistukset koskivat ensisijaisesti valtionhallintoa, mutta keväällä 2019 hyväksytyn tiedonhallintalain myötä vaatimukset laajentuivat koko julkishallintoon. [2.]

4 Tietoturvatestaus

Ohjelmistojen laadunvarmistukseen ja tietoturvaan kuuluu kattava testaus. Ennen kuin tuote tulee hyväksyntään, tulee sille suorittaa tietoturvatestaus todellista käyttötilannetta vastaavassa ympäristössä. Tietoturvatestauksen tarkoituksena on paljastaa puutteet ja aukot järjestelmän tietoturvassa ja täten estää mahdolliset väärinkäyttötapaukset ja parantaa tietoturvan tasoa. Testauksen ja laadunvarmistuksen tulisi keskittyä toistettaviin ja automatisoituihin menetelmiin, koska niillä saavutetaan suurempi testikattavuus, mikä mahdollistaa järjestelmän muutosten tehokkaan ja luotettavan testauksen. [1.]

Koska tietoturvan vaatimukset määritellään jo tuotekehitysprosessin alkuvaiheessa, myös tietoturvatestaaminen pystytään aloittamaan heti. Tietoturvatestaus on siinä mielessä haasteellista, että ohjelmisto saattaa toimia täysin oikein, vaikka siinä olisikin tietoturva-aukkoja. Näin ollen, haavoittuvuuksien löytäminen voi olla vaikeaa. Tietoturvatestaamisen aloittaminen heti tuotekehitysprosessin alkuvaiheessa on tärkeää, koska jälkeenpäin tehtynä se voi olla äärimmäisen kallista, haastavaa tai jopa mahdotonta [3].

Testauksessa käytettävät työkalut toimivat joko automaattisesti tai manuaalisesti. Automatisoidun testauksen etuna on se, että ne ovat nopeita ja helppoja ja säästävät organisaation aikaa sekä rahaa. Automatisoiduissa testauksissa on kuitenkin myös omat ongelmakohtansa. Testausskannerit saattavat saada false-positive-tuloksia. False-positive-tulos kuvaa tilannetta, jossa testitapaus epäonnistuu, vaikka todellisuudessa vikaa ei ole ja ohjelmisto toimii niin kuin pitääkin. Mikäli false-positive-tuloksia on suuri määrä, heikentää se tehokkuutta ja vaikuttaa negatiivisesti kehitys- sekä testaustiimin työskentelyyn. Automatisoituja testausmenetelmiä voi olla esimerkiksi fuzz-testaus, missä järjestelmästä etsitään virheitä syöttämällä sille odottamattomia ja virheellisiä syötteitä [1].

Tehokas haavoittuvuuksien löytäminen vaatii näin ollen sekä automaattisen skannauksen, että löytyneiden haavoittuvuuksien manuaalisen läpikäymisen ja tarkastuksen. Tällaista testaustapaa kutsutaan penetraatiotestaukseksi.

4.1 Testaustyökalut

Tietoturvatestaamiseen käytetään yleensä erilaisia analyysityökaluja, jotka skannaavat ohjelmistoa tunnetuimpien haavoittuvuuksien varalta. Haavoittuvuuksia voivat olla esimerkiksi vanhentuneet ohjelmistoversiot ja -kirjastot, joten yksi suojaustyökalujen tehtävistä on varmistaa, että käytössä olevat versiot ja päivitykset ovat ajan tasalla.

Tässä opinnäytetyössä penetraatiotestaus suoritettiin Burp Suite Professional -testaustyökalulla, josta tarkempi kuvaus kappaleessa 6. On olemassa myös muita suosittuja testaustyökaluja, joista kaksi esimerkkiä alhaalla. Näitä työkaluja ei hyödynnetty tässä työssä, mutta ne ovat Burp Suiten rinnalla yksiä useimmiten käytettyjä työkaluja tietoturvatestaajien keskuudessa.

Kenties tunnetuin ja suosituin tietoturvatestauksissa käytetty työkalu on Kali Linux. Se on avoimen lähdekoodin Linux-distribuutio, joka on suunniteltu penetraatiotestaukseen ja tietosuojan valvontaan tietoturva-alan ammattilaisille. Kali Linux sisältää satoja erilaisia työkaluja ja ohjelmakirjastoja kaikkiin testausvaiheisiin. Kali Linuxia käytetään yleensä virtuaalikäyttöjärjestelmässä. [10.]

Toinen usein käytetty testaustyökalu on Owasp ZAP. Zed Attack Proxy (ZAP) on ilmainen, avoimen lähdekoodin testaustyökalu, jota ylläpidetään OWASP-projektissa. ZAP on suunniteltu erityisesti web-sovellusten testaamiseen ja sillä voidaan suorittaa useita erilaisia testauksia kohteelle. [11.]

4.2 Eettinen hakkerointi

Hakkerit ovat henkilöitä, jotka tunkeutuvat järjestelmään. Tunkeutumisen tarkoituksena on joko pahantahtoisesti tavoitella omaa etua (black hat hacking, krakkeri) tai järjestelmän omistajan suostumuksella etsiä ohjelmasta haavoittuvuuksia (white hat hacking, ethical hacking) [12].

Eettiset hakkerit (ethical hacking) ovat tietoturva-ammattilaisia, jotka yrittävät löytää järjestelmästä haavoittuvuuksia käyttäen samoja tekniikoita, kuin oikeatkin hyökkääjät ja raportoivat löydetyt haavoittuvuudet sekä niiden korjausehdotukset järjestelmän omistajille [13]. Eettinen hakkerointi voi kohdistua yhteen järjestelmään tai laajemmin organisaation tietojärjestelmään ja tietoliikenneverkkoon. Eettisestä hakkeroinnista käytetään myös nimitystä penetraatiotestaus.

5 Penetraatiotestaus teoriassa

Penetraatiotestaus, eli tunkeutumistesti tai pentest, on hyväksytty simuloitu tietoverkkohyökkäys tietojärjestelmää tai ohjelmistoa vastaan. Penetraatiotestauksen ensisijaisena tavoitteena on löytää toimeksiantajan järjestelmästä heikkouksia ja haavoittuvuuksia, joita paha-aikeinen hyökkääjä voisi hyödyntää, ja ilmoittaa toimeksiantajalle näistä haavoittuvuuksista sekä suositelluista toimenpidesuosituksista. Penetraatiotestaus suoritetaan aina toimeksiantajan luvalla eikä penetraatiotestausta saa ikinä suorittaa ilman järjestelmän omistajan lupaa. [14.] Ilman lupaa suoritettava testaus on rinnastettavissa black hat-hakkerointiin ja se on Suomen rikoslain 28 luvun 8 §:n mukaan määritelty laittomaksi.

Monet organisaatiot käyttävät penetraatiotestausta ensisijaisena tai ainoana tieturvatestaustekniikkana. Vaikka penetraatiotestauksella onkin paikkansa tuotekehityksen testausprosessissa, ei se ole tehokas selvittämään tietoturvariskejä yksinään [3]. Tästä syystä on tärkeä ottaa sen ohelle myös muita testausvaiheita aina tuotekehityksen alusta saakka.

5.1 Penetraatiotestauksen vaiheet

Penetration Testing Execution Standard (PTES) jakaa penetraatiotestauksen seitsemään vaiheeseen (kuva 2). PTES on penetraatiotestaamisen suorittamiseen suunniteltu standardi, jonka on koonnut joukko tietoturva-alan ammattilaisia eri teollisuuden aloilta.

Testausvaiheet on vapaasti suomennettu ymmärrettävyyden lisäämiseksi ja kaikissa testausvaiheiden määrittelyissä on käytetty lähteenä PTES:n dokumentaatiota, ellei lähteitä ole erikseen mainittu [15].

Kuva 2. PTES:n kuvaama standardi penetraatiotestaukseen

5.1.1 Määrittely

Ennen penetraatiotestauksen aloittamista on tärkeää määritellä reunaehdot, eli milloin ja minkälaisilla lähtötiedoilla penetraatiotestaus suoritetaan [16]. Lähtökohtaisesti penetraatiotestauksen voi suorittaa kolmella eri tavalla, jotka myös määrittelevät minkälainen hyökkääjä on simulaatiossa kyseessä:

- Valkoinen laatikko (engl. white box). Testaajalle annetaan kaikki mahdollinen tieto toimeksiantajansa ohjelmistosta. Tämän tyyppisen testauksen etuna on se, että simuloidun hyökkääjän täytyy olla sisäpiiriläinen.

- Harmaa laatikko (engl. grey box). Testaajalle annetaan jotain tietoja ohjelmistosta. Simuloitu hyökkääjä voisi tässä tapauksessa olla jokin henkilö, joka tuntee jonkin verran ohjelmistoa tai jolla on osittainen pääsy tietoihin ja tietokantaan.

- Musta laatikko (engl. black box). Testaajalle ei anneta järjestelmästä, verkosta tai sovelluksesta minkäänlaisia lähtötietoja ennen testausta. Tämän testauksen etuna on se, että se simuloi tosielämän hyökkäystä. Kohde voisi olla esimerkiksi julkinen palvelu, johon hyökkää täysin ulkopuolinen henkilö. Musta laatikko on tyypillisin penetraatiotestauksessa käytetty menetelmä.

Lisäksi toimeksiantajan kanssa olisi hyvä keskustella ja sopia, millaisia tavoitteita heillä on testaukseen liittyen, mitkä IP-osoitteet kuuluvat testaukseen ja mitkä jätetään ulkopuolelle, onko testauksen ajankohdalla tai kellonajalla väliä, onhan mahdollisia kolmansia osapuolia informoitu penetraatiotestauksesta ja sen ajankohdasta ja mihin tulisi ilmoittaa mahdollisista kriittisistä löydöistä, jotka vaativat välitöntä huomiota. Kun määrittelyt on suoritettu ja testauksesta ja sen vaiheista ja reunaehdoista on päästy yhteisymmärrykseen, itse testaus voi alkaa.

5.1.2 Tiedonkeruu

Tiedonkeruu kuuluu testauksen valmisteluun. Tiedonkeruun voi suorittaa joko julkisista lähteistä tiedustelemalla (Open Source Intelligence (OSINT)), käyttämällä erilaisia skannereita tai näitä molempia yhdistelemällä. Toimeksiantaja voi myös tarjota kaiken tiedon etukäteen, joten tiedonkeruuta ei tarvita ja pelkkä tiedon analysoiminen riittää.

OSINT-tiedonkeruuvaiheessa kohteesta kerätään ja analysoidaan mahdollisimman paljon tietoa julkisista lähteistä, joita voidaan hyödyntää penetraatiotestauksen aikana. Tällaisia lähteitä voivat olla esimerkiksi yrityksen nettisivut, hakukoneet ja sosiaalinen media. Tiedonkeruuvaiheessa tulee kuitenkin ottaa huomioon toimeksiantajan tavoitteet testaukselle ja näin ollen arvioida aina, onko kerätty tieto oleellista testauksen kannalta.

Tiedonkeruun voi suorittaa myös käyttämällä erilaisia skannereita. Skannerit voivat saada tietoonsa esimerkiksi koodikielet ja niiden versiot, IP-osoitteet, serveritiedot ja laitteiden aukinaiset portit.

5.1.3 Riskianalyysi

Tiedonkeruun jälkeen voidaan luoda riskianalyysi. Riskianalyysi on määrittelyn ja tiedonkeruuvaiheessa saadun datan perusteella luotu hyökkäysmetodi, jonka avulla saadaan selville esimerkiksi suurimmat haavoittuvuudet, mitä kautta hyökkäys olisi viisain suorittaa. Riskianalyysin aikana määritellään myös ensisijaiset ja toissijaiset kohteet.

5.1.4 Haavoittuvuuksien kartoittaminen

Kun toteuttamiskelpoisimmat hyökkäysmetodit on tunnistettu, on selvitettävä kohteen heikot kohdat ja haavoittuvuudet. Nämä haavoittuvuudet ovat voineet syntyä mistä tahansa kehitysvaiheessa tapahtuneesta virheestä tai puutteellisesta suunnittelusta. Haavoittuvuuksien löytämiseen käytetään yleensä erilaisia skannereita, joiden jälkeen haavoittuvuus voidaan tarkastaa vielä manuaalisesti false-positive-tulosten ehkäisemiseksi.

Koska penetraatiotestaaja pyrkii jäljittelemään oikeaa hyökkääjää mahdollisimman hyvin, on tässä vaiheessa oltava valppaana, ettei IDS- ja IPS-järjestelmät havaitse hyökkäystä. Yksi hyvä keino havaitsemisen ehkäisemiseksi on suorittaa testaus hyvin hitaasti ja pienissä osioissa [17].

5.1.5 Hyökkäys

Hyökkäysvaihe keskittyy yksinomaan löydettyjen haavoittuvuuksien hyväksikäyttöön, jotta testaaja pääsee järjestelmään sisälle ohittamalla turvallisuusrajoitukset ja saamaan käsiinsä organisaation arvokkaimmat tiedot ja kohteet.

Testaaja haluaa simuloida oikeaa hyökkäystä mahdollisimman tarkasti ja siksi pysyä myös näkymättömissä mahdollisimman pitkään, mieluiten koko testauksen ajan. Hyökkäysvaiheessa voi tulla vastaan erilaisia turvallisuusrajoituksia, mitkä voivat olla esimerkiksi virustentorjunnat, palomuurit, IDS- ja IPS-järjestelmät sekä HIDS-järjestelmät.

Valitettavan usein hyökkäysvaihe suoritetaan raakaa voimaa käyttäen mahdollisimman nopeasti ilman tarkkuutta ja varmistusta siitä, että haavoittuvuuksien hyödyntäminen onnistuu. Tällainen testaus ei kuitenkaan anna itse testaajalle tai toimeksiantajalle mitään arvokasta tietoa ja testauksesta ei välttämättä selviä suurimmat riskit ja hyväksikäyttökohteet [14]. Testaajan tulee siis valmistautua hyvin ennen hyökkäysvaiheen aloittamista.

5.1.6 Jälkihyökkäys

Jälkihyökkäyksen tarkoituksena on määrittää hyökkäyksen kohteena olevan koneen arvo ja säilyttää sen hallinta myöhempää käyttöä varten. Koneen arvo perustuu siihen, miten arvokasta tietoa se sisältää ja onnistuuko sillä myös entistä syvemmälle tunkeutuminen.

Tässä vaiheessa testaaja on siis sisällä järjestelmässä ja varsinainen datan keräys alkaa. Jälkihyökkäyksen aikana testaaja esimerkiksi kerää tietoa kohteena olevasta järjestelmästä, etsii mielenkiintoisia ja mahdollisesti arvokkaita tiedostoja sekä yrittää nostaa omia oikeuksiaan kohteessa [18]. Lisäksi testaaja voi esimerkiksi hajottaa salasanasuojaukset, joita voi hyödyntää muihin järjestelmiin.

Määrittelyvaiheessa on sovittu toimeksiantajan kanssa, miten kerättyä dataa käsitellään ja millaisia muutoksia järjestelmään on lupa tehdä. Kaikki kerätty data ja tehdyt muutokset dokumentoidaan tarkasti testausraporttiin.

5.1.7 Raportointi

Testausprosessin lopuksi laaditaan kattava raportti. Raportointi on tärkein penetraatiotestauksen vaihe ja siinä tulisi lukea selkeästi ja yksityiskohtaisesti, mitä testaustoimenpiteitä on tehty, miten testausvaiheet on suoritettu ja miten organisaation tulisi korjata penetraatiotestin aikana havaitut haavoittuvuudet. Raportin tulisi myös olla helposti ymmärrettävä, että myös muut kuin tietoturva-ammattilaiset pääsevät perille testauksen vaiheista ja tuloksista.

PTES käy kattavasti läpi, mitä kaikkea raportin olisi hyvä pitää sisällään. Myös monet penetraatiotestaustyökalut muodostavat automaattisesti raportin testauksen löydösten pohjalta.

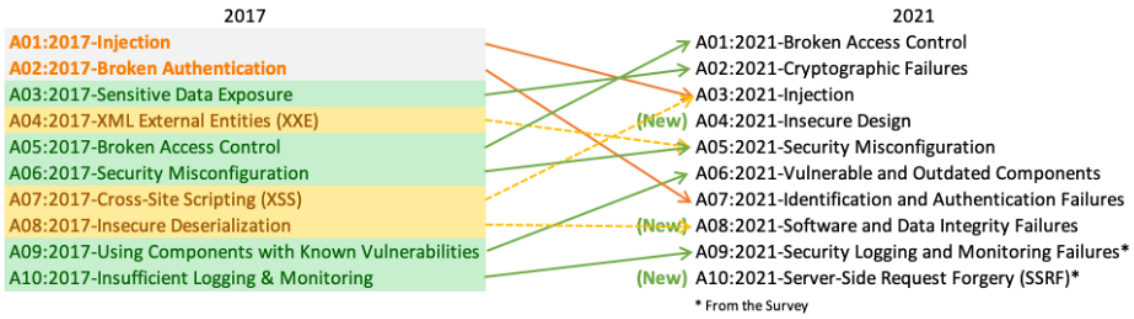
6 OWASP

OWASP® on voittoa tavoittelematon, yhteisövetoinen järjestö, jonka tavoitteena on parantaa eri organisaatioiden ohjelmistojen tietoturvallisuutta. OWASP ylläpitää monia avoimen lähdekoodin ohjelmistoprojekteja, jotka ovat organisaatioiden ja kehittäjien vapaassa käytössä parantaakseen ohjelmistojensa tietoturvaa [19]. Tässä opinnäytetyössä on hyödynnetty OWASP Top 10 -raporttia sekä OWASP Web Security Testing Guidea apuna penetraatiotestauksessa.

6.1 OWASP Top 10

OWASP Top 10 on useiden eri organisaatioiden turvallisuusasiantuntijoiden kokoama raportti, joka kokoaa yhteen kymmenen kriittisintä web-sovelluksiin liittyvää uhkaa ja haavoittuvuutta. Raportti on jokaisen kehittäjän vapaasti käytettävissä, ja OWASP-asiantuntijat ovat sanoneet sen ohjeiden noudattamisen olevan tehokkain askel kohti turvallisempaa ohjelmistokehityskulttuuria. [20.]

OWASP Top 10-luetteloa on ylläpidetty vuodesta 2003 lähtien ja viimeisin versio on julkaistu syksyllä 2021. Se sisälsi merkittäviä muutoksia vuoden 2017 julkaisuun, kuten alla olevasta kuvasta voi päätellä (kuva 3).



Kuva 3. Viimeisimmät muutokset OWASP Top 10 2017 ja 2021 versioiden välillä.

6.2 OWASP Top 10

Alle on listattuna vuoden 2021 OWASP Top 10 -raportin kriittisimmät web-sovelluksiin liittyvät haavoittuvuudet yleisyysjärjestyksessä [20]. Lisäksi listauksessa on avattu muutamalla sanalla, mitä haavoittuvuudet käytännössä tarkoittavat.

Haavoittuvuudet on vapaasti suomennettu ymmärrettävyyden parantamiseksi.

- A01:2021 – Broken Access Control (suom. rikkinäiset käyttöoikeudet). Rikkinäiset käyttöoikeudet liittyvät tunnistamattomiin tai tunnistettuihin käyttäjiin, jotka pääsevät käsiksi web-sovelluksen sisältöön, joihin heillä ei ole oikeutta. Toiminnot voivat olla esimerkiksi tietojen luvaton paljastaminen, poistaminen tai muuttaminen.

- A02:2021 – Cryptographic Failures (suom. salausvirheet). Salausvirheet syntyvät, kun arkaluonteista tietoa ei suojata säilytettäessä tai siirrettäessä riittävän hyvin. Riittämätön suojaus voi johtaa massiiviseen arkaluonteisten tietojen vuotamiseen hyökkäyksen yhteydessä.

- A03:2021 – Injection (suom. injektio). Injektio on haavoittuvuus, joka syntyy, kun hyökkääjä pääsee syöttämään epäluotettavaa tietoa sovellukseen. Hyvin yleinen injektio on esimerkiksi SQL-injektio, minkä avulla hyökkääjä voi päästä käsiksi osaan tai koko tietokantaan.

- A04:2021 – Insecure Desing (suom. epäluotettava suunnittelu). Epäluotettava suunnittelu on laaja haavoittuvuuskategoria, joka keskittyy yleisesti kaikkiin web-sovelluskehityksen elinkaaren suunnittelu- ja arkkitehtuurivirheisiin. Haavoittuvuus liittyy etenkin ohjelmiston riskienhallinnan puutteeseen.

- A05:2021 - Security Misconfiguration (suom. puutteelliset tietoturva-asetukset). Puutteelliset tietoturva-asetukset altistavat hyökkäyksille, jossa tietoturvahyökkäys on mahdollista suorittaa puutteellisten tai puuttuvien tietoturva-asetusten vuoksi. Haavoittuvuuden uhkaan kuuluu myös käyttöjärjestelmien päivittämättömyys.

- A06:2021 - Vulnerable and Outdated Components (suom. haavoittuvaiset ja vanhentuneet komponentit). Web-sovelluskehityksessä käytettävät, haavoittuvuuksille alttiit, komponentit voivat olla esimerkiksi vanhentuneita kirjastoja, lisäosia tai moduuleita.

Pahimmassa tapauksessa vanhentuneiden komponenttien hyväksikäyttö voi johtaa täydelliseen tietomurtoon.

- A07:2021 - Identification and Authentication Failures (suom. tunnistus- ja todennusvirheet). Tunnistus- ja todennusvirheet ilmenevät järjestelmän puutteellisena tai helposti kierrettävänä tunnistautumisena. Yleisimpinä hyökkäyskohteina voidaan pitää käyttäjän luomaa, liian heikkoa salasanaa.

- A08:2021 - Software and Data Integrity Failures (suom. eheysvirheet). Ohjelmistojen eheysvirheet liittyvät ohjelmistopäivityksiin tai epävarmaan CI/CD-järjestelmään varmistamatta ohjelmiston tietojen eheyttä. Tyypillisimmät virheet ovat sovellukseen ladatut laajennukset, kirjastot ja moduulit epäluotettavista lähteistä.

- A09:2021 - Security Logging and Monitoring Failures (suom. puutteellinen loki ja valvonta). Haavoittuvuus ilmenee, kun järjestelmän lokin kirjaukset ovat puutteellisia tai puuttuvat kokonaan. Ilman seurantaa ja lokikirjauksia ei hyökkäyksiä ja tietovuotoja pystytä havaitsemaan tarpeeksi ajoissa tai ollenkaan.

- A10:2021 - Server-Side Request Forgery (SSRF) (suom. palvelinpyynnön väärennös). Haavoittuvuuden ilmetessä hyökkääjät voivat päästä sovelluksen palomuurien takana oleviin suojattuihin järjestelmiin esimerkiksi skannauspalvelimien avulla.

6.3 OWASP Web Security Testing Guide

Turvallisuusalalla testauskriteereitä tai -käytänteitä ei ole tarkkaan määritelty, joten OWASP Testing Project on kehittänyt tietoturvatestaukseen tarkoitetun viitekehyksen. Web Security Testing Guide (WSTG) on web-sovelluskehittäjille sekä tietoturva-alan ammattilaisille kehitetty viitekehys, joka koostuu alan parhaista käytännöistä ja jota ohjelmistokehittäjät ja testaajat voivat hyödyntää web-sovelluksien turvallisuuden testaamiseen [3].

Oppaassa kuvataan yksityiskohtaisesti yleinen testausviitekehys, käytännön toteuttamiseen vaadittavat tekniikat sekä raportointikäytännöt.

7 Burp Suite

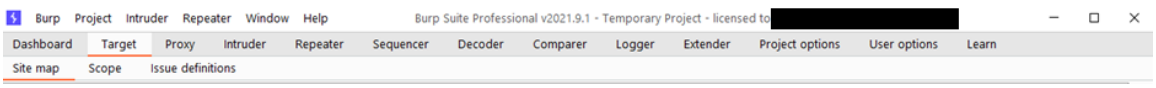
Burp Suite on Java-pohjainen työkalusarja, jota käytetään web-sovellusten turvallisuuden testaamiseen. Yksinkertaistettuna se toimii välityspalvelimena, joka kaappaa kaiken liikenteen käyttäjältä web-sovellukseen ja testaa niitä. Se on tietoturva-ammattilaisten yleisesti käyttämä työkalu, minkä avulla voidaan suorittaa automatisoituja tai manuaalisia tietoturvatestejä [21].

Burp Suite tukee kolmea eri versiota, ilmaista Burp Suite Community Editionia, Burp Suite Enterprise Editionia sekä Burp Suite Professionalia. Tässä opinnäytetyössä tehtävässä penetraatiotestauksessa on käytetty Burp Suite Professionalia ja opinnäytetyön kirjoitushetkellä se maksaa 349 € [21].

Ilmainen Community-versio on mainio työkalusarja tietoturvatestauksesta kiinnostuneelle harrastelijalle tai aloittelijalle, mutta se ei tue niitä työkaluja, mitä penetraatiotestauksessa olisi hyvä käyttää. Tästä syystä tähän opinnäytetyöhön valitsimme Professional-version.

7.1 Burp Suite Professionalin työkalut

Burp Suiten kehittäjäyrityksen PortSwiggerin mukaan, Burp Suite Professional on kehitetty tekemään monimutkaisten web-sovelluksen tietoturvatestaamisesta yksinkertaisempaa ja helpompaa. Web-sovelluksien teknologiat ja suunnittelutrendit uudistuvat vuosi vuodelta ja aloittelevien tietoturvatestaajien kynnys hypätä mukaan on suuri. Tätä varten Professional-versioon on luotu erilaisia työkaluja eri testausvaiheiden helpottamiseen ja nopeuttamiseen (kuva 4). [22.]



Kuva 4. Burp Suite Professionalin työkalupalkki.

Alle listattuna Professional-version ominaisuudet. Ominaisuuksien kuvaukset on haettu PortSwigger:n omilta verkkosivuilta [23]. Ominaisuuksia on avattu myös jonkin verran itse testausvaiheen ymmärrettävyyden parantamiseksi.

- Burp Target. Tämä työkalu sisältää yksityiskohtaisia tietoja kohdesovelluksistasi. Target-välilehdellä on myös mahdollista seurata ja muokata testausprosesseja.

- Burp Proxy. Proxy toimii web-välityspalvelimena selaimesi ja kohdesovellusten välillä ja antaa käyttäjän siepata, tarkastaa ja muokata molempiin suuntiin kulkevaa liikennettä. Proxya voidaan käyttää yhdessä joko Burpin oman sulautetun selaimen tai ulkoisen selaimen kanssa.

- Burp Crawl. Crawl (ent. Spider) on automaattinen indeksointirobotti, joka kartoittaa kaiken sisällön ja toiminnot, jotka se löytää kohdesovelluksesta.

- Burp Repeater. Repeater on työkalu yksittäisten HTTP- ja WebSocket-viestien manuaaliseen käsittelyyn ja uudelleenlähettämiseen sekä sovelluksen vastausten analysointiin.

- Burp Scanner. Scanner on haavoittuvuusskanneri, joka tarkastaa automaattisesti web-sovelluksen haavoittuvuudet. Määrityksistä riippuen skanneri voi indeksoida sovelluksen löytääkseen sen sisällön ja toiminnot ja tarkastaa sovelluksen löytääkseen haavoittuvuuksia.

- Burp Sequencer. Sequencer on työkalu, jolla analysoidaan satunnaisuutta tietokohteiden otannassa. Sitä voidaan käyttää testaamaan web-sovelluksen istuntotunnisteita tai muita tärkeitä tietokohteita, joiden on tarkoitus olla arvaamattomia, kuten anti-CSRF-tunnisteita tai salasanan palautustunnuksia.

- Burp Decoder. Decoder on työkalu, jota käytetään koodatun tiedon muuntamiseen sen todelliseen muotoon tai raakadatan muuntamiseen erilaisiin koodattuihin ja hajautettuihin muotoihin.

- Burp Comparer. Comparer on työkalu, jolla vertaillaan kahta tietokohdetta. Compareria voidaan käyttää esimerkiksi, kun verrataan erityyppisten käyttäjien luomia sivustokarttoja tai välityspalvelimen historiatietoja.

- Burp Intruder. Intruder on työkalu verkkosovelluksia vastaan kohdistettujen räätälöityjen hyökkäysten automatisointiin. Sitä voidaan käyttää automatisoimaan kaikenlaisia tehtäviä, joita voi syntyä testauksen aikana.

- Burp Logger. Logger on työkalu verkkotoiminnan tallentamiseen. Logger tallentaa kaiken Burp Suiten luoman HTTP-liikenteen tutkimista ja analysointia varten.

8 Penetraatiotestaus

Itse testauksen valmistelu voitiin aloittaa, kun tarvittava tiedonkeruuvaihe oli suoritettu. Ennen testausta testauksen valmisteluun kuului testauksen suunnittelu, mikä sisälsi myös kohteen rajauksen sekä asennuksen. Penetraatiotestaus perustuu suurelta osin testaajan omiin tietoihin ja resursseihin, joten koska tiedonkeruuvaihe sekä ohjelmiston ja kohdeympäristön asennukset kestivät suunniteltua kauemmin, jäi itse penetraatiotestaus suppeammaksi, kuin oli työn alussa suunniteltu.

8.1 Testaussuunnitelma

Testaussuunnitelma toteutettiin yhdessä toimeksiantajan kanssa ja käytiin läpi aikataulu, testauskohde, testauksen vaiheet sekä tavoitteet ja testauksessa käytettävät työkalut. Testaussuunnitelmassa otettiin huomioon se, että testausympäristö on paikallinen asennus, joka antaa testaajalle paljon vapauksia määritellä itse, milloin ja mihin testaukset suoritetaan. Mikäli kyseessä olisi ollut esimerkiksi julkisessa ympäristössä oleva kohde, olisi myös laadittu erillinen sopimus, jossa lukisi, että penetraatiotestaukseen on annettu lupa kohteen omistajalta ja testaus olisi tällöin laillinen toimeksianto. Lisäksi tällaisissa testauksissa myös esimerkiksi testauksen ajankohta ja reunaehdot olisi määritelty tarkemmin.

Testaussuunnitelmassa määriteltiin myös, minkälaista pohjatietoa vaaditaan ennen testauksen suorittamista ja nämä kirjattiin ylös. Penetraatiotestauksessa testaajalla olisi oltava mahdollisimman hyvä perusteorian tuntemus tietoturvatestaukseen, web-sovelluksien haavoittuvuuksiin ja testauskäytänteisiin sekä itse testaustyökaluun, joten työn ehdottomasti suurin osuus oli perusteorian, työkalujen ja standardien opiskelu.

Testaussuunnitelmassa päätettiin myös testauksessa käytettävät työkalut. Testaus suoritettiin Burp Suite Professionalin versiolla 2021.9.1 ja sen eri työkaluilla. Testauksessa hyödynnettiin suurilta osin automaattisia skannaustyökaluja, joiden tulokset varmistettiin manuaalisin keinoin. Burp Suiten valinta testauksessa käytettäväksi työkalusarjaksi oli alusta asti hyvin selkeä, sillä se on helppo oppia ja käyttää ensikertalaisena. Lisäksi internetistä löytyy paljon ohjeita sekä kursseja, missä neuvotaan Burp Suiten käyttöönotossa sekä testauksessa.

8.2 Kohde

Testattava kohde määriteltiin toimeksiantajan toimesta. Kohde kuuluu yrityksen tuoteperheeseen, joka on kokonaisuudessaan todella laaja ja käsittää useita eri moduuleita ja lisäosia. Tästä syystä kohde päätettiin rajata ainoastaan päätuotteeseen eli tuotteen runkoon.

Testattava kohde oli yrityksen ASP .Net alustalla toimiva web-sovellus, jonka tietolähteenä toimii SQL. Lisäksi sovellusasennuksessa hyödynnettiin myös IIS-teknologiaa (Internet Information Services), joka on Microsoftin kehittämä palvelinohjelmistokokonaisuus.

8.3 Kohteen asennus

Kohde, eli yrityksen päätuotteen runko, asennettiin toimeksiantajan määräämälle erilliselle tietokoneelle. Kohde asennettiin samoja asennuspaketteja käyttäen kuin mitä yrityksen asiakkaat käyttävät, jotta testausympäristö olisi mahdollisimman todenmukainen. Lisäksi testausympäristöön haettiin data erilliseltä demopalvelimelta varmuuskopioina.

Testausympäristön asennus paikallisesti mahdollisti sen, että testauksen rajaaminen oli helpompaa eikä tarvinnut huolehtia siitä, saisiko testaaja liian arkaluontoista tietoa ulos kohteesta, koska testausympäristöllä oli ainoastaan demoa varten luotua dataa.

Lisäksi testaus pystyttiin suorittamaan vapaammin huolehtimatta siitä, lisätäänkö, muokataanko tai poistetaanko dataa asiakas- tai demoympäristöistä. Mikäli tällainen olisi mahdollista testausympäristössä, tarkoittaisi se sitä, että myös asiakas- ja demoympäristöjen dataa olisi mahdollista muokata hyökkääjän toimesta.

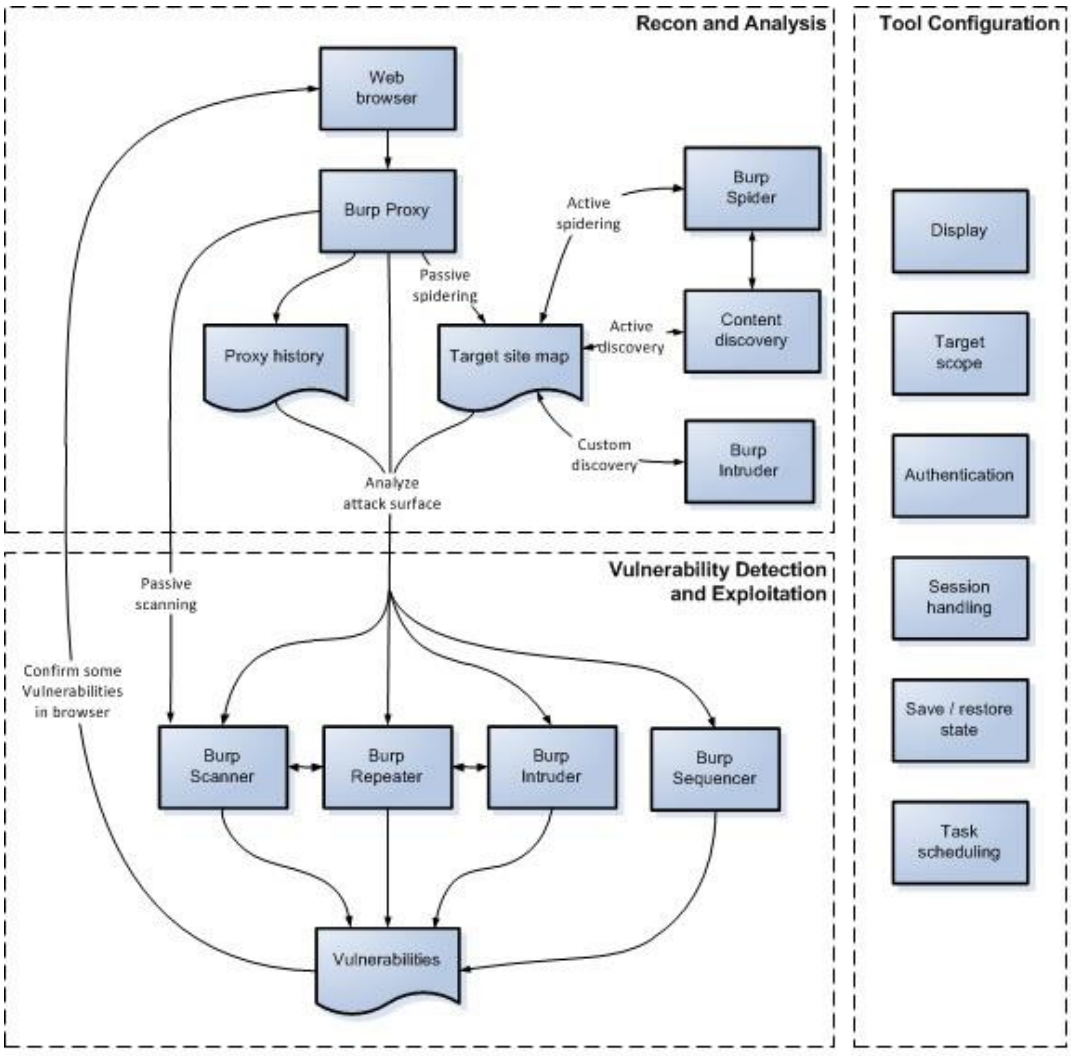
Kohteen asennusvaiheessa ilmeni monia erilaisia ongelmia muun muassa sertifikaattien ja IIS-palvelimien kanssa. Tästä johtuen asennusvaihe kesti suunniteltua kauemmin ja näin ollen itse testaukseen jäi hieman vähemmän aikaa.

8.4 Testaus

Kun työn tutkimuksellinen osuus oli suoritettu, voitiin itse testaus aloittaa. Ensimmäinen varsinainen testausvaihe käsitti testauksen valmistelun ja esityöt. Tähän vaiheeseen sisältyi tietojen

kerääminen kohteesta. Penetraatiotestaus suoritettiin white-box-lähtökohdasta, sillä kohde oli ennestään tuttu kehitys- sekä testaustyön kautta. Tämän lähtökohdan mukaan testauksessa oli mahdollista hyödyntää esimerkiksi sisäänkirjautumistietoja. Koska testaus suoritettiin white-box-lähtökohdista, ei varsinaista tiedonkeräysvaihetta tarvittu, vaan tähän osioon sisältyi hallussa oleviin tietoihin perusteellinen tutustuminen. Tässä vaiheessa oli siis tiedossa esimerkiksi koodikielet ja niiden versiot, servereiden tiedot ja mitä tietovarastoa kohteessa käytetään.

Koko testauksen ajan hyödynnettiin Burp Suiten kehittäjän PortSwigger:in luomia kattavia dokumentaatioita, joissa kuvailtiin penetraatiotestauksen työnkulkua (kuva 5).



Kuva 5. Yleiskatsaus penetraatiotestauksen tärkeimmistä vaiheista

Testauksessa hyödynnettiin Burpin omaa sulautettua selainta, mikä ei vaatinut erillisiä lisämäärityksiä tai -asetuksia. Testauksessa olisi ollut mahdollista käyttää myös erillistä ulkoista selainta,

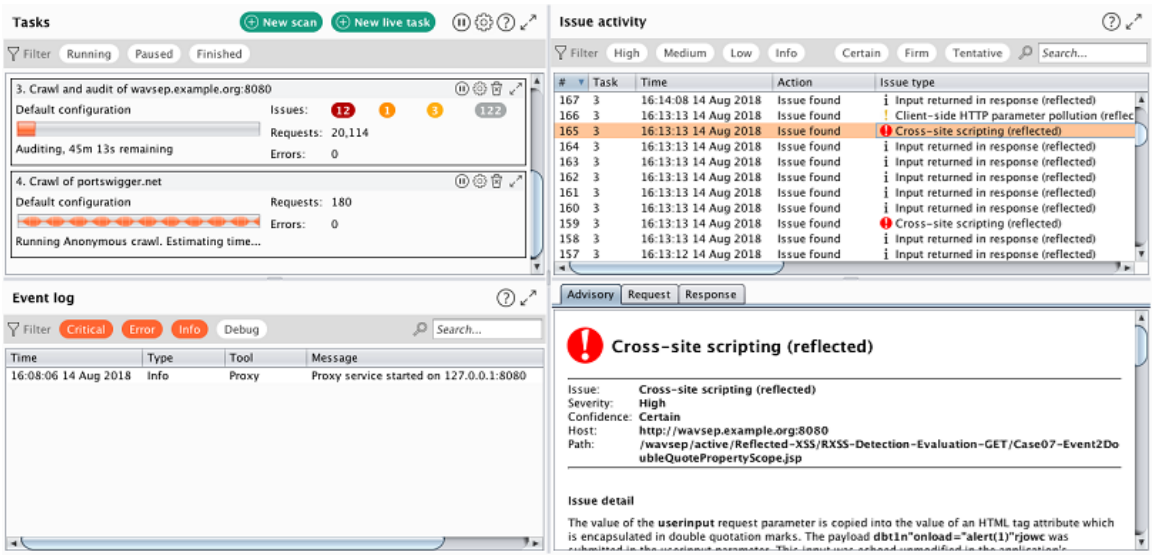
mutta tässä tapauksessa olisi pitänyt suorittaa erilaisia lisävaiheita määrittämään selain toimimaan Burpin kanssa. Lisäksi selaimeen olisi joutunut asentamaan Burpin oman CA-varmenteen.

Kun Burp oli käynnissä ja selain määritetty, voitiin alkaa suorittaa testejä kohteelle. Ensimmäisenä käynnistettiin Proxy-välilehdellä Intercept, eli sieppaus, jonka jälkeen siirryttiin kohteen URL-osoitteeseen. Sieppauksen ollessa päällä jokainen selaimen tekemä HTTP-pyyntö tallennetaan ja sitä voidaan tarvittaessa muokata (kuva 6). HTTP-historia välilehdellä nähtiin kaikki pyynnöt ja vastauksen, mitkä analysoitiin ja valittiin potentiaaliset kiinnostavat kohteet, mitä voitiin tarkastella muun muassa Decoderissa ja Repeaterissa.

Lisäksi testiä varten määritettiin Proxy-asetuksissa kaapattavaksi myös kuvatiedostoja, sillä tiedettiin kohteen sisältävän mahdollisesti arkaluontoisia kuvatiedostoja. Muuten asetukset pysyivät oletuksina.

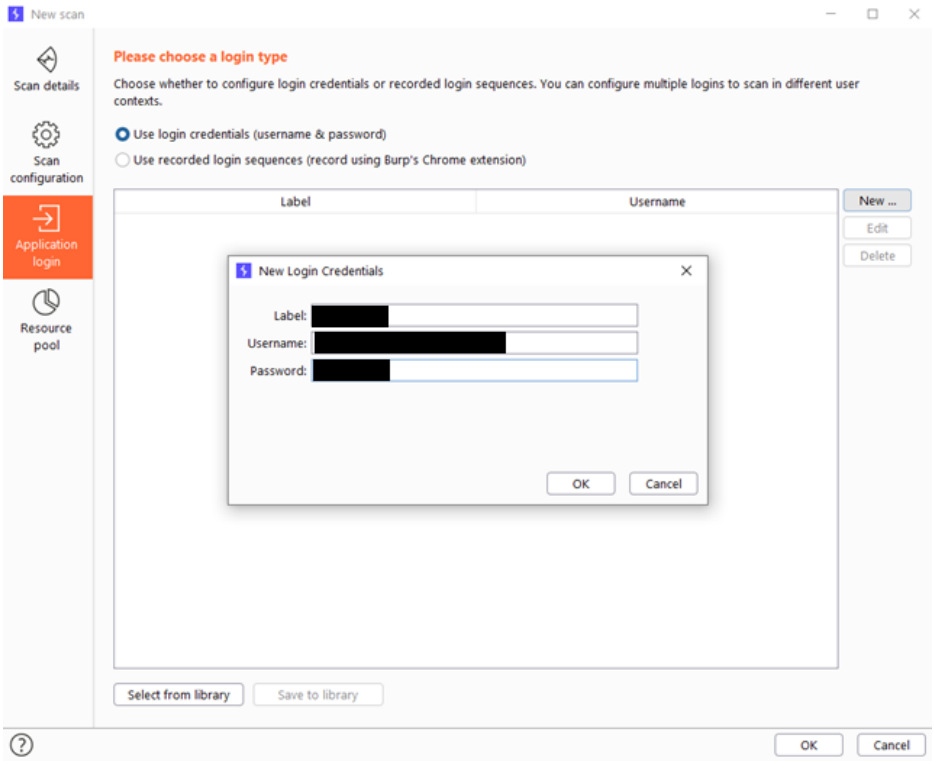
Kuva 6. Käynnissä oleva sieppaus

Kun kohde on käyty läpi perusteellisesti ja sen URL-osoitteet on tiedossa, aloitettiin skannaus. Skannauksen onnistui aloittamaan Dashboard-välilehden kautta (kuva 7). Kuva on haettu kehittäjän nettisivuilta ymmärrettävyyden ja yrityksen tietojen suojan parantamiseksi [24]. Dashboardilla pystyttiin seuraamaan jokaisen skannausvaiheen tapahtumia sekä löydöksiä samanaikaisesti.



Kuva 7. Burp Suiten Dashboard-välilehti (Stuttard D.)

Skannauksen ensimmäistä osiota kutsutaan nimellä Crawl ja tämän vaiheen aikana Burp kartoitti kohteen ja kävi läpi kaikki kohteessa sijaitsevat linkit ja lomakkeet. Näiden tulosten perusteella vastaukset jäsennetään ja pyritään löytämään vielä uusia linkkejä. Crawl-vaiheen asetuksissa määriteltiin myös sisäänkirjautumistiedot, mitkä jo tiedettiin ja Burp pystyi hyödyntämään näitä skannausvaiheessa (kuva 8). Muuten Crawl-vaiheen asetukset jätettiin oletusasetuksiksi.



Kuva 8. Sisäänkirjautumistietojen asetukset

Crawl-vaiheen aikana Burp loi sivustokartan kohteesta, minkä perusteella voitiin aloittaa varsinainen hyökkäysvaihe, eli Audit. Audit-vaiheen aikana Burp yrittää tunnistaa mahdolliset tietoturva-aukot tai indikaattorit, jotka voivat johtaa tietoturva-aukkojen tunnistamiseen. Aktiivisen audit-vaiheen aikana Burp lähettää ja muokkaa jatkuvasti pyyntöjä, joita se lähettää kohteelle, jotta löytäisi mahdollisimman monta haavoittuvuutta kohteesta.

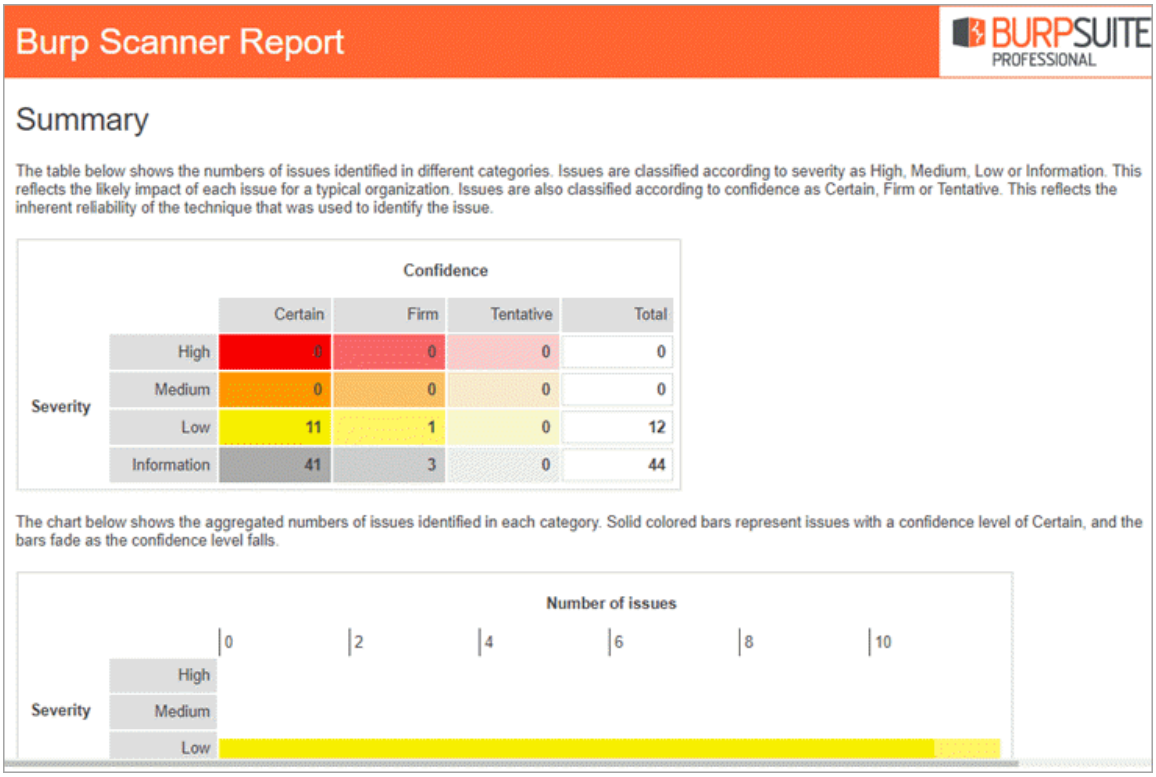
Aktiivisen hyökkäyksen rinnalla Burp suorittaa myös tauotta passiivista skannausta ja crawl-skannausta. Passiivisen skannauksen tarkoituksena on löytää haavoittuvuudet jo olemassa olevien pyyntöjen sisällön perusteella. Crawl-skannauksen tarkoituksena on päivittää ja kartoittaa sivustokarttaa sen perusteella, mitä aktiivinen hyökkäysskanneri löytää.

Kun jokainen skannausvaihe on suoritettu, analysoidaan skannauksen tulokset. Löydetyt haavoittuvuudet pyritään varmistamaan Burpin erilaisten välilehtien moduuleiden avulla. Tässä testauksessa käytettiin Repeater-, Intruder-, Sequencer- sekä Decoder-moduuleita varmistamaan skannauksen aikana ilmenneet löydökset sekä suorittamaan mahdollisten haavoittuvuuksien perusteella kohteen hyväksikäyttö.

Skanneri tallensi kaikki löydökset Issue activity -välilehdelle, josta pystyttiin kokoamaan kattava raportti tuloksista. Burp ehdottaa löydösten perusteella haavoittuvuusluokitusta jokaiselle löydökselle, jotka ovat high, medium, low, information ja false-positive. Haavoittuvuusluokituksia pystyi muokkaamaan sen mukaan, oliko tietty haavoittuvuus esimerkiksi haitallisempi tietylle kohteelle. Tässä vaiheessa pystyttiin myös merkitsemään haavoittuvuusluokitukseen, mikäli löydettiin false-positive tuloksia.

8.5 Raportointi

Raportoinnissa on käytetty apuna Burpin omaa raportointityökalua (kuva 9) [25]. Raportti tallennettiin HTML-muotoon, jotta sitä pystyi tarkastelemaan selaimen kautta. Raportissa kuvaillaan selkeästi haavoittuvuudet sekä niiden vakiokuvaukset ja tiedot, miten haavoittuvuuden voi toistaa, korjausehdotukset sekä haavoittuvuusluokitukset.



Burp Scanner Report

BURPSUITE PROFESSIONAL

Summary

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low or Information. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

Severity	Confidence			
	Certain	Firm	Tentative	Total
High	0	0	0	0
Medium	0	0	0	0
Low	11	1	0	12
Information	41	3	0	44

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.

Kuva 9. Havainnollistava kuva Burpin luomasta haavoittuvuusraportista. Ei alkuperäinen raportti.

Koska testausjakso kesti alle viikon, lähetettiin toimeksiantajalle yksi raportti, mikä sisälsi yhteenvedon koko penetraatiotestauksesta. Jos testaus olisi kestänyt kauemmin, olisi toimeksiantajan kanssa voinut sopia esimerkiksi viikoittaisista raportoinneista tai katselmoinneista.

8.6 Penetraatiotestauksen automatisointi

Työn yhtenä tarkoituksena oli myös selvittää, olisiko penetraatiotestauksen automatisointi toimeksiantajan kohteelle mahdollista. Itse penetraatiotestauksen täydellinen automatisointi olisi mahdotonta, sillä penetraatiotestauksessa todennetaan aina löydökset manuaalisin keinoin ja hyödynnetään näitä haavoittuvuuksia hyökkäykseen. Kuitenkin automatisoitu tietoturva- tai haavoittuvuustestaus olisi mahdollista ja todennäköisesti kannattavaa suorittaa haavoittuvuusskannereilla.

Kohteen automatisoitu skannaus Burp Suitella kattaa OWASP:n 10 kriittisintä haavoittuvuutta ja sen pystyisi suorittamaan kohtuullisella työmäärällä. Se olisi huomattavasti nopeampi prosessi kuin kokonaisvaltainen penetraatiotestaus manuaalisilla tarkistuksilla ja vaatisi testaajalta vä-

hemmän töitä. Automatisoidun testauksen luominen Burp Suitella olisi yksinkertaista ja vaatisi todennäköisesti ainoastaan ensimmäisellä kerralla isomman työmäärän. Testauksen asetukset ja kohde valittaisiin kerran, jonka jälkeen kaikki manuaalisesti muokatut tiedot pystyttäisiin tallentamaan ja näin ollen testaus voitaisiin suorittaa nopeasti uudelleen tallennettuja tiedostoja hyödyntäen. Lisäksi jos kohteeseen tulee suuria päivityksiä tai koontiversioita, tallennettuja asetuksia olisi helppo muokata manuaalisesti vastaamaan kohteen uusien ominaisuuksien tarpeisiin.

Automatisoidun testauksen suurin puute olisi vaillinaiset testaustulokset. Vaikka automatisoidut testit kattaisivatkin OWASP:n kriittisimmät haavoittuvuudet, ei niillä saada testattua kaikkia haavoittuvuuksia, jotka manuaalisen testauksen yhteydessä mahdollisesti havaittaisiin. Tämän lisäksi osa ilmoitetuista haavoittuvuuksista olisi väistämättä false-positive-tuloksia ja testaaja joutuisi käymään läpi suuren määrän testituloksia. Ratkaisu tähän voisi olla jonkinlainen hybridiratkaisu. Yrityksen olisi mahdollista ottaa tuotteelle käyttöön esimerkiksi jokaisen koontiversion jälkeen automaattinen haavoittuvuustestaus ja sen lisäksi suorittaa kerran tai kahdesti vuodessa täydellinen penetraatiotesti.

9 Yhteenveto

Penetraatiotestaus on toiminto, jossa testaaja hyödyntää samoja toimintatapoja kuin pahaaikeiset hakkerit ja yrittää tunkeutua toimeksiantajansa web-sovellukseen käyttäen hyväksi löytämiään haavoittuvuuksia ja saadakseen ulos arkaluontoisia tietoja. Tämän opinnäytetyön tavoitteena oli paneutua penetraatiotestaukseen ja suorittaa toimeksiantajana toimineen yrityksen tuotteelle penetraatiotestaus käyttäen työkaluna Burp Suite Professionalia ja sen lisäosia.

Työ alkoi tutustumalla tietoturvallisuuden ja tietoturvatestauksen perusteoriaan ja pyrittiin ymmärtämään, mitä testauksella on tarkoitus löytää, millaisia eri testausmetodeja on olemassa, mitkä standardit ja oppaat määrittelemät tietoturvatestausta ja miten penetraatiotestaus eroaa muista testausmetodeista. Itse tiedonkeruuvaihe onnistui mutkattomasti, sillä tietoturvallisuudesta ja tietoturvatestauksesta on olemassa paljon materiaalia sekä suomeksi että englanniksi.

Työn alussa huomattiin selkeästi se, miten paljon tietotaitoa ja kokemusta penetraatiotestaajalta vaaditaan testausta suorittaessaan, jotta tulokset olisivat mahdollisimman hyvät ja laaja-alaiset. Tästä johtuen tämän työn kahteen ensimmäiseen vaiheeseen meni suunniteltua enemmän aikaa, joka jouduttiin supistamaan työn kolmannesta vaiheesta eli itse testausvaiheesta. Näin kuitenkin päätettiin, jotta testaustulokset mitä tullaan havaitsemaan, ovat itsessään mahdollisimman laadukkaita ja luotettavia.

Penetraatiotestauksen voi jakaa kolmivaiheiseksi prosessiksi, jossa ensimmäinen vaihe sisältää kaiken tarvittavan tiedonhaun ja informaation keräyksen, toinen vaihe sisältää aktiivisen testauksen ja kolmas vaihe pitää sisällään tulosten analysoinnin ja raportoinnin. Kuten itse aktiivisessa testausvaiheessa huomattiin, tulevat automaattiset työkalut ja erilaiset haavoittuvuusskannerit toimimaan nyt ja tulevaisuudessa suurena apuna itse testauksen aikana. Automaattisista työkaluista on hyötyä etenkin tiedonkeruuvaiheessa, tunnetuimpien tietoturvahaavoittuvuuksien havaitsemisessa ja suuren datamäärän lähettämisessä kohteelle.

Vaikka automatisoidut testausprosessit ovat yleistyneet, ohjaa manuaalinen testaus kuitenkin silti vielä tänäkin päivänä penetraatiotestausprosessia. Testaajan on aina varmistettava testauksen aikana löydetyt haavoittuvuudet. Jäljelle jää myös paljon haavoittuvuuksia, mitä automaattiset skannerit eivät löydä tai ne havaitsevat ne heikosti, jolloin manuaalisen testauksen tärkeys korostuu entisestään.

Lähteet

1. Liikenne- ja viestintävirasto, kyberturvallisuuskeskus. 2018. *Turvallinen tuotekehitys: kohti hyväksyntää.* Haettu 1.10.2021 osoitteesta https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/Turvallinen_tuotekehitys_Suomi_J003_2018.pdf

2. Liikenne- ja viestintävirasto, kyberturvallisuuskeskus. 2019. *Luottamuksen lähteillä.* Haettu 2.10.2021 osoitteesta https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/Luottamuksen_lahteilla.pdf

3. Open Web Application Security Project (OWASP). 2020. *OWASP Web Security Testing Guide version 4.2.* Haettu 4.10.2021 osoitteesta https://owasp.org/www-project-web-security-testing-guide/

4. Helsingin seudun kauppakamari. 2020. *Yritysten rikosturvallisuus 2020.* Haettu 21.10.2021 osoitteesta https://helsinki.chamber.fi/wp-content/uploads/2020/12/yritysten-rikosturvallisuus-2020.pdf

5. Liikenne- ja viestintävirasto, kyberturvallisuuskeskus. 2021. *Näin suojaudut tietomurroilta.* Haettu 2.10.2021 osoitteesta https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/ohjeet-ja-oppaat/nain-suojaudut-tietomurroilta

6. Reinicke C. 2021. *The biggest cybersecurity risk to US businesses is employee negligence, study says.* Haettu 2.10.2021 osoitteesta https://www.cnbc.com/2018/06/21/the-biggest-cybersecurity-risk-to-us-businesses-is-employee-negligence-study-says.html

7. ISO (10/2013). ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements. Haettu 27.9.2021 osoitteesta https://www.iso.org/standard/54534.html

8. ISO/IEC 27002:2013 *Information technology — Security techniques — Code of practice for information security controls.* Haettu 2.10.2021 osoitteesta https://www.iso.org/standard/54533.html

9. ISO/IEC 27701:2019. *Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines.* Haettu 3.10.2021 osoitteesta https://www.iso.org/standard/71670.html

10. Kali Linux dokumentaatio. haettu 14.10.2021 osoitteesta https://www.kali.org/docs/introduction/

11. OWASP ZAP dokumentaatio. Haettu 22.10.2021 osoitteesta https://www.zaproxy.org/getting-started/

12. Liikenne- ja viestintävirasto, kyberturvallisuuskeskus. 2019. *"Palveluistanne löytyi tietoturva-aukko".* Haettu 5.10.2021 osoitteesta https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/palveluistanne-loytyi-tietoturva-aukko

13. Palmer C. C. 2001. *Ethical hacking*. IBM Systems Journal, 40(3), 769-780. doi:10.1147/sj.403.0769

14. Kennedy, D., O'Groman, J., Kearns, D. & Aharoni, M. 2011. *Metasploit. The Penetration Tester's Guide*. San Francisco: No Starch Press, Inc. 2011

15. Penetration Testing Execution Standard dokumentaatio. 2014. Haettu 13.10.2021 osoitteesta http://www.pentest-standard.org/index.php/Main_Page

16. Törhönen V. 2012. *Web-sovelluksen penetraatiotestaus.* Tutkielma. Haettu 28.9.2021 osoitteesta https://wiki.eduuni.fi/display/tuttietoturva/Web-sovellusten+penetraatiotestaus

17. Puhakka J. 2012. *PENETRAATIOTESTAUS OSANA TIETOTURVAN TOTEUTUSTA.* Jyväskylä: Jyväskylän ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö.

18. Weidman G. 2014. *Penetration Testing: A Hands-On Introduction to Hacking.* San Francisco: No Starch Press, Inc. 2014

19. The Open Web Application Security Project (OWASP) omat verkkosivut. Haettu 12.9.2021 osoitteesta https://owasp.org

20. The Open Web Application Security Project (OWASP). 2021. *OWASP Top 10 – 2021.* Haettu 12.10.2021 osoitteesta https://owasp.org/Top10/

21. PortSwigger:n omat verkkosivut. Haettu 13.10.2021 osoitteesta https://portswigger.net/burp/pro

22. Atkinson M. 2021. *Burp Suite Professional: feature roundup.* Haettu 20.10.2021 osoitteesta https://portswigger.net/blog/burp-suite-professional-feature-roundup

23. Portswigger. 2021. *Burp Suite tools-dokumentaatio.* Haettu 14.10.2021 osoitteesta https://portswigger.net/burp/documentation/desktop/tools

24. Stuttard D. 2018. *The new dashboard.* Haettu 10.11.2021 osoitteesta https://portswigger.net/blog/the-new-dashboard

25. Tuntematon kirjoittaja, Software Testing Help. 2021. *How To Use Burp Suite For Web Application Security Testing.* Haettu 10.11.2021 osoitteesta https://www.softwaretestinghelp.com/how-to-use-burp-suite/#Burp_Suite_Report_format