Penetraatiotestaus

– Case Fiarone Oy

Juan Laasonen

2019 Laurea

Laurea-ammattikorkeakoulu

Penetraatiotestaus – Case Fiarone Oy

Juan Laasonen
Tietojenkäsittely
Opinnäytetyö
Marraskuu, 2019

Laurea-ammattikorkeakoulu
Tietojenkäsittely

Tiivistelmä

Juan Laasonen

Penetraatiotestaus - Case Fiarone Oy

Vuosi 2019 Sivumäärä 34

Tämä opinnäytetyö käsittelee penetraatiotestausta ja sen vaiheita. Työn toimeksiantajana toimi turkulainen tietoturvayritys Fiarone Oy, joka tarjoaa tietoturvapalveluita toisille yrityksille. Työssä suoritettiin penetraatiotestaus yrityksen määräämälle kohteelle.

Työn tavoite on tunnistaa ja etsiä kohteesta löytyvät haavoittuvuudet testaamalla. Testaus on suunniteltu noudattamaan OWASP TOP 10-raportin suosituksia. Testaus suoritetiin käyttämällä tunnettuja työkaluja ja hyväksyttyjä tekniikoita.

Työn tietoperustassa avataan laajemmin penetraatio- ja tietoturvatestausta sekä kerrotaan, mitä vaiheita testaamiseen kuuluu. Työssä esitellään tärkeimpiä testauksessa käytettyjä työkaluja. Lisäksi käydään läpi OWASP-raportin kohdat, joihin testaaminen perustui.

Testauksen päätteeksi käytiin palautekeskustelu yrityksen edustajien kanssa, kun he olivat käyneet löydetyt haavoittuvuudet läpi. Keskustelussa purettiin testauksessa saadut tulokset ja yrityksen omat havainnot. Keskustelun jälkeen testicase suljettiin.

Asiasanat: penetraatiotestaus, haavoittuvuus, tietoturvatestaus

Laurea University of Applied Sciences
Degree Programme in Business Information Technology
Bachelor's Thesis

Abstract

Juan Laasonen

Penetration Testing – A Case Study of Fiarone Oy

Year 2019 Pages 34

The purpose of this thesis is to explain penetration testing and all of the phases that it requires. The thesis is commissioned by and the testing is conducted for an information security company based in Turku, called Fiarone Oy. Fiarone Oy provides cyber security solutions for other businesses in Finland.

The objective of this thesis is to perform a penetration test to a specific target set by the commissioner. The testing is planned and performed following the OWASP TOP 10 report and suggestions. Testing is performed using well known tools and methods.

The goal of the testing is to find vulnerabilities in the target system. Once a vulnerability is found, it is to be confirmed and reported to the commissioner.

A feedback session was held with the company at the end of the test case, when they had verified the findings that had been reported to them.

Keywords: penetration testing, vulnerability, information security

Sisällys

1 Johdanto 6
2 Työn tarkoitus ja tavoitteet 6
2.1 Suunnittelu 8
2.2 Toteutus 8
2.3 Raportointi Fiaronelle 9
3 Tietoturva ja tietosuoja 9
4 Tietoturvatestaus 10
5 Penetraatiotestaus 13
5.1 Kohteen rajaus 15
5.2 Testauksen vaiheet 15
5.2.1 Testaussuunnitelma 16
5.2.2 Tietojen hankkiminen 16
5.2.3 Testaus 17
5.2.4 Raportointi 17
5.3 Testauksen tarkoitus 17
6 Työkalut 18
6.1 Kali Linux 18
6.2 Oracle virtual box 19
6.3 Burp Suite 20
6.4 Internetin hakukoneet 22
6.5 Owasp Zap 23
6.6 Muita käytettyjä työkaluja 24
7 OWASP 24
7.1 Injection 27
7.2 Broken Authentication 28
7.3 Sensitive Data Exposure 28
7.4 XLM External Entities 28
7.5 Broken Acces Control 29
7.6 Security Misconfiguration 29
7.7 Cross Site Scripting 30
7.8 Insecure Deserialization 30
7.9 Using Components with Known Vulnerabilities 30
7.10 Insufficient Logging and Monitoring 30
8 Tulokset 30
9 Yhteenveto 31

1 Johdanto

Tämän tutkimuksellisen opinnäytetyön aiheena on suorittaa penetraatiotestaus yhteistyöyrityksen määräämälle kohteelle. Yhteistyöyrityksenä tässä opinnäytetyössä toimii tietoturvayritys Fiarone Oy. Fiaronella on toimipisteet Turussa ja Espoossa. Yritys tarjoaa asiakkailleen tietoturvakonsultointitestausta ja SOC-palveluita.

Tietoturvatestaus on tärkeä osa uuden palvelun, ohjelmiston tai alustan käyttöönottoa. Testaus mahdollistaa haavoittuvuuksien löytymisen jo kehitysvaiheessa ja se nopeuttaa niiden paikkaamista.

Työn testaus suunniteltiin ja toteutettiin käyttäen viimeisintä OWASP TOP 10-haavoittuvuusraporttia (OWASP TOP 10, 2017). Testausjakson aikana kohteelle suoritettiin testejä, joissa testattiin OWASP-raportin listaamia haavoittuvuuksia. Raportti listaa kymmenen sen hetkistä pahinta haavoittuvuutta.

Testeillä pyrittiin selvittämään esiintyykö kohteessa raportin listaamia haavoittuvuuksia. Testaamiseen käytettiin automatisoituja työkaluja, sekä manuaalisia keinoja tulosten varmistamiseen.

Työn tutkimuksellisessa osassa käytetyt työkalut sekä haavoittuvuudet on pyritty avaamaan kattavasti omissa luvuissaan. Selvyyden vuoksi tässä tekstissä käytetään testattavasta kohteesta termiä kohde.

2 Työn tarkoitus ja tavoitteet

Työn testausvaiheen tarkoituksena oli selvittää tarkastettavan kohteen tietoturvan tila ja informoida yhteistyöyritystä mahdollisesti löydetyistä haavoittuvuuksista. Tiedon pohjalta mahdollisiin ongelmakohtiin pystytään puuttumaan jo kehitysvaiheessa. Työ keskittyi seuraavaan kolmeen eri osa-alueeseen.

Ensiksi selvitettiin kohteen haavoittuvuudet käyttämällä saatavilla olevia työkaluja. Kaikki työkalut olivat avoimen lähdekoodin ohjelmia tai ilmaisversioita. Työkalut hyväksytettiin toimeksiantajalla ennen niiden käyttöä.

Toiseksi haavoittuvuuksien löytymisen varmistamiseksi suoritettiin vaadittavat toimenpiteet. Testauksessa käytettiin sekä automaattisia työkaluja että manuaalisia keinoja tulosten varmistamiseksi. Varmistaminen tapahtui toistamalla manuaalisesti automaattisen testin tulos.

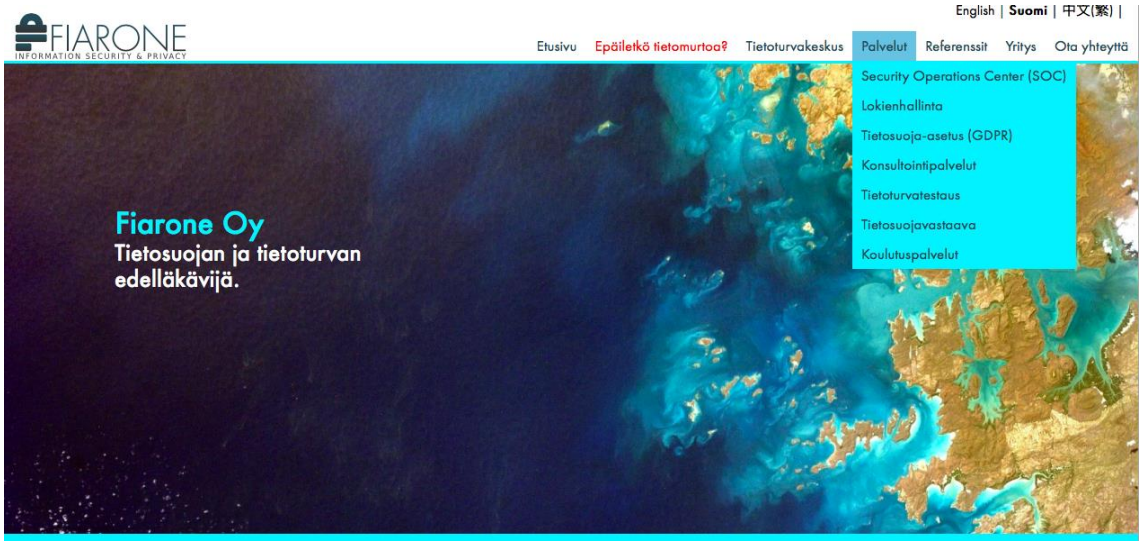
Kolmanneksi testeistä ja tuloksista laadittiin raportti, jossa tulokset esiteltiin toimeksiantajalle. Raportista ilmeni testauksen vaiheet ja selitettiin mitä työkalua oli käytetty. Raporttiin myös kirjattiin miten saadut tulokset olisi mahdollista toistaa.

Työn tutkimuksellisessa osuudessa suoritettiin penetraatiotestaus yhteistyöyrityksen määräämälle kohteelle. Testaus suoritettiin seuraamalla OWASP TOP 10-raportin suosituksia. Raportin suosittelemat kohdat pyrittiin testaamaan. Mahdollisista haavoittuvuuksista tehtiin selvitys yritykselle.

Testauksen tarkoitus oli tuottaa toimeksiantajalle realistinen kuva tarkastettavan kohteen mahdollisista haavoittuvuuksista. Haavoittuvuus on esimerkiksi virhe ohjelmistossa, joka mahdollistaa tietojen muokkaamisen tai niiden vuotamisen ulkopuoliselle osapuolelle.

Testaus suoritettiin käyttämällä automaattisia työkaluja. Tulokset pyrittiin varmistamaan manuaalisin keinoin. Manuaalisella vaiheella pyrittiin poistamaan False Positive-tulokset. False Positive-tuloksia saadaan, kun testauksessa käytetään automaattisia työkaluja. False Positive on ilmoitus haavoittuvuudesta, mitä todellisuudessa ei ole olemassa. Testauksen tuloksista laadittiin lopuksi raportti.

Toimeksiantajana tässä työssä toimi turkulainen tietoturvayritys Fiarone Oy. Fiarone Oy on vuonna 2010 perustettu konsultointiyritys, joka tarjoaa tietoturvapalveluita muille yrityksille. Sen palveluihin kuuluu tietoturvatestaus, GDPR-tarkastukset, tietosuojavastaavan palvelut, tietoturvakoulutusta, lokienhallintaa ja SOC-palveluita.



Kuvio 1: Fiarone Oy

SOC eli Security Operations Center on palvelu, jossa yritys valvoo asiakkaan tietoturvaa seuraamalla sen verkkoliikennettä. Yritys raportoi kaikista mahdollisista uhkista, joita asiakkaaseen saattaa kohdistua. Tieto hyökkäyksestä tulee Fiaronen valvomoon, joka päivystää

kaikkina vuorokauden aikoina, vuoden jokaisena päivänä. Kun mahdollinen uhka havaitaan siitä ilmoitetaan asiakkaalle ja ryhdytään mahdollisiin toimiin uhan pysäyttämiseksi.

Tietosuojavastaava on palvelu, jossa pienet ja keskisuuret yritykset, voivat hankkia tietosuojavastaavan ulkoistettuna palveluna. Tämän lisäksi Fiarone Oy suorittaa tietoturvatestausta yritysten palveluihin, GDPR-tarkastuksia ja koulutusta.

2.1 Suunnittelu

Testausvaihe aloitettiin laatimalla testaussuunnitelma. Suunnitelmassa annettiin toimeksiantajalle tieto testauksen vaiheista ja aikataulusta. Suunnitelma käytiin läpi toimeksiantajan kanssa yhdessä. Suunnitelmasta ilmeni testauksen aikataulu, mahdolliset työkalut ja miten testaus olisi tarkoitus toteuttaa.

Testauksessa pyrittiin käymään läpi OWASP Top 10-tietoturvaraportin kaikki kohdat. Raportti sisältää kymmenen haitallisinta haavoittuvuutta. Raportti ilmestyy muutaman vuoden välien. Tässä opinnäytetyössä on käytetty vuonna 2017 ilmestynyttä versiota, joka on tällä hetkellä viimeisin. Julkaisussa olevat haavoittuvuudet käydään läpi myöhemmin tässä raportissa.

Työssä hyödynnettiin myös OWASP-järjestön julkaisemaa testausohjeistusta, jonka on tarkoitus tukea testaamista yrityksissä. Testausohjeistusta käyttämällä pienet ja keskisuuret yritykset voisivat suorittaa itse tietoturvatestausta palveluihinsa.

Mahdolliset löydökset raportoitiin toimeksiantajalle. Raporttiin pyrittiin kirjaamaan haavoittuvuuden sijainti, miten haavoittuvuus löytyi ja ohjeet, miten testin pystyisi toistamaan. Mahdolliset seuraukset jos haavoittuvuutta ei korjata, ja miten sitä voidaan hyödyntää haitallisiin tarkoituksiin

2.2 Toteutus

Toteutus aloitettiin, kun yhteistyöyritys oli saanut testiympäristön valmiiksi. Yritys tarjosi tarvittavat osoitteet ja turvallisen pääsyn testiympäristöön. Pääsy mahdollistettiin VPN-yhteydellä, ja yritys antoi yhteyden muodostamiseen tarvittavat sertifikaatit ja avaimet.

Työssä toteutettu testaus suoritettiin Black Box- ja White Box-tapojen kombinaatiolla. Testaajalla ei ollut tietoa lähdekoodista tai käytetyistä versioista, mutta ennakkotietoa oli annettu testauksen helpottamiseksi. Tätä metodia kutsutaan nimellä Grey Box.

Testaus suoritettiin seuraamalla laadittua suunnitelmaa ja sen aikataulua. Työn alussa testaaja tutustui kohteeseen ja sen toimintoihin manuaalisesti. Tätä alustavaa tietojen keräämistä täydennettiin käyttämällä Burp Suite-ohjelmaa taustalla. Ohjelma suoritti taustalla omaa skannausta kohteen rakenteesta. Kartoituksen jälkeen pyrittiin tunnistamaa kohteen mahdolliset haavoittuvat kohdat.

Alustavan tutkimuksen jälkeen kohteeseen ajettiin testejä. Testeissä käytettiin automatisoituja työkaluja. Automatisoidut työkalut lyhensivät testausaikaa huomattavasti. Ne kävivät läpi määrätyt testit käyttäen sisään rakennettuja kirjastoja tai käyttäjän valitsemia käskylistoja.

Testausaika vaihteli minuutin ja tunnin välillä, riippuen testistä ja ohjelmasta. Automaattisten testien tulosten perusteella suoritettiin manuaalinen testaus. Manuaalinen testaus oli erityisen tärkeä, sillä löydetyt haavoittuvuudet piti pystyä toistamaan ennen niiden raportointia. Tällä tavalla poistetaan mahdolliset False Positive-tulokset.

2.3 Raportointi Fiaronelle

Raportointi oli tarkoitus suorittaa sitä mukaan kuin todennettuja haavoittuvuuksia löytyi. Lopulta kuitenkin päädyttiin menetelmään, jossa yritykselle laadittiin ainoastaan loppuraportti. Tämä johtui manuaalisen todentamisen hitaudesta.

Manuaalisella todentamisella karsittiin pois väärät hälytykset tuloksista. Joissain tapauksissa manuaalinen testin varmistus ei onnistunut ja raporttiin kirjattiin ainoastaan automaattisten testien tulokset.

3 Tietoturva ja tietosuoja

Tietoturva ja tietosuoja pyrkivät parantamaan organisaation tietopääoman turvallisuutta. Tietoturvan tavoite on turvata tiedon koskemattomuus, eheys ja saatavuus. Tämä tarve on korostunut tiedon muuttuessa digitaaliseen muotoon. (Limnell, Majewski & Salminen, Kyberturvallisuus 2014, 55)

Tietosuojan tavoite on varmistaa henkilötietojen turvallinen käsittely. Tietosuoja on jokaiselle henkilölle kuuluva perusoikeus ja henkilötietoja on käsiteltävä lainmukaisesti, kohtuullisesti ja läpinäkyvästi. (Korpisaari, Pitkänen & Warma-Lehtinen, Uusi tietosuojalainsäädäntö 2018, 34, 89)

Dataa käsittelevät yritykset ja organisaatiot ovat vastuussa hallussaan olevan datan käsittelystä, varastoimisesta ja keräämisestä. Tietoturvaa ja tietosuojaa arvioitaessa on hyvä olla tietoinen seuraavista seikoista:

Kohde eli mitä turvataan. Kohde voi tarkoittaa yrityksen koko tietoverkkoa tai jotain yksittäistä palvelua. Kohde voi myös olla jokin tietty tiedosto. Uhka eli miltä turvataan. Uhka tarkoittaa mahdollisen riskin aiheuttajaa. Tämä voi olla yritysvakooja, rikollinen, harraste hakkeri tai oma työntekijä. Toteutuessaan uhka voi aiheuttaa yritykselle merkittävää vahinkoa. Keino eli miten turvataan. Keino on tapa, jolla kohde turvataan. tämä käsittää kaikki tekniset ja fyysiset turvakeinot, joilla kohdetta pyritään turvaamaan. (Limnell ym. 2014, 37-38)

4 Tietoturvatestaus

Tietoturvatestaus on tärkeä vaihe, joka tulisi suorittaa ennen jokaisen sovelluksen, sivuston tai palvelun käyttöönottoa. Etenkin jos kyseinen palvelu käsittelee tai varastoi arkaluontoista dataa. Arkaluontoisena datana voidaan pitää terveystietoja, henkilötietoja sekä taloudellisia tietoja.

Tietotekniikan nopea kehittyminen ja palveluiden helppous on saanut monet organisaatiot omaksumaan sähköisen tiedonsiirron ja varastoinnin osana toimintaansa. Monet yritykset ovatkin siirtyneet pilvipalveluiden käyttäjiksi. Tämä osaltaan lisää tietoturvan tarvetta.

Testauksessa selvitetään teknisten tietoturvatoimien toiminnan taso. Tekniset toimet ovat esimerkiksi tietoliikenteen valvonta, oikeuksien hallinta ja luvattoman pääsyn esto (Korpisaari ym. 2018, 273). Testaus tulisi suorittaa myös aina, jos kohteeseen tehdään päivityksiä. Päivitykset saattavat luoda uusia haavoittuvuuksia palveluun.

Tietoturvatestaaminen ei ole kerran tehtävä toimi, vaan se tulisi toistaa vähintään kerran vuodessa. Testaus antaa aina selvityksen kohteen sen hetkisestä tilasta, mutta ei tulevaisuudesta. Testaamisen tarkoitus on suojata yrityksen tietojärjestelmissä oleva data ja varmistaa tietojärjestelmien toiminta. (Järvinen, Kyberuhkia ja somesotaa, Digiaikana sinäkin olet etulinjassa, 2018)

Tietoturva kehittyy koko ajan ja uusia haavoittuvuuksia tulee lisää jatkuvasti. Tietoturva ja turvallisuus ovatkin riskien hallintaa. Riskejä ei voi välttää, ne tulee arvioida ja tulosten perusteella yrityksen tulee kohdistaa voimavaransa oikeisiin kohteisiin. (Rousku, Kyberturvaopas, Tietoturvaa kotona ja työpaikalla 2014, 61)

Tietojen turvallinen käsittely on jokaisen palvelun tarjoajan vastuulla. Euroopan Unioni pyrki yhtenäistämään tietosuojaa jäsenmaissaan laatimalla GDPR-asetuksen. GDPR-tulee sanoista General Data Protection Regulation ja se otettiin käyttöön EU:n alueella täysimittaisena 25.05.2018 (Korpisaari ym. 2018, 1). Tämä asetus edellyttää organisaatioita ja yrityksiä huolehtimaan tietojen turvallisesta käsittelystä ja säilytyksestä. GDPR-asetus on askel oikeaan suuntaan tietosuojan parantamisessa.

Asetuksen velvoitteista ei kuitenkaan ole hyötyä, jos tietoja käsittelevät ohjelmat ovat haavoittuvaisia. Tämä korostuu etenkin kun monet yritykset käyttävät kolmannen osapuolen palveluita omassa liiketoiminnassaan. GDPR-asetus velvoittaa yritykset sakon uhalla varmistamaan tietosuojan riittävän tason. Sakko on määrätty GDPR-asetuksen artiklan 83 mukaan. Se on suuruudeltaan joko 10 miljoonaa tai 2 % vuotuisesta liikevaihdosta, tai 20 miljoonaa tai 5 % vuotuisesta liikevaihdosta. Määrä riippuu tapahtuman vakavuudesta ja siitä, onko yritys menetellyt tilanteessa oikein ennen ja jälkeen mahdollisen tietomurron (General Data Protection

Regulation (GDPR)). Suurilla yrityksillä tämä voi johtaa huomattaviin taloudellisiin menetyksiin. Asetus koskee kaikkia Euroopassa toimivia yrityksiä.

Tietoturvan kannalta ongelmalliseksi muodostuu myös käyttäjien toiminta. Monet käyttäjät, jättävät reagoimatta ja raportoimatta mahdollisia uhkia. Syy on joskus tietämättömyys, mutta useammin se johtuu asenteesta. Useat ihmiset ajattelevat jonkun muun kuitenkin hoitavan ilmoittamisen (Järvinen & Rousku, Työpaikantietoturvaopas, tunnista uhat, hallitse riskit 2017, 43). Joskus kyse on tiedon puutteesta ja riittämättömästä koulutuksesta. Koulutuksen tilannetta on pyritty parantamaan yrityksissä järjestämällä henkilöstölle tietoturvakoulutusta. Monet yritykset ovat myös esimerkiksi alkaneet vaatia uusia työntekijöitä suorittamaan pienen GDPR-koulutuksen aloittaessaan työt.

Monet yritykset myös päätyvät ratkaisuun, jossa tietoturvauhkiin suhtaudutaan siirtämällä mahdollinen riski muualle. Tämä tapahtuu hankkimalla vakuutuksen, joka kattaa mahdollisen taloudellisen tappion riskin realisoituessa. Tämä toimintatapa ei ole käyttäjien kannalta hyvä, koska heidän tietonsa vuotavat rikollisille. Vakuutus kuitenkin lieventää yrityksen kokemaa taloudellista iskua.

Riskiä ei voi kuitenkaan koskaan täysin poistaa. Se sisältyy aina toimintaan ja sitä pitää pystyä hallitsemaan riskitietoisuuden avulla. Riskejä voidaan selvittää tekemällä riskiarvioita ja uhkamalleja. Tietoturvaa suunniteltaessa pitää aina olla tietoinen uhasta ja riskeistä sekä mahdollisista haavoittuvuuksista järjestelmässä. (Limnell ym. 2014, 105-110)

Tietoja havittelevia tahoja on useita. Kohde, johon hyökätään usein määrittää myös hyökkääjän. Tavalliset kansalaiset joutuvat tavallisten rikollisten kohteiksi. Yritykset ovat usein haktivistien tai yritysvakoojien kohteena ja valtio on muiden valtioiden vakoilun kohteena. Myös kyberterrorismi on kehittymässä todelliseksi uhkaksi jokaisen maan kansalliselle turvallisuudelle. Supon vuosikirjassa (2017,22) kerrotaan, että Suojelupoliisin tietoon tuli vuonna 2017 useita tapauksia, joissa yrityksiin kohdistuneiden verkkohyökkäysten takana oli valtio. Suomen energiasektori ja teknologiayritykset ovat olleet kohteina. Myös tuotekehitystä tekevät yritykset ovat joutuneet uhreiksi. Tämä lisäksi monet hakkerit ovat ihmisiä, jotka vain haluavat kokeilla hakkerointia, ja tekevät sitä huvin vuoksi. Osa näistä ihmisistä ei edes välttämättä tiedosta rikkovansa lakia.

Tietoturvatestauksella saadaan suurin hyöty kun pyritään selvittämään mahdolliset haavoittuvuudet jo palveluiden, ohjelmien ja sivustojen kehitysvaiheessa. Tämä mahdollistaa uhkien minimoimisen ennen käyttöönottoa ja näin pienentää mahdollista hyökkäyspinta-alaa sekä mahdollisten hyökkääjien määrää. Jälkeenpäin suoritettava testaus auttaa havaitsemaan löytämättä jääneet haavoittuvuudet ja mahdolliset uudet haavoittuvuudet, joita päivitykset ja teknologian kehitys ovat synnyttäneet.

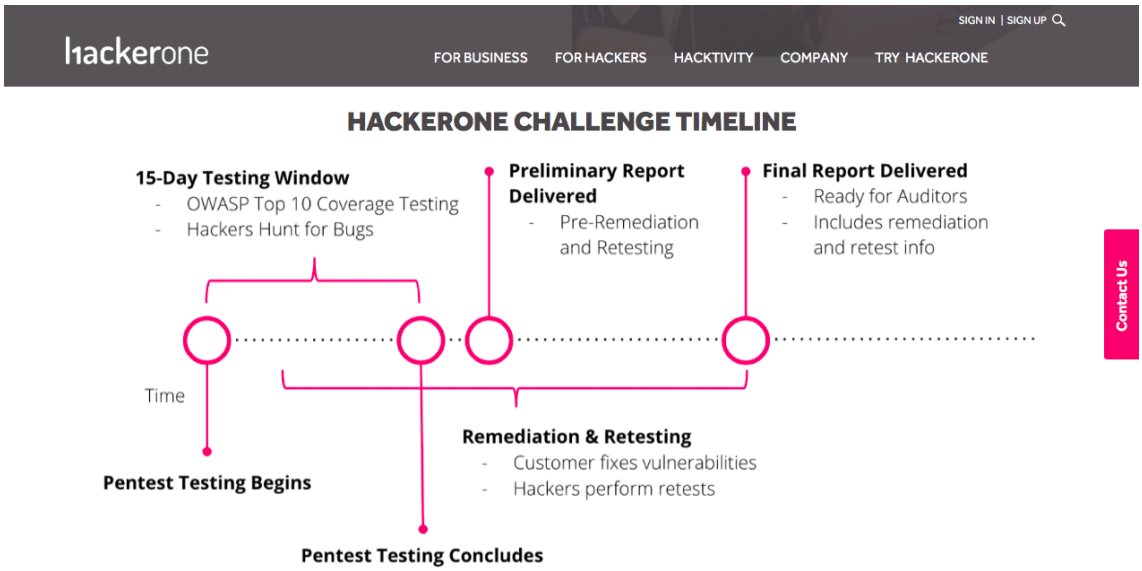
Tietotekniikka kehittyy jatkuvasti ja ohjelmistojen kehittyessä kehittyvät myös taktiikat suorittaa tietoturvamurtoja. Mikään sivusto tai ohjelma ei ole 100 % turvallinen. FBI entinen johtaja Robert Mueller totesi jo vuonna 2012 RSA kyberturvallisuuskonferenssissa pitämässään puheessa, että maailmassa on kahdenlaisia yrityksiä. Yrityksiä, jotka on hakkeroitu ja yrityksiä, jotka tullaan hakkeroimaan.

Tietoturvatestausta voidaan toteuttaa muutamalla eri tavalla. Testauksen voi toteuttaa esimerkiksi auditointina tai penetraatiotestauksena. Auditointimenetelmä tarkoittaa kohteen analysointia automaattisilla työkaluilla, jotka käyvät mahdolliset kohteessa olevat haavoittuvuudet läpi.

Auditoinnissa testaaja ei kuitenkaan käytä manuaalisia keinoja tulosten varmistamiseen. Auditointi on suppea keino kartoittaa kohteen tietoturvan tilanne. Menetelmä paljastaa kohteesta haavoittuvuuksia, mutta niitä jää myös havaitsematta. Auditoinnissa käytettävä työkalu saattaa ilmoittaa haavoittuvuudesta, jota ei oikeasti ole kohteessa. Tästä käytetään aiemmin mainittua termiä False Positive. Auditointi suoritetaan ajamalla kohteeseen testejä automaattisella työkalulla, joka skannaa kohteen ja laatii listan löytämistään haavoittuvuuksista. Auditoinnin päätteeksi laaditaan tuloksista raportti.

Vaihtoehtona voidaan suorittaa penetraatiotestaus. Penetraatiotestauksessa kohteeseen ajetaan samanlainen skannaus, mutta saadut tulokset varmistetaan vielä manuaalisesti. Tämä poistaa False Positive-tulokset ja näin saadaan paljon tarkempi kuva kohteen tietoturvan tasosta ja mahdollisista haavoittuvuuksista.

Penetraatiotestaus voi olla tarkempi, mutta siitä voi koitua yritykselle katkoksia toiminnassa riippuen testattavasta kohteesta. Testaaminen voidaan myös suorittaa Red Team-testauksena, joka on laajempi kokonaisuus kuin penetraatiotestaus. Penetraatiotestauksessa kohde on yleensä tietoverkko tai tietopankki. Red Team-hyökkäyksissä voidaan käyttää tietokoneiden lisäksi fyysisiä keinoja tavoitteiden saavuttamiseen. Red Team-testauksella voidaan testata esimerkiksi asiakkaan kykyä huomata mahdolliset tunkeutumisyritykset, reagointia näihin hälytyksiin ja työntekijöiden kykyä huomata tietojenkalasteluhyökkäys tai kulunvalvonnan toimivuus etuovelta serverihuoneeseen.



Kuvio 2: Hackerone, Bug Bounty-aikajana

Jotkut yritykset ovat hyödyntäneet Bug Bounty-ohjelmia, joissa kohteet annetaan testattavaksi tietylle ryhmälle hakkereita. Löydetyistä haavoittuvuuksista maksetaan mahdollisesti korvaus riippuen löydetyn haavoittuvuuden synnyttämästä uhkasta. Korvaukset liikkuvat maineen ja kymmenien tuhansien välillä.

Bug Bounty-ohjelmat toimivat niin, että sivustot, kuten Hackerone, ilmoittavat rekisteröityneille käyttäjilleen kohteista ja halukkaat voivat pyrkiä löytämään haavoittuvuuksia. Tämä on vaihtoehtoinen tapa, jota voidaan hyödyntää testaukseen.

Tämä lähestymistapa ei ole niin perusteellinen, kuin tietoturvayrityksen suorittama testaus. Yritykset tekevät perusteellista työtä tarkan suunnitelman mukaan. Itsenäisesti testaavat harrastelija, testaavat niitä asioita mitä itse haluavat tai osaavat. Näin ollen moni haavoittuvuus voi jäädä paljastumatta. Tästä huolimatta Bug Bounty-ohjelmat ovat suosittu keino paljastaa mahdollisia haavoittuvuuksia. Yrityksen, kuten Twitter, Steam, Microsoft ja Apple ovat hyödyntäneet bug bounty-metodia. Suomessa ainakin Lähitapiola on hyödyntänyt tätä palvelua.

5 Penetraatiotestaus

Penetraatiotestaus on tietoturvatestauksen muoto, jossa pyritään perusteellisesti selvittämään kohteen tietoturvan tilanne. Testausprosessi voi olla lyhyt tai pitkä riippuen kohteen koosta ja testattavien ominaisuuksien määrästä.

Testi-Case noudattaa aina ennalta määritettyä kaavaa. Ensiksi pidetään aloitustapaaminen, johon osallistuu henkilöitä molemmista yrityksistä. Tapaamisessa määritetään testauksen laajuus ja luovutetaan testausta varten tarpeellinen informaatio. Tämä voi käsittää kohteessa

käytetyt koodikielet, niiden versiot, servereiden tiedot tai muita tarpeelliseksi määritettyjä tietoja. Vähintään testaajalle luovutetaan tarvittava IP-osoite ja mahdollisesti testaukseen luodut tunnukset.

Tämän jälkeen laaditaan ja käydään läpi testaussuunnitelma. Suunnitelmassa ilmenee testauksen aikataulu, käytettävät tekniikat ja työkalut sekä tavoitteet. Tämä tapaaminen on äärimmäisen tärkeä, sillä siinä määritellään testauksen rajat. Testaaja itse ei saa päättää mitä hän lähtee testaamaan.

Testausta voidaan suorittaa joko White Box- tai Black Box-lähtökohdasta. White Box tarkoittaa, että testaajalle on valmiiksi annettu kaikki kohdetta koskeva hyödyllinen tieto. Black Box on tilanne, jossa testaaja ei tiedä kohteesta muuta kuin mahdollisen IP-osoitteen, tai ei edes sitä. Loput tiedot pitää selvittää itse. Näiden kahden metodin yhdistelmää kutsutaan Grey Box-metodiksi, jossa testaajalla on hallussaan jotain tietoa kohteesta mutta ei kaikkea.

Penetraatiotestaus on perusteellinen keino selvittää mahdolliset haavoittuvuudet, ja pyrkiä korjaamaan ne ennen kuin niitä pystytään hyödyntämään vääriin tarkoituksiin. Testaus on kuitenkin vain selvitys sen hetkisestä tilasta. Testaaminen pitäisi toistaa vähintään kerran vuodessa. Testauksen jälkeen asiakkaalla on raportti oman tietoturvansa tasosta ja mahdolliset suositukset parannuksista mitä tulisi tehdä.

Penetraatiotestauksessa kohde on rajattu asiakkaan toimesta, eikä testaaja saa lähteä tekemään testauksia rajauksen ulkopuolelle. On tärkeää, että testaaja ymmärtää tämän rajauksen ja kunnioittaa sitä. Tietoturva-alalla luottamus on merkittävässä asemassa tarjouskilpailuissa.

Penetraatiotestauksen idea on, että testaaja pyrkii saamaan pääsyn tietoihin, joita hänen ei pitäisi saada. Testauksessa simuloidaan oikeaa tietoturvahyökkäystä ja pyritään toimimaan niin kuin oikea rikollinen.

Testaaja saattaa onnistua saamaan testauksen aikana haltuunsa arkaluontoista tietoa, mistä johtuen tietoturvayritykset tekevät usein näitä toimeksiantoja tekeville uusille työntekijöilleen turvallisuusselvityksen. Myös toimeksiantaja voi vaatia turvallisuusselvitystä tulevista testaajista.

Testaajan tulee aina noudattaa maan lakia, eikä testauksen aikana saa käyttää keinoja, jotka voidaan tulkita lainvastaisiksi. Testaajaa voi esimerkiksi käyttää kalasteluviestejä tunnusten selvittämiseen, mutta ei saa uhata asiakkaan henkilökuntaa fyysisellä väkivallalla saadakseen tunnukset. Lain tunteminen ja ymmärtäminen on tärkeää testaajan työssä.

Testaamiseen voi käyttää Windows tai macOS-käyttöjärjestelmiä, sillä niihin on kehitetty tietoturvatestauksessa käytettäviä työkaluja. Useimmat alan ihmisistä ja harrastelijat käyttävät kuitenkin jotain Linux-distribuutiota testaamiseen. Koska raportointi on tärkeässä asemassa,

ovat monet päätyneet ratkaisuun, jossa päivittäiseen työntekoon käytetään PC- tai MAC-konetta ja itse testaamiseen käytetään Linuxia.

Kali Linux-distribuutio on yksi suosituimmista testauksessa käytetyistä käyttöjärjestelmistä. Sen rinnalle on tullut myös muita vaihtoehtoja, kuten Black Arch ja Parrot OS. Monet työkalut kuten Burp Suite, toimivat myös Windowsilla. Tästä huolimatta Linux-jakelun käyttäminen testaamiseen on vakiintunut alan standardiksi. Tämä johtuu siitä, että useimmat työkalut tehdään Linuxille suoraan.

Työkaluja on myös huomattavasti enemmän Linuxille kuin muille käyttöjärjestelmille. Esimerkiksi Kali-jakelupakettiin on esiasennettu noin 600 työkalua testaamiseen ja Black Arch-sisältää valmiiksi jo melkein 2000 erilaista työkalua testaukseen. Työkaluja myös kehitetään jatkuvasti lisää ja monet niistä tulevat ainoastaan Linux-jakeluihin.

5.1 Kohteen rajaus

Testattava kohde määritellään asiakkaan toimesta. Asiakas kertoo, mitä testataan ja mihin asioihin ei saa koskea. Kohde voi sisältää pelkästään yhden uuden lisätyn ominaisuuden tai se voi käsittää koko palvelun. Rajauksella voidaan säästää aikaa ja rahaa. On turha lähteä testaamaan uudestaan jo aiemmin testattua asiaa, jos siihen ei ole tehty muutoksia. On myös turhaa testata osa-alueita, jotka ovat poistumassa. Laajempi testausalue vie enemmän aikaa ja resursseja. Tämä lisää testauksen kustannuksia.

Rajaus tehdään ensimmäisellä tapaamiskerralla. Rajaamisella voidaan esimerkiksi määrittää, että testaaja saa selvittää injektiohyökkäyksen mahdollisuuden. Testaaja ei kuitenkaan saa viedä hyökkäystä niin pitkälle, että saisi arkaluontoista tietoa ulos kohteesta. Testaajan on tärkeää tietää testauksen parametrit ja noudattaa sovittuja sääntöjä. Tietoturvatestauksessa luottamus asiakkaan ja yrityksen välillä on määrittävä tekijä yrityksen menestyksen kannalta.

Rajaus on myös tärkeä testausta suorittavan yrityksen kannalta. Noudattamalla ennalta määritettyjä rajoja, testaaja testaa sen mistä asiakas maksaa. Jos testaaja sattuisi löytämään jotain muutan testauksen aikana, ei asiakkaalla ole velvollisuutta tästä maksaa, ellei sopimuksessa niin erikseen olla sovittu. Tämä ei kuitenkaan tarkoita sitä, että jos varsinaisen testauksen aikana löydetään muita haavoittuvuuksia, jotka eivät ole testausalueella, niistä ei raportoida.

5.2 Testauksen vaiheet

Testaustapaukset suunnitellaan tarkkaan. Jokainen testitapaus on uniikki, ja näin ollen valmista pohjaa ei löydy. Testauksen voi kuitenkin suunnitella seuraamaan OWASP TOP 10-raportin kohtia, jolloin tätä voi käyttää aina pohjana suunnitelmalle. OWASP on laatinut yrityksille

suunnatun testausmanuaalin, jota voi käyttää apuna testauksen suorittamiseen. Tätä ohjekirjaa käytettiin tämän työn tutkimuksellisessa osassa.

5.2.1 Testaussuunnitelma

Suunnitelma laaditaan ja esitellään asiakkaalle ennen testauksen aloittamista. Suunnitelma sisältää arvion aikataulusta. Asiakkaan kanssa sovitaan aina etukäteen aloitus- ja päättymispäivät testaukselle. Käytettävät työkalut ja se, miten testauksessa edetään, ilmoitetaan suunnitelmassa. Suunnitelmasta käy myös ilmi, ketkä henkilöt yrityksestä osallistuvat testaamiseen. Suunnitelmassa on tiedot testattavista asioista, joista on sovittu yhdessä asiakkaan kanssa.

5.2.2 Tietojen hankkiminen

Kun suunnitelma on hyväksytty asiakkaan toimesta, aloitetaan varsinainen testauksen valmistelu. Tämä tarkoittaa tietojen keräämistä kohteesta. Jos asiakas on tarjonnut nämä tiedot jo valmiiksi, niihin tutustutaan tarkoin. Muussa tapauksessa testaajan pitää hankkia kaikkia tarvittavat tiedot itse.

Tiedot voivat käsittää koodikielten versiot ja mitä kieliä on käytetty, IP-osoitteet, Servereiden tiedot tai mitä tietovarannon versiota käytetään kohteessa. Nämä tiedot auttavat testaajaa rajaamaan hyökkäysaluetta ja käytettäviä keinoja tehokkaasti. Tietojen kerääminen helpottaa testaajan työtä valittaessa työkaluja ja metodeja. On esimerkiksi turhaa testata Linux-pohjaista serveriä Windows-serverille kehitetyillä testeillä.

Open Source Intelligence (OSINT) tarkoittaa hyödyllisen tiedon hankkimista julkisesti saatavilla olevista lähteistä. Näitä lähteitä voivat olla tavalliset hakukoneet, kuten Google tai Bing. Myös sosiaalista mediaa voi hyödyntää OSINTin tekemiseen.

OSINT-vaiheessa voidaan käyttää esimerkiksi googlen hakukonetta. Haulla voidaan saada selville kohteen IP-osoitteet ja mitä muita komponentteja kohteessa on käytetty. Haulla voidaan selvittää henkilöstön tietoja ja käyttää näitä tietoja kalasteluhyökkäyksen (Phishing Attack) toteutukseen.

OSINTIN-tekemistä varten on kehitetty paljon työkaluja, jotka hankkivat hyödyllistä tietoa eri lähteistä. Tätä dataa yhdistämällä testaaja voi saada kattavan kuvan kohteesta ja hyödyllistä tietoa hyökkäysten suunnitteluun. Tämä vaihe voi kestää pidempään kuin itse testaus tai hyökkäys. Siinä missä hyökkäys voi kestää muutaman minuutin, voi tiedon keräämiseen mennä viikkoja.

5.2.3 Testaus

Kun tiedonkeruu vaihe on saatu valmiiksi ja testausta varten tarvittavat tiedot on saatu hankittua, testaus voidaan aloittaa. Testauksessa suorittamiseen ei ole universaalisti hyväksyttyä oikeaa tapaa, vaan jokainen testaaja tekee asiat omalla tavallaan. Tässä tekstissä esiintyvät vaiheet on kirjoitettu niin kuin ne suoritettiin tätä testausta toteutettaessa.

Aluksi kohde käydään läpi perusteellisesti. Tämä tarkoittaa kaikkien toimintojen kokeilua ja tutkimista. Samalla kohteelle myös ajetaan skannaus, jolloin automaattinen työkalu käy läpi kohteen ja ilmoittaa, jos se löytää mahdollisia piilotettuja ominaisuuksia. Ohjelman ilmoittamat tulokset kirjataan ylös, jotta niitä voidaan hyödyntää myöhemmin.

Kun kohteesta on tunnistettu mahdollisesti haavoittuvat alueet, ajetaan niihin erilaisia työkaluja käyttämällä automaattisia testejä. Automaattisia työkaluja käytetään ajan säästämiseen. Automaattinen työkalu pystyy suorittamaan satoja erilaisia kokeiluja ja esimerkiksi salasanojen murtaminen olisi mahdotonta ilman automaattista työkalua.

Kun tulokset näistä automaattisista testeistä on saatu, ne pitää vielä varmistaa. Varmistaminen tapahtuu toistamalla testit manuaalisin keinoin. Manuaalinen todentaminen poistaa False Positive-tulokset ja tulos on tällöin luotettavampi.

5.2.4 Raportointi

Riippuen testausjaksosta ja asiakkaan toiveista. Raportointi suoritetaan jokaisen löydetyn haavoittuvuuden kohdalla erikseen, tai koko testausjaksosta laaditaan yksi loppuraportti. Raportissa käydään läpi löydetyt haavoittuvuudet, miten tulos on saatu ja miten sen voi toistaa, sekä mahdolliset keinot korjata ongelma.

Raportointitapa riippuu asiakkaan toiveista. Tähän vaikuttaa myös testausjakson pituus. Jos testijakso kestää viikon, on turha lähettää asiakkaalle useampaa raporttia. Pidemmissä projekteissa voidaan toimittaa esimerkiksi viikkoraportti, jolloin asiakas saa tiedon viikon aikana tehdyistä toimista.

5.3 Testauksen tarkoitus

Penetraatiotestauksen ja auditoinnin tarkoitus on löytää kohteista mahdollisia haavoittuvuuksia ennen kuin niitä otetaan laajamittaiseen käyttöön. Testauksen ansiosta löydetyt ongelmat on helpompi korjata nopeasti jo kehitysvaiheessa. Paljastamalla yleisimmät haavoittuvuudet ja korjaamalla ne pienennetään hyökkäysmahdollisuuksia. Tämä puolestaan pienentää mahdollisien hyökkääjien määrää.

Testauksen voi suorittaa jo palvelun tai kohteen kehitysvaiheessa tai kun tuote on valmis. Jos kyseessä on tuotekehitysprojekti, testauksen kustannukset nousevat projektin elinkaaren

mukana. Alkuvaiheessa se on halvimmillaan ja loppuvaiheessa, kun projekti on lähes valmis se on kalleimmillaan.

Jos kyse on valmiista tuotteesta tai palvelusta, testaus olisi hyödyllinen tehdä kerran vuodessa, jotta voidaan varmistua tietoturvan tilasta. Testaustulos on aina tilannekuva sen hetken tilasta.

Testauksen kannattavuus saadaan tekemällä riskianalyysi, jossa käy ilmi arvio taloudellisista menetyksistä mahdollisen hyökkäyksen sattuessa. Riskianalyysin tai uhka-arvion avulla voidaan verrata testauksesta koituvat kustannukset mahdolliseen taloudelliseen menetykseen. Useimmiten testaus tulee yritykselle halvemmaksi vaihtoehdoksi.

Testauksen tavoite on pienentää yrityksen mahdollisia tietoturvauhkia ja antaa perusteellinen selvitys sen hetkisestä tilasta. Jokainen yritys ja organisaatio joutuu elämään riskin kanssa ja ainut keino pienentää tätä riskiä, on pyrkiä hallitsemaan sitä.

6 Työkalut

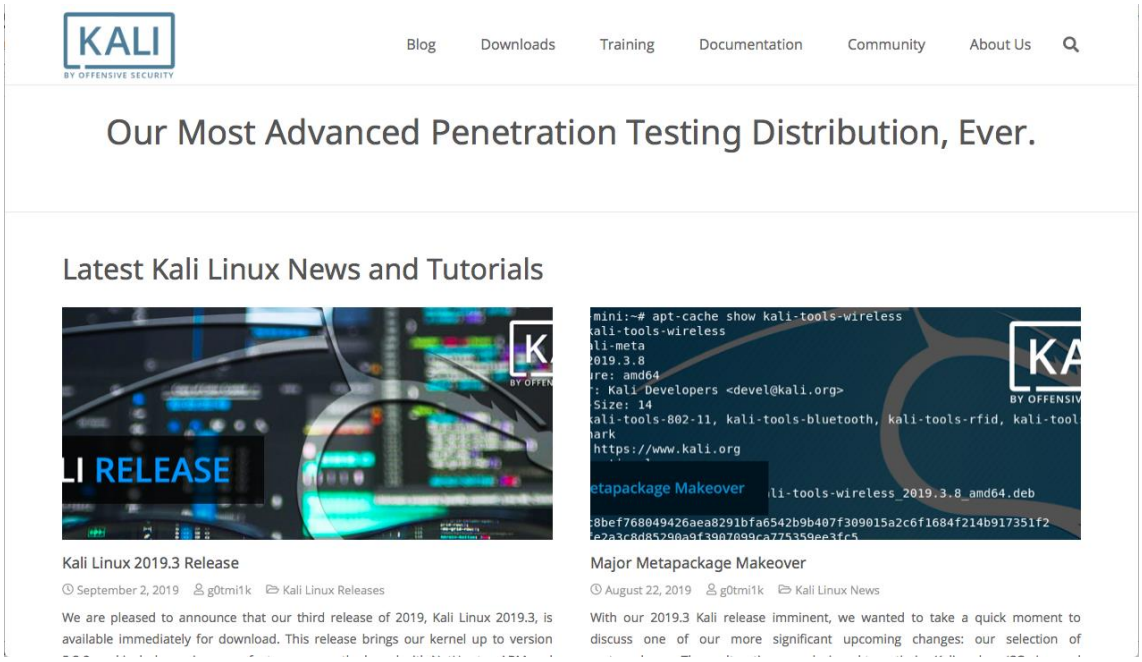
Tämän työn tutkimuksellisessa osassa käytettiin seuraavia työkaluja testauksen suorittamiseen. Kaikki käytetyt työkalut ovat avoimen lähdekoodin ohjelmia, tai ohjelmista on käytetty niiden ilmaisversioita. Testauksessa hyödynnettiin Windows 10-käyttöjärjestelmää, sekä Kali Linux-distribuutiota josta on tarkempi kuvaus alla.

6.1 Kali Linux

Kali Linux on Linux-distribuutio, joka on suunniteltu penetraatiotestausta ja tietoturva-ammattilaisia varten. Siinä on satoja valmiiksi asennettuja ohjelmia, joilla erilaisia testauksia voi suorittaa. Valmiiksi asennettu ohjelmakirjasto sisältää työkaluja kaikkiin tietoturvatestauksen vaiheisiin.

Suurin osa ohjelmista on avoimen lähdekoodin ohjelmia ja niitä voi käyttää vapaasti. Ne muutamat ohjelmat, jotka ovat maksullisia kuten Burp Suite, sisältävät ilmaisen rajoitetun version. Kali Linux on suosituin distribuutio tietoturvatestauksen alalla.

Kalille on kehitetty muutama kilpailija, mutta ne eivät ole vielä saavuttaneet niin merkittävää suosiota kuin Kali. Suurimmaksi haastajaksi voidaan odotella Black Arch-distribuutiota, jota kehitetään kuuluisan Arch Linuxin-pohjalta. Toinen vaihtoehto Kalin-tilalle on Parrot OS, joka pohjautuu Kalin-tavoin Debianiin ja on suunniteltu tietoturvatestaukseen.



Kuvio 3: Kali Linux-kotisivu

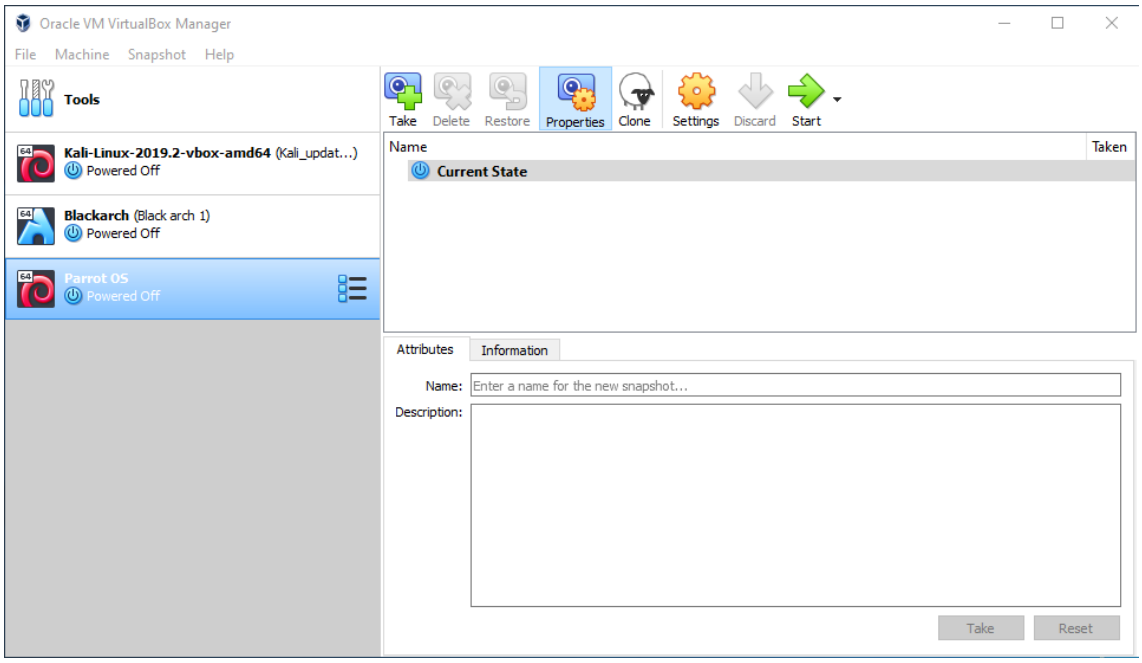
Kali on helppo käyttää ja ottaa käyttöön. Sen voi asentaa suoraan koneelle tai käynnistää USB-tikulta. Tikulta käynnistäessä aiemman session tiedot eivät kuitenkaan säily ohjelmassa.

Tässä työssä Kali asennettiin virtuaalikoneeksi käyttäen Kalin sivuilta ladattavaa OVA-tiedostoa. Tämä tiedosto on erityisesti suunniteltu Oracle Virtual Box-virtuaalikoneohjelmistolle ja se on äärimmäisen helppo ottaa käyttöön. Kun Kali Linux-asennettiin virtuaalikoneeksi, myös kaikki järjestelmään tehdyt muutokset pysyivät sessiosta toiseen, eikä esimerkiksi päivityksiä tarvinnut ladata aina uudestaan.

6.2 Oracle virtual box

Virtual Box on työkalu, jolla voi luoda ja hallita virtuaalikoneita vaivattomasti. Sitä voidaan käyttää macOS-, Windows, Linux- ja Solaris-käyttöjärjestelmissä. Siinä voidaan hallita lukuisia eri virtuaalikoneita, yhdeltä isäntäkoneelta.

Virtual Boxin-avulla käyttäjällä voi olla omalla koneellaan useampia käyttöjärjestelmiä samanaikaisesti. Tämä on hyödyllistä etenkin testauksissa, joissa testaaja käyttää yhtä käyttöjärjestelmää testaamiseen ja toista muihin työtehtäviin tai jos haluaa vain tutkia jonkin käyttöjärjestelmän ominaisuuksia ennen sitoutumista.



Kuvio 4: Virtualbox

Virtual Boxia-voidaan hyödyntää harjoitteluympäristön luomiseen. Valmiita ympäristöjä löytyy jo verkosta ja niitä voidaan hyödyntää Virtual Boxin-avulla. Virtual Box on avoimen lähdekoodin ohjelma, jota julkaistaan GPL-lisenssillä (General Public License).

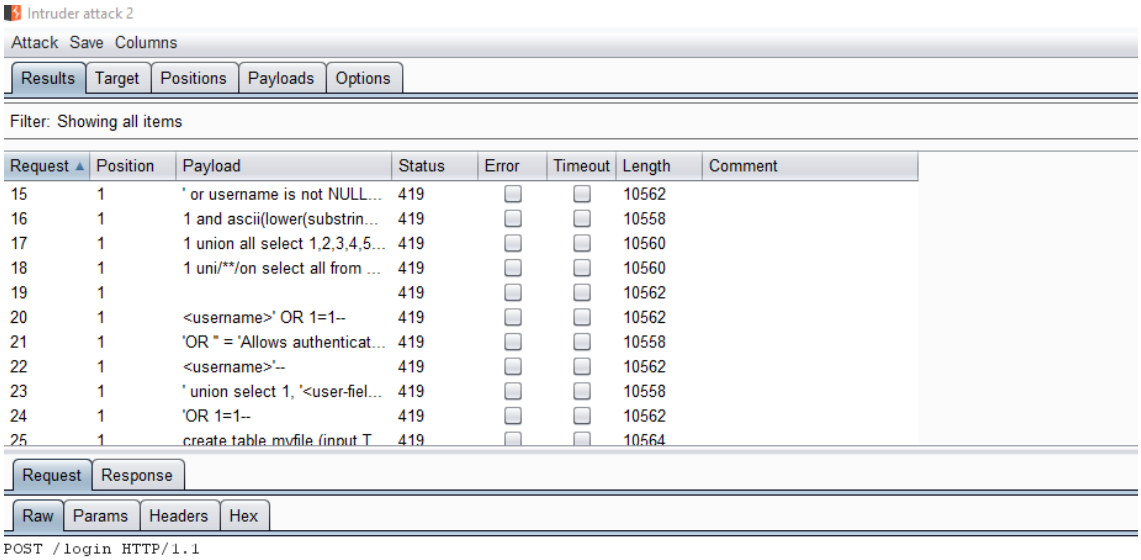
6.3 Burp Suite

Burp Suite on tietoturvatestaukseen kehitetty työkalu. Siinä on useita ominaisuuksia, joita voidaan käyttää testaamisessa. Yksi tärkeimmistä ominaisuuksista on välityspalvelin ominaisuus, joka mahdollistaa testattavan kohteen kartoittamisen. Burp Suite-myös mahdollistaa jokaisen HTTP- ja HTTPS- kutsun tarkastelun ja muokkaamisen ennen niiden läpimenoa.

Burp sisältää Intruder-ominaisuuden, joka mahdollistaa joidenkin hyökkäysten ja haavoittuvuuksien testaamisen automaattisesti. Automatisointi säästää testaajalta paljon aikaa.

Repeater on toinen tärkeä ominaisuus Burpissa. Repeater-mahdollistaa yhden pyynnön testaamisen useaan kertaan manuaalisesti. Tämä on erittäin kätevä ominaisuus esimerkiksi, kun pyritään selvittämään antaako kohde virhekoodin vai hyväksyykö se pyynnön. Sen avulla voidaan myös todentaa manuaalisesti Intruderin-automaattisesti löytämät mahdolliset haavoittuvuudet.

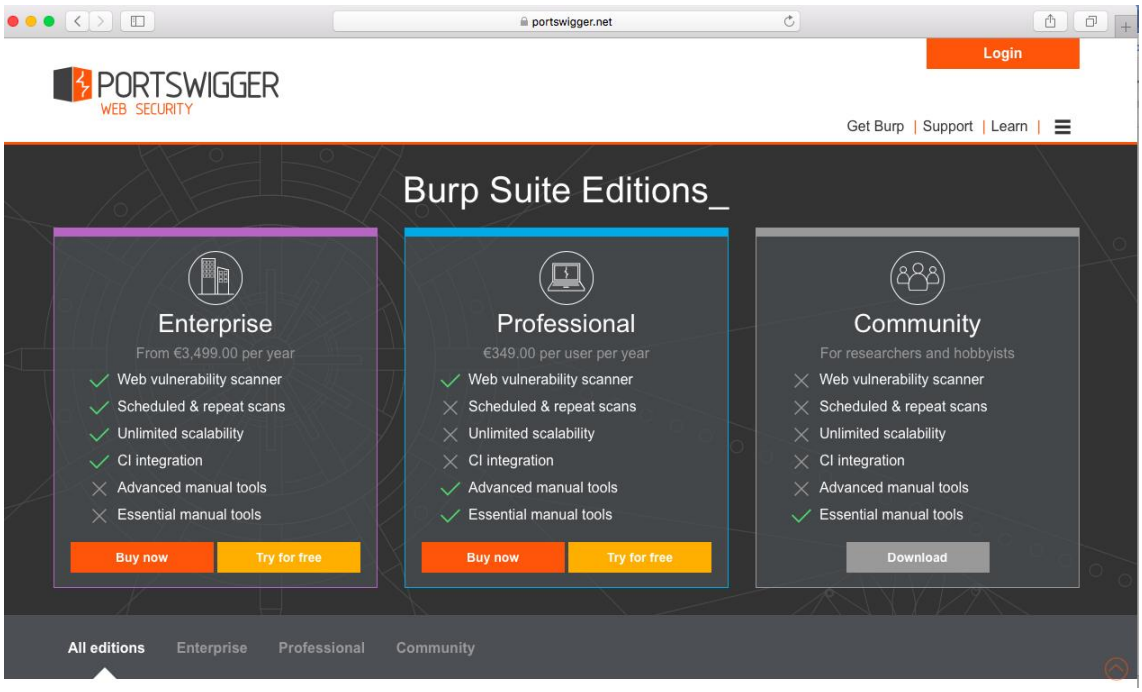
Burp-omaa koodin käännöstyökalun, jolla voidaan purkaa koodattuja tietoja ja kääntää niitä selväkieliseen ja luettavaan muotoon.



Kuvio 5: SQL-injektio hyökkäys Burp Suitella

Spider on ominaisuus Burpissa, joka niin sanotusti ryömii kohdesivuston läpi ja etsii kaikki mahdolliset URL-osoitteet, joita käyttäjä ei pysty näkemään. Spider on tärkeä työkalu, kun kartoitetaan kohdetta. Spiderin-avulla saadaan selville mahdollisesti piilotettuja polkuja ja tiedostoja. Spider-auttaa kartoittamaan kohdetta tehokkaasti ja nopeasti.

Yksi tärkeimmistä ominaisuuksista, jota Burp Suiten-Ilmaisessa versiossa ei ole, on Skanneri. Skanneri nimensä mukaan skannaa kohdesivuston ja ilmoittaa testaajalle mitä haavoittuvuuksia missäkin kohteen osassa mahdollisesti on. Tämä auttaa testaajaa löytämään nopeammin testattavat kohdat kohteesta. Skanneri antaa kattavan raportin löytämistään tuloksista, josta testaajan on helppo edetä eteenpäin.



Kuvio 6: Burp Suite-versiot

Burp on saatavilla Windowsille, MacOs:lle ja Linuxille. Se on mahdollista ladata joillekin Android puhelimille. Burp Suiten-ammattilaisille tarkoitettu versio maksaa noin 350 € vuodessa. Jos tekee paljon testausta tai harrastaa esimerkiksi Bug Bounty-ohjelmia, on se kannattava hankinta. Harrastelijalle riittää myös ilmainen versio. Yrityksille on tarjolla Enterprise-vaihtoehto.

6.4 Internetin hakukoneet

Internetin hakukoneet, kuten Google, Bing, DuckDuckgo ja Yahoo sisältävät valtavan määrän tietoa. Tätä tietoa kannattaa käyttää hyväkseen testausta suoritettaessa. Hakukoneiden avulla voi etsiä nopeasti tietoa haavoittuvuuksista ja testaustavoista, joita kohteeseen voi soveltaa.

Kuvio 7: DuckDuckGo-hakukone

Tiedon hakeminen on nopeaa. Jos käyttäjä on perehtynyt hakukoneiden ominaisuuksiin, tulosten laatu kasvaa huomattavasti. Jos testaaja saa esimerkiksi selville mitä PHP-versiota kohteessa käytetään, tai mikä SQL-ohjelma on käytössä, voi hakukoneista etsiä näiden tunnettuja haavoittuvuuksia nopeasti.

Hakukoneet ovat erittäin hyödyllisiä etenkin, kun suoritetaan tiedonkeruuta tulevaa testausta varten. Googlella on myös kehittäjätyökalu, jota hyödyntämällä voidaan hankkia hyödyllistä tietoa kohteesta, jos kohteena on verkkosivu.

6.5 Owasp Zap

Owasp Zap on Burp Suiten tapaan työkalu, jolla voidaan suorittaa useita eri testauksia kohteelle. Työn tutkimuksellisessa osassa hyödynnettiin tämän työkalun skannaus ominaisuutta. Ohjelma skannasi kohteen, ja ilmoitti löytämänsä mahdolliset haavoittuvuudet. Tämän ohjelman käyttö johtui Burp Suiten-vastaavan ominaisuuden löytymisestä vain maksullisessa versiossa. Toinen käytetty ominaisuus oli Fuzzer, jonka nopeus Owasp Zap:issa oli paljon parempi kuin Burp Suitella.

Owasp Zap-ajaa kohteeseen skannaukset, ja ilmoittaa löytämänsä mahdolliset haavoittuvuudet. Haavoittuvuudet ilmoitetaan uhkatason, vaikutuksen ja todennäköisyyden mukaan. Löydetyt haavoittuvuudet pyrittiin todentamaan manuaalisesti. Skannerit antavat hälytyksiä, mutta jotkin niistä eivät ole oikeita haavoittuvuuksia. Owasp Zap-ilmoittaa suoraan raportissa, miten todennäköisenä se pitää haavoittuvuuden olemassa oloa. Raportissa myös näkyy värikoodilla mahdollisen haavoittuvuuden uhkataso.

Fuzzer on työkalu, jota voidaan käyttää, esimerkiksi käyttäjätunnusten tai salasanojen murtamiseen. Sitä voidaan käyttää ajettaessa tuhansien salasanojen ja käyttäjänimien listoja läpi tavoitteena löytää oikea salasana oikealle käyttäjänimelle.

6.6 Muita käytettyjä työkaluja

Burp Suite oli keskeisimmässä roolissa tässä projektissa. Muita työkaluja käytettiin tukemaan toimintaa. Tämä usein johtui jonkin ominaisuuden puutteesta Burp Suiten-ilmaisversiossa.

Muita työssä käytettyjä työkaluja olivat Nikto, Netcat, Commix ja SqlMap. Nikto on skannaustyökalu, jolla voidaan löytää kohteesta mahdollisia haavoittuvuuksia. Netcatia sanotaan tietoturvatyökalujen sveitsiläiseksi linkkuveitseksi, sillä siinä on useita toimintoja, joilla voi tukea testaamista. Netcat-pystyy luomaan yhteyden kohteeseen ja muokkaamaan parametreja. Sillä voidaan testata TCP/IP-järjestelmiä. Commix- ja SqlMap-ovat automaattisia työkaluja, joita käytetään injektiohaavoittuvuuksien havaitsemiseen ja testaamiseen.

7 OWASP

OWASP (The Open Web Application Security Project) on voittoa tavoittelematon järjestö, jonka tavoitteena on parantaa sovellusten ja ohjelmistojen turvallisuutta. OWASP ylläpitää monia projekteja joiden tarkoitus on edistää turvallisuutta, kuten OWASP TOP 10 ja OWASP Testing Guide. Molempia näitä käytettiin tämän testausvaiheessa.

OWASP TOP 10 on tietoturvaraportti, jossa käydään läpi kymmenen kriittisintä haavoittuvuutta. Testing Guide on ohjeistus penetraatiotestaamiseen ja sitä voidaan käyttää kehyksenä yrityksen tietoturvatestauksessa.

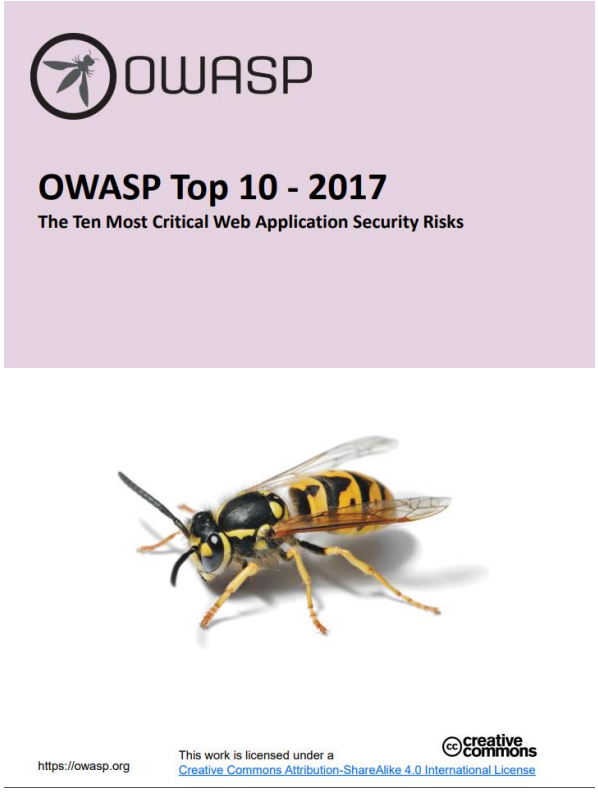
Kuvio 8: OWASP Testing Guide

OWASP on julkaissut älypuhelinversion TOP 10-raportistaan, jossa keskitytään yksinomaan älypuhelimia ja tabletteja koskeviin uhkiin. Raportin lisäksi myös mobiililaitteille on laadittu oma testausohje. Seuraava projekti tulee keskittymään ohjelmointirajapintojen turvallisuusuhkiin ja se on tarkoitus julkaista 2019 vuoden aikana.

Kuvio 9: Owasp API Security Top 10

Ohjelmointirajapintoja ja Web-applikaatioita koskevat raportit sisältävät päällekkäisyyksiä jonkun verran. Tämä johtuu siitä, että API-raportti keskittyy pelkästään ohjelmointirajapintojen haavoittuvuuksiin ja OWASP TOP 10-raportti käsittää laajemman kokonaisuuden web-sovellusten testaamiseen.

OWASP TOP 10 on raportti, johon on koottu kymmenen kriittisintä web-sovellusten tietoturva riskiä. Tätä raporttia päivitetään ja julkaistaan muutaman vuoden välein ja sen on tarkoitus toimia ohjeistuksena testauksessa haavoittuvuuksista, jotka pitää vähintään testata. Tätä raporttia pidetään alalla standardina ja ohjeistuksena pahimmista uhkista.

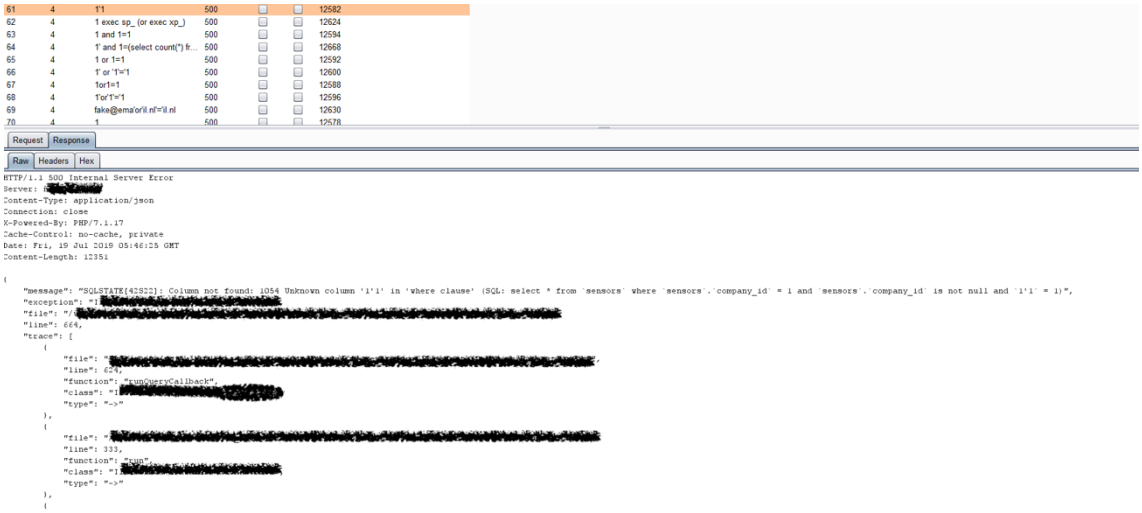


Kuvio 10: Owasp Top 10-raportti

7.1 Injection

Injektio haavoittuvuus tarkoittaa, että mahdollinen hyökkääjä pystyy syöttämään koodia sovelluksen pyytämiin kenttiin. Hyvä esimerkki tästä on sisäänkirjautumissivut. Syötetty koodi tekee sille määritetyn toiminnon ja hyökkääjän on näin mahdollista saada haltuunsa arkaluontoista tietoa.

Injektiohyökkäys on määritelty suurimmaksi uhkaksi OWASPIN-raportissa. Se on helppo toteuttaa ja siitä saatava informaatio voi olla erittäin tuhoisaa yrityksen kannalta. Injektiohyökkäyksellä voidaan saada haltuun esimerkiksi käyttäjien nimiä, salasanoja, sähköpostiosoitteita ja luottokorttitietoja.



Kuvio 11: Burp Suite SQL-Query Response

Yllä olevassa kuvassa on esimerkki Burp Suiten-antamasta vastauksesta, kun kohteeseen yritetään suorittaa injektiohyökkäystä.

7.2 Broken Authentication

Rikkinäinen autentikaatio tarkoittaa sovelluksen heikosti tai väärin toteutettua autentikointia. Tämä mahdollistaa salasanojen, istuntotunnisteiden ja avaimien haltuun saamisen tai muiden käyttäjien tunnusten käyttämisen.

Broken Authentication-hyökkäyksen tavoitteena on hankkia pääsy sivustolle, jolle käyttäjälle ei ole oikeutta. Tähän voidaan käyttää esimerkiksi Credential Stuffing-hyökkäystä, jossa kirjautumiskenttiin syötetään suuri määrä eri yhdistelmiä käyttäjänimistä ja salasanoista, kunnes oikea löytyy.

7.3 Sensitive Data Exposure

Arkaluontoisella datalla tarkoitetaan sovelluksiin tallennettuja arkaluonteisia tietoja, kuten taloudellisia tietoja, maksukorttitietoja tai työntekijöiden tietoja. Sovelluksien haavoittuvuudet antavat hyökkääjille mahdollisuuden varastaa tai tehdä muutoksia näihin tietoihin.

7.4 XLM External Entities

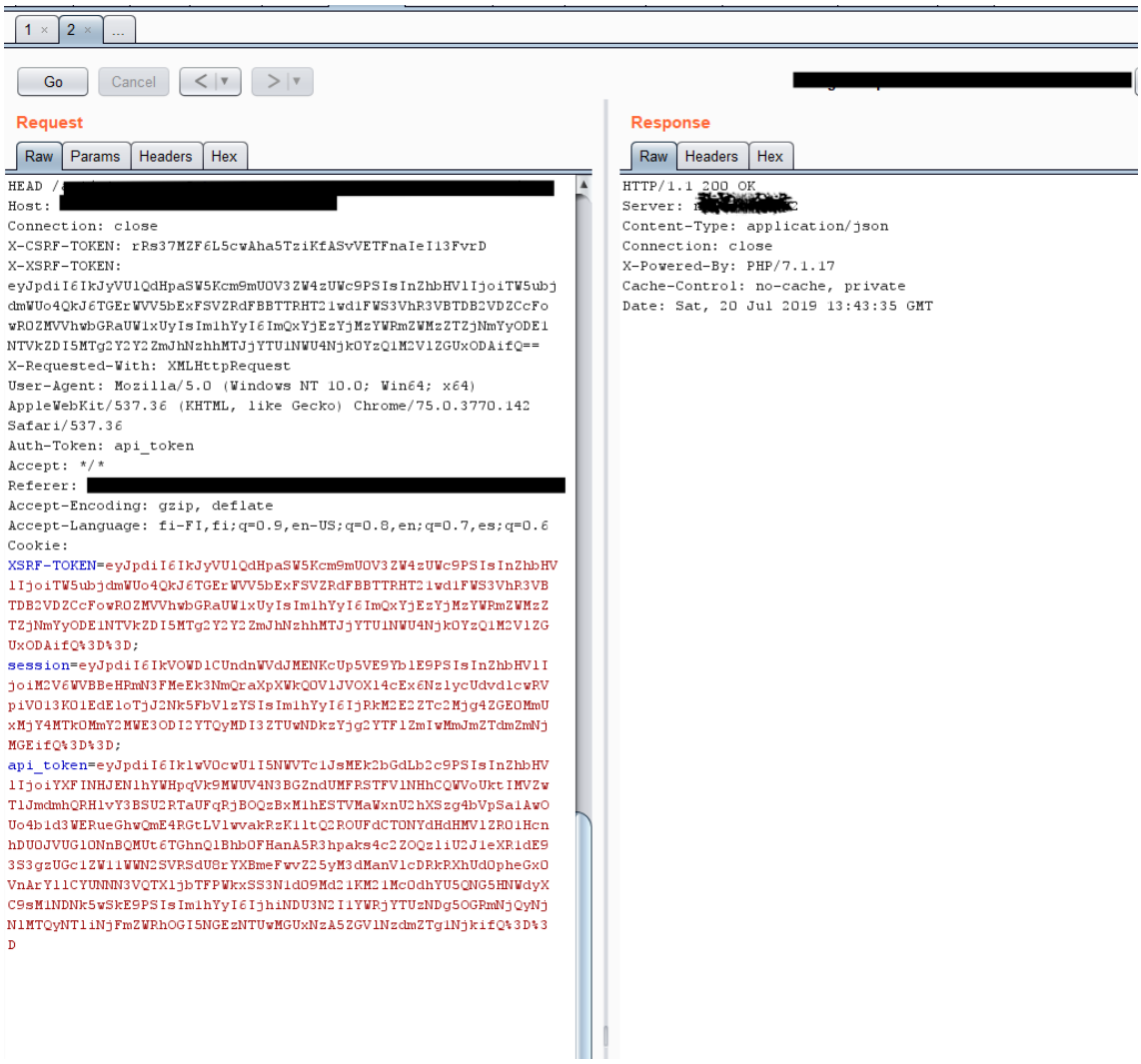
Hyökkääjä pystyy hyödyntämään tätä heikkoutta, jos he pystyvät syöttämän XML-dokumentin mukana haitallista sisältöä.

XML-dokumenteista ei tarkisteta ulkoisia viittauksia. Tämän haavoittuvuus on yleinen, kun XML-jäsennin on virheellisesti konfiguroitu. Tämä hyökkäys mahdollistaa monia hyökkäyksiä, kuten arkaluontoisen datan tarkastelun ja palvelunesto hyökkäykset.

7.5 Broken Acces Control

Väärin asetetut rajoitukset käyttäjien oikeuksissa mahdollistavat hyökkääjien saada korkeamman tason oikeuksia sovelluksen sisällä. Tämä haavoittuvuus mahdollistaa hyökkääjän tehdä toimintoja, joihin hänellä ei pitäisi olla oikeuksia.

Nämä toiminnot voivat olla esimerkiksi käyttäjien lisääminen, poistaminen tai tietojen muokkaaminen. Alla olevassa selvitetään pääsynhallinnan ohituksen heikkouden mahdollisuutta Burp Suitella



Kuvio 12: HEAD access controll bypass testing

7.6 Security Misconfiguration

Turvallisuusasetusten väärä konfiguraatio tai päivitysten asentamatta jättäminen on yleinen haavoittuvuus. Tämä mahdollistaa hyökkääjälle useita hyökkäysvektoreita. Hyökkääjän on

mahdollista selvittää avoinna olevia portteja ja käytössä oleva versio, jonka perusteella voidaan valita hyökkäysmetodit.

Tämä on vakava haavoittuvuus. Jos mahdollinen hyökkääjää saa tietoonsa, että kohde käyttää vanhaa versiota kohteesta. Hyökkääjä voi tehdä nopean Google-haun ja selvittää kaikki haavoittuvuudet, jotka kyseistä versiota vaivaavat. Tämän jälkeen niiden hyödyntäminen on helppoa.

7.7 Cross Site Scripting

XXS-on haavoittuvuus, joka mahdollistaa hyökkääjän syöttää koodia sovellukseen, jonka sovellus suorittaa osan ohjelmaa. Tämä mahdollistaa istunnon kaappaamisen tai tietojen varastamisen ohjaamalla käyttäjä haitalliselle sivustolle.

XXS-on erityisen vaarallinen, koska hyökkääjän on helppo selvittää sen mahdollisuus. Automaattiset työkalut kertovat XXS-heikkouksista nopeasti, myöskään manuaalinen kokeilu ei ole vaikeaa ja vie vain hetken.

7.8 Insecure Deserialization

On haavoittuvuus, joka ilmenee, kun sovellus ei varmenna vastaanotettua dataa. Tämä mahdollistaa hyökkääjän kaapata ja muuttaa sovellukseen menevän datan sisältöä. Tämä hyökkäys on hankalampi toteuttaa, mutta onnistuessaan se on erittäin tuhoisa.

7.9 Using Components with Known Vulnerabilities

Haavoittuvuuksia löydetään jatkuvasti komponenteista ja ohjelmistoista. Jos näitä haavoittuvuuksia ei korjata, hyökkääjä voi hyödyntää tunnettuja haavoittuvuuksia suorittaessaan hyökkäystä.

Tekemällä tutkimustyötä (OSINT) voidaan selvittää mitä komponentteja kohteessa käytetään, onko niihin ajettu viimeisimmät päivitykset ja mitä haavoittuvuuksia niissä voi olla. Näillä tiedoilla hyökkääjä voi alkaa toteuttamaan hyökkäystä.

7.10 Insufficient Logging and Monitoring

Riittämätön lokien hallinta mahdollistaa hyökkääjille suuremman aikaikkunan hyökkäyksen suorittamiseen. Mitä enemmän hyökkääjällä on aikaa, sitä suurempaa tuhoa hän saa aikaan. Nopealla reagoinnilla hyökkäys voidaan pysäyttää heti alkuun. Tämän lisäksi myös todisteiden kerääminen on helpompaa, sillä hyökkääjällä ei ole aikaa tuhota jälkiään.

8 Tulokset

Testauksen aikana saadut tulokset pyrittiin toistamaan ja varmistamaan käyttäen hyväksi manuaalisia keinoja. Kaikkia tuloksia ei kuitenkaan pystytty todentamaan ja raporttiin kirjattiin

ainoastaan automaattisten skannausten tulokset. Tämä vaihe pyrittiin tekemään, jotta voitaisiin varmistua löydöksen aiheellisuudesta. Tämä ei kuitenkaan aina ollut mahdollista.

Kaikki testauksen aikana saadut tulokset raportoitiin eteenpäin yhteistyöyritykselle. Työn tutkimuksellisessa osassa testatusta kohteesta ei kuitenkaan löytynyt vakavia haavoittuvuuksia. Vaikka testauksessa ei onnistuttu löytämään mitään vakavia haavoittuvuuksia, oli testaus aiheellinen ja se paljasti muutaman haavoittuvuuden, jotka tulee korjata.

Yrityksen edustajan kanssa käydyn keskustelun perusteella ymmärrettiin, että löydetyt haavoittuvuudet eivät olleet kriittisiä tai ne olivat False Positive-tuloksia. Tämä ei kuitenkaan ollut yllätys, sillä kohteen oli tarkoitus olla tietoturvallinen jo valmiiksi, eikä suuria yllätyksiä pitänytkään löytyä.

9 Yhteenveto

Penetraatio testauksen tärkeys tulee korostumaan teknologisen kehittymisen myötä. Internet of Things eli IoT-laitteiden tuleminen markkinoille, niiden nopea yleistyminen ja laajat käyttömahdollisuudet teollisuudesta lääketieteeseen ja tavallisiin kuluttajiin on luonut uusia hyökkäysalueita ja testauskohteita.

Myös pilviteknologian käytön yleistyminen tulee lisäämään tietoturvatestauksen tarvetta entisestään. Mitä enemmän yritykset ja yksityiset kansalaiset hyödyntävät pilviteknologiaa, sitä enemmän se myös kiinnostaa verkkorikollisia. Jos pilveen tallennetaan kaikki yrityksen tiedostot ja palvelu vaarantuu, voi pahimmassa tapauksessa koko yritys joutua lopettamaan toimintansa.

Jokaisen yrityksen ja organisaation johtoryhmän tulisi ymmärtää, että testaaminen on paitsi tärkeää, se pitää toistaa tasaisin väliajoin. Usein mietitään testauksesta koituvaa kulua, mutta ei ajatella mahdollisesta hyökkäyksestä koituvia kustannuksia. Kun näitä lukuja verrataan toisiinsa, on testauksesta koituva kulu usein pienempi.

Testaaminen on prosessi, joka vaatii usean vaiheen ja tarkat rajaukset. Testitapaukset tulee suunnitella huolella ja testaajan tulee olla tietoinen omista taidoistaan. Kaiken tekemisen tulee myös noudattaa lakia. Testaajan tulee ottaa testatessa huomioon paikallisen maan laki ja kansainväliset lait suorittaessaan testausta. Testaajalla on myös velvollisuus raportoida, jos hän löytää testauksen kohteena olevasta tietojärjestelmästä laitonta materiaalia ja vikoja tai puutteita. Raportointi tehdään joko asiakkaalle tai viranomaisille riippuen löydetystä materiaalista.

Luottamus on tärkeää tietoturva-alalla. Tämä on merkittävää etenkin, kun puhutaan testaamisesta, missä testaajalla on usein mahdollisuus päästä käsiksi asiakkaan arkaluontoiseen tietoon. Asiakkaan pitää pystyä luottamaan testaajaan. Monet tietoturvayritykset teettävät

työntekijöilleen turvallisuusselvityksiä ja henkilöt, joilla on mahdollisesti rikosrekisterimerkintä, eivät pääse näihin positioihin.

Teknologian kehittyessä ja tiedon digitalisoituessa kasvaa mahdollinen hyökkäyspinta-ala. Verkkorikokset ovat lisääntyneet ja niillä tavoitellut voitot voivat olla suuria. Tämä lisää tietoturvan tarvetta yrityksissä. Testaaminen on vain yksi palapelin pala taistelussa verkkorikollisuutta vastaan.

Lähteet

Painetut

Andreasson, Ari, Riikonen, Jaana & Ylipartanen Arto. 2017. Osaava Tietosuojavastaava.

Järvinen, Petteri & Rousku, Kimmo. 2017. Työpaikantietoturvaopas, tunnista uhat, hallitse riskit

Järvinen, Petteri. 2014. NSA Näin meitä seurataan.

Järvinen, Petteri. 2018. Kyberuhkia ja somesotaa, Digiaikana sinäkin olet etulinjassa.

Korpisaari, Päivi, Pitkänen, Olli & Warma-Lehtinen, Eija. 2018. Uusi Tietosuojalainsäädäntö

Limnell, Jarno, Majewski, Klaus & Salminen, Mirva. 2014. Kyberturvallisuus.

Peltomäki, Juha & Norppa, Kati. 2015. Rikos Meni Verkkoon, Näkökulmia kyberrikollisuuteen ja verkkoturvallisuuteen.

Rousku, Kimmo. 2014. Kyberturvaopas, Tietoturvaa kotona ja työpaikalla.

Sähköiset

General Data Protection Regulation (GDPR). Article 83 GDPR. Verkkodokumentti. https://gdpr.eu/article-83-conditions-for-imposing-administrative-fines. Viitattu 28.10.2019

Mueller, Robert. 01.03.2012. Speech at RSA Cyber Security Conference. https://archives.fbi.gov/archives/news/speeches/combating-threats-in-the-cyber-world-outsmarting-terrorists-hackers-and-spies. Viitattu 14.09.2019

Suojelupoliisi. Supon Vuosikirja 2017. Verkkodokumentti https://www.supo.fi/instancedata/prime_product_julkaisu/intermin/embeds/supowwwstructure/75374_Supo_2017_FIN_www.pdf?57559a3bf3fad688. Viitattu 05.10.2019

Kuviot

Kuvio 1: Fiarone Oy 7

Kuvio 2: Hackerone, Bug Bounty-aikajana 13

Kuvio 3: Kali Linux-kotisivu 19

Kuvio 4: Virtualbox 20

Kuvio 5: SQL-injektio hyökkäys Burp Suitella 21

Kuvio 6: Burp Suite-versiot 22

Kuvio 7: DuckDuckGo-hakukone 23

Kuvio 8: OWASP Testing Guide 25

Kuvio 9: Owasp API Security Top 10 26

Kuvio 10: Owasp Top 10-raportti 27

Kuvio 11: Burp Suite SQL-Query Response 28

Kuvio 12: HEAD access controll bypass testing 29