Maiju Valkeinen

Web-sovellusten manuaalisen penetraatiotestauksen erilaiset ohjeet

Tietotekniikan kandidaatintutkielma

18. toukokuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Maiju Valkeinen

Yhteystiedot: maiju.k.valkeinen@student.jyu.fi

Ohjaaja: Tytti Saksa

Työn nimi: Web-sovellusten manuaalisen penetraatiotestauksen erilaiset ohjeet

Title in English: Different instruction for manual penetration testing of web applications

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikka

Sivumäärä: 20+0

Tiivistelmä: Penetraatiotestaukselle ei ole vielä virallista standardia, mutta monet toimijat ovat julkaisseet omia ohjeitaan tietoturvatestaukseen. Tässä kanditutkielmassa vertaillaan kolmea seuraavaa ohjetta, Offensive Securityn julkaisema Kali Linux Revealed, NIST erikoisjulkaisu 800-115 ja OSSTMM 3. Tutkielman tarkoitus on selvittää miten niiden esittelemät testausmetodologiat poikkeavat toisistaan, ja miten ne soveltuvat manuaalisen web-penetraatiot testauksen opetteluun. Tutkielman tuloksena on että kaikkien kolmen ohjeen testausmetodologiat ovat toteutukseltaan samankaltaisia, vaikka niiden käyttämä termistö tai etenemiskaavio poikkeavatkin toisistaan ja ne poikkeavat lukijakunnna osaamistasossa.

Avainsanat: Penetraatiotestaus, kirjallisuuskatsaus, kanditutkielma

Abstract: Penetration testing does not yet have official standard, but many companies, offices and organisation have published their own guidelines for information security testing. In this bachelor's thesis three different manuals are being compared; Kali Linux Revealed by Offensive Security, NIST special publication 800-115 and OSSTMM 3. Testing methodologies presented by these manuals are being compared, as well as their suitability as teaching method for manual penetration testing of web applications. We find that all three manuals testing methodologies follow same kind execution, even when they differ in terminology or in progress charts and require skill level.

Keywords: Penetration testing, book review, Bachelor's Thesis

Kuviot

Taulukot

Sisältö

1 Johdanto

Monet ihmisten päivittäin käyttämistä ohjelmista ovat web-pohjaisia, eli ne toimivat täysin verkossa tai käyttävät verkkoyhteyttä osittain toiminnassaan. Tällaiset järjestelmät ja niiden sisältämä data tarvitsevat hyvän suojauksen tietoturvahyökkäyksiltä. Yksi tapa kehittää ja ylläpitää järjestelmien suojausta, on tietoturvasuojauksien testaus ja uhkien kartoitus. Tietoturvan testausta eettisen hakkeroinnin keinoin kutsutaan penetraatiotestaukseksi, koska sillä pyritään läpäisemään ohjelmiston suojaus. Tässä tutkimuksessa kartoitetaan kirjallisuuskatsauksena eri asiantuntijoiden julkaisemia ohjeita web-ohjelmistojen penetraatiotestaukselle.

Penetraatiotestaukselle ei ole vielä olemassa virallista standardia. Useat tahot kuten Offensive Security (Hertzog, O'Gorman ja Aharoni 2017), Institute for Security and Open Methodologies (lyh. ISECOM) (Herzog 2010) ja amerikkalainen National Institute of Standards and Technology (lyh. NIST) (Scarfone ym. 2008) ovat julkaisseet omat ohjeensa penetraatiotestauksen toteuttamiselle. Joillain toimialoilla on vaatimuksia ja standardeja, mitä tietoturvan kuuluu täyttää, kuten maksukorttialan kansainvälinen tietoturvastandardi PCI DSS, mutta edellä mainitussa ohjeessakaan ei määrätä penetraatiotestauksesta, kuin että sen metodologia tulee olla laajasti alalla hyväksytty (*Payment Card Industry Data Security Standard, v3.2.1* 2018). Eri asiantuntijoiden käytännöt, kuten testausvaiheitten määrä, testauksen suunnittelu ja löydettyjen turvallisuusuhkien raportointi poikkeavat toisistaan. Tämän tutkielman tarkoituksena on tarkastella eri asiantuntijoiden julkaisemien ohjeiden eroavaisuuksia ja yhtäläisyyksiä sekä vertailla niissä esiteltyjä penetraatiotestauksen metodologioita. Tutkielmassa keskitytään web-ohjelmistojen manuaaliseen black box -tietoturvatestauksekseen, mutta käsitellään myös hyvin lyhyesti automaattitestausta ja sen heikkouksia.

2 Penetraatiotestaus

Penetraatiotestaus tai tietoturvatestaus on tietoturva-alan ammattilaisten toteuttamaa eettistä hakkerointia, jossa kartoitetaan järjestelmän tietoturvan haavoittuvuuksia ja pyritään simuloimaan tietoturvahyökkäyksen tapahtumia (Scarfone ym. 2008). Penetraatiotestauksessa asiakkaan luvalla heidän järjestelmäänsä tehdään oikeaa tietoturvahyökkäys (Hertzog, O'Gorman ja Aharoni 2017). Hyökkäyksessä yritetään päästä tilanteeseen, jossa olisi mahdollista varastaa järjestelmän käyttöoikeudet, dataa, estää järjestelmän käyttö tai vastaavaa, ilman että aiheutetaan oikeaa vahinkoa järjestelmälle, sen käytölle tai datalle. Hyökkäyksen onnistuminen ja toteutustavat raportoidaan testauksen päätteeksi takaisin asiakkaalle. Haavoittuvuudet ovat ominaisuuksia tai virheitä ohjelmiston koodissa tai sen toiminnassa, joita pahantahtoiset käyttäjät voivat käyttää hyväkseen päästäkseen sisään ohjelmaan, saadakseen heille kuulumatonta tietoa tai estääkseen ohjelmiston normaalin käytön (Hertzog, O'Gorman ja Aharoni 2017).

Penetraatiotestauksessa käytettävät työkalut toimivat joko automaattisesti tai manuaalisesti. Esimerkiksi fuzzing-testaus on tietoturvantestaustekniikka, jossa ohjelmalle annetaan satunnaisia syötteitä, jotka saavat järjestelmän antamaan tulosteen joka ilmaisee tietoturvahaavoittuvuudesta ohjelmankoodissa (Stuttard ja Pinto 2011). Fuzzing-menetelmää voidaan tehdä manuaallisesti, jolloin testaaja kartoittaa järjestelmän, syöttää itse datan ja tulkitsee tulosteen haavoittuvuuksien löytämiseksi. Automaattinen työkalu suorittaa samat edellä mainitut toiminnot, mutta nimensä mukaisesti automaattisesti.

2.1 Automaattitestaus vs. manuaalitestaus

Automaattisia testaustyökaluja ovat muun muassa haavoittuvuusskannerit (engl. web vulnerability scanner), jotka käyvät järjestelmän läpi etsien tietyn tyyppisiä haavoittuvuuksia (Doupe, Cova ja Vigna 2010), kuten edellisessä kappaleessa kuvatusti automatisoidulla fuzzing-testauksella. Tällaiset työkalut ovat nopeita, helppoja, eikä niiden käyttö vaadi erityistä koulutusta. Automaattisten työkalujen huono puoli on, etteivät ne välttämättä havaitse monimutkaisempia tietoturvahaavoittuvuuksia ja ne tekevät valepositiivisia haavoit-

tuvuus löytöjä (engl. false positives). Kokeellisessa tutkimuksessa eri skannerien tekemistä haavoittuvuushavainnoista valepositiivisten havaintojen määrä on lähes puolet (Doupe, Cova ja Vigna 2010). Tehokkaampi haavoittuvuuksien löytäminen vaatii tulosteen manuaalista läpikäymistä, mutta tämä on kallista ja aikaa vievää. Manuaalisen testauksen tehokkuus riippuu testaajan osaamistasosta ja tietoturvahaavoittuvuuksien tuntemuksesta, minkä takia testaajan tulee olla koulutettu ammattilainen.

Automaattisia haavoittuvuusskannereita hyödynnetään yritysten rutiininomaisessa tietoturvatarkastuksessa ja järjestelmien haavoittuvuusanalyyseissä (Scarfone ym. 2008). Ne ovat myös hyödyllisiä välineitä penetraatiotestauksen alkuvaiheessa, kun testattavasta järjestelmästä kerätään tietoa. Tätä tullaan käsittelemään lisää kappaleessa 3.

2.2 White box- ja black box-testaus

White box ja black box testaus ovat termejä, joita käytetään kuvaamaan testaajalle annettuja ennakkotietoja testattavasta järjestelmästä. White box -testauksessa testaajalla on käytössään järjestelmän lähdekoodi, jonka analysoinnilla kartoitetaan tietoturva-aukkoja ohjelmistossa, ja black box-testaus tarkoittaa vastaavasti, ettei testaaja tunne ohjelmiston rakennetta tai lähdekoodia (Hertzog, O'Gorman ja Aharoni 2017). Penetraatiotestaus on aina lähtökohtaisesti black box-testausta, koska ennakkotietojen ja järjestelmän saatavuuden kannalta penetraatiotestaajan ja pahatahtoisen käyttäjän lähtökohdat ovat black box-testauksessa samat.

2.3 ROE eli Rules Of Engagement

Ennen kuin testaus voi alkaa, testaajan ja asiakkaan on määriteltävä yhteisymmärryksessä tavoitteet, rajoitukset ja säännöt tietoturvatestauksen toteutukselle, joista käytetään lyhennettä ROE. Lyhenne tulee englanninkielisestä nimestä Rules of Engagement (vapaasti suomennettuna taistelun säännöt) (Hertzog, O'Gorman ja Aharoni 2017). Säännöt ovat tärkeä osa sopimusta testaajan ja järjestelmän omistajan välillä, jota ilman testaajilla ei ole oikeutta koskea järjestelmään. ROE:n kirjoittamiselle on olemassa valmiita pohjia, kuten NIST:n erikoisjulkaisussa 800-115 liitteessä B (Scarfone ym. 2008), mutta Hertzogin ym. (2017) mukaan kaikkein paras suoja testaajien oikeusturvalle on asianajajan tekemä sopimus.

3 Eri julkaisijoiden ohjeet

Seuraavissa kappaleissa tullaan käymään läpi kolme eri penetraatiotestauksen ohjetta. Kappaleissa käytetään pääasiassa lähteenä niissä läpikäytävää ohjetta, ellei toisin mainita. Ensimmäinen ohje, Kali Linux Revealed (Hertzog, O'Gorman ja Aharoni 2017) on Offensive Security -yrityksen julkaisema ohje, joka käsittelee suurimmaksi osaksi Kali Linux -distribuutio asentamista ja käyttöä penetraatiotestauksessa, mutta ohjeistaa myös penetraatiotestauksen toteutuksessa. NIST erikoisjulkaisu 800-115 on Amerikan valtion julkaisema virallinen ohje penetraatiotestaukseen (Scarfone ym. 2008). OSSTMM 3 on kolmas versio ISECOM:n julkaisemasta ja OSSTMM -projektin tuottamasta vertaisarvioidusta tietoturva-auditointiohjeesta (Herzog 2010).

3.1 Kali Linux Revealed - Mastering the Penetration Testing Distribution

Kali Linux -käyttöjärjestelmä on yleisesti käytetty työväline penetraatiotestauksessa, koska sen asennuspaketti sisältää monia verkon analysointiin ja tietoturvatestaukseen tarvittavia työkaluja. Kali Linuxin kehittäjä Offensive Security tarjoaa sertifiointikoulutuksia penetraatiotestaukseen, ja on myös julkaissut oppaita sekä kirjallisuutta aiheesta. Yritys suosittelee kali.training-sivustollaan aloittamaan tietoturvatestaukseen tutustumisen lukemalla tässä kappaleessa käsiteltävä ohjeen Kali Linux Revealed - Masterin the Penetration Testing Distribution (vapaasti suomennettuna: Kali Linux paljastettu - Penetraatiotestausjakeluversion hallitseminen), joka on ladattavissa ilmaiseksi kyseisellä sivustolla ("The Ultimate Kali Linux Manual and Course" 2020).

Ohje alkaa Kali Linuxin esittelyllä, ja käyttöjärjestelmän asennus- ja käyttöohjeilla. Vaikka nämä ovat tärkeä osa penetraatiotestausta edeltävää toimintaa, ei tutkielmassa niihin keskitytä enempää kuin mitä seuraavassa kappaleessa kerrotaan.

Ennen testausta tehtäviin toimiin kuuluu käyttöjärjestelmän ja tarvittavien työkalupakettien huolellinen asentaminen. Tämä on tärkeä vaihe, jotta testauksesta saatu data ei sotkeudu

toisesta testistä saadulla datalla, testaus sujuisi mahdollisimman sujuvasti, ja että testaaja pystyy suorittamaan kaikki tarvitsemansa analyysit, esimerkiksi tarkastelemaan verkon ja laitteen ominaisuuksia.

Tässä ohjeessa penetraatiotestauksen metodologia on esitetty etenevän seuraavissa vaiheissa: Tiedonkeruu (engl. information gathering), haavoittuvuuksien tutkiminen (engl. vulnerability discovery), haavoittuvuuksien hyödyntäminen (engl. exploitation), pivotointi ja datan vuotaminen (engl. pivoting and exfiltration) sekä raportointi (engl. reporting). Vaiheiden englanninkieliset nimet vastaavat Kali Linuxin työkaluluettelon otsikointia.

Tiedonkeruulla tarkoitetaan tässä yhteydessä järjestelmän kartoitusta tavallisen käyttäjän tasolta, siihen tutustumista ja järjestelmän käyttämän verkon pintapuolista tarkastelua. Tiedonkeruuvaiheessa ei tehdä vielä sivustoon tai sovellukseen tunkeutuvia testejä. Haavoittuvuuksien tutkimisvaiheessa järjestelmää tutkitaan aktiivisemmin, normaalia käyttöä tunkeutuvammin menetelmin, kuten haavoittuvuusskannauksilla, ohjelmiston takaisinmallinnuksella, selvittämällä palvelimen ominaisuudet sekä web-applikaatioiden ja tietokantojen analysointia Kali Linuxin työkaluilla. Nimensä mukaisesti, tässä vaiheessa yritetään löytää mahdollisimman paljon tietoturvahaavoittuvuuksia, joiden avulla on mahdollista murtautua järjestelmään testauksen seuraavassa vaiheessa.

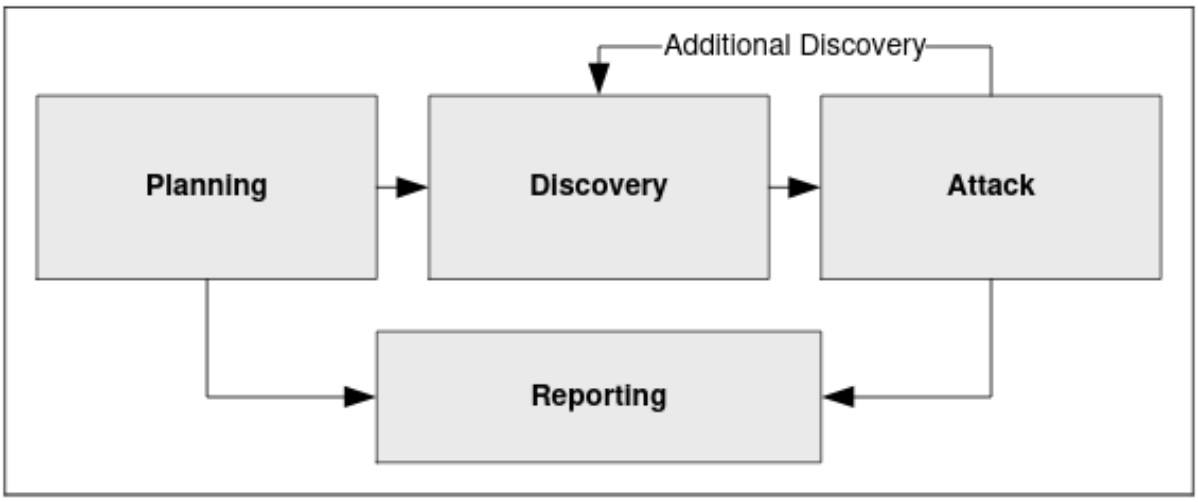
Haavoittuvuuksien hyödyntämisvaiheessa on tarkoitus aloittaa järjestelmän tietoturvan penetrointi käyttämällä hyväksi aiemmin löydettyjä haavoittuvuuksia. Tässä vaiheessa voi hyödyntää edellisessä vaiheessa mainittujen analyysityökalujen lisäksi Kalin salasanahyökkäys- ja haavoittuvuuksien hyödyntämistyökaluja. Seuraava vaihe eli pivotointi ja datanvuoto voidaan toteuttaa vasta, kun testaaja on päässyt sisään järjestelmää. Pivotoinnissa murtaudutun koneen kautta yritetään päästä sisään suojatun verkon muihin osiin, kuten toisiin koneisiin tai palvelimiin. Pivotoinnin avulla voidaan saada lisää tietoa kohteesta, kuten muita haavoittuvuuksia ja sensitiivistä dataa. Löydetty data pyritään testatessa kopioimaan tai muuten vuotamaan ulos järjestelmästä. Ohjeessa tähän testivaiheeseen suositellaan käytettäväksi aiemmin mainittujen työkalujen lisäksi Kalin kalastelu- ja huijausviesti- sekä tietoturva-aukon läpäisyn jälkeen -kategorian (engl. post-exploitation) työkaluja. Haavoittuvuuksien tutkimista, hyödyntämistä, pivotointia ja datavuotoa toistetaan niin kauan, kunnes ei enää löydetä uusia haavoittuvuuksia ja tietoturvauhkia.

Penetraatiotestauksen päätyttyä, kaikki testauksen aikana löydetyt, oikeiksi todennetut haavoittuvuudet, niiden löytymistavat sekä muut suoritetut testit dokumentoidaan loppuraporttiin, joka luovutetaan järjestelmän haltijalle.

3.2 NIST erikoisjulkaisu 800-115

NIST on Amerikan valtion kauppaministeriön alaisuudessa toimiva yksikkö, joka julkaisee tieteellisiä standardeja. NIST:n erikoisjulkaisu 800-115 "Technical Guide to Information Security Testing and Assessment"(vapaasti suomennettuna Tekninen opas informaatioturvallisuuden testaukseen ja arviointiin) perustuu NIST:n informaatiotieteen laboratorion (engl. Information Technology Laboratory) tutkimustuloksiin. Sen käyttöä suositellaan myös tietoturvan, tietoverkkojen tai järjestelmien kehityksessä ja ylläpidossa työskenteleville henkilöille kattavaksi oppaaksi tietoturvan testaukseen ja arviointiin. Ohjetta kutsutaan usein muissa julkaisuissa lyhenneteellä NIST SP 800-115 (*Payment Card Industry Data Security Standard, v3.2.1* 2018)(Hertzog, O'Gorman ja Aharoni 2017).

Scafone ym. (2008) esittelevät ohjeessaan penetraatiotestaukseen nelivaiheisen mallin, jonka etenemiskaavio on esitetty kuviossa 1. Mallin vaiheet ovat suunnittelu (engl. planning), havainto (engl. discovery), hyökkäys (engl. attack) ja raportointi.



Kuvio 1. Nelivaiheisen penetraatiotestin eteneminen (Scarfone ym. 2008)

Suunnitteluvaiheessa määritetään ja dokumentoidaan ROE, testauksen tavoitteet ja luvalli-

suus järjestelmän haltijan kanssa. Ennen tämän valmistumista ei tehdä mitään varsinaista testausta. Havainnointivaiheeseen kuuluu kaksi osaa, tiedon keruu ja haavoittuvuusanalyysi. Tiedon keruussa yritetään löytää mahdollisimman paljon tietoa järjestelmästä kuten nimipalvelin ja IP-osoitteet, työtekijöiden nimiä ja muita yhteystietoja sekä tietoa sen laitteista. Näitä tietoja voi kerätä muun muassa porttiskannauksien ja DNS-kyselyjen avulla tai tutkimalla yrityksen sivustoja, roskiksia tai kävelemällä sen tiloissa. Tiedonkeruussa voi löytyä jo haavoittuvuuksia, joita on mahdollista hyödyntää muissa vaiheissa. Haavoittuvuusanalyysi voidaan toteuttaa automaattisilla skannereilla, mutta valepositiivisten ja havaitsemattomien havaintojen vuoksi ohjeessa suositellaan tutkimaan manuaalisesti tiedonkeruussa löydettyjen järjestelmän tietoja tunnettujen haavoittuvuuksien varalta.

Hyökkäysvaiheessa yritetään murtautua sisään järjestelmään aiemmin löydettyjen haavoittuvuuksien avulla. Mikäli tässä onnistutaan, vahvistetaan haavoittuvuus todelliseksi uhkaksi, ja selvitetään menetelmä tietoturvauhan ehkäisyksi. Tietomurron onnistuttua testausta jatketaan selvittämällä, onko haavoittuvuuden kautta mahdollista pivotoitua muihin laitteisiin, tai onko testaajan mahdollista kasvattaa käyttäjäoikeuksiaan järjestelmässä. Järjestelmän sisällä, testaaja voi löytää uusia laitteita tai mahdollisia haavoittuvuuksia, joita tulee tutkia myöhemmin uudella havainnointivaiheella. Testauksen ajan otetaan selvää, kuinka laajaan osaan järjestelmää testaajan on mahdollista päästä sisään ja kuinka paljon tietoa hän pystyisi varastamaan, tuhomaan tai muokkaamaan. Hyökkäysvaihe aloitetaan niin kauan alusta, kun kaikki haavoittuvuudet on todennettu tai uusia haavoittuvuuksia ei enää löydy.

Raportointivaihetta tehdään saman aikaisesti testauksen muiden vaiheitten kanssa. Suunnitteluvaiheessa dokumentoidaan ROE ja kuluarvio. Havainto- ja hyökkäysvaiheissa uudet haavoittuvuudet kirjataan ylös ja niistä tiedotetaan järjestelmän ylläpidolle. Testin pääteeksi raportoidaan löydetyt haavoittuvuudet, ne luokitellaan todennäköisyyden ja vaarallisuuden mukaan sekä niille annetaan ohjeita uhkien lieventämiseksi.

3.3 OSSTMM 3 – The Open Source Security Testing Methodology Manual

ISECOM on puolueeton, voittoa tavoittelematon tiedeyhteisö, jonka tavoitteena on parantaa tietoturvatietoutta projekteilla ja heidän tarjoamillaan sertifiointikoulutuksilla. Heidän julkaisemansa The Open Source Security Testing Methodology Manual (vapaasti suomennettuna Avoimen lähteen turvallisuustestaus metodologia opas) on CC BY-NC-ND 3.0 lisenssin alainen teos, eli se on vapaasti jaettavissa ja käytettävissä viitaten alkuperäiseen teokseen, mutta ei kaupallisessa tarkoituksessa (Alecu 2012). Ohjeessa esitetty testausmetodologia on ISECOM:n oman OML 3.0-lisenssin (Open Methodology License) alaisuudessa.

Ohjeessa käytetään eri terminologiaa testausvaiheille kuin kahdessa aiemmassa ohjeessa. Tämä on Herzogin (2010) mukaan siksi, että tekniikan kehittyessä kielen on kehityttävä, ja he haluavat määritellä termit tarkemmin mitä niiden nykyinen tarkoitus mahdollistaa. OSSTMM-metodologia sisällyttää järjestelmän ympäristön laajemman tietoturvantestauksen, kuin tavallinen penetraatiotestausmetodologia. OSSTMM-ohjeen avulla testaajaa voi toteuttaa järjestelmän ja sen omistaman yrityksen tietoturvan täydellisen auditoinnin. Herzogin (2010) mukaan ohjetta voi käyttää penetraatiotestaukseen, vaikka hän myös kertoo pelkän penetraatiotestauksen olevan hyödyllinen työkalu vain silloin, kun halutaan korjata virheitä. Ohjeen ja auditoinnin tavoitteena on auttaa turvallisen järjestelmän kehittämisessä ja laadukkaassa sekä luotettavassa tietoturvatestauksessa.

OSSTMM 3 sisältää opastuksen tietoturvatestaukseen testauksen kohteena ollessa ihmiset, fyysiset rakenteet, langaton yhteys, tietoliikenne ja tietoverkot. Kuviossa 2 on esitetty ohjeen testausmetodologian etenemisvaiheet, ja kaikki edellä mainitut testausohjeet noudattavat sitä. Kaikkia metodologian vaiheita ei tarvitse käyttää testauksessa, ellei olla tekemässä täydellistä auditointia. Web-penetraatiotestaukseen sovelletaan kappaleen 11. Data Network Security Testing-ohjetta (vapaasti suomennettuna Tietoverkon turvallisuustestaus).

Testauksen apuna on käytettävissä ISECOM:n STAR-lomake (Security Test Audit Report), joka sisältää kaikki kysymykset ja testit, joihin auditoinnissa halutaan löytää vastaus. Täytetty lomake ei kuitenkaan korvaa kunnollista loppuraporttia.

Testausmetodologian vaiheet jaetaan neljään vaiheeseen, jotka on kuviossa 2 esitetty eri värisinä. Nämä vaiheet ovat vapaasti suomennettuina käynnistysvaihe (engl. induction phase) kuviossa keltaisella, vuorovaikutusvaihe (engl. interaction phase) oranssilla, tutkimusvaihe (engl. inquest phase) vaaleansinisellä ja väliintulo (engl. intervention phase) vaaleanpunaisella ja sinisellä.

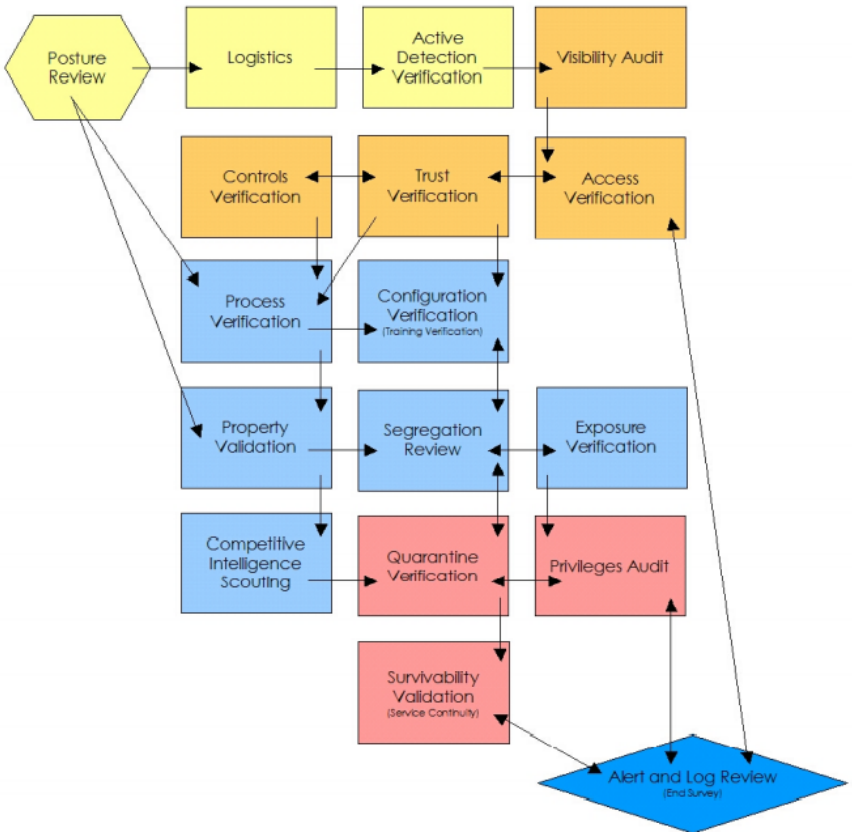
Käynnistysvaiheessa testaaja määrittää testaustilanteen lähtöasetelman (engl. posture review), testaukseen vaadittavat logistiset järjestely (engl. logistics) ja tekee aktiivisen havainnoinnin varmistuksen (engl. active detection verification). Lähtöasetelmassa tutustutaan yritykseen, otetaan selvää järjestelmästä ja käytössä olevista ohjelmistoista. Logistisessa järjestelyssä varmistetaan testauskohteen rajaus, saatavuus ja aikavyöhykkeet sekä luotettavan verkkoyhteyden saatavuus. Aktiivisen havainnoinnin varmistuksessa huolehditaan, että yrityksessä ollaan valmiita testauksen aloittamiseen, ja testauksen kannalta oleellisia henkilöitä on informoitu sen aloittamisesta.

Vuorovaikutusvaiheessa tarkastetaan näkyvyyttä (engl. visibility audit) ja saatavuutta (engl. access verification) ulkopuolelta käsin sekä järjestelmän tapaa tunnistaa käyttäjät, josta käytetään termiä luottamuksen tarkistus (engl. trust verification). Järjestelmän näkyvyyttä testataan verkko- ja porttiskannereilla, selvittäen mitä tietoa järjestelmästä on mahdollista saada selville sen ulkopuolelta. Saatavuudella tarkoitetaan tässä tapoja, joilla järjestelmää voi käyttää sen ulkopuolelta, ja sen vahvistus on haavoittuvuuksien etsimistä verkkoyhteyden välityksellä. Luottamuksen tarkistus pyritään selvittämään, onko mahdollista päästä järjestelmään, niin että se luulee sinua joksikin toiseksi. Luottamusta testataan kalastelu- ja huijausviesteillä, joilla pyritään saamaan muiden käyttäjien tunnukset järjestelmään, tai testaamalla järjestelmän käyttäjä- ja salasanajärjestelmän toimivuutta.

Tutkimusvaiheessa tehdään prosessin tarkastus (engl. process verification), rakenteellinen tarkastus (engl. configuration verification) ja riskiarvio (engl. exposure verification). Prosessin tarkastuksella tarkoitetaan yrityksen tietoturvakäytäntöjen, kuten huollon testausta. Ovatko kaikki ohjelmistot ja sertifikaatit päivitetty? Rakenteellisessa tarkistuksessa kerätään tietoa siitä, kuinka järjestelmän toiminnasta. Onko toiminnassa tarpeettomia ominaisuuksia, jotka voivat olla haavoittuvuuksia? Onko järjestelmän valvojan oletussalasana vaihdettu tai vanhat käyttäjät poistettu? Riskiarviossa selvitetään, onko yhden käyttäjän mahdollista pääs-

tä käsiksi kaikkiin järjestelmän tiedostoihin, eli kuinka paljon dataa vaarantuisi, jos henkilöstön tilille murtaudutaan. Penetraatiotestauksessa riskiarvio on mahdollista toteuttaa vasta, kun järjestelmä on penetroitu, joka ohjeen mallin mukaisesta tapahtuu vasta seuraavassa vaiheessa.

Vuorovaikutusvaiheessa järjestelmän ominaisuuksiin vaikutetaan tavoilla, jonka tarkoituksena on saada aikaan tietoturvan penetrointi. Tämä vaihe sisältää eristyksen tarkastuksen (engl. quarantine verification), oikeuksien auditoinnin (engl. priviledges audit), selviämismahdollisuuden validointi (engl. survivability validation) ja raportointi. Eristyksen tarkastuksella tarkoitetaan tietoturvasuojauksen testausta, eli miten hyvin se pystyy estämään haittaohjelmien käytön ja leviämisen. Oikeuksien tarkastuksessa hyödynnetään luottamus tarkastuksessa saatua tietoa, ja pyritään murtautumaan järjestelmään käyttäjien kirjautumisen kautta. Onko ulkopuolisen mahdollista päästä sisään ja saada samat tai korkeammat oikeudet kuin tavallinen käyttäjä. Selviämismahdollisuuden validoinnilla pyritään selvittämään, mitkä mahdollisuudet yrityksellä tai järjestelmällä on selvitä laajamittaista ja pahantahtoista hyökkäystä kuten palvelunestohyökkäystä vastaan. Raportoinnissa käydään läpi järjestelmän huomaamat hyökkäysyritykset, löydetyt uhat ja testien tulokset (engl. alert and log review) ennen kuin testaus voidaan virallisesti päättää.



Kuvio 2. OSSTMM 3:n tietoturvatestauksen etenemisvaihekaavio. (Herzog 2010) Kaavion etenemisjärjestystä käytetään kaikissa OSSTMM:n tietoturvatestauksissa, mutta kaikkia vaiheita ei käydä läpi pelkässä penetraatiotestauksessa.

4 Ohjeiden vertailu

4.1 Ohjeet

Kali Linux Revealed oli helppolukuinen ja antaa hyvän käsityksen tietoturvatestauksesta lukijoille, joille aihe ei ole vielä tuttu. Penetraatiotestausta käydään läpi ohjeessa hyvin yksinkertaistetusti ja useimmat ohjeet testausvaiheisiin neuvovat lukijaa tutustumaan Kali Linuxin testaustyökaluihin. Kirja ei ole kaikkein kattavin Offensive Securityn julkaisema penetraatiotestausopas. Ohjeen lopussa kirjoittajat ilmaisevatkin sen olevan vain johdanto penetraatiotestaukseen (Hertzog, O'Gorman ja Aharoni 2017), koska kyseinen ohje on ensisijaisesti Kali Linuxin käytön aloittamiseen, ja luettavissa ilmaiseksi. Offensive Security tarjoaa sertifiointikoulutusta penetraatiotestaukseen, ja he ovat julkaisseet myös muita, laajempia ja tarkempia opaskirjoja tietoturvatestaukseen.

NIST:n erikoisjulkaisussa penetraatiotestauksen vaiheet ja siinä käytettävät menetelmät on kuvailtu tarkasti. Toisin kuin Kali Linux Revealed -ohjeessa, erikoisjulkaisussa käsitellään myös penetraatiotestauksen fyysinen puoli, kuten social engineering, yrityksen roskien kaivelu ja yritykseen soluttautuminen tiedon keräystarkoituksessa. Ohjeessa oletetaan lukijan tietävän tietotekniikasta ja verkkoprotokollista, mutta muuten siinä opastetaan penetraatiotestauksessa tarvittavista menetelmistä, termeistä, dokumenteista ja analyyseista. Ohjeen liitteissä annetaan useita lähteitä niin tietoturvatestauksen omatoimiselle jatko-opiskelulle kuin avoimen lähdekoodin testaustyökaluille.

OSSTMM 3 oli hyvin laaja ohje, ja sen testauksien tavoitteet on kuvattu hyvin tarkasti. Siinä ei aina esitellä menetelmiä, joilla testauksia on tarkoitus toteuttaa. Oppaan lukijalta oletetaan todella hyvää tietoturva- ja tietoverkko-osaamista. Ohjeen konseptit ovat haastavampia sisäistää, koska sillä on oma termistönsä testaukselle ja tietoturvan laadun määrittelylle. Pelkkään penetraatiotestaukseen se on hyvin laaja, mutta ohjeessa on alussa niin kutsuttu quick start -ohje, eli pikaohje, jonka avulla lukija voi hyödyntää julkaisun termistön opittuaan käyttöoppaana niin web-penetraatiotestaukseen kuin yrityksen työskentelypäätteiden turvallisuuden tarkastukseen, kunhan on sisäistänyt sen filosofian.

4.2 Nelivaihetestaus

Vaikka tässä tutkielmassa esiteltyjen ohjeiden penetraatiotestausvaiheet poikkeavat määrältään toisistaan, noudattavat ne etenemiskaavaltaan samoja vaiheita kuin Scafonen ym. (2008) ohjeessa esitelty nelivaihetestausmalli (Kuvio 1.). Sekä Kali Linux Revealed että NIST esittelemät testausmetodologiat ovat hyvin samankaltaista. Scarfonen esittämässä nelivaihemallissa Hertzogin ym. (2017) metodologianvaiheet 'haavoittuvuuksien hyödyntäminen' sekä 'pivotointi ja datan vuotaminen' on yhdistetty hyökkäysvaiheeksi. OSSTMM-ohjeen metodologia (Kuvio 2.) noudattaa myös nelivaihetestausta, mutta sen vaiheet on jaoteltu eri tavalla kuin muissa. Herzogin (2010) mallissa ensimmäinen vaihe vastaa suunnittelua, mutta toisessa ja kolmannessa vaiheessa suoritetaan enimmäkseen tiedon keruuta. Neljäs vaihe sisältää sekä penetraatiotestauksen että raportoinnin. OSSTMM metodologia jakautuu vielä vaiheiden sisällä 17 eri tarkistus- ja testausvaiheeseen, joista kaikki eivät ole tarpeellisia web-penetraatiotestauksessa.

Kappaleen lopussa olevassa taulukossa (taulukko 1.) on edellä mainittujen ohjeiden vaiheet sovitettuna noudattamaan Scafonen (2008) nelivaiheista mallia. Taulukossa on lisätty kaikkiin ohjeisiin ROE:n määrittelyn, silloinkin jos se on käyty läpi ohjeessa, mutta ei ole esitelty osana testausmetodologiaa. Kaikissa ohjeissa ROE esitellään vähimmäisvaatimuksena testaajan ja järjestelmän haltijan välisestä sopimuksesta, koska ilman sitä kyseessä olisi luvaton hakkerointi.

	Kali Linux Revealed	NIST 800-115	OSSTMM 3
1.	Asenna Kali Linux ROE	Suunnittelu ROE	Lähtöasetelman määrittely Logistiset järjestelyt Aktiivisen havainnoinnin varmistus ROE
2.	Tiedonkeruu Haavoittuuksien tutkiminen	Tiedonkeruu Haavoittuvuusanalyysi	Näkyvyyden auditointi Saatavuuden vahvistus Luottamuksen tarkistus Prosessien tarkistus Rakenteellinen tarkistus
3.	Haavoittuvuuksien hyödyntäminen Pivotointi Datavuoto Testauksen toisto	Hyökkäys Paluu tiedonkeruun vaiheeseen	Riskiarvio Eristyksen tarkistus Oikeuksien auditointi
4.	Raportointi	Raportointi	Raportointi

Taulukko 1. Ohjeiden menetelmät sovitettuna nelivaiheiseen testaukseen. 1. Suunnittelu, 2. tiedonkeruu, 3. penetraatio ja 4. jälkitoimet

5 Yhteenveto

Tutkielmassa todettiin, että eri julkaisijoiden ohjeiden metodologiat noudattavat hyvin samankaltaista etenemismallia. Vaikka ohjeiden testausvaiheitten määrät ja nimeämiset voivat vaihdella, penetraatiotestauksen toteutus pysyy pääkohdittain samana. Ohjeet eivät ole ennakkotietovaatimukseltaan samantasoisia, eivätkä ne käsittele penetraatiotestausta yhtä tarkasti. Seuraavassa vertailussa olisi tarpeen tarkastella jotain muita Offensive security-yrityksen julkaisemia oppaita, jotka käsittelevät testausta huolellisemmin, sekä muita tietoturvasertifiointikoulutuksen järjestävien tahojen ohjeita. Oppiakseen tietoturvatestauksesta kaikki ohjeet ovat lukemisen arvoisia, tässä nimenomaisessa järjestyksessä. Kali Linux Revealed opastaa tietoturvatestauksen työkalujen käyttöön, NIST SP 800-115 syventää lukijan tietämystä tietoturvatestauksesta ja sen menetelmistä ja OSSTMM 3 ohjeistaa lukijaa ajattelemaan testausta tietoturvan kehityksen menetelmänä.

Lähteet

Alecu, Felician. 2012. “Information Technology Trends, Creative Commons Licenses”. *Oeconomics of Knowledge* 4 (5): 2–7.

Doupe, Adam, Marco Cova ja Giovanni Vigna. 2010. “Why Johnny Can’t Pentest: An Analysis of Black-Box Web Vulnerability Scanners”. Teoksessa *Detection of Intrusions and Malware, and Vulnerability Assessment,* toimittanut Christian Kreibich ja Marko Jahnke, 111–131. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-14215-4.

Hertzog, Raphaël, Jim O’Gorman ja Mati Aharoni. 2017. *Kali Linux Revealed: Mastering the Penetration Testing Distribution.* OffsecPress. ISBN: 978-0-9976156-0-9.

Herzog, Pete. 2010. *OSSTMM 3, The Open Source Security Testing Methodology Manual, Contemporary Security Testing and Analysis.* Saatavilla WWW-muodossa, `https://www.isecom.org`, viitattu 5.2.2020.

Payment Card Industry Data Security Standard, v3.2.1: Requirements and Security Assessment Procedures. 2018. Saatavilla WWW-muodossa, `https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss`, viitattu 28.3.2020.

Scarfone, Karen, Murugiah Souppaya, Amanda Cody ja Angela Orebaugh. 2008. *Special Publication 800-115 Technical Guide to Information Security Testing and Assessment.* Saatavilla WWW-muodossa, `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf`, viitattu 5.2.2020.

Stuttard, Dafydd, ja Marcus Pinto. 2011. *The Web Application Hacker’s Handbook: Finding and Exploiting Security Flaws.* 2. painos. John Wiley & Sons.

“The Ultimate Kali Linux Manual and Course”. 2020. Viitattu 16. huhtikuuta 2020. `https://kali.training/`.