Jussi Hartikainen

NESSUS – TIETOTURVAN TARKASTAMISOHJELMISTO

Tietojärjestelmätieteen
kandidaatintutkielma
15.11.2004

Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
Jyväskylä

TIIVISTELMÄ

Hartikainen, Jussi Petteri

Nessus – Tietoturvan tarkastamisohjelmisto/Jussi Hartikainen
Jyväskylä: Jyväskylän yliopisto, 2004.
22 s.
Kandidaatintutkielma

Tässä tutkielmassa tarkastellaan Nessus ohjelman toimintaa. Nessus on GPL (general public licence) lisenssin alla julkaistu vapaan lähdekoodin ohjelma. Nessuksen pääsuunnittelija on Renaud Deraison.

Nessus projekti pyrkii tarjoamaan internet-yhteisölle ilmaisen, tehokkaan, ajan tasalla pysyvän ja helposti käytettävän tietoturvaskannerin. Tietoturvaskanneri on ohjelmisto joka tarkastaa annetun verkko-osoiteavaruuden ja selvittää onko kohteeseen mahdollista tehdä tietomurto (ihmisen tai ohjelman, kuten madon, toimesta) tai käyttää sitä muuten vääriin tarkoituksiin. [1]

Internetistä on saatavilla ilmaisia erillisiä verkon tutkimistyökaluja, mutta tarvetta on ohjelmalle joka automatisoi tietoturvan tarkastamista ja tekee sen helpommaksi käyttäjille. Aiheen käsittely perustuu yleisen toiminnallisuuden kuvaamiseen ja ohjelman mahdollisiin käyttötarkoituksiin käyttäen olemassa olevia lähteitä.

Tutkielman keskeisenä tuloksena on havainnollinen esitys Nessus ohjelman toiminnalle. Johtopäätös on, että myös ilmainen tietoturvaskanneri voi olla käytännöllinen.

AVAINSANAT: Nessus, tietoturva, haavoittuvuus

SISÄLLYSLUETTELO

1 JOHDANTO

Tämä tutkimus kuuluu aihealueeseen tietoturva Internetissä. Verkossa toimivista laitteista löytyy jatkuvasti haavoittuvuuksia. Näiden haavoittuvuuksien havaitseminen ja poistaminen mahdollisimman nopeasti on tärkeää. Nessus ohjelma on suunniteltu löytämään tietoturvariskejä ja auttamaan niiden korjaamisessa. Ihanne olisi tietoturvan saavuttaminen täydellisesti, mutta todellisuudessa pyritään saavuttamaan mahdollisimman optimaalinen tilanne olemassa olevilla resursseilla.

Keskeisiin käsitteisiin kuuluu skannaaminen jolla tarkoitetaan tässä yleisesti sitä tarkastelua minkä Nessus tekee. Tietoturvaskanneri on ohjelmisto joka tarkastaa annetun verkko-osoiteavaruuden ja selvittää onko kohteeseen mahdollista tehdä tietomurto tai käyttää sitä vääriin tarkoituksiin [1].

Nessuksen tekemä skannaus sisältää porttiskannauksen sekä erilliset haavoittuvuustestit mitkä se tekee porttiskannauksen perusteella. Eli jos porttiskannauksessa ei havaita Web-palvelinta, niin kyseisiä testejä ei myöskään suoriteta.

Porttiskannaus (port scan) on toimenpide, jolla selvitetään kohdejärjestelmän aktiiviset portit [2]. Portti tarkoittaa tässä yhteydessä TCP/IP protokollan mukaista porttia (TCP tai UDP). Portin numero kertoo yleensä sen tyypin. Porttiosoite voi olla väliltä 1 - 65535. Portit jaotellaan siten, että portit välillä 1 - 1023 ovat alaportteja ja portit välillä 1024 - 65535 ovat yläportteja. Esimerkiksi TCP-portti 80 on yleisesti käytetty HTTP liikenteeseen. Näin ei kuitenkaan aina ole ja on mahdollista, että esimerkiksi Web-palvelin kuuntelee jossain muussa kuin portissa 80.

Tutkielmassa aihetta lähestytään kirjallisuuskatsauksen muodossa. Painettua kirjallisuutta on saatavilla mutta ei kovin runsaasti. Tämän tutkielman lähteet ovat www-sivustoja ja artikkeleita.

Luvussa 2 käsitellään Nessuksen asentamista ja peruskäyttöä. Luvussa 3 esitellään muutamia edistyneille käyttäjille sopivia ominaisuuksia. Luvussa 4 kerrotaan ohjelman mahdollisista käyttötarkoituksista. Luvussa 5 kerätään yhteen tutkimuksen tärkeimmät tulokset.

Tutkimuksen näkyvin tulos on selväkielinen dokumentti jota voidaan käyttää esimerkiksi Nessuksen käytön aloittamisessa. Toinen tärkeä tulos on, että ilmainen ohjelma voi olla hyvä ja käyttökelpoinen.

2 NESSUS YLEISESITTELY

Nessus projekti pyrkii tarjoamaan internet-yhteisölle ilmaisen, tehokkaan, ajan tasalla pysyvän ja helposti käytettävän tietoturvaskannerin [1]. Ajan tasalla pysyvyys tarkoittaa sitä, että Nessukseen tehdään jatkuvasti uusia skriptejä eli plugineja (plug-in) joilla varsinaisia haavoittuvuuksia testataan.

Nessus ei tee skannausta luottaen oletusportteihin eikä versionumeroihin, kuten monet muut tietoturvaskannerit. Se löytää esimerkiksi Web-palvelimen vaikka se kuuntelisi portissa 1234 ja yrittää todella käyttää hyväkseen mahdollista haavoittuvuutta. Nessus ei luota versionumeroon jonka palvelinohjelmisto antaa. [1] Tämä tarkoittaa sitä, että joissakin tapauksissa Nessus aloittaa tietyn hyökkäyksen todetakseen sen mahdollisuuden, mutta ei pyri aiheuttamaan mitään vahinkoa kohdejärjestelmään [7].

Yksi Nessuksen vahvuuksista on sen modulaarinen arkkitehtuuri. Asiakasohjelmistolla ohjataan palvelinohjelmistoa, joka tekee varsinaisen skannauksen. Tämän kaltainen arkkitehtuuri mahdollistaa palvelimen sijoittamisen erilaisiin strategisiin paikkoihin verkossa, joista skannauksen suorittaminen avaa erilaisia näköaloja. [2] Esimerkiksi skannaaminen palomuurin ulkopuolelta sisäverkkoon antaa täysin erilaisen raportin kuin skannattaessa sisäverkkoa palomuurin sisäpuolelta.

Palvelinohjelmisto on saatavilla Unix käyttöjärjestelmille ja asiakasohjelmisto sekä Windowsille että Unixille. Tenable Network Security Inc. ([5]) tarjoaa Nessus käännöksen Windows ympäristöön. Tämän ohjelman nimi on NeWT. NeWT lisenssi on ilmainen ja rajoittaa skannaamisen paikalliseen aliverkkoon. NeWT Pro lisenssi maksaa tällä hetkellä 6000 US dollaria. Saatavilla on myös Knoppix STD (security tools distribution), ns. Linux Live-cd (Linux joka käynnistyy suoraan cd:ltä eikä vaadi asennusta) jolla Nessuksen toimintaa voi

kokeilla vaivattomasti [6]. Garzan ja Rothin tekemän arvostelun [9] mukaan Knoppix STD oli helpoin tapa testata Nessusta.

Seuraavissa kappaleissa käsitellään Nessus ohjelman asennusta ja sen käyttöä pohjautuen suurimmaksi osaksi Harry Andersonin kirjoittamiin kolmeen artikkeliin ([2], [3], [4]).

2.1 Asennus

Nessus palvelinohjelmiston asennukseen tarvitaan Unix kone (toimii MacOS X, FreeBSD, Linux, Solaris jne. ympäristöissä) [1]. Tässä tarkastellaan asentamista Linux käyttöjärjestelmään. Asennus vaatii pääkäyttäjän oikeuksia.

Helpoin tapa asentaa Nessus on antaa komento (edellyttää lynx ohjelman olemassaoloa): ”lynx –source http://install.nessus.org | sh”. Tätä pidetään vaarallisena koska suoritetaan komentoja suoraan internetistä. Jos olisi meneillään ns. DNS-huijaus, saatettaisiin ajaa vahingollista ohjelmakoodia Nessus asennuksen sijaan.

On mahdollista myös ladata nessus-installer.sh asennusohjelma ohjelman kotisivuilta ja suorittaa paikallisesti komentamalla: ”sh nessus-installer.sh”. Tämä on turvallisempi tapa suorittaa asennus.

Ohjelman asennus onnistuu myös lataamalla neljä lähdekoodin sisältämää tar pakettia ja kääntämällä ne oikeassa järjestyksessä (nessus-libraries, libnasl, nessus-core ja nessus-plugins). [1]

Nessus voi hyödyntää skannauksessa muutamien ylimääräisten (eivät ole siis pakollisia) ohjelmien toimintaa: NMAP on monipuolinen porttiskanneri, Hydra on heikkojen salasanojen testaaja ja Nikto on Web-palvelimen skannaaja, joka tarkastaa kohteen monenlaisien haavoittuvuuksien, kuten esim. CGI-haavoittuvuuksien varalta [2].

2.2 Asennuksen jälkeen

Seuraavaksi täytyy lisätä käyttäjät jotka saavat käyttää palvelinta. Se tapahtuu komennolla ”nessus-adduser”. Käyttäjälle määritellään tapa jolla käyttäjä tunnistetaan (salasanalla tunnistamista suositellaan) sekä mahdollisia rajoituksia koskien skannattavia IP-osoitteita. Rajoituksen antaminen voi olla järkevää jos on syytä epäillä väärinkäytöksiä. Julkisten osoitteiden rajaaminen pois mahdollisista skannauskohteista on perusteltua jos tarkoitus on käyttää ohjelmaa yksityisen sisäverkon skannaamiseen. Käyttäjiä voidaan lisätä useampiakin jolloin yhtä palvelinta voitaisiin käyttää suuressa organisaatiossa joissa useat järjestelmänvalvojat käyttäisivät Nessusta omien aliverkkojensa skannaamiseen.

Käyttäjien lisäämisen jälkeen on luotava sertifikaatti, jolla mahdollistetaan asiakkaan ja palvelimen välisen liikenteen salaus. Tämä tapahtuu komennolla ”nessus-mkcert”. [2]

Ennen skannauksen aloittamista on tärkeää päivittää pluginit. Harry Anderson vertaa ensimmäisessä artikkelissaan plugineja virustorjuntaohjelmistojen virustunnistetietoihin. Nessus ei havaitse haavoittuvuutta, jos sillä ei ole käytössään pluginia joka tarkastaa kohteen kyseisen haavoittuvuuden varalta [10]. Pluginien päivitys onnistuu helposti antamalla komento: ”nessus-update-plugins” [2]. Päivitys kannattaa tehdä säännöllisin väliajoin tai ajastaa päivitys tehtäväksi automaattisesti.

Seuraavaksi palvelinohjelmisto (ns. daemoni eli taustalla ajettava ohjelma) voidaan käynnistää komentamalla: ”nessusd –D” . Palvelinohjelmiston mukana asennetaan samalla graafinen asiakaskäyttöliittymä joka lähtee käyntiin komennolla ”nessus” (tämän käyttäminen vaatii, että koneessa on graafinen käyttöympäristö). Palvelimen ohjaaminen onnistuu tarvittaessa myös komentoriviltä, tällöin käytetään optiota –q. Jos palvelinta halutaan käyttää muusta kuin palvelinkoneesta, niin silloin käytetään asiakaskoneen

käyttöjärjestelmälle sopivaa versiota asiakasohjelmistosta. Windowsille tämän asiakasohjelmiston nimi on NessusWX. [2]

2.3 Skannauksen valmistelu

Asiakasohjelmistolla täytyy ensin luoda yhteys palvelimeen. Asiakasohjelmistolle täytyy kertoa palvelimen IP-osoite sekä käyttäjä ja salasana jolla kirjaudutaan. Oletusportti 1241 on valmiina. Muodostettava yhteys on SSL (Secure Sockets Layer) suojattu. Palvelimelta ladataan lista asennetuista plugineista ja ensimmäisellä kerralla ladataan palvelimelta SSL sertifikaatti joka on hyväksyttävä jos aiotaan käyttää palvelinta. Sertifikaatilla varmennetaan, että jatkossa käytetään oikeaa palvelinta. [2]

2.3.1 Pluginien valinta

Pluginien valinta on tärkeä tehtävä koska ohjelman toiminnallisuus on pitkälti kiinni plugineista. Pluginit on jaoteltu toisaalta kategorian mukaan ja toisaalta vaaralliseen ja ei vaaralliseen luokkaan. Erilaisia kategorioita ovat esimerkiksi palvelunestohyökkäys, palomuurit, takaovet, Windows, RPC (remote procedure call) ja FTP. Eli jos haluttaisiin testata kohteen haavoittuvuutta pelkästään palvelunestohyökkäyksille, niin voitaisiin valita pelkästään kyseiset pluginit. Voidaan valita myös kaikki pluginit tai kaikki paitsi vaaralliset pluginit. [2]

Valintaa voi tehdä yksitellenkin, mutta se on työlästä sillä plugineja oli vuoden 2004 marraskuussa n. 5500 kappaletta. Listaa kahdestakymmenestä tärkeimmästä haavoittuvuudesta pitää yllä SANS Institute – Computer Security Education and Information Security Training internetsivusto (www.sans.org). Sivustolta löytyy myös paljon muuta hyödyllistä informaatiota tietoturva-asioista.

Pluginien valinnassa täytyy ottaa huomioon skannattavan kohteen asema. Jos skannataan tuotantokäytössä olevaa palvelinta niin vaarallisia plugineja ei tulisi valita, ja lisäksi olisi syytä valita optio ”safe-checks”. Se poistaa plugineista vaarallisia osioita ja tekee skannauksesta passiivisiin menetelmiin, kuten versionumeroiden tarkastamiseen (banner grabbing), perustuvan. [2]

2.3.2 Porttien skannaus

Porttien skannaamisella kartoitetaan kohdejärjestelmän aktiiviset portit. Käytännössä se tarkoittaa sitä, että käydään kohdejärjestelmän portteja läpi yksitellen. Porttiin lähetetään viesti (connect()-kutsu, tietynlaisen lipun omaava TCP paketti, nolla-tavuinen paketti, ICMP echo request-paketti tai jotakin muuta) ja odotetaan vastausta. Vastauksesta voidaan tulkita portin tila sekä mahdollisesti portissa kuunteleva sovellus. Porttien skannaamisella voidaan myös yrittää tunnistaa kohteen käyttöjärjestelmä.

Jos skannataan esimerkiksi portit 1-5000 ja kohteessa on Web-palvelin portissa 8080, niin Nessus ei löydä sitä. Tässä tapauksessa Web-palvelimeen kohdistuvia testejä ei suoritettaisi ollenkaan, mikäli portissa 8080 oli ainut kohteen Web-palvelin. Nessus tekee siis pluginien valintaa dynaamisesti eli turhia testejä ei suoriteta [2].

Parhaana porttiskannerina pidetty Nmap (”Network Mapper”) antaa laajan valikoiman erilaisia optioita koskien porttien skannaamista ja sitä voidaan käyttää jos Nessuksen sisäänrakennettu porttiskannneri ei kata skannaustarpeita. [3] Nmap on ilmainen työkalu ja se on saatavilla lähdekoodeineen ohjelman kotisivuilta osoitteesta http://www.insecure.org/nmap/.

Porttien skannaamiseen tulee kiinnittää erityistä huomiota jos skannataan palomuurin takana olevaa järjestelmää. Monet palomuurit estävät suuriman osan porttiskannauksista. Porttien skannaamiseen käytettävä aika, ts. siirtymäaika kahden portin välissä tulisi olla melko suuri jos aiotaan välttää skannauksen pysähtyminen palomuuriin.

2.3.3 Kohteiden valinta

Viimeisenä tärkeänä tehtävänä ennen skannauksen aloittamista on valita skannauksen kohde. Kohteita voi olla usempia ja silloin voidaan antaa erillisiä IP-osoitteita, IP-osoitteiden väli tai aliverkko. [2] Jos kohteita on useampia, ne on järkevää jaotella loogisiin ryhmiin jolloin tulosten seuraaminen on helpompaa.

On huomattava, että esimerkiksi porttiskannaus pankin tai muun virallisen tahon järjestelmään saattaa aiheuttaa rikostutkinnan ja isot sakot skannaajalle. Tämän vuoksi on syytä muistaa, että Nessuksen käyttäjälle voidaan antaa rajoitus (käyttäjän lisäämisvaiheessa, palvelimen asennuksen jälkeen) skannata vain esim. yrityksen sisäverkkoa. Kohdetta valittaessa on joka tapauksessa muistettava pyytää kohdekoneen hallitsijalta lupa skannaukseen ja tarkka aika milloin skannaus voidaan suorittaa.

2.4 Skannauksen aloitus

Seuraavaksi varsinainen skannaus voidaan aloittaa. Unix asiakasohjelmalla klikataan ”Start Scan” nappia ja NessusWX:llä (Windows asiakasohjelma) valitaan haluttu kohde ja painetaan Enter-näppäintä tai klikataan hiiren oikealla napilla ja valitaan ”Execute”. Oikein käytettynä Nessus voi löytää ongelmia ja tarjota niihin ratkaisuja. Väärinkäytettynä Nessus saattaa kaataa kohdejärjestelmän, aiheuttaa tietojen menetystä ja skannaajalle ongelmia.

Ohjelman käyttäjän vastuu on siis suuri ja ensimmäinen skannaus tulisi suorittaa omassa eristetyssä ympäristössä. [2]

2.5 Raportointi

Kun skannaus on päättynyt, niin on aika analysoida tulokset. Tuloksia voi analysoida suoraan asiakasohjelman tarjoamassa näkymässä tai tallentamalla ne jonkin muun ohjelman avulla katseltavaksi. Raportin voi tallentaa mm. ASCII, HTML, XML tai LaTeX muotoon. On myös mahdollista tallentaa SQL tiedostoksi joka voidaan tallentaa SQL-tietokantaan (löytyy myös MySQL tuki). [4]

Windows ja Unix asiakasohjelmien ero raportin tallentamisen suhteen on syytä huomata. Windows asiakas tallentaa tulokset automaattisesti mutta Unix asiakkaalla tallennus täytyy tehdä itse [4].

Tulokset joita Nessus antaa saattavat olla myös vääriä [4]. Tietoturvaskanneri ei voi olla täydellinen ohjelma ja myös kaupalliset tietoturvaskannerit antavat vääriä tuloksia. Ohjelman käyttäjän vastuulla onkin tunnistaa mitkä tulokset ovat mahdollisesti virheellisiä ja vaikuttaa siihen, että virheellisiä tuloksia saataisiin mahdollisimman vähän.

Garzan ja Rothin tekemän arvostelun mukaan Nessuksen raportointi-ominaisuudet ovat samankaltaiset kuin Internet Scanner ohjelman mutta huonommat kuin kaupallisilla ohjelmilla kuten Qualys ja Foundstone. Nessuksen heikkous raportoinnissa johtuu osittain siitä, että ei käytetä tietokantaa, jossa olisivat kaikkien edellisten skannausten tulokset. Näin ollen on siis vaikea arvioida onko tietoturvaa parannettu vai ei. Tähän ongelmaan on kuitenkin saatavilla kolmannen osapuolen tarjoama ratkaisu; Inprotect (http://sourceforge.net/projects/inprotect/) tarjoaa mahdollisuuden räätälöidä Nessukseen tietokannan johon voidaan säilöä tuloksia. [9]

2.5.1 Väärien tulosten karsinta

Nessus tuottaa joskus vääriä tuloksia. Tämä voi johtua esimerkiksi siitä, että plugini tekee tarkastuksen luottaen kohteessa olevan ohjelmiston versionumeroon (ns. banner grabbing menetelmä). Versionumeroon ei voida luottaa täysin koska aina versionumeroa ei muuteta kun ohjelmaan ajetaan korjauspäivitys haavoittuvuutta varten. Nessus raportoi kyllä käyttäjää tästä, eli ilmoittaa että kyseessä saattaa olla väärä hälytys koska käytettiin ns. ”Safe Checks” optiota tai kyseinen plugini perustui pelkästään passiiviseen versionumeron tarkastukseen. [4]

Mahdollinen väärä tulos saattaa ilmetä myös, jos Nessus saa kohteelta odottamattoman vastauksen. Esimerkiksi web-haavoittuvuuksia skannattaessa (yleensä portissa 80) web-palvelin saattaa palauttaa jotain muuta kuin standardin mukaisen ”HTTP 404” virheilmoituksen silloin kun sivua ei löydy. Tällöin Nessuksen on vaikea tulkita sitä oikein. [4]

2.5.2 Epäilyttäviä merkkejä

On syytä epäillä, että tuloksessa on jotain väärää, jos Nessus ilmoittaa jonkin tietyn palvelun haavoittuvuudesta, vaikka kohteessa ei tiettävästi olisi kyseistä palvelua. Jos Windows koneesta löytyy Unix haavoittuvuus, niin jotain on pielessä. Tämä voi tarkoittaa, että tulos on väärä, käyttöjärjestelmän tunnistusosio on epäonnistunut tai Unix haavoittuvuus on siirtynyt Windowsiin esim. kun on saatettu kyseisen Unix haavoittuvuuden omaava ohjelma Windowsissa toimivaksi. Käyttöjärjestelmän tunnistaminen onnistuu Nessukselta yleensä hyvin mutta aina se ei onnistu. Tähän voi olla syynä esimerkiksi se, että järjestelmänvalvojat muuttavat tiettyjä asetuksia tarkoituksellisesti tehdäkseen käyttöjärjestelmän tunnistamisen vaikeammaksi. [4]

2.5.3 Ratkaisun etsintä

Raportin analysoinnin jälkeen on aika etsiä ratkaisuja löydettyihin ongelmiin. Useimpiin ongelmiin Nessus tarjoaa valmiin ratkaisun tai linkin olemassa olevaan korjauspäivitykseen. Jos Nessus ei tarjoa valmista ratkaisua niin useimmiten se tarjoa kuitenkin viitteen johonkin tietokantaan jossa ylläpidetään löytyneitä haavoittuvuuksia.

Tällaisia viitteitä ovat Bugtraq (http://www.securityfocus.com/archive/1) ID(BID (http://www.securityfocus.com/bid/bugtraqid/)) ja Common Vulnerability Exposure (http://www.cve.mitre.org/) (CVE (http://www.cve.mitre.org/cve/)) numerot. BID on viitenumero joka luodaan kun haavoittuvuus lähetetään Bugtraq:n listalle. Sama numero toimii SecurityFocus:n tietokannassa johon voidaan tehdä hakuja. Tietokannassa listataan miksi tietty haavoittuvuus toimii, haavoittuvat versiot ja ratkaisut ynnä muuta relevanttia.

CVE numero toimii avaimena Mitren ylläpitämässä Common Vulnerability tietokannassa. Täältä löytyy myös jonkin verran tietoa tietystä haavoittuvuudesta, mutta pääpaino on eritellä ja identifioida haavoittuvuuksia sekä tarjota yhteys eri haavoittuvuustietokantojen välille. [4]

Nessus tarjoaa yleensä muutakin tietoa kuin pelkästään löydetyt haavoittuvuudet, kuten tiedottavia tuloksia, mm. traceroute (ohjelma, joka selvittää reitin kohteeseen) tuloksen ja arvauksen kohteen käyttöjärjestelmästä.

Kun haavoittuvuuteen on löydetty ratkaisu ja se on korjattu, niin on järkevää suorittaa uusintatesti eli ajaa sama skannaus uudelleen. Korjauspäivitykset ja ohjeet ongelman korjaamiseksi eivät aina välttämättä toimi halutulla tavalla ja saattavat avata uusia haavoittuvuuksia. Tämän vuoksi skannaus korjauksen jälkeen on erittäin suositeltavaa.

3 NESSUS EDISTYNEILLE

Enemmän hyötyä Nessuksesta saadaan irti kun käytetään hyväksi sen edistyneimpiä ominaisuuksia, kuten omien pluginien kirjoittamista. Skannaamisesta voidaan saada myös mielekkäämpää jos käytetään esimerkiksi session tallennusta, hyödynnetään edellisen skannauksen tuloksia tai ajetaan skannaus tausta-ajona.

3.1 Skriptien kirjoittaminen

Skriptien eli pluginien kirjoittaminen on mahdollista C-ohjelmointikielellä ja NASL (The Nessus attack scripting language) skriptikielellä. Skriptien kirjoittamiseen suositellaan NASL skriptikieltä, koska NASL kielellä kirjoitetut skriptit ovat helpommin siirrettäviä ja jaettavia. C-kieltä tulisi käyttää vain kun NASL kielen ilmaisuvoima ei riitä [1].

Syntaksiltaan NASL on C-kielen kaltaista, mutta muilta ominaisuuksiltaan ei. NASL-kielessä tyypityksestä ei tarvitse huolehtia eikä myöskään muistin varaamisesta tai sen vapauttamisesta. Muuttujia ei tarvitse määritellä ennen kuin niitä käyttää. NASL on pyritty suunnittelemaan niin, että sillä olisi helppoa ja nopeaa kirjoittaa turvallisuustesti. [1]

Nessuksen parissa toimiva aktiivinen yhteisö huolehtii siitä, että uuteen haavoittuvuuteen kirjoitetaan skripti erittäin nopeasti. Joku kirjoittaa skriptin, jonka jälkeen ylläpitäjä (Renaud Deraison) tarkistaa sen ja lisää tietokantaan. Tämänkaltaisen skriptin kirjoittamisesta ei siis normaalin ylläpitäjän tarvitse huolehtia.

Halutessaan Nessukseen voisi kirjoittaa skriptin, joka räätälöi sen johonkin tiettyyn tehtävään. Esimerkiksi opiskelijaverkon ylläpitäjä voisi kirjoittaa skriptin joka käyttää hyväkseen ainoastaan Peer2Peer kategorian alla olevien

skriptien ominaisuuksia ja näin ollen yhdistää niistä kattavan testipaketin, jolla havainnoida tietoverkon väärinkäyttöä.

3.2 Istunnon tallennus

Skannaaminen on aikaa vievää ja verkkokaistaa kuluttavaa toimintaa. Jos skannaus jostain syystä jumittuu, olisi mielekästä olla aloittamatta koko skannausta alusta. Istunnon (session) tallennus tuo helpotuksen tähän. Istunnon tallennusta käytettäessä voidaan käyttää tietoa edellisestä skannauksesta ja aloittaa kohdasta ennen jumittumista. [1] Isommassa ympäristössä (skannattaessa useita koneita) istunnon tallennus on lähes välttämätön ominaisuus.

3.3 Knowledge Base

Knowledge Basen käyttö on myös hyödyllinen ominaisuus. Tämän avulla esimerkiksi jo kerran skannattu kohde voidaan tarkastaa vain uusien haavoittuvuuksien osalta. Näin säästetään aikaa ja tietoliikennekaistaa. Myös pluginit voivat käyttää hyväkseen KB:ta ja jakaa tietoa sen kautta. Tämä vähentää samojen asioiden tekemistä moneen kertaan. KB:lle voidaan määritellä elinikä, ts. aika kuinka kauan sitä käytetään eli jos aika on hyvin pitkä saattaa kohdejärjestelmässä jo olla tapahtunut joitakin muutoksia joita ei huomata jos luotetaan vanhaan KB:een. [1]

3.4 Skannaaminen tausta-ajona

Skannaus voidaan suorittaa myös tausta-ajona eli ilman yhteyttä asiakasohjelmiston ja palvelinohjelmiston välillä. Tämä on hyödyksi jos halutaan ajastaa skannaus suoritettavaksi aina tietyin väliajoin. Tällä on merkitystä esimerkiksi ylläpidon kannalta. Uuden haavoittuvuuden löytyessä Nessus voidaan määritellä lähettämään tulos sähköpostilla ylläpitäjälle.

Voidaan myös ottaa välillä yhteys palvelimeen ja ladata tulokset asiakasohjelmaan. [1]

3.5 Paikallinen skannaus

Kun skannataan kohdetta ilman muuta tietoa kuin IP-osoite, ei saada kaikkea tietoa mitä ehkä haluttaisiin sekä saatetaan saada vääriä tuloksia. Nessus versiosta 2.1.0 lähtien on mahdollisuus käyttää paikallisia turvallisuustestejä. Tämä koskee Unix tyyppisiä koneita. Aluksi luodaan SSH-avainpari. Jokaiseen skannattavaan järjestelmään on luotava käyttäjätili. Julkinen avain kopioidaan luodun käyttäjätilin hakemistoon. Salainen avain pidetään koneessa josta skannausta ohjataan eli Nessus asiakaskoneessa.

Tällaisella menettelyllä Nessus voi kirjautua järjestelmään ja saadaan yksityiskohtaisempaa tietoa kohteesta. Voidaan helposti pitää kirjaa kaikista puuttuvista korjauspäivityksistä ilman mahdollisia vääriä tuloksia tai riskiä kohteen jumiutumisesta. Skannaaminen voidaan suorittaa tällä tavalla nopeammin ja myös useammin.

3.6 Windows kohteen skannaus

Jotkin pluginit (Windows) tarvitsevat tunnuksen jolla on pääsy rekisteriin, esim. järjestelmänvalvojan tunnuksen. Kun Nessukselle annetaan käyttäjänimi ja salasana skannattavaan kohteeseen, skannaus antaa enemmän tietoa, ts. löytää enemmän haavoittuvuuksia. Se voi löytää rekisteristä tietoa joistain viruksista tai päivityksistä joiden olemassaolo ei selviäisi ilman pääsyä rekisteriin. Skannattavalle Windows kohteelle olisi hyvä luoda tietynlainen käyttäjätili Nessusta varten koska järjestelmänvalvojan tunnuksella skannattessa on riskinsä (rekisteriin tarvitaan vain lukuoikeus) ja tunnusta ei välttämättä ole käytettävissä.

4 KÄYTTÖTARKOITUS

Nessus voi toimia erilaisissa käyttötarkoituksissa. Ensisijaisesti se voi olla osa tietokoneverkon ylläpidon työvälineistöä. Osana tietoturvariskien hallintaa sillä voisi olla tärkeä rooli. Sitä voitaisiin käyttää myös tietoturvakursseilla oppilaitoksissa yhtenä opetusvälineenä, kuten Pohjois-Kentuckyn yliopistossa on tehty [8].

4.1 Ylläpitäjän työväline

ATK-suunnittelijan huoli tietoturvasta on ilmeinen. Esimerkiksi Sasser-madon aiheuttamilta ongelmilta olisi voitu välttyä Nessuksen avulla. Nessus voisi toimia yhtenä osana tietoturvapolitiikkaa. Käytännössä Nessus palvelin voitaisiin sijoittaa muiden palvelimien joukkoon josta sitä käskettäisiin tekemään tarkastuksia aina tarpeen mukaan. Se voisi myös tehdä skannauksen automaattisesti sopivin väliajoin ja ilmoittaa tuloksista ylläpitäjän sähköpostiin.

Tällainen järjestely olisi järkevää tilanteessa, jossa verkossa on hyvin erilaisia koneita, kuten Microsoft Windowsin eri versioita (joitakin päivitetään automaattisesti, joitakin satunnaisesti ja joitakin ei ollenkaan), erilaisia Unix koneita ja Macintosheja. Tällaisessa tilanteessa potentiaalisten tietoturva uhkien havaitseminen on tärkeää koska systemaattinen tietoturvapäivitysten jakelu erilaisille koneille on vaikeaa. Järkevää olisi, että kaikille koneille olisi luotu tietty käyttäjätili ainoastaan Nessuksen käytettäväksi. Edellä mainittiin mitä hyötyjä tällä saavutetaan.

Windows päivityksiä ei ehkä aina haluta asentaa automaattisesti niiden mahdollisten sivuvaikutusten vuoksi, tällöin on erityisen tärkeää havaita kuinka potentiaalisia tietoturva riskejä päivityksen asentamatta jättäminen synnyttää ja mitkä niistä on pakko korjata. Esimerkiksi ilman Windowsin

rekisteriin pääsyä ei välttämättä saada tietoa mitä päivityksiä kohteeseen on asennettu.

4.2 Opetusväline

Pohjois-Kentuckyn Yliopistossa on tietoturvaa opetettu osana Unix-ylläpito kurssia. Myöhemmin tietoturva on eriytetty omaksi kurssikseen. Opetukseen yritetään sisällyttää käytännön harjoituksia koska suurin osa opiskelijoista siirtyy opiskelujen jälkeen suoraan työhön.

Laboratorioharjoituksissa on käytetty kolmea ilmaista ohjelmaa: COPS (The Computer Oracle and Password System), Nutcracker ja Nessus. Näistä ensimmäinen tutkii mm. väärin suojattuja systeemitiedostoja. Toinen ohjelma on yksinkertainen salasanan murtaja. Kaikki nämä työkalut ovat siis UNIX ohjelmia. Laboratorioharjoitukset suoritettiin eristetyssä Linux luokassa.

Opiskelijat saavat näillä ohjelmilla harjoitella tietoturvan parantamista käytännössä. Tietokoneisiin on istutettu tahallaan virheitä joita oppilaat löytävät käyttäen näitä apuohjelmia. Oppilaiden täytyy selvittää mistä vika johtui ja miksi työkalu löysi vian.

Käytännön harjoitukset tietoturvakurssilla ovat varmasti erittäin hyödyllisiä. Tietoturva-asiantuntijoita tarvitaan työelämässä ja on hyvä, että opetus on käytännönläheistä jolloin työhön siirtyminen helpottuu.

5 YHTEENVETO

Tässä tutkielmassa tarkasteltiin Nessuksen asentamista ja käyttöä sekä pohdittiin sen mahdollisia käyttötarkoituksia. Nessuksen asentaminen ja käyttäminen on helppoa aloittaa koska verkosta löytyy kattava ja hyvä englanninkielinen dokumentointi. Ohjelman ympärillä toimii aktiivinen käyttäjäyhteisö ja mahdollisiin ongelmiin saa apua keskusteluryhmistä.

Ohjelman asentaminen vaatii perustuntemusta Windows ja Unix ympäristöistä. Menestyksekäs käyttäminen vaatii käyttäjältä paljon perehtymistä ohjelman erilaisiin asetuksiin sekä perustietoja tietotekniikasta ja tietoliikenteestä. Erilaisten käsitteiden, kuten porttiskannaamisen, tunteminen on välttämätöntä halutunlaisen toiminnan aikaansaamiseksi.

Kyseessä on ilmainen työkalu jolla on paljon käyttöä. Sitä voidaan käyttää yrityksissä ja yhteisöissä tietoturvan parantamiseen. Sillä on myös potentiaalinen käyttötarkoitus opetusvälineenä tietoturvakursseilla kuten Pohjois-Kentuckyn esimerkki osoittaa.

LÄHDELUETTELO

[1] Deraison R. 2004. The Nessus Project. Saatavilla www-osoitteessa <www.nessus.org>.

[2] Anderson H. 2003. Introduction to Nessus. SecurityFocus. Saatavilla www-osoitteessa <http://www.securityfocus.com/infocus/1741>.

[3] Anderson H. 2003. Nessus, Part 2: Scanning. SecurityFocus. Saatavilla www-osoitteessa <http://www.securityfocus.com/infocus/1753>.

[4] Anderson H. 2004. Nessus, Part 3: Analysing Reports. SecurityFocus. Saatavilla www-osoitteessa <http://www.securityfocus.com/infocus/1759>.

[5] Tenable Network Security Inc. 2002. Vulnerability Management and Intrusion Detection. Saatavilla www-osoitteessa <http://www.tenablesecurity.com/newt.html>.

[6] Knoppix STD 2004. Knoppix STD 0.1 security tools distribution. Saatavilla www-osoitteessa <http://www.knoppix-std.org/>.

[7] Mick Bauer, 2004. Paranoid penguin: seven top security tools. Linux Journal, 2004(118), 12.

[8] Charles E. Frank, Gregory A. Wells. 2002. Laboratory exercises for a computer security course. The Journal of Computing in Small Colleges 17(4), 51-54.

[9] Garza V. R., Roth J. R. 2003. No-frills security scanning (review). InfoWorld. Saatavilla www-osoitteessa (vaatii ilmaisen rekisteröitymisen) http://www.infoworld.com/infoworld/article/03/09/05/35TCvuln_1.html.

[10] Mick Bauer, 2001. Paranoid penguin: checking your work with scanners, part II: Nessus. Linux Journal, 2001(86), 11.