

Verkkovarmenteiden testausraportti

Tatu Erkinjuntti, Jussi Isosomppi, Eino Kupias, Saku Kähäri

Raportti

Tietojenkäsittelyn koulutusohjelma

2018

Sisällys

[1 Esipuhe 1](#_Toc532399585)

[2 Testausraportti: Välimieshyökkäys ja SSL-salatun liikenteen murtaminen 2](#_Toc532399586)

[2.1 Välimieshyökkäys (kevyt) 2](#_Toc532399587)

[2.2 Välimieshyökkäys (korkeatasoisempi) 3](#_Toc532399588)

[3 Testausraportti: Sertifikaattipalvelimeen kohdistuva palvelunestohyökkäys 5](#_Toc532399589)

[3.1 Teoria 5](#_Toc532399590)

[3.2 Käytäntö 5](#_Toc532399591)

[3.3 Lopputulos 5](#_Toc532399592)

[4 Testausraportti: OCSP-viestiliikenteen häirintä 6](#_Toc532399593)

[4.1 Testaus: OCSP-vastausten muuttaminen 6](#_Toc532399594)

[4.2 Testaus: OCSP-vastausten täysi blokkaaminen 6](#_Toc532399595)

[4.3 Lopputulos 7](#_Toc532399596)

[5 Testausraportti: Välimieshyökkäys Firefoxiin etänä 8](#_Toc532399597)

[5.1 Testaus 8](#_Toc532399598)

[5.2 Lopputulos 8](#_Toc532399599)

# Esipuhe

Tämä dokumentti sisältää verkkovarmenteiden testausraportit kootusti.   
Testaukset suoritettiin Haaga-Helian Ammattikorkeakoulun laboratorioluokassa (5004). Ellei toisin mainita, hyökkäyskoneessa on käytetty Kali Linux käyttöjärjestelmää.   
Kohdekoneiden käyttöjärjestelminä on ollut Windows 10 tai Antergos Linux.

# Testausraportti: Välimieshyökkäys ja SSL-salatun liikenteen murtaminen

**Työkalut**

Bettercap on Linux-ohjelma, joka on suunniteltu verkkoliikenteen kaappaamiseen. Siihen integroidut toiminnallisuudet, kuten SSLsniff ja SSLstrip, mahdollistavat (tietyin varauksin) myös lähtökohtaisesti salatun liikenteen pakettien kaappaamisen. Bettercapin käyttömahdollisuudet verkkoliikenteen kartoittamisessa ja kaappaamisessa ovat laajat, aina yksinkertaisesta verkon kartoituksesta sertifikaattien automaattiseen väärentämiseen ja verkkosivujen salatun liikenteen huomaamattomaan kaappaamiseen asti. Lisätietoa Bettercapistä löytyy projektin GitHub-sivulla: <https://github.com/bettercap/bettercap>

**Skenaario – Oletukset**

Testauksemme oletti seuraavaa alkutilanteesta:

• Hyökkääjällä on pääsy kohdeverkkoon

• Kohdehenkilöt eivät suhtaudu vainoharhaisesti tietoturvaan

• Verkonvalvontaa tai virustorjuntaa ei tässä yhteydessä huomioida

## Välimieshyökkäys (kevyt)

Tarkoituksenamme oli luoda tilanne, jossa yrityksen käyttäjien verkkoliikennettä kohtaan voidaan hyökätä niin, etteivät käyttäjät itse edes huomaa, että jokin on vialla. Luomassamme tilanteessa kohdeverkkoon asetetaan ARP-hyökkäystä ajava palvelin, joka ohjaa verkkoliikenteen kulkemaan itsensä kautta. Palvelin voidaan määrittää välittämään liikennettä esim. Bettercapin oman proxyn avulla. Tässä tilanteessa liikenne kulkee proxyn kautta muokkaamattomana, ja ainoastaan salaamatonta verkkoliikennettä voidaan lukea.

Kokeilimme välimieshyökkäyksen toteuttamista muutamalla eri kokoonpanolla,   
yhtä (omaa) maalikonetta vastaan.

**Tulokset olivat pääosin positiiviset:**

* Noin 90% ajasta kohdekoneella ei ollut havaittavissa mitään erikoista tietoliikenteessä.
* Noin 10% ajasta kohdekoneella ei saanut lainkaan yhteyttä internetiin, pääosin huonosti määritettyjen proxy-asetusten takia.
* Salaamattomasta liikenteestä saatiin kaapattua suoraan ladatut verkkosivut, yksittäiset kuvatiedostot jne.
* Salatusta liikenteestä saatiin tietoon yhteyden kohde, protokolla sekä sivulatausten tiheys.

## Välimieshyökkäys (korkeatasoisempi)

Aiempaa välimieshyökkäystä voidaan kehittää entisestään liittämällä mukaan salatun liikenteen purku. Bettercapin tarjoamalla SSLstrip-toiminnolla voidaan purkaa tätä salausta niin, että ladatut sivut käyttävät Bettercapin luomaa sertifikaattia alkuperäisen sisään. Tämän jälkeen liikenne puretaan Bettercap:ia ajavalla palvelimella, ja salataan uudelleen alkuperäisellä avaimella. Jotta tämä hyökkäys toimii käytännössä, vaaditaan Bettercapin root-sertifikaatin asentaminen kohdekoneen selaimeen (johon vaaditaan koodin ajamista kohdekoneessa, mutta kuitenkin vain käyttäjän oikeuksilla).

**Tässä tapauksessa hyökkäys etenee siis seuraavalla tavalla:**

* Kohdeverkossa toimii ARP-hyökkäystä ajava palvelin (joko sinne itse salaa asennettu tai olemassa olevasta koneesta saastutettu)  
  + - ARP-hyökkäyksen kautta saadaan selville verkossa olevien koneiden tietoja (IP-osoitteita, hostnameja, tietoliikennettä)
    - Kerättyjen tietojen perusteella voidaan tunnistaa yksittäisiä työntekijöitä (kiinnostuksen kohteet, profilointi vs. selaushistoria)
* Valitaan kohteeksi sopivia työntekijöitä, joiden vastuut ovat sopivan suuria ja tietotaso riittävän alhainen  
  + - Kohteet voidaan valita avoimesti saatavilla olevan tiedon (yrityksen kotisivut, työntekijän sosiaalinen media jne) sekä kerätyn selaushistorian perusteella
    - Kohteita vastaan voidaan hyökätä jollain keinolla, joka mahdollistaa oman koodin ajamisen tämän koneella (kuten sähköpostin liitetiedostot)
* Kun kohdekone on saastutettu Bettercapin haitallisella root-sertifikaatilla, voidaan purkaa salattua liikennettä lennosta

**Hyökkäyksen ongelmat:**

* Root-sertifikaatin asentaminen kohdekoneeseen vaatii jonkinasteista pääsyä. Vaikka tähän riittää saastuneen linkin tai liitetiedoston avaaminen, sulkee tämä silti suuren osan kohteista pois.
* Kaikilla nettisivuilla on tämän jälkeen saman yrityksen takaama sertifikaatti. Jos käyttäjä klikkailee usein sertifikaatin tietoja auki (tai käyttää selainta, joka näyttää sertifikaatin myöntäjän osoitepalkissa), alkaa tämä nopeasti epäilyttää.
* Sertifikaattien väärentämistä vastaan on kehitetty Certificate Pinning, jossa verkkosivun tiedoissa määritetään sertifikaatit, joilla sen liikenne saadaan salata. Testauksen perusteella suuret kansainväliset palvelut (Google, Facebook, Amazon) käyttävät tätä palvelua, kun taas suomalaisista palveluista suurimmatkaan (Yle, Helsingin Sanomat) eivät ole ottaneet sitä käyttöön. Certificate Pinningiä käyttävät sivustot huomaavat hyökkäyksen välittömästi, eivätkä toimi lainkaan.

# Testausraportti: Sertifikaattipalvelimeen kohdistuva palvelunestohyökkäys

## Teoria

Eräänä keinona sertifikaattijärjestelmän häirintään nousi esiin sertifikaattipalvelimeen (Certificate Authority, CA) kohdistuva palvelunestohyökkäys (Distributed Denial of Service, DDoS). Tässä tarkoituksena on häiritä sertifikaatteja vahvistaneen juuripalvelimen toimintaa niin paljon, että sen toiminta hidastuu huomattavasti tai jopa estyy kokonaan. Tämä johtaisi tilanteeseen, jossa OCSP-viestit eivät koskaan saa vastausta (ja näin ollen johtavat virheeseen), eikä uusia sertifikaatteja voida myöntää tai vanhoja muokata.

Tällaiseen lopputulokseen johtavan hyökkäyksen toteuttaminen lienee vaikeaa, muttei missään tapauksessa mahdotonta. Yleisimmät sertifikaatteja myöntävät tahot ovat varmasti suojanneet palvelunsa tällaista hyökkäystä vastaan, mutta esimerkiksi pienen organisaation sisäistä CA-palvelinta vastaan hyökkääminen saattaisi hyvinkin olla mahdollista.

## Käytäntö

Vaikka CA:ta vastaan kohdistuva DDoS-hyökkäys on periaattessa helppo käsittää ja toteuttaa, kohtasimme tässä kohtaa totuuden tietoturvatutkimuksen rajoista: lainsäädäntö rajoittaa erittäin tarkasti mahdollisuuksia. Käytännössä ainoa laillinen tapa tämän teorian testaukseen olisi ollut kokonaan oman infrastruktuurin pystyttäminen. Periaatteessa projektin aikana kerrytetyt tiedot CA-palvelinten ja sertifikaattijärjestelmien toiminnasta riittivätkin kyseisenlaisen ratkaisun hahmottamiseen, ei sen toteuttaminen käytännössä ollut kuitenkaan sen vaatiman ajan, vaivan ja riskin arvoista.

## Lopputulos

Sertifikaattipalvelimeen kohdistuva palvelunestohyökkäys vahingoittaisi kyseisen palvelun käyttäjiä hetkellisesti, mutta vaatisi valtavan määrän resursseja ja olisi lainsäädännölliseltä kannalta erittäin riskialtista.

# Testausraportti: OCSP-viestiliikenteen häirintä

**Työkalut**

Käytimme tässä testauksessa työkaluina Burp Suitea sekä OWASPin Zed Attack Proxyä (ZAP). Näiden verkkoliikenneanalyysityökalujen avulla teimme hallittua liikenteen uudelleenohjausta ja suodatusta, jonka tavoitteena oli häiritä tai kokonaan estää Online Certificate Status Protocol (OCSP) -viestien kulkua.

**Tavoitteet**

Tällä liikenteen häirinnällä oli tarkoitus joko saada vanhentuneet tai käytöstä poistuneet sertifikaatit näyttämään käyttäjän silmissä toimivilta, tai aiheuttaa virheitä täysin kelvollisten sertifikaattien käytöstä. Näistä tavoitteista ensimmäinen soveltuu hyvin mm. tietoturvarikkeen keston pidentämiseen (tietoturvarikkeen takia kuoletettu sertifikaatti näyttää käyttäjälle edelleen täysin toimivalta), ja jälkimmäinen lähinnä tietoliikenteen häirintään (verkkosivujen lataaminen johtaa jatkuviin tietoturvavirheilmoituksiin).

## Testaus: OCSP-vastausten muuttaminen

Koitimme aluksi muuttaa OCSP vastauksia dynaamisesti niin, että jokainen sivunlataus johtaisi "vanhentuneesta/epäkelvosta sertifikaatista" kertovaan virheilmoitukseen. Vaikka saimmekin napattua OCSP-viestit suoraan tietoliikenteestä (ne lähetetään salaamattomina), emme onnistuneet muokkaamaan niitä dynaamisesti. Tästä johtuen viesteihin upotettu aikakoodi esti tämän lähestymistavan toiminnan.

## Testaus: OCSP-vastausten täysi blokkaaminen

Siirryimme pian kokeilemaan OCSP-viestien täyttä estämistä, jonka oli tarkoitus johtaa samaan lopputulokseen. Käytännössä tämän toteuttaminen oli huomattavan helppoa (kaikki "ocsp.\*" -osoitteisiin lähetetyt paketit pudotettiin), josta ei kuitenkaan seurannut minkäänlaista muutosta verkkosivustojen toimintaan. Epäilemme tästä syyksi työkalujen vaatiman oman root-sertifikaatin käyttöä, joka todennäköisesti sai OCSP-vastauksensa niin, ettei proxymme päässyt siihen puuttumaan.

## Lopputulos

OCSP-vastausten muuttaminen osoittautui niin hankalaksi tehtäväksi, ettemme sitä lähteneet aikarajoitteiden vuoksi liiemmin testaamaan. Jos sen kautta haluttaisiin saavuttaa tuloksia, olisi oman pyyntöjä muokkaavan koodin kirjoittaminen välttämätöntä. Vain reaaliaikaisesti automaattisesti muokattavat pyynnöt saattaisivat onnistua.

OCSP-liikenteen täysi estäminen taas voisi toimia, mikäli se suoritettaisiin pelkällä välityspalvelimella, jonka käyttö ei vaatisi sertifikaatteihin puuttumista. Tässäkin tapauksessa aikarajoitteet estivät syvällisemmän tutustumisen.

# Testausraportti: Välimieshyökkäys Firefoxiin etänä

**Työkalut ja tavoitteet**

Tämän testauksen pääosaan nousi Mozilla Network Security Services (NSS), joka on suunniteltu organisaation Firefox-selainten etähallintaan. NSS:n avulla on mahdollista hallita selainten sertifikaatteja, joten se oli selkeä valinta tässä yhteydessä. Tarkoituksena oli luoda Bettercapin avulla oma root-sertifikaatti, joka lisättäisiin kohdekoneen selaimeen hyökkäysyrityksiä naamioimaan. Tarkoituksena oli automatisoida tämä prosessi niin, että se olisi mahdollista suorittaa esimerkiksi peukaloidun selaimen liitetiedoston kautta.

## Testaus

Vaikka Mozillan oma NSS-työkalu vaikuttikin melko helppokäyttöiseltä, oli sen toimimaan saaminen kuitenkin erittäin hankalaa. Vaikka saimmekin lopuksi sertifikaatin lisättyä suunnitellulla tavalla, osoittautui tämän prosessin paketoiminen hyökkäykseksi kuitenkin liian haastavaksi tehtäväksi. Sertifikaatin lisääminen oli hankalaa myös siksi, että kyseiseen prosessiin vaadittiin satunnaisesti generoidun profiilitunnuksen selvittäminen ja kohdistaminen.

## Lopputulos

Emme tutkineet tätä haavoittuvuutta kovin pitkää aikaa, sillä totesimme sen olevan kaiken jälkeen sittenkin liian tehoton. Tässä yhteydessä jouduimme punnitsemaan hyökkääjän kannalta teon hyödyllisyyttä, riskialttiutta ja monimutkaisuutta. Hyökkäys olisi kuitenkin todennäköisesti jäänyt kiinni heti virusturvan huomatessa sen.