Tampereen ammattikorkeakoulu

Web-Sovellustestaus

Joonas Heikkilä

OPINNÄYTETYÖ
Joulukuu 2020

Tieto- ja viestintätekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintätekniikka
Ohjelmistotekniikka

Joonas Heikkilä:
Web-Sovellustestaus

Opinnäytetyö 34 sivua
Joulukuu 2020

Opinnäytetyö opastaa aiheesta kiinnostuvaa keskiverto tietokoneenkäyttäjää web-sovellustestauksen alkuvaiheissa. Lukija tarvitsee käyttöjärjestelmään asti valmiin Linux pohjaisen tietokoneen parhaan tuloksen saavuttamiseksi. Opinnäytetyön tarkoitus on olla ohje uusille harjoittelijoille, jotka eivät vielä osaa käyttää Burp Suite Prota.

Opinnäytetyön keskeinen työkalu on Burp Suite Pro. Suuri osa työtä on Burp Suite Pron käyttöönotto ja yleiset toiminnot. Käydään läpi myös muita testausta täydentäviä työkaluja, kuten esimerkiksi SQLmap, SSLscan ja Nmap. Työkalujen käyttöönoton jälkeen katsotaan yleisiä merkittäviä havaintoja mitä web-sovellustestauksessa tulee vastaan.

Opinnäytetyön pohjalta keskiverto tietokoneenkäyttäjä voi tehdä web-sovellustestausta. Työ käsittelee työkaluja pinnallisella tasolla ja, jotta lukija pystyy testaamaan keskiverto testaajan tasolla, tarvitsee hänen lukea asiasta enemmän esimerkiksi OWASP ASVS standardista. Tätä opinnäytetyötä voisi kehittää selittämään työkaluja enemmän ja syvällisemmin. Voisi lisätä myös matalan tason havaintoja ja harvinaisempia korkean tason havaintoja.

Asiasanat: Linux, web-sovellustestaus, Burp Suite Pro, SQLmap, SSLscan, Nmap

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Information and communication engineer
Software engineering

Joonas Heikkilä:
Web application penetration testing

Bachelor's thesis 34 pages
December 2020

This paper guides average computer user, that is interested in the subject, to help with early phases of web application penetration testing. For this guide to be most effective, you need working Linux machine. The purpose of this paper is to guide new trainees to the use of Burp Suite pro and other tools.

The paper uses Burp Suite Pro as a main tool for web application penetration testing. Most of the paper is how to setup Burp Suite Pro and what features does Burp Suite Pro have. The paper also looks at other tools that assist the tester for example SQLmap, SSLscan and Nmap. After the tools there are significant findings that are present in web applications often.

With this paper average computer user can do web application penetration testing. This paper just scratches the surface on the features and uses of the tools, and for the reader to be effective in web application penetration testing they need to read more information for example in OWASP ASVS standards. This paper could be improved on explaining the tools better. One could also add low-level findings and less common high-level findings to the paper.

Key words: Linux, web application penetration testing, Burp Suite Pro, SQLmap, SSLscan, Nmap

SISÄLLYS

LYHENTEET JA TERMIT

DNS	Domain name system
DVWA	Damn Vulnerable Web Application
HTTPS	Hypertext Transfer Protocol Secure
KPMG	Klynveld Peat Marwick Goerdeler
MAC	Message Authentication Code
TLS	Transport Layer Security
VPN	Virtual Private Network
XSS	Cross Site Scripting

1 JOHDANTO

Tämä opinnäytetyö käsittelee web-testausta tietoturvan näkökulmasta. Työssä käydään läpi testauksen työkaluja, niiden käyttöönottoa sekä käyttöä.

Läpi käydään myös Web-sovellustestauksessa yleisesti käytettyjä sovelluksia, sekä niiden mahdollisia haavoittuvuuksia. Tämän työn web-sovellustestaus tehdään ohjelmalla Burp Suite Pro. Burp Suite Pro toimii välityspalvelimena, joka mahdollistaa kutsujen näkemisen ja muokkaamisen. Burp Suite Pro on kuitenkin maksullinen ohjelma, ja tähän käytetään KPMG (Klynveld Peat Marwick Goerdeler) tarjoamaa lisenssiä.

Web-sovellustestaaminen on laitonta ilman sovelluksen haltian lupaa. Lain mukaan ”Joka käyttämällä hänelle kuulumatonta käyttäjätunnusta taikka turvajärjestelyn muuten murtamalla oikeudettomasti tunkeutuu tietojärjestelmään” ” on tuomittava tietomurrosta sakkoon tai vankeuteen enintään kahdeksi vuodeksi” (Rikoslaki 2015/368 § 8). Kannattaa huomioida myös, että pelkästään yrityksestä voidaan jo rankaista. Lain mukaan lievän tietoliikenteen häirinnän mukaan myös porttien skannaus julkisessa verkossa on laitonta (Rikoslaki 1995/578 § 7). Internettiä kuitenkin skannataan jatkuvasti, joten ei kannata luottaa Suomen lakiin estämään oman palvelimen joutumista skannauksen kohteeksi. Käytämme aina auktorisointikirjettä, jossa meille annetaan lupa tehdä muuten rangaistavia hyökkäyksiä ja skannauksia.

Työssä on käytetty Googlen julkista DNS (domain name system) sivua (google dns). Web-sovellustestausympäristönä on käytetty dvwa:ta (*Damn Vulnerable Web Application*).

Opinnäytetyön toisessa kappaleessa käydään läpi web-sovellustestauksen rajoituksia, riskienhallintaa ja aikataulua kuinka paljon itsellä kuluu aikaa projektin suorittamiseen. Kolmannessa kappaleessa kerrotaan yleisesti mitä web-sovellustestaus on ja miksi web-sovellustestausta tehdään. Käydään läpi myös standardeja, joiden pohjalta web-sovelluksia testataan. Neljännessä kappaleessa käydään läpi web-sovellustestauksen yleisiä työkaluja ja esimerkkejä, kuinka

niitä voi käyttää. Viidennessä kappaleessa käydään läpi Burp Suite Pro:n käyttöönottoa ja ominaisuuksia. Kuudennessa kappaleessa käydään läpi yleisiä haavoittuvuuksia, joita web-sovellustestauksessa tulee vastaan, sekä kuinka ne havaitaan. Viimeisenä kappaleena on yhteenveto opinnäytetyöstä.

2 TYÖN TAVOITE JA TARKOITUS

Työn tavoite on opastaa keskiverto tietokoneen käyttäjä, joka aiheesta on kiinnostunut, web-sovellustestauksen alkuvaiheista eteenpäin. Työn tarkoitus on opastaa uusia henkilöitä web-sovellustestauksen alussa.

Web-sovellustestaus on verkossa toimivan sovelluksen testausta, tämä käydään seuraavassa kappaleessa tarkemmin läpi. Web-sovellustestauksen tarkoitus on tehdä jokaisen internetin käyttäjän arjesta turvallista. Tämänhetkisen maailmantilanteen takia suuri osa joutuu käyttämään web-sovelluksia useita kertoja päivässä. Tämä tuo enemmän arkaluontoista tietoa Internettiin, ja tätä myötä myös pahantahtoisia henkilöitä, jotka yrittävät saada tätä arkaluontoista tietoa omiin käsiin rahallisen hyödyn toivossa (Security magazine 2020).

2.1 Rajoitukset

Web-sovellustestauksessa rajataan ulos yleensä koodikatselmointi, palvelimien sijainti ja fyysinen turvallisuus sekä hallinnolliset tietoturvakontrollit pois. Verkkoyhteyden saaminen voi olla rajoittava tekijä VPN (Virtual Private Network) sovelluksen takia. VPN sovellus luo salatun yhteyden tietokoneen ja VPN palvelimen välille. Tästä yhteys usein pääsee VPN palvelimen sisäverkkoon ja tätä kautta saa yhteyden tarkastettavaan palvelimeen. VPN sovellus voi esimerkiksi rajoittaa mitä käyttöjärjestelmää voidaan testauksessa käyttää.

Web-sovellustestausta ei aloiteta ennen kuin on saatu asiakkaalta allekirjoitettu auktorisointikirje, joka takaa, että asiakas ei voi syyttää tietomurrosta. Auktorisointikirje kertoo mitä menetelmiä työssä voidaan käyttää ja asiakas lisää myös tarkistettavat kohteet ja mahdolliset muut testauksessa vastaan tulevat resurssit, kuten puhelinnumerot, IP osoitteet ja verkkosivujen osoitteet.

2.2 Riskienhallinta

Web-sovellustestauksessa riskejä on vähän. Sovelluksen voi saada kaadettua, mikä voi johtaa jonkinlaiseen tiedon menetykseen. Tämä on kuitenkin pieni on-

gelma testiympäristöissä. Injektiohyökkäyksessä saadaan palvelimelle syöttökenttien avulla koodia, joka lähtee suoritukseen, tai saa aikaan vikatilan sovelluksessa. Injektiohyökkäyksellä voidaan saada tietokanta vastaamaan, mutta tarkoituksena ei ole rikkoa ympäristöä, joten ”*Drop tables”*-tyyppisiä komentoja ei suoriteta (drop tables). ”Drop tables”-tyyppinen komento poistaa mahdollisesti koko tietokannan.

2.3 Aikataulu

Oman arvion mukaan, web-sovellustestauksen aikataulu vaihtelee paljon sovelluksen koosta ja toiminnoista. Sovellus, joka näyttää vain staattista dataa ei vie paljoa aikaa. Tällaisen voi tarkistaa hyvin päivässä. Kun taas esimerkiksi Gmail web-sovelluksen testauksessa menisi noin 2 viikkoa kahdelta henkilöltä.

Omissa projekteissa palaverit vievät oman aikansa. Näihin varataan yleensä yksi päivä. Usein palavereissa on mukana kaksi henkilöä ja tämä tuplaa kuluvat tunnit.

3 MITÄ ON WEB-SOVELLUSTESTAUS

Web-sovellustestaus on verkossa toimivan sovelluksen tarkastusta. Tähän tarvitaan yhteys sovellukseen. Tämä voi olla julkiverkon yli, VPN yhteydellä tai sovelluksen tuottajan tiloista sisäverkon yli. Web-sovelluksilla on yleensä edustapalvelin, taustapalvelin ja erillinen tietokantapalvelin. Näistä käyttäjälle näkyy edustapalvelin, ja tätä web-sovellustestauksessa testataan. Ideaalista olisi, jos taustapalvelimesta ei näkyisi mitään tietoa edustapalvelimelle. Tämä on kuitenkin melko vaikeaa, sillä sovelluskirjastot ja niiden versionumerot vaikuttavat käyttäjän selaimen toimintaan. Sovellus ei välttämättä toimi oikein, jos selain ei tiedä voiko kirjastoa käyttää viimeisimmän ominaisuuden toistamiseen. Esimerkiksi jQuery kirjaston toiminta on riippuvainen versionumerosta (jquey versionumero).

Web-sovellustestaus kattaa web-sovelluksen ohjelmiston tietoturvan. Web-testausta tehdään tietoturvan takia, jotta tavallinen kansalainen voi kirjautua turvallisesti esimerkiksi Kelan sivuille ilman, että hänen tarvitsee miettiä välimieshyökkäyksiä ja henkilötietojen vuotamista vääriin käsiin. Välimieshyökkäyksessä käyttäjän ja sovelluksen välinen liikenne kiertää pahantahtoisen henkilön laitteen läpi. Laitteessa liikennettä voidaan muuttaa reaaliajassa tai ottaa talteen ja etsiä arkaluontoista tietoa myöhemmin. Ilman web-testausta sivut olisi tehty turvallisen näköiseksi, että saadaan asiakkaille turvallinen olo. Todellisuudessa nämä olisivat vain temppuja. Esimerkiksi lisäämällä liitetiedoston lataamiseen rajoitus tiedostopäätteistä tai tiedoston koosta, mutta ei oikeasti tarkasteta näitä. Web-testauksella varmistetaan, että haavoittuvuudet on oikeasti ratkaistu ja korjattu (relevant software).

3.1 Kaikki vihreät lukot eivät ole samanarvoisia

Selaimen osoite palkissa on lukko, joka tarkoittaa, että yhteys sovellukseen on salattu. Tämä salaus voi olla heikko, joka mahdollistaa välimieshyökkäyksen. Vahvan salauksen kanssa pahantahtoinen henkilö ei saa salattua liikennettä purettua, ja tätä kautta välimieshyökkäys estetään. Esimerkiksi HTTPS salaus on vain niin hyvä kuin web-sovelluksen palvelin sen vaatii olevan. Tämä ei näy käyttäjälle onko salaus heikko TLS_RSA_3DES_EDE_CBC_SHA vai vahvempi

TLS_ECDHE_ECDSA_AES_256_GCM_SHA384 (jscape 2018). Edellä mainituissa TLS (Transport Security Layer) tarkoittaa protokollaa liikenteessä. Seuraavana on avainten vaihto algoritmi. Molemmat RSA ja ECDHE ovat salausavaimen vaihtamisen algoritmeja. Seuraavana vahvemmassa salaustavassa on autentikointi algoritmi ECDSA (jscape 2018), joka ottaa tarkistussumman avaimesta ja tuo turvallisuutta tätä kautta. Seuraavana molemmissa salaustavoissa tulee salausalgoritmeja 3DES, EDE ja CBC, ja AES ja GCM. Nämä salausalgoritmit ovat symmetrisiä, joka tarkoittaa, että salaus on yhtä nopea purkaa kuin salata. Vahvemmassa salaustavassa 256 tarkoittaa bittien määrää. Lopuksi tulee SHA, joka on MAC (Message Authentication Code) -algoritmi. Tällä algoritmilla määritellään tulevan liikenteen oikeellisuuden tarkistusta. Tämä estää myös VPN palveluiden väitteen epäturvallisista julkisista verkoista. VPN suojaa salasanan samalla tavalla kuin HTTPS yhteys. Tämä olisi ongelma vain, jos web-sovellus, johon kirjaudut sisään ei tukisi HTTPS yhteyttä. Näitä palveluita ei vuonna 2020 ole montaa.

3.2 Standardit

Web-sovellus katsotaan läpi *Open Web Application Security Project Application Security Verification Standard* (OWASP ASVS 2020) standardin mukaan. Tämä kattaa Lähes kaikki mahdolliset haavoittuvuudet ja puutteet mitä sovelluksesta voi löytyä. Esimerkiksi autentikointi vaatimukset, jossa käydään läpi salasanavaatimuksia ja elinikää, ja injektio hyökkäykset, jossa käydään läpi eri syöttökentät ja laitetaan niihin vaarallisia merkkejä tai syötteitä, jotka paljastavat tietoa tai aiheuttavat haittaa sovellukselle tai muille käyttäjälle. Tämän lisäksi kokemuksen perusteella voi katsoa tarkemmin eri osa-alueita kuten XSS tai injektio hyökkäyksiä. Usein sovelluksessa on monta kohtaa, jotka eivät sovellu ASVS standardin tarkastuksiin, joten täytyy itse muokata tarkistuksen kohteita ja sovelluksen vaatimuksia. Tähän vaikuttaa esimerkiksi onko sovellus sisäverkossa, onko sovelluksella rajattu käyttäjäkunta yms.

Web-sovellustestauksessa käytetään useampia käyttäjiä. Helpoin raja menee todennetun ja todentamattoman välissä. Tämän jälkeen tarvitsee esimerkiksi lukijan, kirjoittajan ja pääkäyttäjän tason käyttäjät. Tämä toki riippuu sovelluksesta, onko olemassa eri tason käyttäjiä.

4 KÄYTETTEÄVÄT TYÖKALUT

4.1 Burp Suite Pro

Burp Suite Pro on web-sovellustestauksen päätyökalu. Sillä voi testata kaikki havainnot mitä web-sovellustestauksessa tulee vastaan. Burp Suite on oma sovellus, joka pyörii koneella. Se luo välimieshyökkäyksen selaimen ja verkkoon lähtevän liikenteen väliin. Tämän avulla voidaan muokata kutsuja, joita selain tekee, ja tehdä havaintoja tätä kautta. Burp Suitesta on ilmaisversio, mutta joitain ominaisuuksia on heikennetty verrattuna maksulliseen versioon. Tämä työkalu käydään paremmin läpi seuraavassa kappaleessa (Burp Suite).

4.2 SQLmap

SQLmap on automaattinen työkalu taustapalvelimen tutkimiseen. Useilla eri vivuilla voidaan saada tunnetuistakin palveluista vikoja esille. Pyyntö voidaan kopioida tekstitiedostoon ja syöttää tiedosto työkalulle vivulla -r. Muita hyviä vipuja on *-dbms* jolla voi asettaa tietokannan mallin, esimerkiksi *MySQL*, *PostgreSQL* yms. Vivulla *-level* 1-5 voidaan asettaa testien määrää. Tämä voi suurilla arvoilla jäädä kiinni palomuuriin ja joudut jäähylle. Myös löytyy *-risk* 1-3 jolla voidaan valita, kuinka vaarallisia komentoja ohjelma tekee. Kuitenkaan ”*Drop tables*”-tyyppisiä komentoja ei tehdä ikinä. SQLmapin saa ilmaiseksi GitHubista (SQLmap).

4.3 SSLscan

SSLscan on työkalu, jolla voi tarkistaa HTTPS salauksen menetelmän ja tavat. Työkalua ajetaan komentoriviltä. Sopivia parametrejä on --xml=<tiedosto> joka tekee xml muotoisen tiedoston raportista. SSLscannin saa ilmaiseksi GitHubista (SSLscan).

Tähän voi myös käyttää Qualysin verkkosivua https://www.ssllabs.com/ssltest/, josta löytyy sama testi. Tämän voi kuitenkin suorittaa vain julkiverkon palveluihin, joten kannattaa olla SSLscan asennettuna testikoneella.

4.4 Selaimet ja laajennokset

Web-sovellustestauksessa käytetään useampaa selainta samaan aikaan, jotta saadaan kaksi eristettyä istuntoa. Selaimesta pitää liikenne uudelleenohjata Burp Suitelle, tämä onnistuu parhaiten laajennoksilla. esimerkiksi *Foxyproxy* on hyvä laajennos tähän tarkoitukseen. Sillä voi nopeasti painaa uudelleenohjauksen päälle ja pois. Windowsissa voidaan myös kääntää koko käyttöjärjestelmän verkkoliikenteen välityspalvelimelle, mutta kaikki sovellukset eivät tottele tätä käskyä ja toiset taas eivät toimi välityspalvelimen kanssa.

4.5 Soap UI ja Postman

Soap UI ja Postman on API (Application Programming Interface) kutsujen tekemiseen helpottavia työkaluja. Ne tarjoavat graafisen käyttöliittymän, joka tekee kutsujen muokkaamisesta helpompaa. Molemmat työkalut ovat ilmaisia (Soap UI ja Postman).

Soap UI ja Postman on enemmän ohjelmoijien työkaluja, mutta näistä voi myös uudelleenohjata liikenteen Burp Suitelle. Usein mallisanomat tulevat tiedostoissa, jotka tukevat vain Soap UI:ta tai Postmania, joten ne on hyvä olla asennettuna, varsinkin jos joutuu työskentelemään sisäverkossa, jossa ei ole pääsyä internettiin.

4.6 Nmap

Nmap on Porttiskannaustyökalu, jolla voi tarkistaa protokollatason haavoittuvuuksia. Vivuilla -p- valitaan kaikki portit skannaukseen mukaan. Muita hyviä vipuja on -Pn, joka poistaa ping skannauksen ja kohtelee jokaista porttia auki. Tämä on hyvä olla käytössä, sillä useissa verkoissa on ping liikenne poistettu käytöstä. –script=vuln vivulla saadaan skannattua yleisiä haavoittuvuuksia protokollatasolla. Esimerkiksi palvelunestohyökkäykset löytyvät tällä työkalulla (Nmap).

Nmap ei ole varsinaisesti työkalu itse web-sovelluksen testaamiseen, mutta usein joutuu etsimään mistä portista palvelu vastaa. Julkisissa palveluissa portti on lähes aina 80 tai 443, mutta sisäverkkojen testipalvelimilla nämä portit voivat olla jo toisen palvelun käytössä, joten oikea portti tarvitsee löytää muilla konsteilla.

4.7 Käyttöjärjestelmä

Me KPMG:llä käytämme testauksessa Debian 10 pohjaista Linux käyttöjärjestelmää, jossa on kaikki työkalut asennettuna valmiina. Asennettuna on myös VM Workstation, jonka avulla saadaan esimerkiksi ajettua VPN sovellus Windows koneessa ja tehtyä testaus tätä kautta. VM Workstation on Virtualisointiohjelma, jolla voi suorittaa virtuaalisia tietokoneita. Näissä voi olla käytössä eri käyttöjärjestelmä kuin isäntä laitteessa. Tämä voi olla hyödyksi myös testikoneen puhtaana pitämisessä. Voidaan esimerkiksi luoda jokaiselle projektille oma virtuaalinen tietokone, joka projektin jälkeen poistetaan ja sen mukana kaikki data.

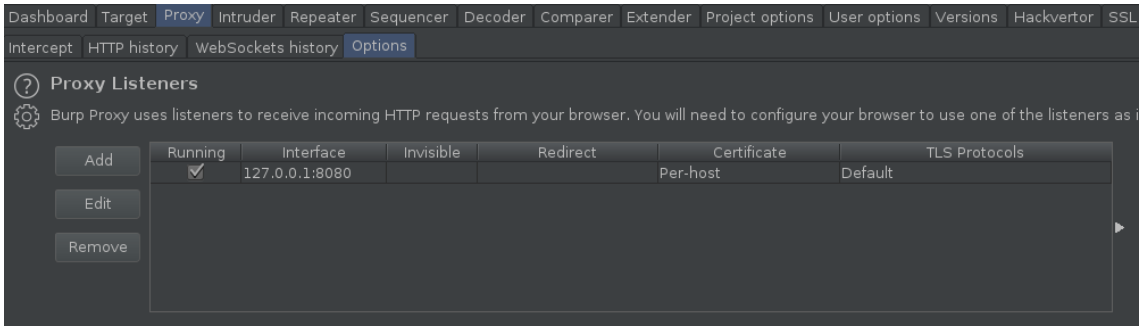
5 MENETELMÄT

5.1 Burp Suite Pro:n käyttöönotto

Testauksen aloittamisessa tarvitaan tapa, jolla kääntää liikenne selaimelta Burp Suitelle. Tämä onnistuu esimerkiksi *Foxyproxy* selainlaajennoksella (kuva 1). Voidaan tehdä uusi välityspalvelin, jonka osoite on localhost (127.0.0.1) ja portti 8080, joka on yleinen uudelleenohjausportti.

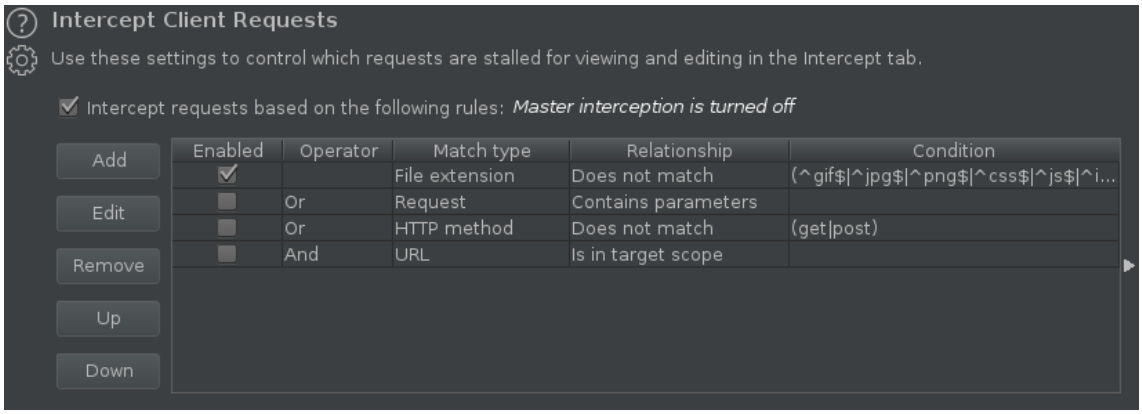
Kuva 1 Välityspalvelimen asetukset FoxyProxy ohjelmassa

Tämän jälkeen asetetaan Burp Suitesta kuuntelija päälle (kuva 2).

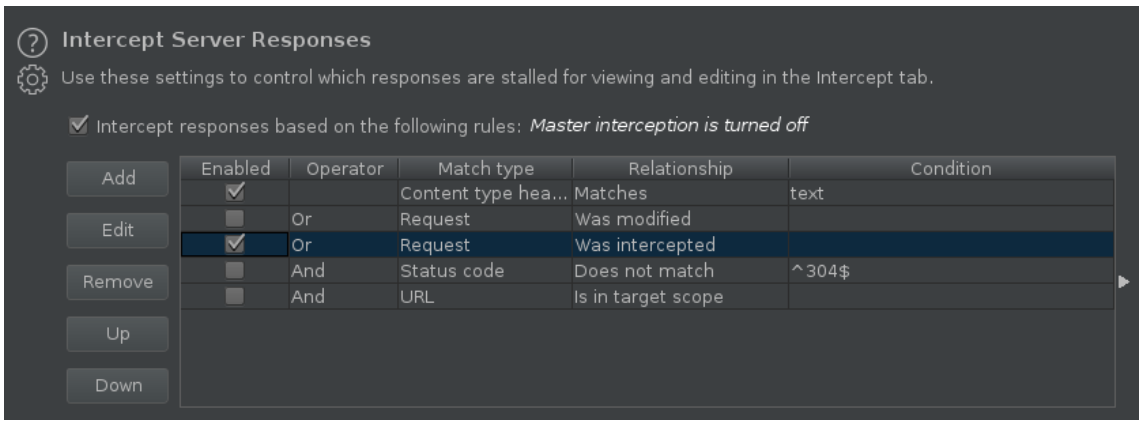


Kuva 2 Välityspalvelimen käyttöönotto Burp Suitessa

Seuraavana on hyvä laittaa päälle *"intercept client request"* (kuva 3) ja *"intercept server responses"* (kuva 4). Näiden avulla voidaan reaaliajassa muokata selaimelle tulevaa ja lähtevää dataa.

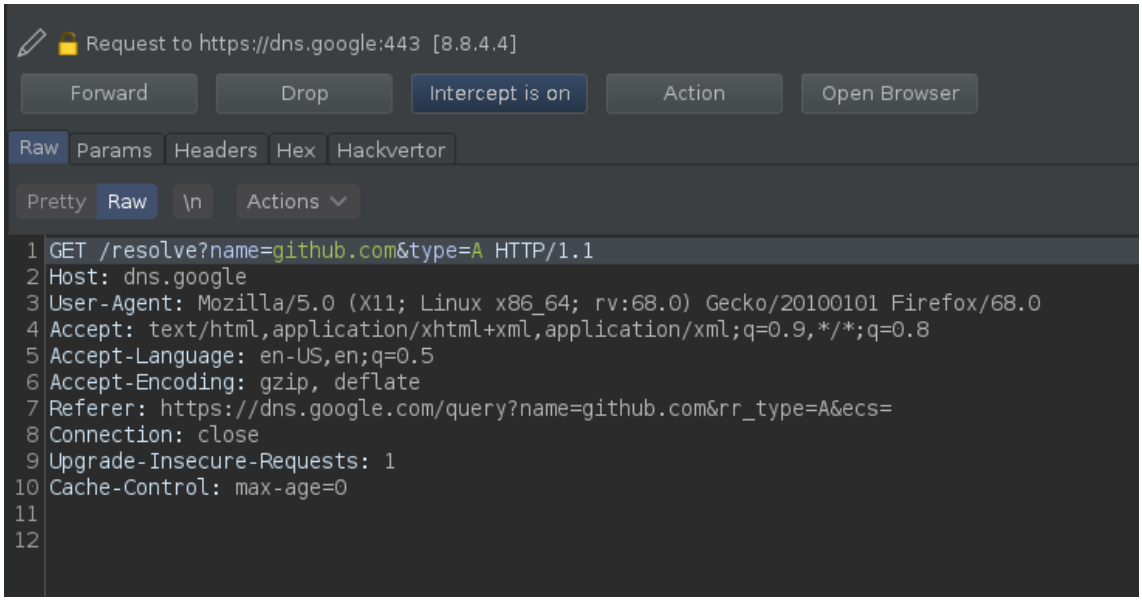


Kuva 3 Välityspalvelimen asetusten muokkaus, lähtevät kutsut



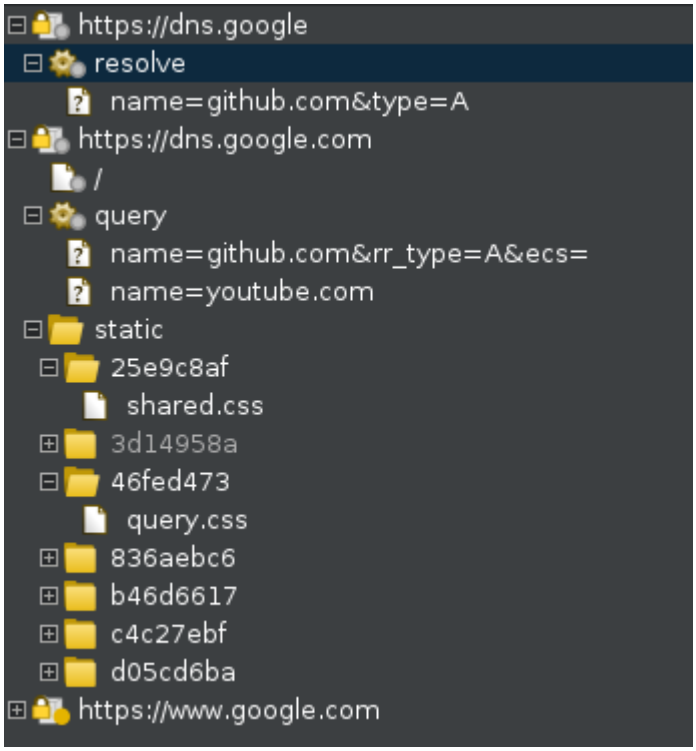
Kuva 4 Välityspalvelimen asetusten muokkaus, palaava kutsu

Seuraavaksi tarvitsee ottaa *Intercept* pois päältä (kuva 5). Tämä estää selaimelle tiedon menemisen ja saa selaimen näyttämään siltä, ettei yhteys toimisi.



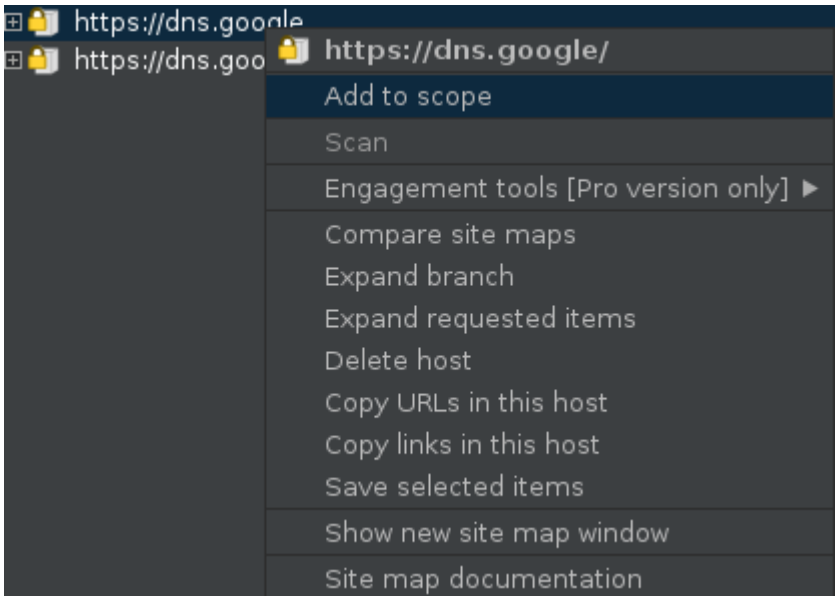
Kuva 5 Interceptin toiminnallisuudet ja mallikutsu

Nyt pitäisi näkyä *(Target > Site map)* -välilehdellä liikenne joka selaimella on luotu (kuva 6).



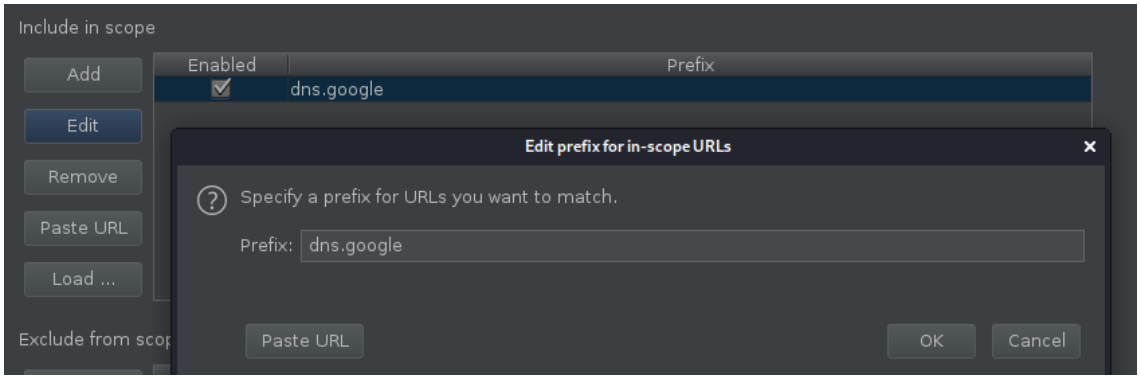
Kuva 6 Kohteet, jotka välityspalvelin on nähnyt

Tämän jälkeen kannattaa lisätä sivusto, jota testataan ”piiriin” (”*Add to scope”)* -napista (kuva 7). Kannattaa olla tarkkana, ettei tule laitettua pelkkää kansiota piiriin, joka estäisi muiden kansioiden analysoinnin ja talteen oton.



Kuva 7 Lisää sivu piiriin

Piiriin voi myös lisätä manuaalisesti (*target > scope)* -välilehdeltä (kuva 8).

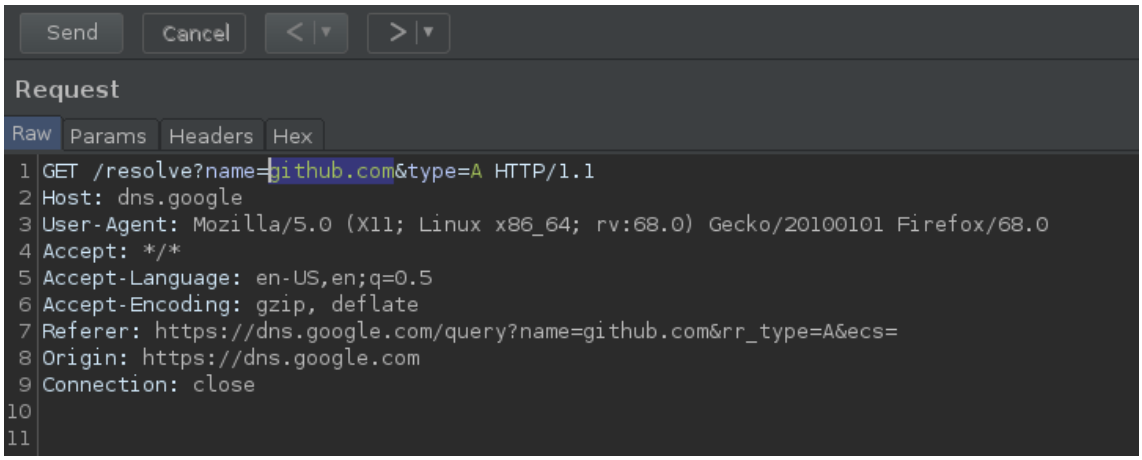


Kuva 8 Lisää sivu piiriin manuaalisesti

5.2 Burp Suite ominaisuuksia

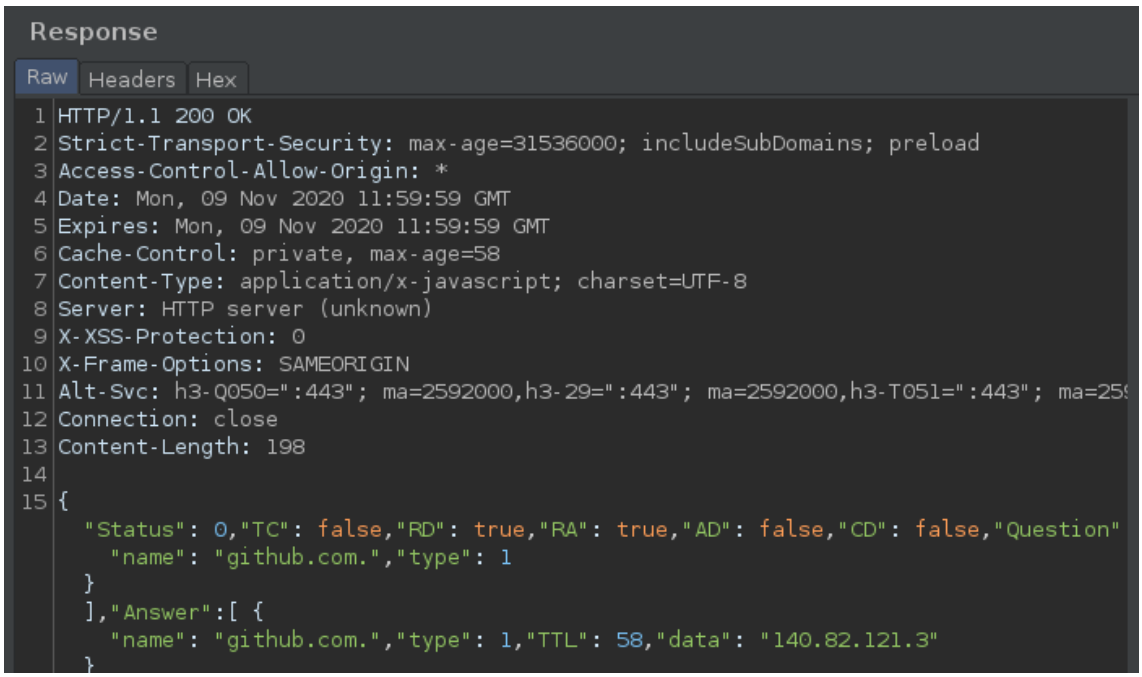
5.2.1 Repeater

Repeater on ominaisuus, jolla pystyy muokkaamaan ja toistamaan pyynnön helposti (kuva 9). Tämän avulla on helppo varmistaa löydettyjä havaintoja. Myös roolirajojen kokeileminen on helppo suorittaa Repeatterissa ottamalla tunnistetiedot pois pyynnöstä.



Kuva 9 Pyynnön lähetys Repeaterilla

Näet myös vastauksen nopeasti viereisestä laatikosta (kuva 10).



Kuva 10 Pyynnön vastaanottaminen Repeaterilla

5.2.2 Intruder

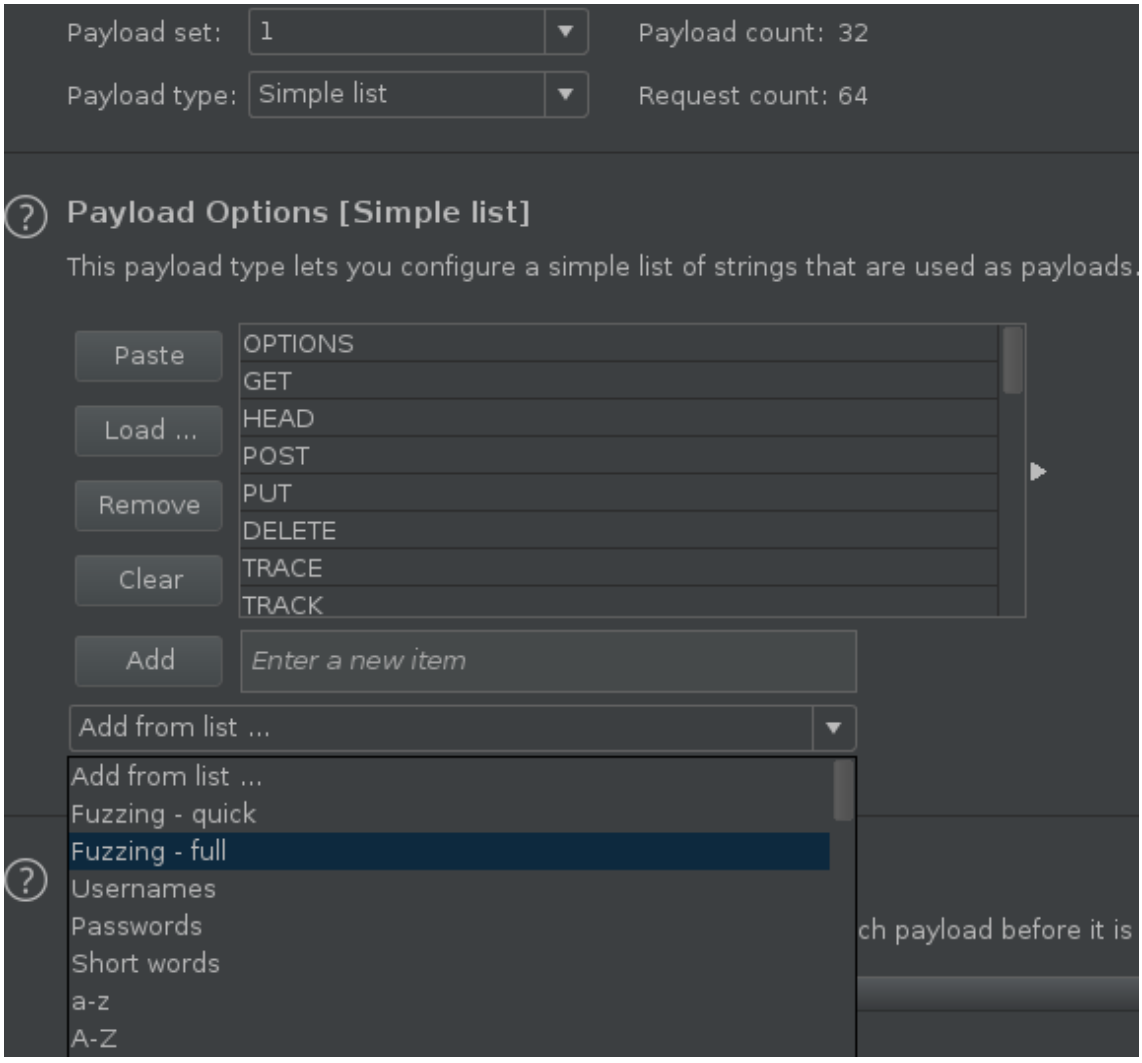
Intruder on väsytyshyökkäys työkalu (kuva 11). Voidaan valita muokattavat kohteen pyynnöstä ja listata mitä kaikkea siihen kohtaan kokeillaan laittaa. Esimerkiksi voidaan ajaa kaikki HTTP verbit tämän kautta helposti (GET, POST, PUT…). Nämä asetukset toki ovat yleensä sivu kohtaisia, joten tätä on hyvä testata useammassa sivussa.

Kuva 11 Välilehtiä, joita Intruderissa on

Positions välilehdellä voidaan vaihtaa mitä kohtaa kutsusta muokataan (kuva 12). § merkillä valitaan kohta mitä muokataan hyökkäyksessä.

Kuva 12 Kutsun osat, joihin väsytyshyökkäys kohdistetaan, näkyy korostetusti

Payloads välilehdessä voidaan päättää mitä korostettuihin kohtiin yritetään syöttää (kuva 13). Aluksi valitaan, laitetaanko lista vai esimerkiksi numeroita kenttään. Seuraavana on lista, johon voidaan itse kirjoittaa lasteja tai valita listasta, jotka löytyvät Burp Suitesta valmiina. Voidaan myös ladata lista tiedostosta, tähän voi ladata esimerkiksi IntruderPayloads nimisen GitHub arkiston (Intruder payloads).

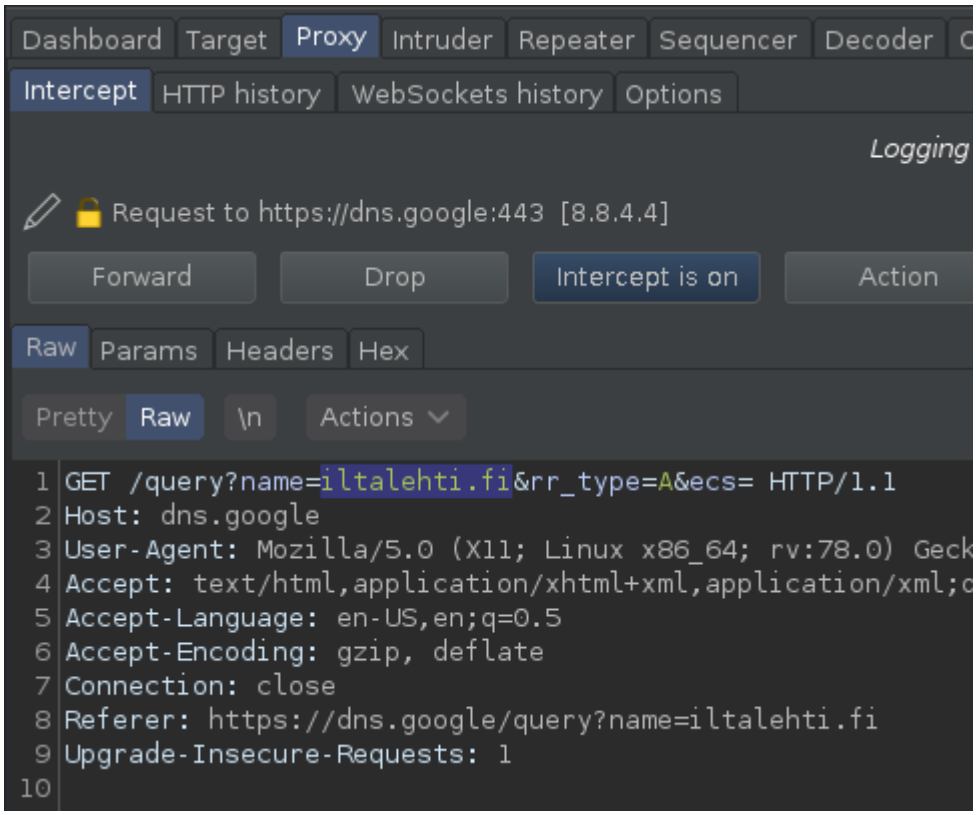


Kuva 13 Intruderin payloads välilehti

5.2.3 Intercept

Interceptillä voi muokata reaaliajassa pyyntöjä (kuva 14). Tämä on hyvä esimerkiksi sisäänkirjautumisen muokkaamisessa. Yleensä tällä muokataan palvelimelle lähteviä pyyntöjä, mutta voidaan myös muokata takaisinpäin tulevia pyyntöjä. Näillä voi joskus saada sovelluksen sekaisin ja pyytämään käyttäjälle vääriä oikeuksia tai ylimääräistä dataa, vaikkei oikeuksia oikeasti olekaan. Interceptillä

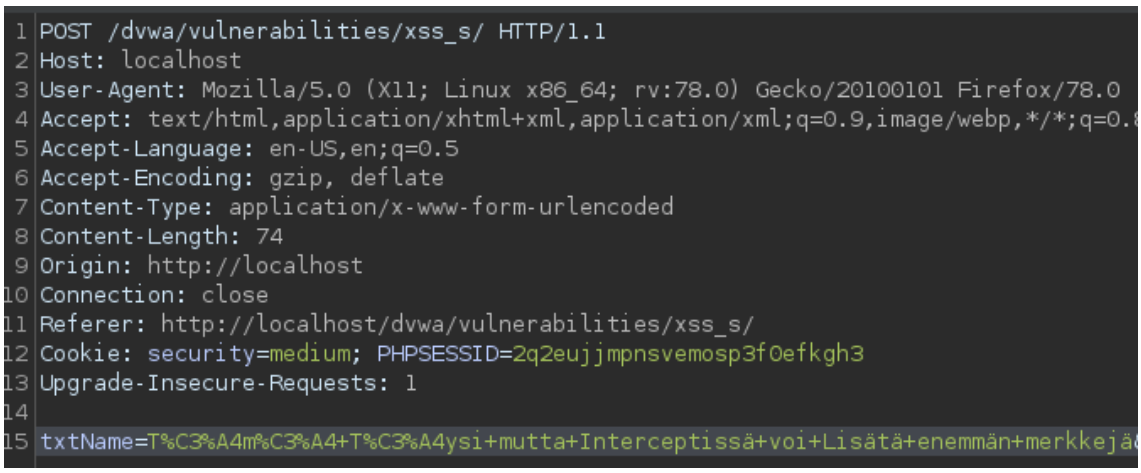
voi myös ohittaa selaimen päässä tapahtuvan kenttien syötteen tarkistuksen, ja tätä kautta saada injektiohyökkäyksiä aikaan.



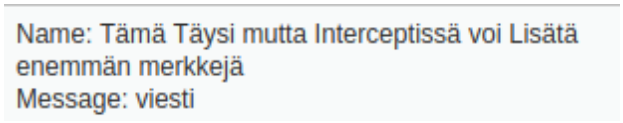
Kuva 14 Intercept syöttökentän muokkaus

Voidaan myös lisätä kenttään lisää merkkejä, vaikka käyttöliittymässä olisi laitettu jo maksimi määrä merkkejä (kuva 15; kuva 16; kuva 17).

Kuva 15 Name kenttä on täysi



Kuva 16 Interceptillä lisätään merkkejä nimi kenttään

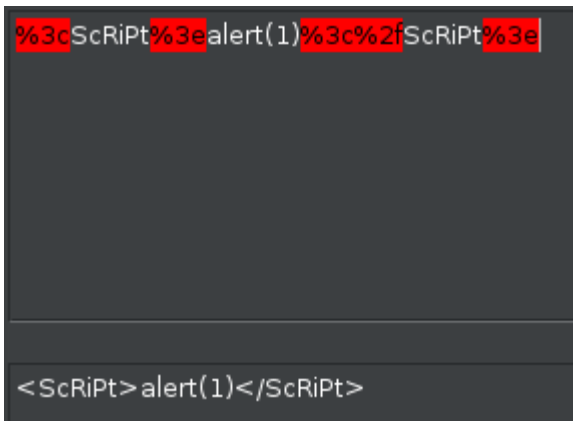


Kuva 17 Kuitenkin merkit menivät läpi palvelimelle

Tämä havainto on melko yleinen. Yleensä joko selaimessa tai palvelimessa ei tarkisteta syötettä ja voi saada sovelluksen sekaisin tämän takia.

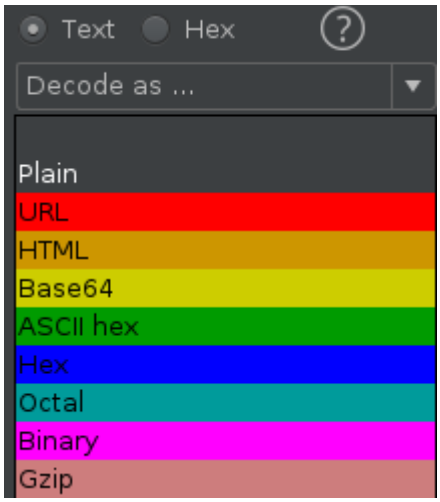
5.2.4 Decoder

Decoder on yksinkertaisesti dekooderi. Sen avulla voi katsoa esimerkiksi base64 koodattujen ”*authorization bearer*” otsakkeen sisälle ja tarkistaa salaustavan mitä sen luonnissa käytetään, oikeudet mitä otsakkeessa luvataan ja voimassa-oloajan. Base64 koodauksen voi lukea lisää Mozillan sivuilta (Mozilla 2020). Koodaus onnistuu molempiin suuntiin. Voidaan salata omia viestejä esimerkiksi URL koodauksella, jolla voi päästä ohi joistakin kenttien tarkastuksista. Usein näissä kentissä on vain selaimen päässä mustalistattu tietyt merkit. voidaan korvata ”<” ja ”>” merkkien eston ” %3C” ja ”%3E” URL koodatuilla vastikkeilla (kuva 18). Tämä kääntyy takaisin alkuperäiseen muotoon, kun palvelin näkee sen pyynnössä.



Kuva 18 Decoder kääntämässä "<", ">" ja "/" merkkejä, jotta saataisiin ohitettua selaimen päässä tehtävä tarkistus

Tekstiä voidaan kääntää usealle eri ”kielelle” (kuva 19).



Kuva 19 lista kielistä joihin Burp Suite osaa kääntää

5.2.5 Extender

Extender on applikaatiokauppa, josta voidaan ladata lisäominaisuuksia suoraan Burp Suiteen. Itse käytän usein seuraavia lisäosia:

- SSL scanner
    - Voidaan tarkistaa ”*cipher suitet*" ja TLS salaustavat helposti ja saada raportin yhdestä ohjelmasta muiden havaintojen kanssa samaan tiedostoon.
- Autorise
    - Tämä työkalu toistaa kaikki pyynnöt mitä tehdään web-sovelluksessa, mutta ottaa niistä keksit ja muut todentavat tekijät pois.
- Retire.js

  - Tämä työkalu etsii sovelluksesta vanhentuneita kirjastoja ja taustapalvelimella käynnissä olevia sovelluksia, joista jää jälki web-sovellukseen.
- SAML Raider
  - Tällä työkalulla voidaan lukea SAML sanoman sisältöä ja muokata sitä. Voidaan myös tehdä hyökkäyksiä kuten allekirjoituksen poistamista ja muita osoittaja haavoittuvuuksia mitä kyseiseen standardiin kuuluu.

6 YLEISIMPIÄ HAVAINTOJA

Tässä kappaleessa käydään läpi yleisimpiä havaintoja mitä minulle on tullut vastaan. Havainnoilla on merkittävä vaikutus sovelluksen toimintaan, jos ne löytyvät sovelluksesta.

6.1 Kenttien syötteiden tarkistus

Tämä havainto voi johtaa puhelinnumeron sisältämään kirjaimia, tai pahimmassa tapauksessa pysyvään XSS (Cross Site Scripting) hyökkäykseen. XSS hyökkäys voi esimerkiksi lähettää keksit hyökkääjälle, joiden avulla hyökkääjä voi esiintyä uhrin henkilöllisyydellä. Helpoin tapa tehdä XSS hyökkäys on käyttää ”*script*” HTML viittaus. Tämä viittaus on usein kielletty, joten pitää ottaa käyttöön muita viittauksia kuten ”*img*” tai ”*a*”.

6.1.1 Havaitseminen

Kenttien syötteiden tarkistus on yleensä toteutettu jollakin tasolla. Tämä on usein selaimen päässä. Tämä on kuitenkin helppo ohittaa Burp Suiten Repeater tai Intercept ominaisuuksilla. Kuvat 15-17 havainnollistavat tätä.

6.2 Roolirajat

Usein sovelluksissa on useamman tason käyttäjiä. Esimerkiksi pääkäyttäjä, käyttäjä, jolla on muokkausoikeus ja lukijan roolissa oleva käyttäjä. On tärkeää, että nämä käyttäjät pysyvät omien oikeuksien sisäpuolella. Näin ei kuitenkaan aina ole. Vastaan on tullut sovelluksia, joissa ylläpitäjän asetukset ovat piilotettu käyttöliittymässä, mutta jos tiedät mitä asetuksia löytyy ja minne tehdä kutsu, se onnistuu jokaiselta käyttäjältä. Joskus roolirajan rikkominen voi olla toisen käyttäjän kielen vaihto, mutta joskus sen avulla voi lukea monien muiden käyttäjien tietoja ilman lupaa. Rooliraja on myös tunnistautuneen ja tunnistautumattoman käyttäjän välillä. Useimmissa julkisen verkon palveluissa pääset eteenpäin tunnistautumatta.

6.2.1 Havaitseminen

Roolirajoihin on Burp Suitessa lisäosa Autorise, mutta tämä ei aina löydä oikeata yksilöivää tekijää kutsussa, ja tämän takia tekee paljon vääriä havaintoja. Helpoin tapa testata tätä on kokeilla pääkäyttäjän kutsuja ilman oikeuksia ja kokeilla saman tasoisen toisen käyttäjän tietojen näkemistä. Myös voi katsoa todentamattomana sovelluksen toimintaa. Tähän vaikuttaa paljon onko sovellusta mahdollista käyttää ilman todennusta vai vaaditaanko aina ensimmäisenä kirjautuminen.

6.3 Virustorjunta

Vieläkään useassa sovelluksessa ei ole toteutettu virustorjuntaa, vaan luotetaan siihen, että käyttäjät, jotka lataavat tiedostot palveluun, ovat tarkistaneet ne itse. Osassa ympäristöissä, joissa virustorjunta on toteutettu, ovat asetukset laitettu pieleen tai suorituskyky rajoitettu, että virustorjunnalla ei ole mitään virkaa.

6.3.1 Havaitseminen

Tämän havainnot löytämiseen tarvitaan eicar testivirus. Eicar on teksti, joka sisältää kiellettyjä merkkejä ja sanoja, jotka pitäisi jäädä kiinni virusturvaan, jos sellainen on käytössä (Eicar 2006). Toki vaaditaan, että sovelluksesta löytyy jokin lataus mahdollisuus, että saadaan eicar tiedosto palvelimelle. Kannattaa tehdä muutama eri versio eicar testiviruksesta, esimerkiksi .docx, .txt, pdf jne. Latauksessa voidaan tarkistaa tiedoston muoto. Tämän pystyy ohittamaan kenttien syötteen tarkistuksen tapaan tai laittamalla tiedoston perään toisen päätteen esimerkiksi
eicar.txt;eicar.img. Tämä onnistuu huijaamaan pelkän selaimen tarkistuksen ilman Burp Suiten apua.

6.4 Datan lähettäminen tuntemattomana

Web-sovelluksilla kerätään usein dataa tunnistamattomilta käyttäjiltä, esimerkiksi kyselyillä ja palautteilla. On kuitenkin tärkeää, että yksi henkilö ei pysty automaattityökaluilla lähettämään miljoonia palautteita, kuvia tai mielipiteitä. Tämä voi pie-

nimmillään esimerkiksi täyttää kyselyn väärällä tiedolla ja tekee siitä käyttämättömän. Tai pahimmillaan tämä voi kaataa taustajärjestelmän (automaattiset vastaukset 2016).

6.4.1 Havaitseminen

Tämä on melko helppo havaita. Voi olla palautelaatikko yms., johon voi tunnistautumatta lähettää dataa. Täytyy kuitenkin varmistaa Burp Suitella, että palaute lähtee kaikkien tunnistavien tietojen poiston jälkeen.

7 YHTEENVETO

Tässä opinnäytetyössä käytiin läpi Web-sovellustestauksen perusteita, työkaluja, joilla testausta tehdään ja huomattavia haavoittuvuuksia web-sovelluksissa. Työssä paneuduttiin tarkemmin Burp Suite Pro:n toimintaan ja käyttöönottoon. Mainittiin myös muita sovelluksia, jotka helpottavat testausta. Lopuksi käytiin läpi yleisiä havaintoja web-sovellustestauksesta.

Työn tavoitteena oli saada aiheesta kiinnostunut keskiverto tietokoneenkäyttäjä alkuun web-sovellustestauksessa. Opinnäytetyössä onnistuttiin ottamaan käyttöön Burp Suite pro ja muita testausta helpottavia työkaluja. Näillä ohjeilla keskiverto tietokoneenkäyttäjä pääsee alkuun testauksessa ja voi alkaa erikoistumaan tiettyyn osa-alueeseen.

Tätä työtä käytetään uusien harjoittelijatason henkilöiden koulutukseen web-sovellustestauksen maailmaan. Harjoittelijoilla on käytössä valmis käyttöjärjestelmä, johon kaikki työkalut on asennettu, mutta niitä ei välttämättä olla otettu käyttöön vielä.

Web-testaus on tärkeä osa valtion toimintaa sillä ilman web-testausta et voisi luottaa erimerkiksi Kelan sivuun tukia hakiessasi tai verohallinnon sivuun verokorttia uusiessasi. Nämä verkkosivut ovat turvallisia jokaiselle käyttäjälle, koska web-sovellustestauksella on todettu, ettei mahdollisia haavoittuvuuksia ole jäänyt sovellukseen. Jokaisen yksilön ei tarvitse osata havaita onko lataamassasi tiedostossa virus tai käykö koneesi hitaammin sivustolla koska XSS ohjaa koneen resursseja esimerkiksi kryptovaluutan louhimiseen, kiitos web-testauksen.

Web-sovellustestaus tekee maailmasta paremman paikan, yksi sivu kerrallaan!

LÄHTEET

Automaattiset vastaukset. 22.6.2016. Kyselyt netissä. Luettu 8.12.2020. https://medium.com/salesforce-ux/on-the-internet-nobody-knows-youre-a-bot-participant-327dd0da5ce7

Burp Suite. Web-sovellustestaustyökalu. https://portswigger.net/burp

Damn Vulnerable Web Application. Testiympäristö. Luettu 25.11.2020. http://www.dvwa.co.uk/

Drop tables. Sarjakuva sanitoinnista ja injektiohyökkäyksestä. Luettu 9.12.2020. https://xkcd.com/327/

Eicar. 7.9.2006. Testivirus. Luettu 15.11.2020. https://www.eicar.org/?page_id=3950

Google DNS. DNS palvelu. dns.google

Intruder payloads. Lista injektiohyökkäyksistä. https://github.com/1N3/Intruder-Payloads

Jscape. 16.5.2018. Cipher suitet ja SSL salaus. Luettu 29.11.2020. https://www.jscape.com/blog/cipher-suites

JQuery versionumero. 14.2.2018. Miksi versionumeroita ei saa piilottaa. Luettu 6.12.2020. https://stackoverflow.com/a/48784157

Mozilla. 25.2.2020. Base64 koodaus. Luettu 4.12.2020. https://developer.mozilla.org/en-US/docs/Glossary/Base64

Nmap. Porttiskannaustyökalu. https://github.com/nmap/nmap

Owasp asvs. 29.10.2020. Web-sovellustestaus standardi. Luettu 4.12.2020. https://owasp.org/www-project-application-security-verification-standard/ https://github.com/OWASP/ASVS/tree/v4.0.2#latest-stable-version---402

Postman. API testaustyökalu. https://www.postman.com/downloads/

Relevant software. Mitä on web-sovellustestaus. Luettu 29.11.2020. https://relevant.software/blog/penetration-testing-for-web-applications/

Rikoslaki 21.4.1995/578 § 7. Lievä tietoliikenteen häirintä. https://www.finlex.fi/fi/laki/ajantasa/1889/18890039001#L38

Rikoslaki 10.4.2015/368 § 8. Tietomurto. https://www.finlex.fi/fi/laki/ajantasa/1889/18890039001#L38

Security magazine. 14.8.2020. Hakkerit muuttanut tapojaan toimia koronaviruksen takia. Luettu 6.12.2020. https://www.securitymagazine.com/articles/93086-the-future-of-hacking-covid-19-shifting-the-way-hackers-work-and-who-they-target

Soap UI. API testaustyökalu. https://www.soapui.org/downloads/soapui/

SSLscan. Web-sovellustestaustyökalu. https://github.com/rbsec/sslscan

SQLmap. Web-sovellustestaustyökalu. https://github.com/sqlmapproject/sqlmap

LIITTEET

Liite 1. Auktorisointikirje 1 (3)

Auktorisointikirje

Allekirjoittamalla tämän auktorisointikirjeen Asiakas antaa KPMG Oy Ab:lle (jäljempänä KPMG) luvan suorittaa seuraavat tekniset tietojärjestelmien tarkastukset (mikäli kyseinen testaus kuuluu Asiakkaan tilauksen/sopimuksen piiriin):

- ☐ 1. Ulkoinen haavoittuvuustarkastus
- ☐ 2. Ulkoinen tunkeutumistestaus (mahdollisten haavoittuvuuksien hyväksikäyttö/verfiointi)
- ☐ 3. Palvelinten alustatarkastukset (palvelinten tietoturva-asetusten analysointi)
- ☐ 4. Sisäinen haavoittuvuustestaus
- ☐ 5. Sisäinen tunkeutumistestaus (mahdollisten haavoittuvuuksien hyväksikäyttö/verfiointi)
- ☐ 6. Palvelunestohyökkäysten testaus
- ☐ 7. Muu alla määritelty tekninen testaus

Muun teknisen testauksen tiedot:

KPMG: TÄYTÄ TÄHÄN TIEDOT

Testauksen kulku on esitetty projektisuunnitelmassa ”SUUNNITELMAN NIMI”.

Tarkastuksen kohteet (verkkoavaruudet, järjestelmät, puhelinnumerot jne….) ovat seuraavat:

ASIAKAS TÄYTÄÄ TÄHÄN KOHTEET MAHDOLLISIMMAN TARKASTI

Asiakas vakuuttaa, että se on edellä mainittujen laitteiden ja järjestelmien omistaja ja/tai haltija ja että sillä on oikeus valtuuttaa KPMG testaamaan näiden järjestelmien tietoturvallisuutta sovittuna ajankohtana. Asiakas vastaa itse järjestelmien testausoikeudesta ja kohteeseen liittyvien kumppaneiden tiedottamisesta niissä tapauksissa, joissa tarkastuksen kohde on kolmannen osapuolen omistamassa tai hallinnoimassa ympäristössä tai resurssissa.

Auktorisoimalla KPMG:n suorittamaan sovitut testaukset Asiakas ymmärtää, että:

- Mikäli tarkastukseen sisältyy ulkoinen testaus, se suoritetaan KPMG:n testausverkosta IP -osoitteista 62.236.206.1-14, ellei asiakkaan kanssa muuta sovita.
- Sisäinen testaus suoritetaan asiakkaan sisäisestä verkosta, sovituista pisteistä käsin.
- Tekninen tietoturvallisuuden testaus ei voi osoittaa järjestelmän olevan aukottomasti turvallinen. Tekninen testaus voi ainoastaan osoittaa järjestelmän olevan turvaton.

Asiakas tiedostaa ja hyväksyy, että KPMG voi saada testauksen aikana pääsyn testauksen kohteena oleviin järjestelmiin tai muihin yksilöityihin ja erikseen mainittuihin järjestelmiin, sekä niiden sisältämiin tietoihin.

Asiakas sitoutuu olemaan esittämättä vaatimuksia KPMG:lle tai sen tytäryhtiöille tai niiden osakkaille tai työntekijöille koskien mahdollisia välillisiä, että välittömiä vahinkoja, jotka voivat aiheutua haavoittuvuustestauksen toteuttamisesta.

Henkilö on vastuussa projektin etenemisestä ja hän voi ohjata tehtäviä edelleen muille erikseen sovittaville KPMG:n työntekijöille tarjouksen ja sopimuksen sallimissa puitteissa. Projektin aikana vastuuhenkilö on tavoitettavissa numerosta +358 40 XXX.

Toiminnalliset rajoitukset

Haavoittuvuustestauksen aikana käytettävät menetelmät ja sovellukset saattavat (väliaikaisesti) rajoittaa tietojärjestelmän turvallisuutta. Allekirjoittamalla tämän auktorisointikirjeen, asiakas antaa KPMG:lle luvan käyttää seuraavia menetelmiä haavoittuvuustestauksen suorittamisen aikana (mikäli kuuluu tarjotun tarkastuksen piiriin):

☐ Verkkoliikenteen kuuntelu järjestelmässä, jossa testaus suoritetaan;

☐ Takaoviohjelmien asentaminen testauksen kohteena olevaan järjestelmään;

☐ Troijalaisten lähettäminen sähköpostilla tai vastaavalla menetelmällä;

☐ Nk. Logger -ohjelmien asentaminen kohteena olevaan järjestelmään (logger ohjelmat mahdollistavat salasanojen ja näppäinpainallusten kuuntelemisen);

☐ Kohteena olevan järjestelmän asetusten muuttaminen ja muutosten suorittaminen (tietyt hyökkäykset vaativat näitä toimenpiteitä);

☐ Sosiaaliset menetelmät tietojen kalastelussa (Social Engineering);

☐ Palvelunestohyökkäykset

Asiakkaan yhteyshenkilö

Asiakas määrittelee toimeksiannon ajaksi yhteyshenkilön, joka pystyy tarvittaessa antamaan järjestelmiin liittyvää yksityiskohtaista tietoa ja johon voi ottaa yhteyttä ongelmatilanteissa. Yhteyshenkilön kanssa sovitaan myös testauksen aloittamisesta ja päättämisestä.

Testauksen yhteyshenkilön tiedot

Nimi: _____

Sähköposti: _____

Puhelin: _____

Asiakkaan hyväksyntä	
Allekirjoitus:	
Nimen selvennys	
Asiakas:	
Tehtävänimike:	
Päiväys:	