WEB APPLICATION FIREWALLS

Veli-Pekka Vainio

Opinnäytetyö
Maaliskuu 2013

Tietotekniikan koulutusohjelma
Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖN
KUVAILULEHTI

Tekijä(t) VAINIO, Veli-Pekka	Julkaisun laji Opinnäytetyö	Päivämäärä 18.03.2013
	Sivumäärä 104	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty (x)

Työn nimi
WEB APPLICATION FIREWALLS

Koulutusohjelma
Tietotekniikan koulutusohjelma

Työn ohjaaja(t)
NARIKKA, Jorma

Toimeksiantaja(t)
JYVSECTEC
VATANEN, Marko

Tiivistelmä

Opinnäytetyö oli osana JYVSECTEC-kyberturvallisuushanketta, jonka tarkoituksena on rakentaa kyberturvallisuuden kehitykseen, testaukseen ja koulutukseen soveltuva ympäristö.

Työn tavoitteena oli tutkia OSI-mallin 7. kerroksella eli ohjelmistokerroksella toimivia palomuureja ja tehdä vertailua, kun vastakkain ovat avoimen lähdekoodin tuote sekä kaupallinen tuote. Kaksi erilaista Web Application Firewall -tuotetta valittiin ja rakennettiin suljettuun laboratorioverkkoon. Hyökkäyksiä tehtiin haavoittuvaiselle web-ohjelmistolle ja havaitut haavoittuvuudet paikattiin Web Application Firewall -tuotteiden avulla. Tuotteiden vertailut perustuivat aina hyökkäyksien estokyvystä dokumentointiin ja käytettävyyteen.

Tuloksien pohjalta oli nähtävissä Web Application Firewall:ien hyödyllisyys, kun haavoittuvaisia web-palvelimia kohti suoritettiin nykyaikaisia ja hyvin yleisiä hyökkäyksiä. Tulokset osoittivat myös hyvin selkeästi kaupallisen ja avoimen lähdekoodin tuotteiden eroja.

Opinnäytetyön toteutusta hyödynnetään myöhemmin JYVSECTEC-tietoturvahankkeen parissa.

Avainsanat (asiasanat)
WAF, Web Application Firewall, 7 layer, tietoturva, SQL-injektio, XSS, LFI, F5, ModSecurity

Muut tiedot

Author(s)
VAINIO, Veli-Pekka

Type of publication
Bachelor´s Thesis

Date
18.03.2013

Pages
104

Language
Finnish

Permission for web publication
(x)

Title
WEB APPLICATION FIREWALLS

Degree Programme
Information Technology

Tutor(s)
NARIKKA, Jorma

Assigned by
JYVSECTEC
VATANEN, Marko

Abstract

The bachelor's thesis was a part of JYVSECTEC security development project. The purpose of JYVSECTEC security development project is to build an environment for developing security, testing and education.

The purpose of the thesis was to research the firewalls of layer 7 of the OSI model and compare them with each other. There was a commercial and also an open source product of Web Application Firewall. Two Web Application Firewall products were chosen and built in a closed laboratory network. Attacks were made against a vulnerable web platform. Detected vulnerabilities were patched using Web Application Firewall products. The comparison of products was carried out using different kinds of methods such as the ability to secure the vulnerable web platform and documentation.

Based on the results, it was possible to see the necessity of Web Application Firewalls when attacks were made against the web platform using modern and common attack techniques. The results also indicated the difference between the commercial and open source Web Application Firewall.

The implementation of thesis will be later used as a part of JYVSECTEC security development project.

Keywords
WAF, Web Application Firewall, 7 layer, security, SQL injection, XSS, LFI, F5, ModSecurity

Miscellaneous

SISÄLTÖ

KUVIOT

LYHENTEET

ASM	Application Security Manager
CRLF	Carriage Return/Line Feed
CRS	Core Rule Set
HTTP	Hypertext Transfer Protocol
IPS	Intrusion Prevention System
LFI	Local File Inclusion
LTM	Local Traffic Manager
OSI	Open Systems Interconnection
OWASP	The Open Web Application Security Project
PHP	PHP: Hypertext Preprocessor
RFI	Remote File Inclusion
SP	Space
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
TMOS	Traffic Management Operating System
URL	Uniform Resource Identifier
VLAN	Virtual LAN
WAF	Web Application Firewall
WWW	World Wide Web
XSS	Cross-Site Scripting

1 TYÖN LÄHTÖKOHDAT

1.1 Toimeksiantaja

Työn toimeksiantajana toimi JYVSECTEC, joka on kyberturvallisuuden kehittämishanke. Hankkeen tarkoituksena on rakentaa kyberturvallisuuden kehitysympäristö, joka toimii muun muassa testaus- ja koulutusalustana. Valmis ympäristö havainnollistaa kyberturvallisuuden toimintamalleja ja tilannesimulointia sekä mahdollistaa organisaatioiden yhteistyön ympäristössä. (JYVSECTEC 2012.)

Kehitysympäristön työkaluilla on mahdollista tehdä laajat laite- ja ohjelmistotestaukset aina kokonaisiin järjestelmätestauksiin asti. Testausmenetelmät antavat organisaatioille mahdollisuuden turvalliseen ja syvälliseen testaukseen osana käyttöönottoa. (JYVSECTEC 2012.)

1.2 Tavoitteet

Työn lähtökohtana oli tutustua OSI-mallin 7. kerroksella toimiviin palomuureihin; muun muassa niiden toimintatapaan, tehokkuuteen, hyödyllisyyteen ja käyttöönottoon. Sekä kaupallisista että avoimen lähdekoodin Web Application Firewall –ratkaisuista tuli valituksi yksi tuote, joita verrattiin keskenään. Ainakin toinen tuotteista oli tarkoitus rakentaa JYVSECTEC:n kehitysympäristöön. Pääasiallinen tavoite oli tutkia palomuuriratkaisuiden toimintaa ja kannattavuutta. Tämän lisäksi perehdyttiin erilaisiin hyökkäystekniikoihin, joita vastaan WAF toimii. Myös penetraatiotestausta sivuttiin opinnäytetyössä. Tällöin pyrittiin läpäisemään valittujen tuotteiden tarjoama suoja.

2 WEB APPLICATION FIREWALL (WAF)

2.1 Yleistä

Yli 70 prosenttia hyökkäyksistä internetissä kohdistuu OSI-mallin 7. kerrokselle eli sovelluskerrokselle (application layer). Tällöin hyökkäys kohdistuu suoraan ohjelmistoon. WAF:it kehitettiin antamaan organisaatioille lisäsuojaa; havaitsemaan ja estämään 7. kerroksen hyökkäykset, ennen kuin ne edes saavuttavat ohjelmistoja. (ModSecurity n.d.)

Maaliskuussa 2010 julkaistiin Älypää.com-sivuston yli 125 000 käyttäjän tietokanta julkisesti internetissä. Se sisälsi käyttäjien tunnukset, salasanat ja sähköpostiosoitteet. Monilla Älypään käyttäjillä käyttäjätunnus ja salasana saattoi olla sama myös muihin palveluihin. Tämä johti muun muassa Facebook-tunnusten luvattomaan käyttöön. Epäillään, että tietokantaan on voitu päästä käsiksi SQL-injektion avulla, mikä on mahdollista puuttuvan tai väärin tehdyn syöttötiedon tarkistuksen vuoksi. Luultavasti tiedot olivat säilötty kaiken lisäksi selväkielisinä. (Cert-fi 2010.)

2.2 Toimintaperiaate

WAF voi olla laite, pelkkä sääntöihin perustuva suodatin tai serveriin asennettava lisäosa. Sen toiminta perustuu sääntöihin, joiden perusteella hyökkäykset havaitaan. Sääntöjä on kuitenkin jatkuvasti päivitettävä uusia, mukautettuja hyökkäyksiä varten. (OWASP - Web Application Firewall 2012.)

WAF monitoroi ohjelmistoille tulevaa sekä tarvittaessa myös lähtevää liikennettä. Liikenteen käytöstä ja loogisuutta tutkitaan kyselyiden ja vastauksien perusteella. Täten sen on mahdollista estää myös tuntemattomia uhkia epätavallisien toimintei-den tai tiettyä kaavaa noudattavien hyökkäysten varalta. IPS ei ole kykenevä tulkitsemaan ohjelmistokerroksen datavirtaa näin laajasti. (McMillan 2009.)

2.3 Tekniikat

WAF toteutetaan verkkoon yleensä välityspalvelimen (reverse proxy) periaatteella ennen suojattavia ohjelmistoja. Tällöin koko verkon liikennettä ei käsitellä, mikä antaa mahdollisuuden tehokkaalle sovelluskerroksen tutkimiselle. (McMillan 2009.)

Reverse proxy –tekniikkaa hyödynnettäessä WAF:lla on oma IP-osoite, ja kaikki yhteydet päättyvät siihen. WAF hoitaa sen jälkeen itse yhteydet web-palvelimiin. Transparent proxy on toimintatapana samanlainen kuten edellä mainittu reverse proxy, mutta sillä ei ole IP-osoitetta. Tällöin WAF on helppo asentaa, kun verkolta ei vaadita muutoksia. Tosin niin kattavia ominaisuuksia transparent proxy -toteutus ei kata. 2. kerroksen silta-moodissa (Layer 2 bridge) WAF sijoitetaan myös palomuurin ja web-palvelimien väliin. Kun WAF toimii käytännössä kytkimenä, se ei vaadi suuria muutoksia verkolta. Se on myös tehokas, mutta taas ominaisuuksiltaan vajaampi ratkaisu. (Beechey 2009.)

Verkon monitorointi –mallissa WAF tarkastelee liikennettä sille määritetystä portista, eikä sen läpi kulje liikennettä. Kyseessä on lähinnä testaukseen tarkoitettu malli, vaikkakin sen avulla voi myös liikennettä evätä. WAF:in voi asentaa myös web-palvelimen yhteyteen (host based). Tällöin palvelimen kuorma kasvaa, mutta vikasietoisuus suurenee. Toimintojen määrä ei ole myöskään niin kattava. (Beechey 2009.)

Vuonna 2011 Imperva alkoi tarjota ”Cloud WAF” –palvelua. Se tarjoaa suojan muun muassa ajankohtaisia uhkia ja OWASP Top 10 –uhkia vastaan sekä lisää web-sivuston suorituskykyä. Palvelun käyttöönottokaan ei vaadi ohjelmisto- tai laitteistomuutoksia – vain nimipalveluosoitteiden muutoksen. (Imperva Releases Cloud-based Web Application Firewall Service for Mid-Sized Businesses 2011.)

2.4 Virtual patching

Nimitystä “virtuaalipaikkaus” (virtual patching) käytettiin ensimmäistä kertaa IPS-valmistajien keskuudessa. Se ei ole ominainen termi web-ohjelmistoille, mutta sitä käytetään yleensä WAF:ien keskuudessa. Myös muita nimityksiä virtuaalipaikkauksesta on käytetty. Oleellisinta on kuitenkin ymmärtää, mistä virtuaalipaikkauksessa on kysymys. (OWASP - Virtual Patching Best Practices 2011.)

Virtuaalipaikkausta voidaan ajatella turvakerroksena, joka estää haavoittuvuuden hyväksikäytön. Liikennevirta kulkee turvakerroksen läpi, joten se kykenee estämään haitallisen liikenteen pääsyn web-ohjelmistolle. Etuna virtuaalipaikkauksessa on, että web-sivuston lähdekoodia ei tarvitse muuttaa haavoittuvuuden paikkaamiseksi. Totta kai myös web-sivuston lähdekoodi tulee korjata, mutta se ei ole aina niin yksinkertaista. (OWASP - Virtual Patching Best Practices 2011.)

Kaupallisten tuotteiden osalta paikkausta haavoittuvuuteen voidaan joutua odottamaan pitkään. On myös mahdollista, että kaupallisen tuotteen valmistaja on luopunut tuotteen kehityksestä ja paikkausta haavoittuvuuteen ei ole koskaan edes tulossa. Avoimen lähdekoodin tuotteissa paikkaus voi olla jo saatavilla, mutta haavoittuvuuden korjaus vie siitä huolimatta aikansa. (OWASP - Virtual Patching Best Practices 2011.)

Ei ole mitenkään uutta, että yritykset siirtävät web-sivuston kehityksen alihankkijalle. Alihankkija voi olla Suomessa, mutta yksi suosituimmista maista on myös Intia, jossa työvoima on halpaa. Haavoittuvuuksien korjaus voi olla hyvin hidasta, kun web-ohjelmisto on alihankkijan vastuulla. Olen huomannut tämän henkilökohtaisesti joidenkin web-sivustojen kohdalla raportoidessani löytämiäni haavoittuvuuksia, joissa olen ollut tietoinen alihankkijoiden käytöstä. Korjaukset on saatettu suorittaa kuukausia myöhemmin. Alihankkijoista koituvat kokonaiskustannukset voivat myös yllättää. Nortion kirjoittamasta uutisesta käy ilmi, että eräässä tapauksessa kustannukset olivat Intiassa suuremmat huonon työnlaadun vuoksi kuin Suomessa. (Nortio 2012.)

3 OWASP

OWASP-säätiö perustettiin joulukuussa 2001. Sen tarkoituksena on edesauttaa organisaatioiden turvallista ohjelmakehitystä, ohjelmien hankkimista ja käyttöä sekä ylläpitoa. OWASP tarjoaa tukea ilmaisten työkalujen, dokumenttien ja keskustelualueiden muodossa, mitkä ovat kaikkien käytössä. (OWASP - About The Open Web Application Security Project 2013.)

OWASP ei ole kaupallinen säätiö, joten se kykenee tarjoamaan puolueettomia, käytännöllisiä ja taloudellisia käytänteitä ohjelmistoturvallisuuteen liittyen. Tämä takaa myös säätiölle pitkäikäisen toiminnan, kun tärkeintä ei ole raha, vaan tiedon jakaminen ja opastus siitä kiinnostuneille. (OWASP - About The Open Web Application Security Project 2013.)

OWASP Top 10:n avulla on tarkoitus tunnistaa kaikki kriittisimmät riskit, joita organisaatiolla voi olla web-ohjelmistoissa. OWASP Top 10 julkaistiin ensimmäisen kerran vuonna 2003, mikä jälkeen sitä on päivitetty vuosien varrella aina sen mukaan, miten riskit muuttuvat. Kyseessä ei ole vain ohjelmistoturvallisuusprojekti, vaan sillä pyritään saamaan tietoturva osaksi koulutusta aina työkaluihin asti. (OWASP – OWASP Top 10 2012.) OWASP Top 10:n sisältö on luettavissa liitteessä 1 ja viimeisimmät muutokset liitteessä 2.

4 TIETOTURVAMALLIT

Tietoturvamalleja on kahdenlaisia: Positiivinen ja negatiivinen tietoturvamalli. Ne ovat käytökseltään hyvin erilaisia, mutta rakenteeltaan samanlaisia. Molemmat tietoturvamallit kuitenkin pohjautuvat ennalta määrättyihin sääntöihin. (Murphy, A. & Salchow, K 2007.)

Positiivisen tietoturvamallin ideana on hyväksyä vain liikenne, jonka se tuntee. Kun tietoturvamalliin lisätään uusi sääntö, kasvaa tunnetun liikenteen määrä ja täten hyväksymisaste. Jos sääntöjä ei ole ollenkaan, ei liikennettä hyväksytä ollenkaan. (Murphy, A. & Salchow, K 2007.)

Negatiivinen tietoturvamalli taas hyväksyy liikenteen, jota se ei tunnista haitalliseksi. Tällöin sääntöjen lisäyksellä saadaan muutettua hyväksytyn liikenteen määrää tiukemmaksi. Ilman sääntöjä, mitään estotoimia ei suoriteta. Liikennettä verrataan tunnettuihin kaavoihin, joita esiintyy hyökkäyksissä. Tämän perusteella tehdään päätös, onko liikenne turvallista päästää eteenpäin. Negatiivisessa tietoturvamallissa ei oteta kantaa ohjelmiston toimintaan. Vain haitalliset piirteet liikenteessä huomioidaan. (Murphy, A. & Salchow, K 2007.)

Tietoturvamallien paremmuudesta on eri mielipiteitä. Yleensä molempien tietoturvamallien puolestapuhujat ovat oikeassa. Teoreettisesti. Käytännössä hyvä tietoturva on negatiivisen ja positiivisen tietoturvamallin välissä. Molemmat toimivat tällöin yhdessä. Kun positiivisen tietoturvamallin puolta löysennetään esimerkiksi uusien ohjelmien vuoksi, kiristetään negatiivisen tietoturvamallin puolta. Kumpikaan tietoturvamalli ei yksistään voi tarjota parasta ratkaisua kaikkiin tilanteisiin. Ottaen huomioon käyttötarkoituksen ja vaatimukset, on molempien tietoturvamallien avulla mahdollista määrittää toimiva kokonaisuus teorian ja käytännön väliltä. (Murphy, A. & Salchow, K 2007.)

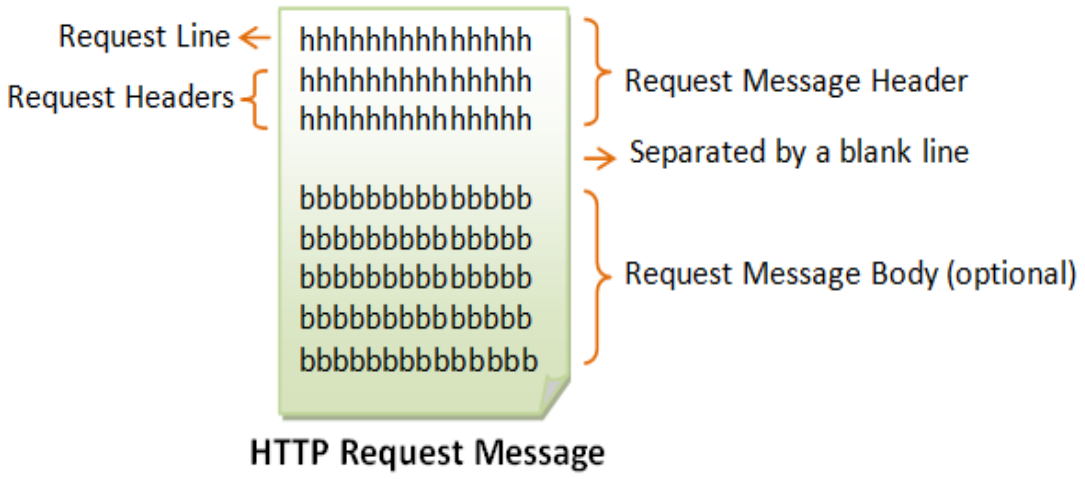
5 HTTP-PROTOKOLLA

5.1 Yleistä

HTTP-protokolla (The Hypertext Transfer Protocol) on OSI-mallin sovelluskerroksella (7. kerros) toimiva protokolla, joka on ollut WWW:n (World Wide Web) käytössä jo vuodesta 1990. Se hyödyntää liikennöintiin TCP/IP-protokollaa oletusportin ollessa 80. Myös muutkin luotettavat protokollat käyvät HTTP:n alustaksi. HTTP:n toiminta on käytännössä kyselyä asiakkaan toimesta ja vastailua palvelimen toimesta. (Hypertext Transfer Protocol – HTTP/1.1 – Overall Operation 1999.)

5.2 HTTP-pyyntö

Kuviossa 1 on esiteltynä HTTP-pyynnön rakenne. Kokonaisuutena pyyntö muodostuu otsakkeesta (Request Headers) ja sisällöstä (Request Message Body), joista sisältö ei ole pakollinen. Myöskään otsakkeen kaikki tiedot eivät ole pakollisia. (Hock-Chuan 2009). Jotta otsakkeen eri osat tunnistetaan, ne erotellaan SP-merkein (Space). Rivin päättymistä kuvataan CRLF-merkein (Carriage Return/Line Feed). (Hypertext Transfer Protocol – HTTP/1.1 – Request 1999.)

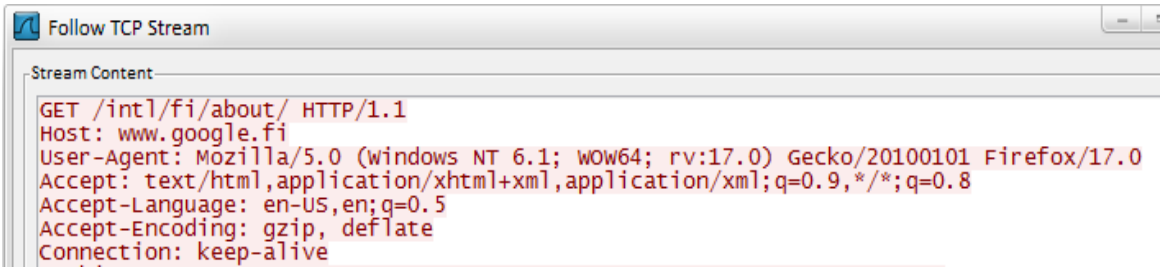


KUVIO 1. HTTP-pyyntö (Hock-Chuan 2009)

Kuviossa 2 on esitetty HTTP-pyyntö eräälle Googlen sivulle. Ensimmäisenä on Request-Line, joka koostuu seuraavasti:

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

GET /intl/fi/about/ HTTP/1.1



KUVIO 2. HTTP-pyyntö Googlen palvelimelle

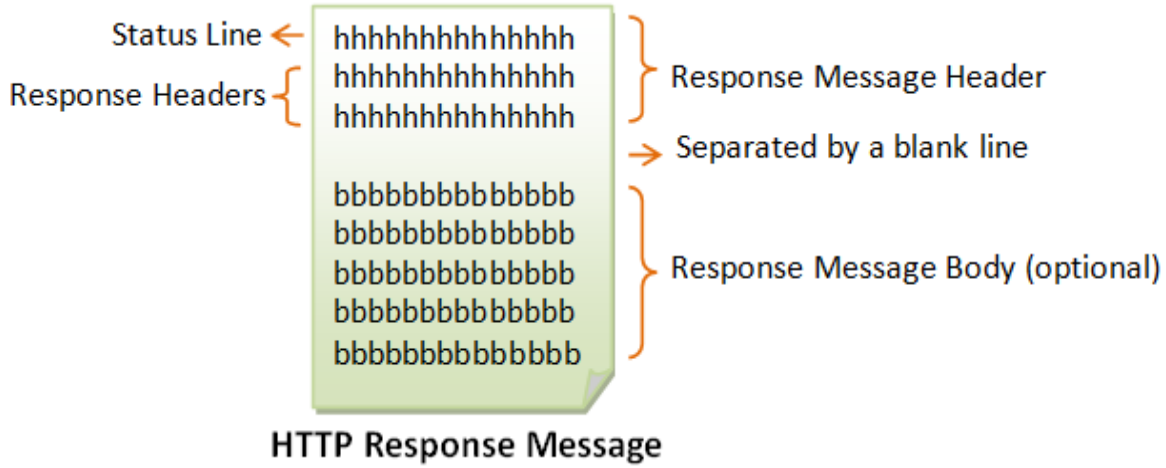
Käytettynä tapana lähettää pyyntö serverille on GET. Muita vaihtoehtoisia tapoja on esimerkiksi POST tai HEAD (Hypertext Transfer Protocol – HTTP/1.1 – Method Definitions). Seuraavana on pyydetty kohde (Request-URI) ja käytetty HTTP-versio. HTTP-pyynnön Request headers –osa muodostaa loppuosan otsakekentästä, jotka ovat kuvion 2 pyynnössä seuraavat:

Host: www.google.fi
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:17.0) Gecko/20100101 Firefox/17.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/*;q=0.8*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive

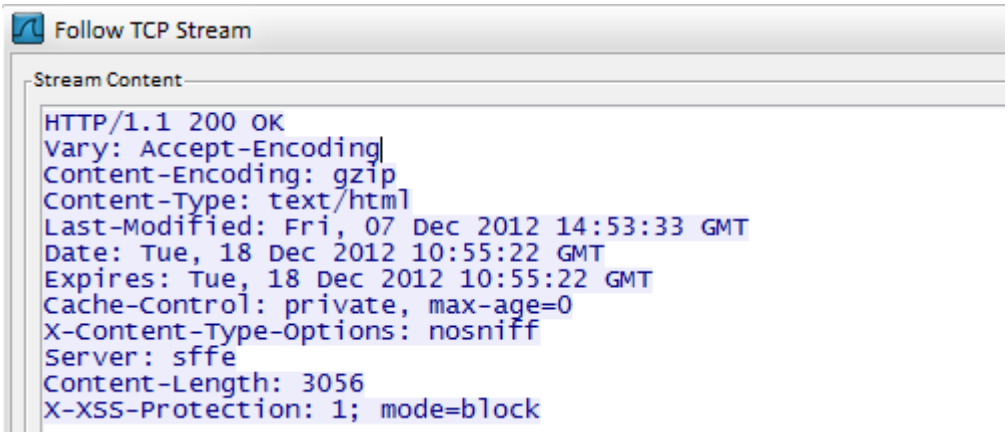
Host-kentässä on pyydetyn sivuston tarjoaja. Loppu otsakkeesta koostuu muun muassa käytetystä selaimesta ja siitä, millaista sisältöä hyväksytään takaisin palvelimelta. (Hypertext Transfer Protocol – HTTP/1.1 – Request 1999.) Kuvion 2 pyynnössä ei ole sisältöä.

5.3 HTTP-vastaus

HTTP-vastaus on rakenteeltaan hyvinkin samankaltainen kuin HTTP-pyyntö. Kuviossa 3 on esitelty sen rakenne. Vastaus sisältää myös otsakkeen (Response Headers) ja sisällön (Response Message Body). Kuviossa 4 on Googlen palvelimen vastaus kuviossa 2 esitettyyn pyyntöön.



KUVIO 3. HTTP-vastaus (Hock-Chuan 2009)



KUVIO 4. HTTP-vastaus Googlen palvelimelta

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

HTTP/1.1 200 OK

Ensimmäinen osa HTTP-vastausta on Status-Line, joka ilmoittaa HTTP:n protokolla-version, numeerisen tilakoodin ja tilakoodia vastaavan tekstiselityksen takaisin asiakkaalle. Loput otsaketiedot ilmoittavat muun muassa web-palvelinohjelmiston ja että vastaus on jo valmiiksi erääntynyt. (Hypertext Transfer Protocol – HTTP/1.1 – Response 1999.)

Jos sivustolla käydään uudelleen, eikä selaimen välimuistia ole tyhjennetty, antaa Googlen palvelin vastaukseksi “304 Not Modified”, ilmoittaen sivuston pysyneen samana sitten viime käynnin. Tähän hyödynnetään kuviossa 4 olevaa ”Last-Modified” –arvoa. Vastauksen mukana ei lähetetä sisältöä uudelleen, vaan se ladataan selaimen välimuistista, jolloin säästytään turhalta liikennöinniltä. (HTTP Caching. n.d.) Kuviossa 5 on esitettynä Googlen palvelimen vastaus, kun sivu on vastaava kuin välimuistissa.

```
HTTP/1.1 304 Not Modified
Date: Tue, 18 Dec 2012 11:17:04 GMT
Expires: Tue, 18 Dec 2012 11:17:04 GMT
Cache-Control: public, max-age=0
X-Content-Type-Options: nosniff
Server: sffe
X-XSS-Protection: 1; mode=block
```

KUVIO 5. Googlen palvelimen vastaus toistamiseen

5.4 Muuta

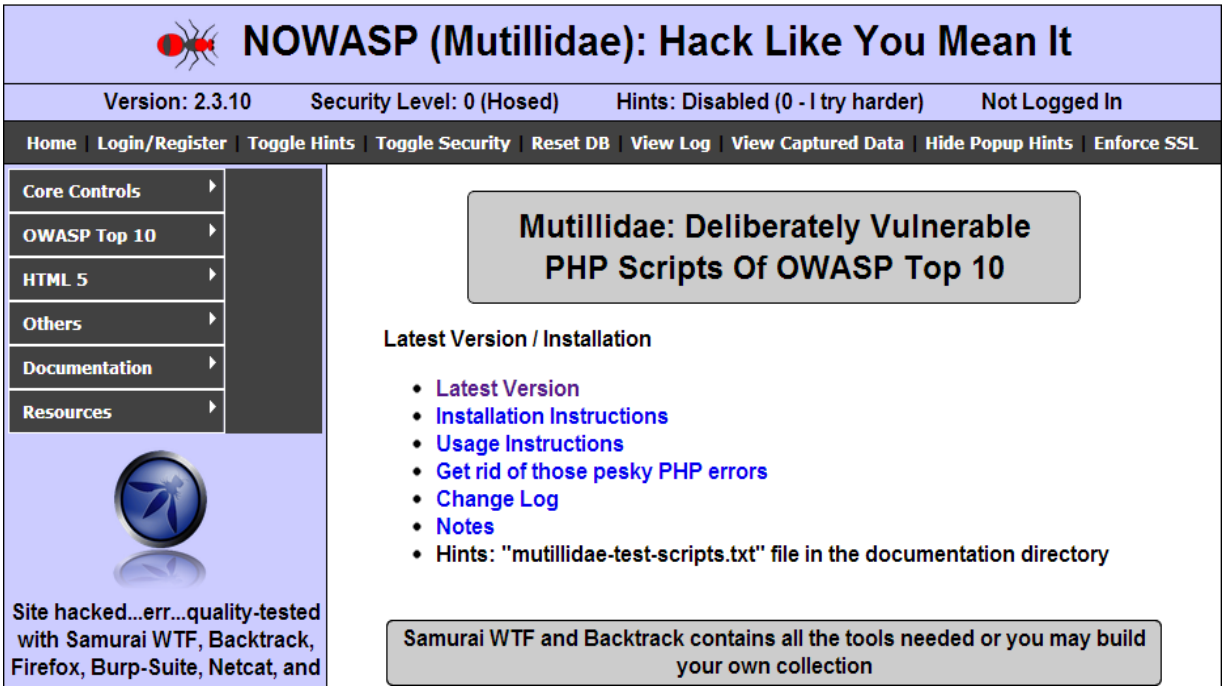
HTTP-protokollan rakenne on hyvä tuntea, sillä edellä mainittuja pyyntöjä ja vastauksia WAF tulkitsee hyökkäyksien estämiseksi. Hyökkäämiseen käytetyt parametrit voidaan sijoittaa kehyksessä myös otsakkeisiin eikä vain sisältöön. Tällöin ei riitä, että pelkkä sisältöosio luettaisiin. Myöhemmin suoritettavissa hyökkäyksissä hyödynnetäänkin koko HTTP-kehystä paremman lopputuloksen saamiseksi hyökkääjän näkökulmasta.

6 TESTAUSYMPÄRISTÖ

6.1 Mutillidae

Yhtenä tärkeänä osa-alueena on testausympäristö, jolla on mahdollisuus havainnollistaa sekä testata WAF:ien toimivuus haavoittuvassa ympäristössä. Internet tarjoaa valtavasti helppokäyttöisiä, haavoittuvia alustoja tietoturvasta kiinnostuneille.

Mutillidae on avoimeen lähdekoodiin perustuva web-ohjelmisto, joka sisältää lukuisia haavoittuvuuksia. Sen avulla haavoittuvuuksien paikantaminen on helppoa ja hyvin opettavaista. Mutillidae on helppo asentaa ja se toimii sekä Windows- että Linux-käyttöjärjestelmillä. Koska kyseessä on nimenomaan haavoittuvaksi tehty ohjelmisto, ei sitä pidä ajaa tuotantoverkossa, vaan eristettynä sisäverkossa. (Mutillidae n.d.) Kuviossa 6 on esitelty Mutillidaen käyttöliittymää.



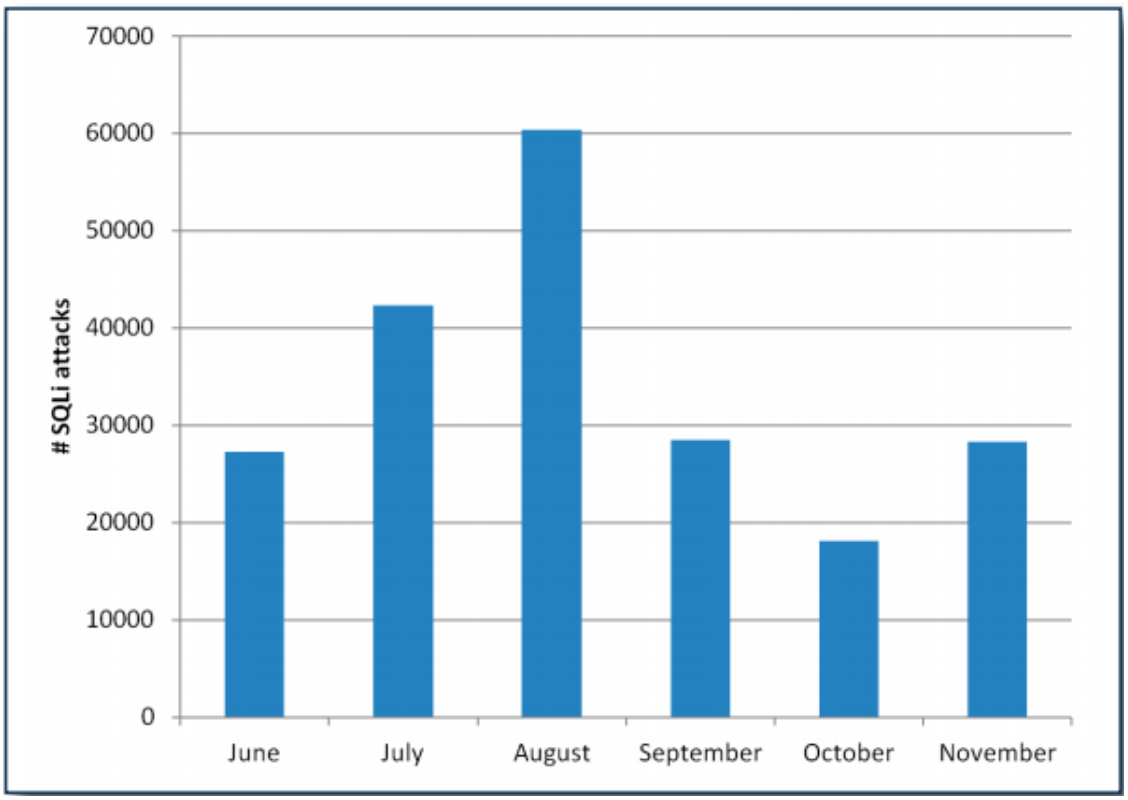
KUVIO 6. Mutillidaen käyttöliittymä

6.2 Hyödynnettävät haavoittuvuudet

WAF:ien toimintaa tarkasteltaessa hyödynnetään yleisiä web-ohjelmistoihin käytettäviä hyökkäyksiä: SQL-injektio, Cross-Site Scripting ja Local File Inclusion. Kaikki hyökkäykset suoritetaan W3af-ohjelmistolla, joka on penetraatiotestaukseen kehitetty työkalu. Apuna käytetään myös Burp Suite –työkalua. SQL-injektiossa ajetaan tarkemmat skannaukset Sqlmap-ohjelmalla.

6.2.1 SQL-injektio

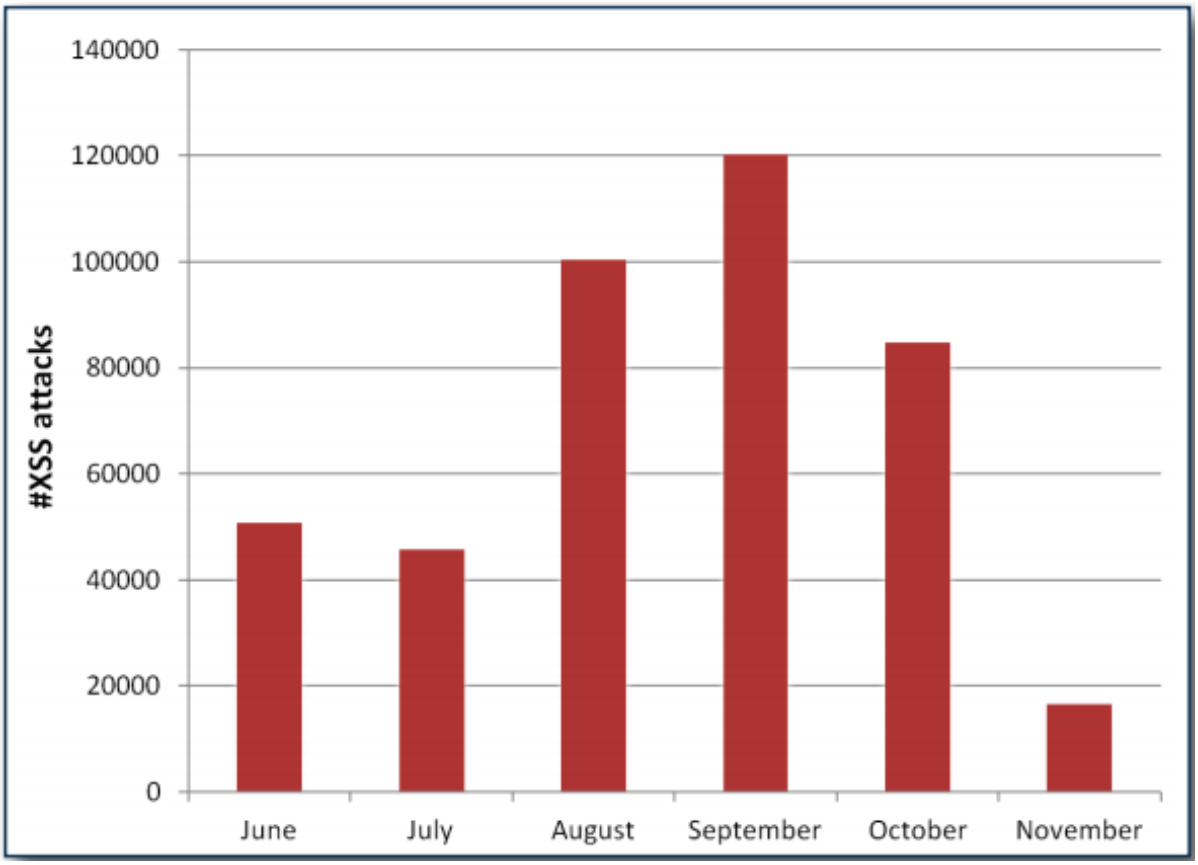
SQL-injektio on mahdollinen, kun tietokantapalvelimelle lähetettävää dataa ei tarkisteta. Kun haavoittuvuus on paikannettu, voi SQL-komentoja yhdistää ja täten saada haltuunsa arkaluonteista materiaalia suoraan web-ohjelmiston avulla. Joissain tapauksissa SQL-komentojen avulla voi suorittaa myös käyttöjärjestelmän komentoja. (Imperva – SQL Injection. n.d.) Kuviossa 7 on nähtävissä Impervan keräämää tilastotietoa SQL-injektiohyökkäyksistä.



KUVIO 7. SQL-injektioiden määrä heinäkuusta marraskuun vuonna 2011. (Imperva – Web Application Attack Report Edition #2.)

6.2.2 Cross Site Scripting (XSS)

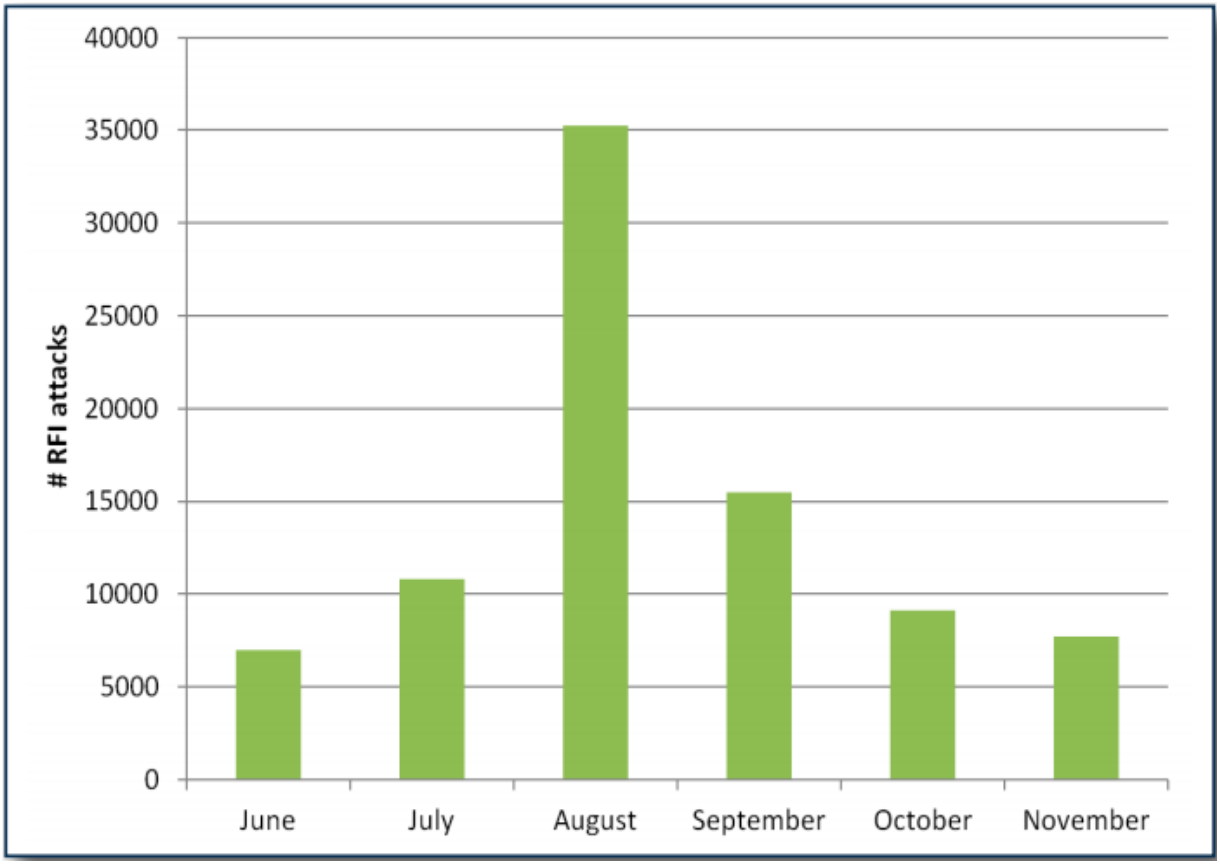
Cross-Site Scripting –hyökkäyksillä (XSS) ei pyritä hyökkäämään serverille, vaan käyttäjän selaimeen, jossa ohjelmakoodia myös suoritetaan. Haitallinen ohjelmakoodi lisätään jollekin web-sivulle, josta käyttäjien selain lataa koodin ja suorittaa sen. Koska selain kuvittelee haitallisen ohjelmakoodin olevan luotettavaa, sillä on pääsy evästeisiin, istunnon tiedostoihin sekä muuhun arkaluontoiseen dataan, jota kyseinen sivusto käsittelee käyttäjän kanssa. (Imperva – Web Application Attack Report Edition #2.) Kuviossa 8 on nähtävissä Impervan keräämää tilastotietoa XSS-injektiohyökkäyksistä



KUVIO 8. XSS-hyökkäyksien määrä heinäkuusta marraskuuhun vuonna 2011. (Imperva – Web Application Attack Report Edition #2.)

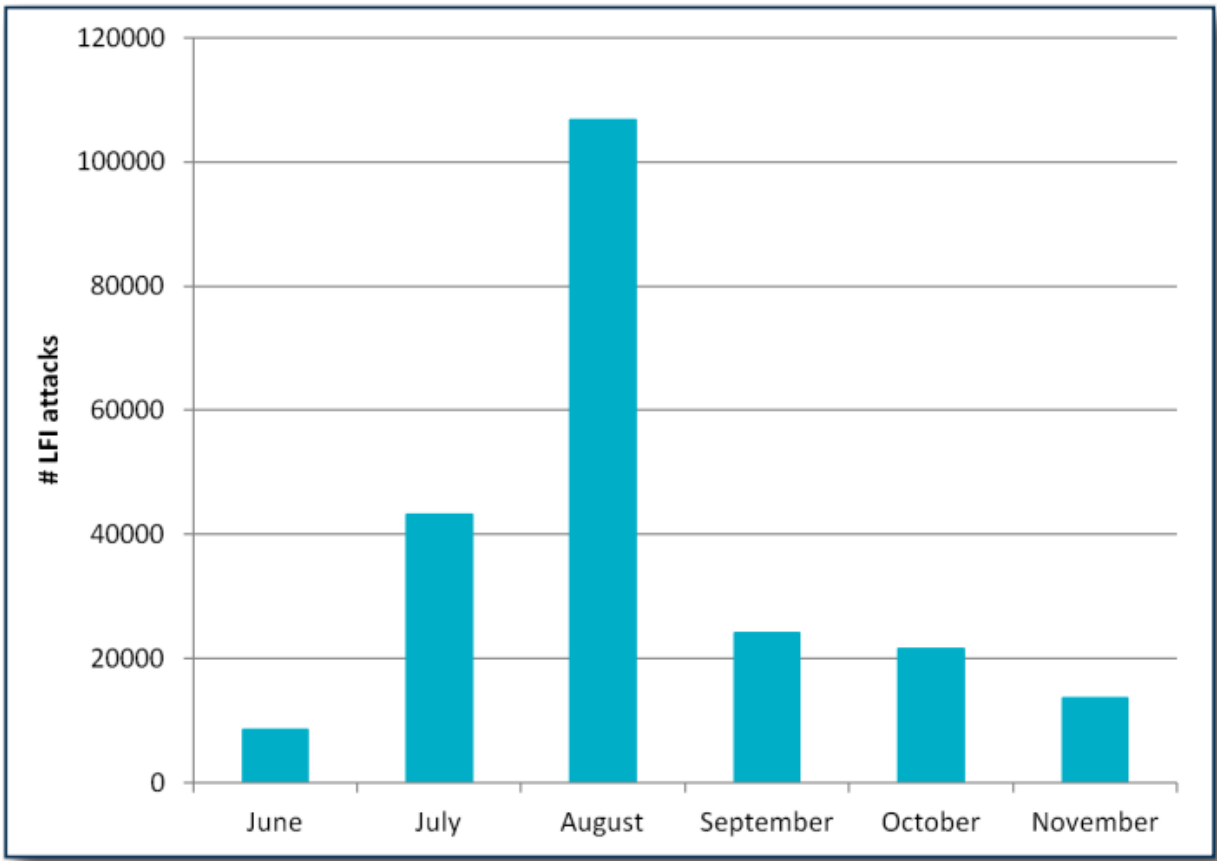
6.2.3 File Inclusion

Remote File Inclusion –hyökkäystapaa (RFI) käytetään web-ohjelmistoissa, jotka pyytävät käyttäjän toimenpiteitä esimerkiksi URL-osoitteen tai parametrin valintaan, mutta pyydettyä dataa ei vahvisteta. Hyökkääjä voi täten sisällyttää pyyntöön ulkoisen tiedoston, jonka avulla haitallista ohjelmakoodia suoritetaan palvelimella. Pahimmassa tapauksessa tämä mahdollistaa sen, että hyökkääjä ottaa haltuun koko palvelimen. (Imperva – Hacker Intelligence Initiative, Monthly Trend Report #1. 2011.) Kuviossa 9 on nähtävissä Impervan keräämää tilastotietoa RFI-injektiohyökkäyksistä



KUVIO 9. RFI-hyökkäyksien määrä heinäkuusta marraskuuhun vuonna 2011. (Imperva – Web Application Attack Report Edition #2.)

Local File Inclusion –hyökkäys (LFI) on hyvin samanlainen ulosanniltaan kuin RFI-hyökkäys, mutta se sisällyttää palvelimen omia tiedostoja haavoittuvuudessa. Vaikkakin RFI-hyökkäykset ovat yksinkertaisempia suorittaa, pyrkivät hyökkääjät käyttämään LFI-hyökkäystä. Tähän on syynä PHP-ohjelmointikielen version 5.2 muutos, joka oletuksena ei enää salli etäältä tapahtuvaa tiedoston sisällytystä. Noin 10 % palvelimista enää hyödyntää vanhempaa versiota PHP:stä. (Imperva – Remote and Local File Inclusion Vulnerabilities 101 2012.) Kuviossa 10 on nähtävissä Impervan keräämää tilastotietoa LFI-injektiohyökkäyksistä



KUVIO 10. LFI-hyökkäyksien määrä heinäkuusta marraskuuhun vuonna 2011. (Imperva – Web Application Attack Report Edition #2.)

7 TUOTTEIDEN VALINTA

Varsinaisia valintaperusteita WAF-tuotteille toimeksiantaja ei antanut, joten valinta tehtiin saatavuuden, ajantasaisuuden, mutta toki myös ominaisuuksien perusteella. Testattavaksi valittiin yksi avoimen lähdekoodin sekä yksi kaupallinen tuote.

Avoimen lähdekoodin tuotteista valittiin ModSecurity sen jatkuvan kehityksen, dokumentoinnin ja ohjelmistotuen puolesta. Muita vaihtoehtoja olivat muun muassa Naxsi, joka toimii vain Nginx-ohjelmiston kanssa sekä Ironbee, joka on vielä kehitysasteella.

Ironbee on ModSecurityn perustajan, Ivan Risticin, uusin projekti. Tällä hän pyrkii menemään vielä pidemmälle kehityksen suhteen kuin mitä ModSecurity tarjoaa. (Ristic 2011.) Ensisilmäyksellä Ironbeen projekti vaikutti hyvinkin mielenkiintoiselta.

Kaupallisten tuotteiden kohdalla mietittiin vaihtoehtoja, joissa kyseinen tuote voidaan asentaa virtuaalikoneeseen itsenäisesti. Yleensä tällaisissa hankinnoissa ollaan yhteydessä tuotteen valmistajiin, joiden kanssa valitaan sopivin vaihtoehto tarpeiden mukaan. Alustavia valintoja olivat Trustware Webdefend, Riverbed Stingray ja Impervan SecureSphere Web Application Firewall. Kaikista tuotteista löytyi virtuaalinen versio. Loppujen lopuksi kaupalliseksi tuotteeksi tuli kuitenkin F5:n BIG-IP ASM, jonka saatavuus Suomesta oli suhteellisen hyvä verrattuna muihin tuotteisiin.

8 MODSECURITY

8.1 Yleistä

Vuonna 2002 julkaistiin ModSecurityn ensimmäinen versio, joka lähti liikkeelle harrastusprojektista. Kuten ohjelmistoprojektien kanssa aina käy, vei myös ModSecurityn saaminen käytettävään kuntoon vielä kuukausia julkaisupäivän jälkeen. Ohjelma sai tuulta alleen ja alkoi nousta tietoturvapiireissä tietoisuuteen. Forrest Research testasi eri WAF-tuotteita vuonna 2006. ModSecurity pärjäsi testeissä erittäin hyvin. Pian tämän jälkeen Breach Security kiinnostui ModSecuritystä ja osti yrityksen Ivan Risticiltä. Loppuvuodesta 2006 koko ohjelmisto kirjoitettiin uusiksi ja versionumeroksi tuli 2.0. (Ristic 2012. ModSecurity Handbook.)

Tutustuttaessa eri avoimen lähdekoodin vaihtoehtoihin, osoitti ModSecurity olevansa kypsimpiä WAF-ohjelmistoja, joita on tarjolla avoimen lähdekoodin tuotteista.

8.2 Ominaisuudet

ModSecurityn ollessa avointa lähdekoodia ovat sen muokkausmahdollisuudet hyvin laajat. Tosin tämä vaatii omistautumista ja aikaa. Alla on lueteltuna tärkeimpiä ominaisuuksia:

- HTTP-liikennevirran seuranta reaaliaikaisesti ja sen tutkintamahdollisuus.
- ”Virtuaalipaikkaus”, joka mahdollistaa haavoittuvuuden korjaamisen ilman, että ohjelmaan itse tarvitsee koskea. Virtuaalipaikkausta voi hyödyntää useilla protokollilla, mutta parhaiten HTTP:n kanssa sen hyvän liikennevirran tuntemisen vuoksi. Virtuaalipaikkaus ei vaadi isoja investointeja, on helppo suorittaa ja hyödyt näkyvät heti.
- HTTP-liikenteen kirjaaminen tietoturvatarkoituksiin on hyvin vähäistä web-palvelimissa. ModSecurity mahdollistaa lokien kirjaamisen aina raakadatasta lähtien.

- Jatkuva passiivinen tietoturvan arviointi, joka reaaliaikaisesti monitoroi itse järjestelmää. Sen avulla on mahdollista tunnistaa poikkeavuudet ja heikkoudet järjestelmässä, ennen kuin on liian myöhäistä. Web-ohjelmien kovettamisella voidaan jo ennaltaehkäistä hyökkäyksiä vähentämällä HTTP-protokollan hyväksyttäviä ominaisuuksia, kuten pyyntötapoja ja –otsikoita.

(Ristic 2012. ModSecurity Handbook.)

8.3 Säännöt

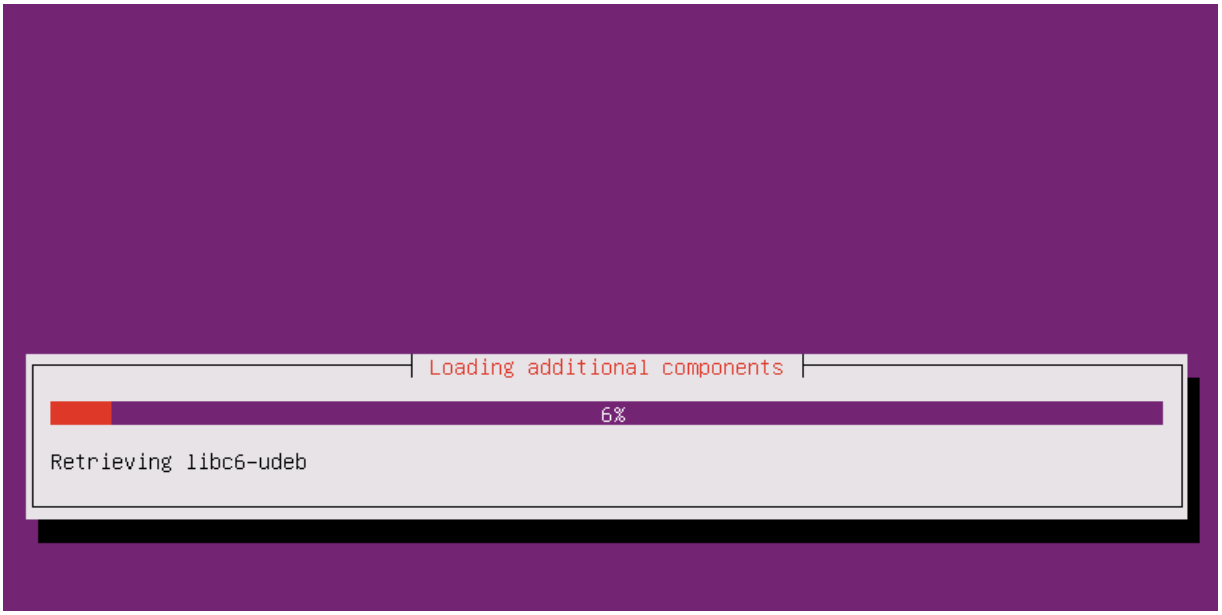
ModSecurityn toiminta perustuu ennalta määritettyihin sääntöihin. Sääntöjä ei ole valmiina ohjelmistossa, vaan ne ladattava ja otettava käyttöön. Sääntöpaketteja on kaksi erilaista: Avoimen lähdekoodin OWASP ModSecurity Core Rules Set (CRS) ja kaupallinen Trustwaren ylläpitämä sääntöpaketti. (ModSecurity Rules and Support Services n.d.) Molempia paketteja ylläpitää Trustware Spiderlabs (OWASP ModSecurity Core Rule Set Project 2012).

Säännöt kattavat haavoittuvuudet kategorioittain, kuten SQL-injektiot ja XSS-haavoittuvuudet. OWASP:n CRS-paketti takaa hyvän perus suojan hyökkäyksien havaitsemiselle ja estämiselle. Se ei kuitenkaan kykene torjua jatkuvalla syötöllä tulevia, erityisesti muodostettuja hyökkäyksiä. Näitä hyökkäyksiä vastaan Trustwave ylläpitää kaupallista sääntölistaansa, jonka tarkkuus on CRS-pakettia parempi. Tarkkuuden lisääntyessä myös ”false-positivien” määrä vähenee. (ModSecurity Rules and Support Services n.d.)

8.4 Käyttöönotto

8.4.1 Asennus

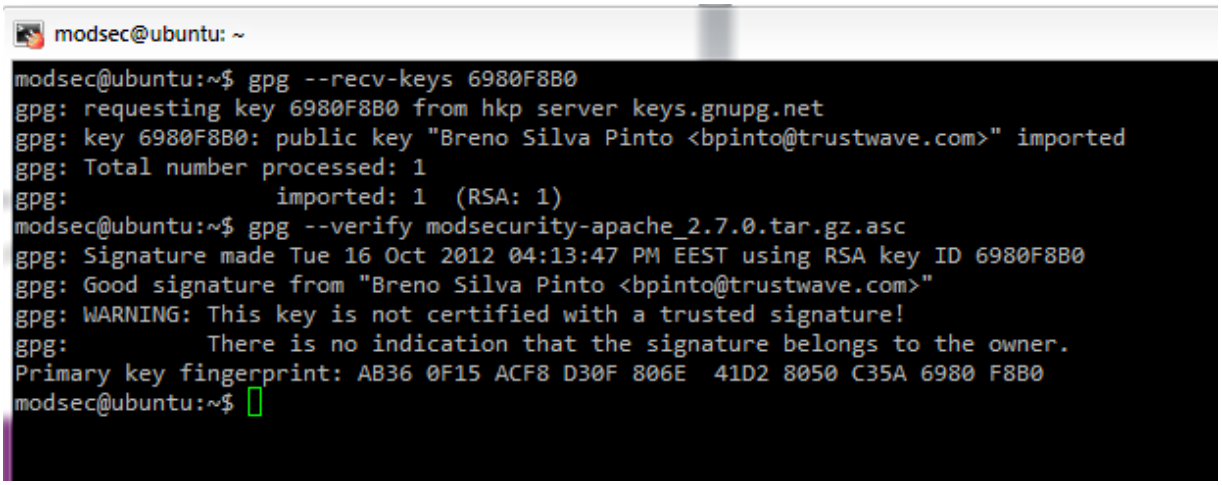
ModSecurity asennetaan web-palvelimen (Apache) yhteyteen, jonka käyttöjärjestelmänä toimii Ubuntu 12.10. Asennusalusta on virtualisoitu VMware-ohjelmistolla, joka on yhteensopiva myös toimeksiantajan järjestelmään, johon toteutus myöhemmin tulee. Kuviossa 11 on esitelty Ubuntu-käyttöjärjestelmän asennusprosessia.



KUVIO 11. Ubuntu-käyttöjärjestelmän asennus

Apache asennetaan suoraa Ubuntun ohjelmavarastosta (repository). Viimeisin sieltä löytyvä versio on 2.2.22, jossa on paikattu kaikki tärkeimmät haavoittuvuudet (Apache httpd 2.2 vulnerabilities).

ModSecurityn asennus suoritetaan suoraan lähdekoodeista, koska Ubuntun pakettienhallinnassa ei ole vielä viimeisintä versiota ohjelmasta. Jotta varmistutaan asennuspaketin eheydestä ja siitä, että sitä ei ole muokattu sen teon jälkeen, tarkastetaan paketin allekirjoitus. Kuviossa 12 on esitelty allekirjoituksen tarkistus.



KUVIO 12. ModSecurity-asennuspaketin allekirjoituksen tarkistus

Asennusta varten ModSecurity tarvitsee seuraavat riippuvuudet: automake, g++, apache2-threaded-dev, dpkg-dev, libxml2, pkg-config, libxml2-dev, libcurl4-openssl-dev ja liblua5.1-dev. Osa asennettavista riippuvuuksista on valinnaisia, mutta tarpeellisia esimerkiksi monimutkaisempia sääntöjä kirjoittaessa. Riippuvuudet asennetaan Ubuntun pakettienhallintajärjestelmällä, apt-get:llä. ModSecurityn kääntämisen ja asennuksen jälkeen Apache tarvitsee konfiguroida käyttämään ModSecurityä. Enää vuorossa on vain sääntöjen lataus ja muokkaus.

8.4.2 Core Rules Set (CRS)

Pelkkä ModSecurity ei siis tarjoa turvaa ilman sääntöjä. Kaikki säännöt ovat kommentoitu hyvin tekijöiden toimesta, jotta niiden ymmärtäminen ja muokkaaminen olisi helppoa ylläpitäjille. Core Rules Set –paketista otettiin käyttöön base rules –osio, joka sisältää seuraavat säännöt:

modsecurity_crs_20_protocol_violations.conf
modsecurity_crs_21_protocol_anomalies.conf
modsecurity_crs_23_request_limits.conf
modsecurity_crs_30_http_policy.conf
modsecurity_crs_35_bad_robots.conf
modsecurity_crs_40_generic_attacks.conf
modsecurity_crs_41_sql_injection_attacks.conf
modsecurity_crs_41_xss_attacks.conf
modsecurity_crs_42_tight_security.conf
modsecurity_crs_45_trojans.conf
modsecurity_crs_47_common_exceptions.conf
modsecurity_crs_49_inbound_blocking.conf
modsecurity_crs_50_outbound.conf
modsecurity_crs_59_outbound_blocking.conf
modsecurity_crs_60_correlation.conf

Kaikkia sääntöjä ei hyödynnetä hyökkäyksien estämisessä, koska hyökkäykset kohdistuvat vain tiettyihin hyökkäystekniikoihin.

8.4.3 ModSecurityn asetukset

Seuraavaksi käydään läpi hakemistopolussa /etc/apache2/ sijaitseva modsecurity.conf-asetustiedosto, joka on tarkoitettu vain ModSecurityn ohjelman asetuksiin.

Oletuksena ModSecurity on vain tarkkailevassa tilassa. Se kirjoittaa lokitiedostoihin havaitut hyökkäykset, mutta ei estä liikennettä. Tällä halutaan edesauttaa, että alkuvaiheessa verkon liikenne ei estyisi, ennen kuin määritykset on saatu sopiviksi. Jos kuormitus käy liian suureksi, käsitellään dataa vain osittain. (Ristic 2012. ModSecurity Handbook.) Asetus jätetään aluksi tarkkailevaan tilaan (DetectionOnly).

SecRuleEngine DetectionOnly

HTTP-pyyntö (HTTP request) koostuu osittain pakollisen otsikon lisäksi valinnaisesta sisällöstä. SecRequestBodyAccess-toiminto määrittää, annetaanko ModSecuritylle lupa tutkia myös HTTP-pyyntöjen sisältö (HTTP body). Jos ominaisuus otetaan pois käytöstä, ei HTTP:n POST-parametreja voida havaita. Myöskään datan puskurointia ei voida tehdä, mikä on oleellista hyökkäyksien estämisen kannalta. Asetus jätetään käyttöön.

SecRequestBodyAccess On

Ennen ModSecurityn 2.5-versiota huomattiin, että HTTP-pyyntöjen sisällön koon rajoittaminen on epäkäytännöllistä, jos palvelinta käytetään tiedostojen lähettämiseen. Tämän vuoksi sisällön koon rajoittamiseen on nykyään kaksi toimintoa: SecRequestBodyLimit ja SecRequestBodyNoFilesLimit. SecRequestBodyLimit:n avulla voidaan määrittää, kuinka suuret sisällöt HTTP-pyynnöissä hyväksytään tiedostot mukaan lukien, jolloin ei ole syytä epäillä kohdistetusta hyökkäyksestä. SecRequestBodyNoFilesLimit taas määrittää sisällön koon ilman tiedostoja, jolloin kokorajoitus voidaan laskea turvalliselle tasolle. SecRequestBodyInMemoryLimit on suorituskykyyn vaikuttava toiminto, joka määrittää sijainnin datan puskuroinnille riippuen HTTP-pyynnön

sisällön koosta. Koska testialustalla ei ole tarvetta lähetellä tiedostoja, jätetään asetukset oletusarvoihin.

SecRequestBodyLimit 13107200
SecRequestBodyNoFilesLimit 131072
SecRequestBodyInMemoryLimit 131072

HTTP-vastaus koostuu myös otsakkeesta ja sisällöstä. Jos SecResponseBodyAccess on käytössä, käsitellään myös palvelimelta asiakkaalle lähtevät HTTP-vastaukset. Oletuksena kyseinen asetus on pois päältä, koska hyökkäyksien torjumisen osalta se ei ole oleellisin asia ja myös palvelimen resursseja säästyy. Yhtä lailla myös HTTP-vastauksen sallittua kokoa voi muuttaa omalla asetuksella. Yleensä ModSecurityä käytetään vastauksen tarkkailuun, kun halutaan estää muun muassa tietovuodot ja asetusvirheet. Asetus otetaan käyttöön, jotta tietovuodot havaitaan paremmin.

SecResponseBodyAccess On

On myös oleellista määrittää lokien kirjaaminen. Debug-loki on hyödyllinen ongelman ratkonnan suhteen. Oletuksena sen kirjaustaso on kolme (3), jolloin kaikki oleelliset tapahtumat kirjataan. Audit-loki pitää sisällään koko tapahtuman, paitsi asiakkaalta tulevan vastauksen sisällön, joka voi olla hyvinkin suurikokoinen. Oletusasetus audit-lokin pitämiselle on, että vain oleelliset tapahtumat kirjataan, koska muuten lokitiedoston koko kasvaisi valtavaksi. Lisäksi HTTP-virhesivuilmoituksen 500-599 antavat pyynnöt kirjataan. Lokitiedostojen hakemistopolut muutetaan seuraavasti:

SecDebugLog /var/log/modsec_debug.log
SecAuditLog /var/log/modsec_audit.log

ModSecurityn asetustiedosto sisältää lisäksi muun muassa asetuksia ”multi-part/form-data”-määritykselle. Sitä suositellaan hyödynnettävän web-sivustoilla HTML-kaavakkeiden yhteydessä, kun sisältönä on muuta kuin ASCII-dataa (Forms in HTML documents n.d). Näihin asetuksiin ei ole tarvetta tehdä muutoksia.

```
SecRule MULTIPART_STRICT_ERROR "!@eq 0" \
"id:'200002',phase:2,t:none,log,deny,status:44,msg:'Multipart request body \
failed strict validation: \
PE %{REQBODY_PROCESSOR_ERROR}, \
BQ %{MULTIPART_BOUNDARY_QUOTED}, \
BW %{MULTIPART_BOUNDARY_WHITESPACE}, \
DB %{MULTIPART_DATA_BEFORE}, \
DA %{MULTIPART_DATA_AFTER}, \
HF %{MULTIPART_HEADER_FOLDING}, \
LF %{MULTIPART_LF_LINE}, \
SM %{MULTIPART_MISSING_SEMICOLON}, \
IQ %{MULTIPART_INVALID_QUOTING}, \
IP %{MULTIPART_INVALID_PART}, \
IH %{MULTIPART_INVALID_HEADER_FOLDING}, \
FL %{MULTIPART_FILE_LIMIT_EXCEEDED}'"

SecRule MULTIPART_UNMATCHED_BOUNDARY "!@eq 0" \
"id:'200003',phase:2,t:none,log,deny,status:44,msg:'Multipart parser detected a
possible unmatched boundary.'"
```

Myös hakemistopolut omiin sääntötiedostoihin voidaan määrittää ModSecurityn asetustiedostoon, jolloin omat säännöt ladataan Apachen käynnistyksen yhteydessä. Asetustiedoston loppuun on määritelty rules.conf-tiedosto seuraavasti:

```
## Custom rules
Include /etc/apache2/rules.conf
```

8.4.4 CRS-paketin asetukset

On hyvä tapa olla muokkaamatta CRS-paketin sääntöjä, vaan pitää ne ennallaan. Sääntömuutoksille tehdään oma asetustiedosto, johon muutokset kasataan. Tällöin on helppo päivittää CRS-tiedostot myöhemmin, kun muutokset ovat erillisissä tiedostoissa. Sääntöjen alustavat asetukset tehdään modsecurity_crs_10_setup.conf-tiedostoon. Yksittäisten sääntöjen ottaminen pois päältä voidaan kuitenkin tehdä suoraan CRS-tiedostoihin, jotka sijaitsevat /etc/apache2/crs-kansiossa. Säännöt ja asetukset kirjataan seuraavasti:

- /etc/apache2/rules.conf – Omat säännöt
- /etc/apache2/modsecurity_crs_10_setup.conf – CRS-paketin sääntöjen asetukset
- /etc/apache2/crs/modsecurity_custom_exceptions.conf – CRS:ien muutokset

Seuraavaksi käydään läpi modsecurity_crs_10_setup.conf-tiedoston sisältö. Oleellisin asetus modsecurity_crs_10_setup.conf-tiedostossa on SecDefaultAction, joka määrittää, miten toimitaan havaittaessa haitallista datavirtaa. Tämä on vakioasetus, mutta se on muutettavissa sääntökohtaisesti. Oletusasetus on haitallisen toiminnon estäminen ja siitä kirjoittaminen lokitiedostoon.

SecDefaultAction "phase:1,deny,auditlog"

Oletuksena CRS-paketin säännöt toimivat perinteisen tavan mukaan eivätkä ne osaa jakaa tietoa keskenään. Jokainen sääntö käydään vain yksitellen läpi. CRS-paketin asetustiedostoon voidaan määrittää käyttöön havainnointitila, jossa jokainen sääntö kasvattaa datavirrassa havaittujen poikkeamien kokonaispistemäärää. Havainnointitila on varmempi, koska useampi sääntö on osana sitä antamassa pisteytyksiä. Myös kynnysarvot on määritettävissä web-sivuston mukaan.

Havainnointitilaa ei kuitenkaan otettu käyttöön, koska se huomattavasti monimutkaisempi konfiguroida perinteiseen tapaan verrattuna. Koska alla oleva asetusrivi on ”kommentoitu”, ei asetus ole käytössä.

```
#SecAction \
 "id:'900004', \
 phase:1, \
 t:none, \
 setvar:tx.anomaly_score_blocking=on, \
 nolog, \
 pass"
```

GeoIP-tietokanta on ominaisuus, joka tekee haun IP-osoitteille. Haun tulokset kertovat IP-osoitteen maantieteellisen sijainnin kaupungin. Jotta ominaisuus on käytettävissä, on ladattava kolmannen osapuolen tietokanta, joka sisältää tarvittavan GeoIP-datan. GeoIP-ominaisuutta voi käyttää esimerkiksi lisäämään maantieteellinen data hyökkäyslokeihin. Tällöin on helppo seurata, tuleeko liikennettä maista, joista sitä ei odottaisi tulevan. Myös tietyn maan IP-osoitealueen voi estää kokonaan. Tämä voi tulla tarpeeseen, jos web-sivustoa väärinkäytetään jatkuvasti tietystä maasta. Ominaisuuden käyttöönotto tapahtuu määrittämällä hakemistopolku GeoIP-datatiedostoon. Ominaisuudelle ei nähty tarvetta eikä sitä otettu käyttöön.

```
#SecGeoLookupDb /opt/modsecurity/lib/GeoLiteCity.dat
```

Modsecurity_crs_10_setup.conf sisältää asetuksia, jotka vaikuttavat CRS-paketin sääntöön modsecurity_crs_30_http_policy.conf. Määrityksiä ovat esimerkiksi sallitut HTTP-tavat, HTTP-pyynnön sisältötyyppi, sallitut HTTP-versiot ja kielletyt tiedostopäätteet. Myös kiellettyjä otsakkeita voidaan määrittää. Sopivat asetukset ovat aina sivustoriippuvaisia. Testialustan osalta muutoksia ei tehty kuin sallitun HTTP-version osalta. Vain versio 1.1 hyväksytään. Alla ovat määritetyt asetukset kokonaisuudessaan.

```
SecAction \
  "id:'900012', \
  phase:1, \
  t:none, \
  setvar:'tx.allowed_methods=GET HEAD POST OPTIONS', \
  setvar:'tx.allowed_request_content_type=application/x-www-form-
urlencoded|multipart/form-data|text/xml|application/xml|application/x-
amf|application/json', \
  setvar:'tx.allowed_http_versions=HTTP/1.1', \
  setvar:'tx.restricted_extensions=.asa/ .asax/ .ascx/ .axd/ .backup/ .bak/ .bat/ .cdx/
.cer/ .cfg/ .cmd/ .com/ .config/ .conf/ .cs/ .csproj/ .csr/ .dat/ .db/ .dbf/ .dll/ .dos/ .htr/
.htw/ .ida/ .idc/ .idq/ .inc/ .ini/ .key/ .licx/ .lnk/ .log/ .mdb/ .old/ .pass/ .pdb/ .pol/
.printer/ .pwd/ .resources/ .resx/ .sql/ .sys/ .vb/ .vbs/ .vbproj/ .vsdisco/ .webinfo/
.xsd/ .xsx/', \
  setvar:'tx.restricted_headers=/Proxy-Connection/ /Lock-Token/ /Content-Range/
/Translate/ /via/ /if/', \
  nolog, \
  pass"
```

Brute force –hyökkäyksen avulla hyökkääjä yrittää yleensä arvata salasanoja ja täten päästä käsiksi käyttäjätunnuksiin (OWASP – Brute force attack 2009). ModSecurityssä on myös brute force –hyökkäyksen esto. Määritettäviä asetukset ovat kirjautumissivun URL-osoite, aikaikkuna monitoroinnille, raja hyökkäyksien määrälle. Lisäksi estoaika määrittää, kuinka kauan havaittua hyökkääjää pidetään estettynä. Brute force –hyökkäyksen estoa ei otettu käyttöön.

```
#SecAction \
  "id:'900014', \
  phase:1, \
  t:none, \
  setvar:'tx.brute_force_protected_urls=/login.jsp /partner_login.php', \
  setvar:'tx.brute_force_burst_time_slice=60', \
  setvar:'tx.brute_force_counter_threshold=10', \
  setvar:'tx.brute_force_block_timeout=300', \
  nolog, \
  pass"
```

Alla on esitettynä rules.conf-tiedoston sisältö, joka sisältää käyttäjän omatekemiä sääntöjä. Käyttäjän kirjoittaessa web-sivustolla sanan ”TURSKA” johonkin sisältökenttään tai osoiteriville ja tiedot lähetetään eteenpäin, tulee vastauksena HTTP 503 –virhesivu. Kyseinen tapahtuma myös kirjataan lokitiedostoihin. Esitetty esimerkki on hyvin yksinkertainen ja vain luovuus on rajana sääntöjä kirjoittaessa.

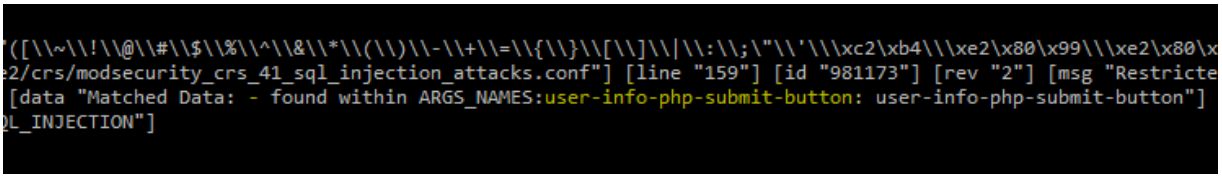
```
## Custom rules

SecRule ARGS TURSKA \
  "id:'2',phase:1,log,deny,status:503"
```

Viimeinen asetustiedosto on modsecurity_custom_exceptions.conf, jonka on tarkoitettu poikkeamille. Jos jokin sääntö tarvitsee muokkausta, tehdään se tähän tiedostoon. Alla on esitetty poikkeamatiedoston sisältö. Ensiksi sääntö 981173 poistetaan ja sen jälkeen sen sisältö muutetaan tarpeen mukaan. Poikkeaman syytä käsitellään tarkemmin sivulla 30.

```
SecRuleRemoveById 981173
SecRule ARGS_NAMES|ARGS|XML:/* "([\~\!\@\#\$\%\^\&\*\(\)\-
\+\=\{\}\[\]\|\:\;\"\'\Ã\\\\\`\<\>].*?){5,}"
"phase:2,t:none,t:urlDecodeUni,block,id:'1',rev:'2',ver:'OWASP_CRS/2.2.6',maturity:'9
',accuracy:'8',msg:'Restricted SQL Character Anomaly Detection Alert - Total # of
special characters exceeded',capture,logdata:'Matched Data: %{TX.1} found within
%{MATCHED_VAR_NAME}:
%{MATCHED_VAR}',tag:'OWASP_CRS/WEB_ATTACK/SQL_INJECTION',setvar:tx.anom
aly_score=+%{tx.warning_anomaly_score},setvar:tx.sql_injection_score=+1,setvar:'tx
.msg=%{rule.msg}',setvar:tx.%{rule.id}-
OWASP_CRS/WEB_ATTACK/RESTRICTED_SQLI_CHARS-
%{matched_var_name}=%{tx.0}"
```

Mutillidaen user-info-sivulla on mahdollista lukea haavoittuvuuden avulla myös muidenkin käyttäjätietoja. ModSecurity kuitenkin havaitsee kyseisen lomakkeen käytön haitalliseksi, kun tiedot lähetetään eteenpäin, vaikka haitallisia toimintoja ei vielä tehdä. Kuviossa 13 on ModSecurityn audit-lokista löytyvä ilmoitus haitallisesta toiminnasta.



KUVIO 13. ModSecurityn hyökkäysloki

Oletusasetuksena SQL-injektiolta suojaava sääntö estää neljä (4) erikoismerkkiä ja sitä suuremmat määrät argumentteja. Mutillidaen lähdekoodissa erikoismerkkejä on juurikin neljä, joten säännön perusteella tämä estetään. Vaihtoehtoina on joko lähdekoodin tai säännön muuttaminen, joista säännön muuttaminen onnistuu ModSecurityssä helposti. Koko sääntö poistetaan, jonka jälkeen se luodaan muuten vastaavaksi, mutta erikoismerkkien määrää nostetaan. Täten kyseinen Mutillidaen ominaisuus saadaan käytettäväksi, johon kohdistetaan hyökkäys myöhemmin. Kuviossa 14 on esitetty selaimelta lähtenyt HTTP-pyyntö palvelimelle.

KUVIO 14. Lähetettävä HTTP-pyyntö

9 F5 BIG-IP LTM/ASM

9.1 Yleistä

F5 Labs perustettiin vuonna 1996 ja vuotta myöhemmin julkistettiin heidän ensimmäinen tuotteensa – BIG-IP-kuormantasaaja. Vuoden kuluttua julkistuksesta liikevaihto oli noussut yli neljällä miljoonalla dollarilla. Läpimurto tehtiin, kun vuonna 2004 esiteltiin täysin uusi teknologia: TMOS (Traffic Management Operating System). Välityspalvelimena toimiva käyttöjärjestelmä pystyi tutkimaan ja muokkaamaan sen läpi kulkevaa liikennettä, sekä ohjata sitä optimaalisesti kuormantasausmielessä. F5:n tuotteet ovatkin suunnattu yrityskäyttöön ja F5:llä on toimintaa jopa 59 maassa. F5:n liittyessä pörssiin ennen 2000-luvun taitetta, sen nimeksi muuttui F5 Networks. (About F5 n.d.)

9.2 Ominaisuudet

F5 Application Security Manager (ASM) on ominaisuuksiltaan hyvin kattava. Mukana on muun muassa toiminto, jonka avulla tietoturvakäytäntö rakennetaan perustuen liikenteen tarkkailuun. Viruksentorjuntaskannaukset voidaan suorittaa automaattisesti ja mahdollisuus saada automaattisia päivityksiä IP-osoitteiden mustista listoista, joita käytetään haitallisiin tarkoituksiin. Kuten myös Modsecurity, F5 ASM suojaa OWASP Top 10 –uhkilta ja nollapäivähaavoittuvuuksilta. F5 ASM sisältää jo valmiiksi erilaisia suojausmalleja eri ohjelmistoille, joita voi muokata tarpeiden mukaan. (BIG-IP Application Security Manager – DATASHEET 2013) Tämä helpottaa tuotteen käyttöönottoa heti alussa.

Kun uusia hyökkäystunnisteita päivitetään tai lisätään suodatuksia, on aina vaarana, että ei-haitallista liikennettä suodattuu mukana. Tähän on apuna ”Staging”-ominaisuus. Tällöin liikennettä ei suodata uusilta säännöiltä. Liikenne päästetään läpi ja tapahtumat kirjataan lokiin. Kyseessä on ihanteellinen ominaisuus tuotantoympäristöä varten, kun halutaan varmistaa, että suojaustaso pysyy korkeana muutoksista huolimatta. (BIG-IP Application Security Manager – DATASHEET 2013.)

9.3 Käyttöönotto

F5 BIG-IP LTM (Local Traffic Manager) on kokonaisuus, johon ASM on saatavilla moduulina. Myös itsenäisiä laitteita on saatavilla (BIG-IP Application Security Manager – DATASHEET 2013).

LTM muodostaa välityspalvelimen, jonka kautta liikenne kulkee palvelimen ja käyttäjän välillä. Sen vastuulla on hoitaa muun muassa kuormantasaus ja yhteyden tilan valvominen. ASM vastaanottaa liikenteen LTM:ltä ja tutkii sen konfiguraation mukaisesti. Tarvittaessa ASM blokkaa liikenteen, jos haitallista datavirtaa havaitaan. ModSecurityn toimintaperiaate poikkeaa tässä tapauksessa F5 LTM/ASM:n toiminnasta täysin. ModSecurity on vain moduulina Apachen yhteydessä eikä hoida liikennettä välityspalvelimen tavoin.

Saatu tuote on liikenteen käsittelykyvyltään rajoitettu, kuitenkaan karsimatta ominaisuuksista. Kyseessä on siis oivallinen ratkaisu laboratorioympäristöihin ja testaukseen. Tuotetta ajetaan virtualisoidusti VMware EXSi:ssä.

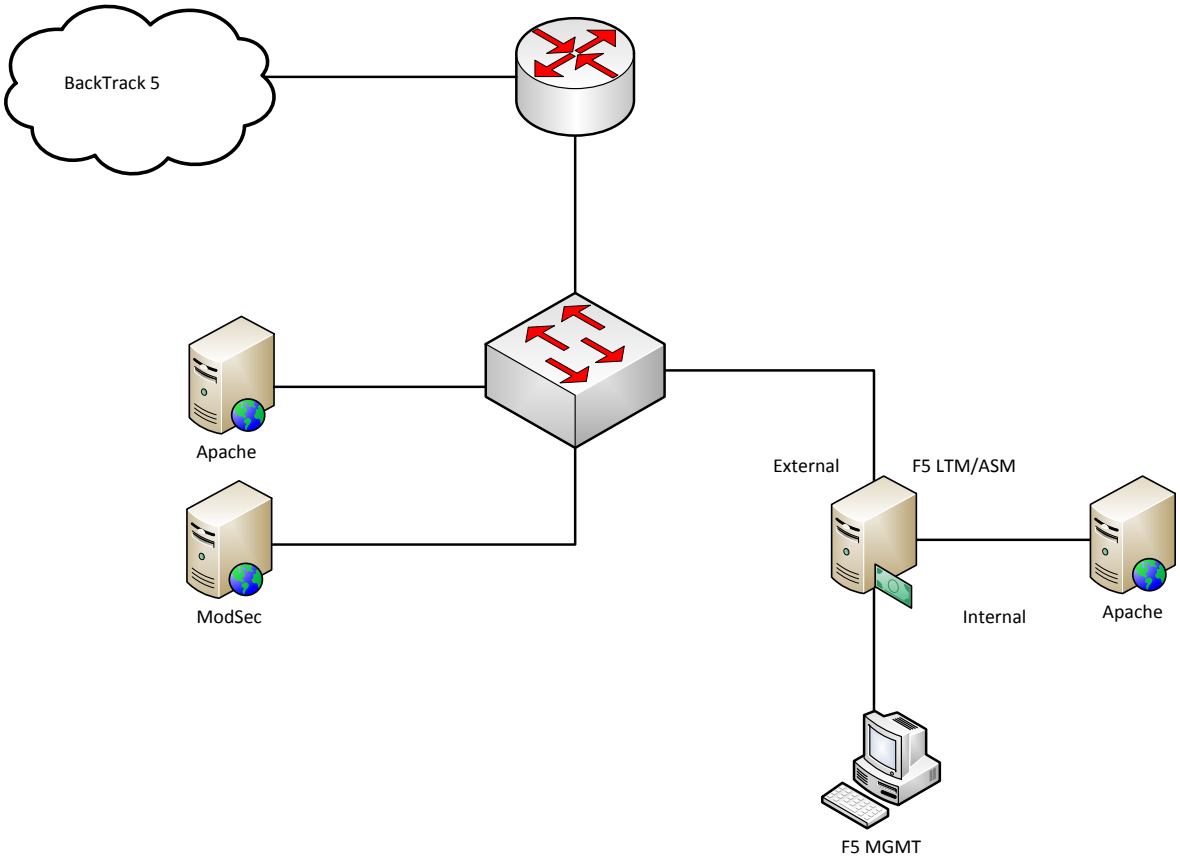
9.3.1 Verkon rakenne ja rajapinnat

F5 ASM sijoitetaan verkkoon kuvion 15 osoittamalla tavalla. Rajapintojen käyttöönotto tapahtuu VMwaren ohjelmiston kautta. Hallintaa varten on määritelty kokonaan oma rajapinta ja verkko.

Käytettäville VLAN:ille määritetään IP-osoitteet reititystä varten seuraavasti:

- External vlan = xxx.xxx.xxx.xxx/24
- Internal vlan = xxx.xxx.xxx.xxx/24

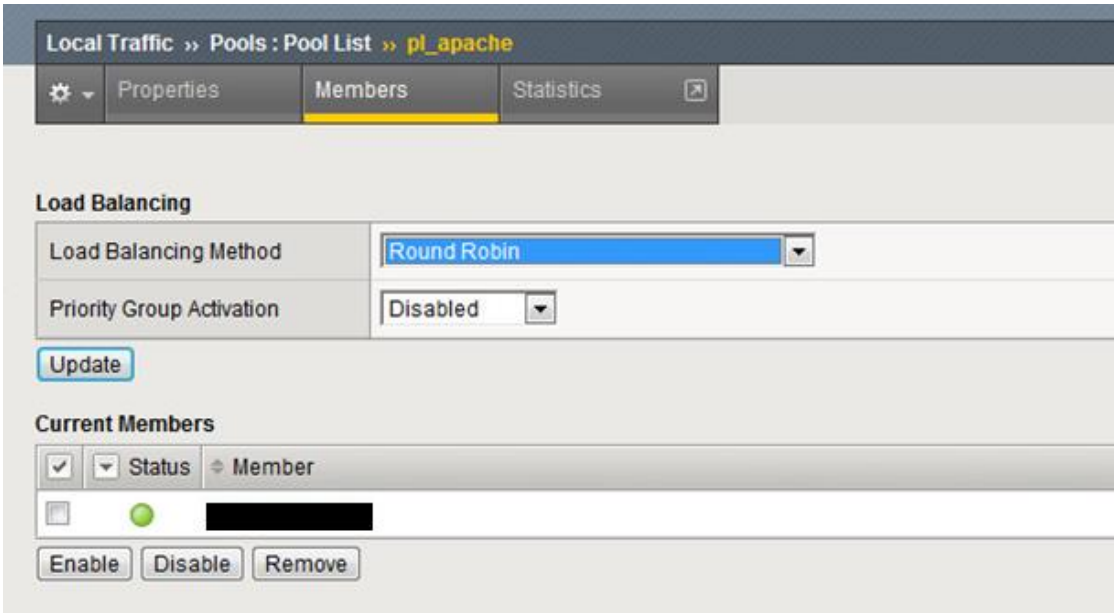
External-verkko on tässä tapauksessa ulkoverkko ja Internal-verkko on sisäverkko, josta löytyy suojattava web-palvelin. Hyökkäykset suoritetaan ulkoverkosta käsin. Kuviosta 15 on nähtävissä myös täysin suojaamaton Apache-palvelin sekä ModSecurityn suojaama palvelin. Kuva verkon topologiasta on nähtävissä myös liitteessä 3.



KUVIO 15. Verkon rakenne

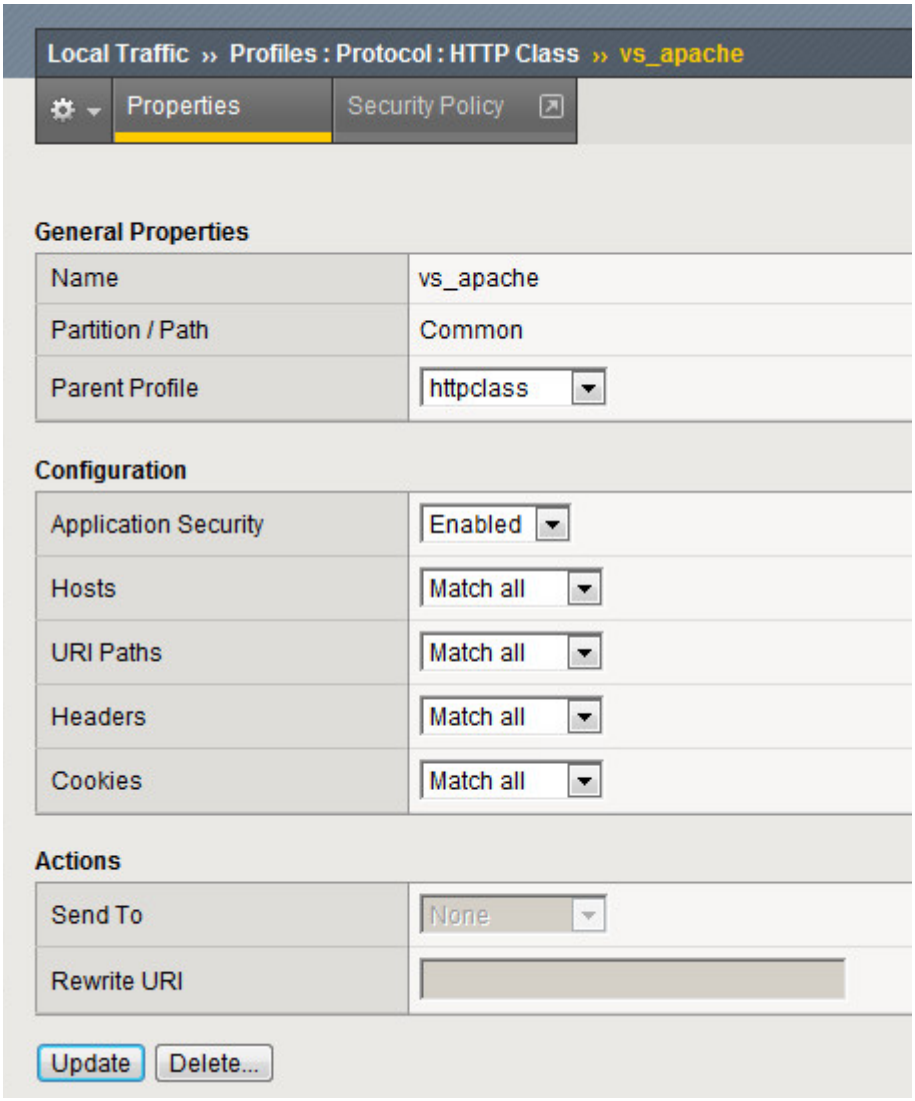
9.3.2 Yleiset asetukset

Verkon ollessa toimintakuntoinen, tulee vielä tehdä vähän määrityksiä, ennen kuin tietoturvakäytäntöä pääsee rakentamaan. Ensimmäinen vaihe on määritellä ”pooli”. Tämä sisältää palvelimet, joita halutaan suojata tietoturvakäytännöllä. Poolissa on aina jonkinlainen kuormantasaus päällä. Oletuksena käytetään ”Round Robin” –tasausta, joka jakaa liikenteen poolin palvelimille vuoropyynnöin. Tässä tapauksessa kuormantasauksella ei ole merkitystä ja asetus jää oletustilaansa. Kuviossa 16 on nähtävissä käytettävä kuormantasaus ja pooli.



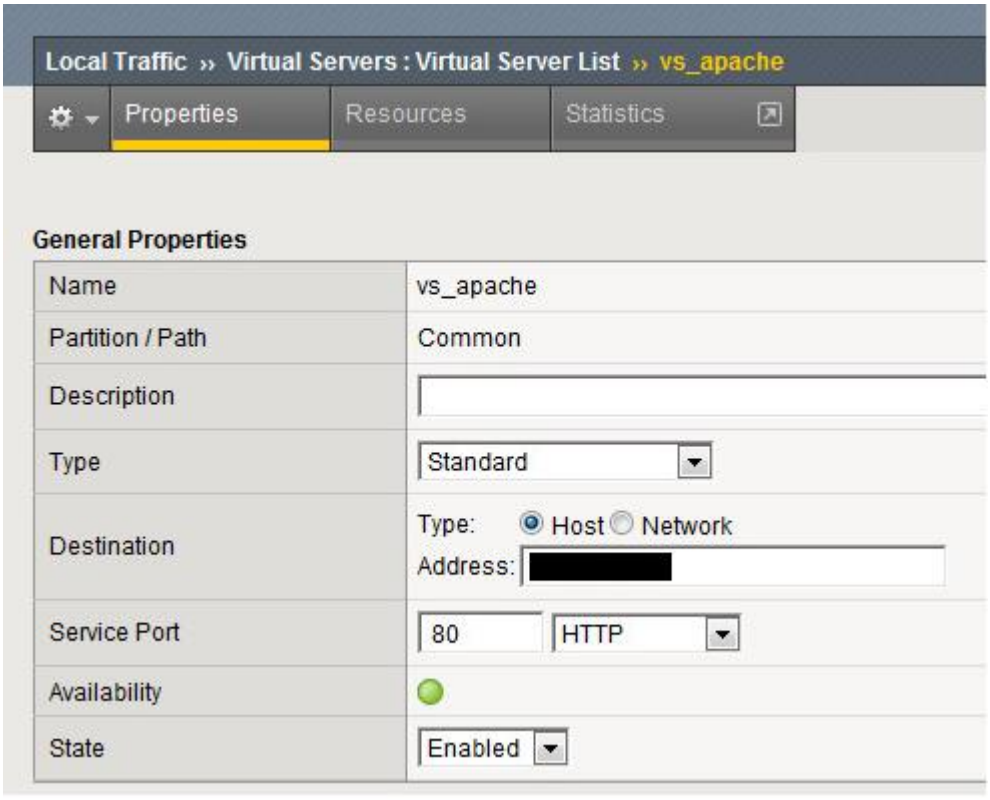
KUVIO 16. Pooli ja kuormantasaus

Kun pooli on määritelty, on aika lisätä ”HTTP Class” –profiili. Tämän profiilin avulla haluttu liikenne ohjataan LTM:lta ASM:lle tarkasteltavaksi. Määritykset voi tehdä aina yksilöidysti IP-osoitteiden perusteella tai jopa evästetasolla. Profiileita on mahdollista tehdä useampia tarkempia määrityksiä varten. Web-sisällön estämisen suhteen tulee miettiä, onko kannattavampaa käyttää HTTP Class –profiilia vai iRule-toimintoa. Oletuksena kaikki liikenne siirretään ASM:lle, ellei toisin määritetä. Kuviossa 17 on määriteltynä käytettävä profiili.



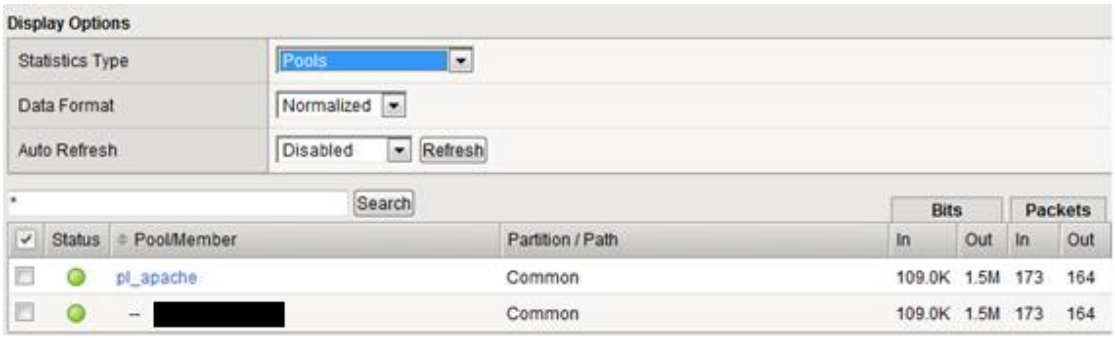
KUVIO 17. HTTP Class -profiilin asetuksia

Jotta sisääntuleva liikenne saadaan käsiteltyä, tulee määrittää käyttöön ”virtual server” -asetus. Samalla huolehditaan, että ASM:lle siirrettävä liikenne on aikaisemmin määritetyn profiilin mukaista. Virtual server saa oman IP-osoitteensa, johon yhdistetään haluttaessa web-sivulle. Muita oleellisia määrityksiä on palvelun portti (tässä tapauksessa 80), profiilin valinta ja oletuspooli. Kuviossa 18 on otos käytettävästä konfiguraatiosta.



KUVIO 18. Konfiguraatio virtual server -toimintoa varten

Edellä mainittujen toimenpiteiden jälkeen, yhteys suojattavalle web-palvelimelle toimii moitteetta. Kuviossa 19 on esitelty poolin liikestatistiikkaa. Kun liikennöinti on saatu kuntoon, voidaan alkaa turvata haavoittuvaista web-palvelinta eli perehdytään tarkemmin ASM-moduuliin.



KUVIO 19. Poolin liikennestatistiikkaa

9.4 Tietoturvakäytännön rakentaminen

Oletuksena ASM-moduuli ei ole käytössä, vaan se pitää käydä aktivoimassa erikseen. Tämän jälkeen valikkorakenne muuttuu parin valikon verran, tuoden ASM:n ominaisuudet näkyviin. Kyseinen vaihe löytyy kyllä F5:n dokumentaatiosta, mutta hieman mutkien kautta.

Suositeltavin ratkaisu F5 ASM:n kanssa olisi käyttää automaattisesti luotavaa tietoturvakäytäntöä. Tällöin opitaan millaista liikennettä kulkee web-palvelimelle ja sieltä ulospäin. Kun liikennettä seurataan useiden päivien ajan oppien ei-haitallinen datavirta, on kasassa toimiva tietoturvakäytäntö. F5 ASM:n ideana on hyödyntää positiivista tietoturvamallia, jonka tukena on vielä negatiivisen tietoturvamallin allekirjoitustietokanta. Kuten Modsecuritykin, F5 ASM ei estä liikennettä, ennen kuin se määrätään niin tekemään.

Kuviossa 20 on esitettynä tietoturvakäytännön yhteenveto. Toisin kuin tuotantoympäristössä, joissa liikenteen kulku on ratkaisevaa, ”Staging-Tightening Period” –asetus on muutettu seitsemästä päivästä nollaan.

Security Policy Properties Configuration

Security Policy Name	vs_apache2
Application-Ready Security Policy	None
Application Language	Western European (iso-8859-1)
Staging-Tightening Period	0 days
Dynamic Session ID in URL	Disabled
Security Policy is Case Sensitive	Yes

Attack Signatures Configuration

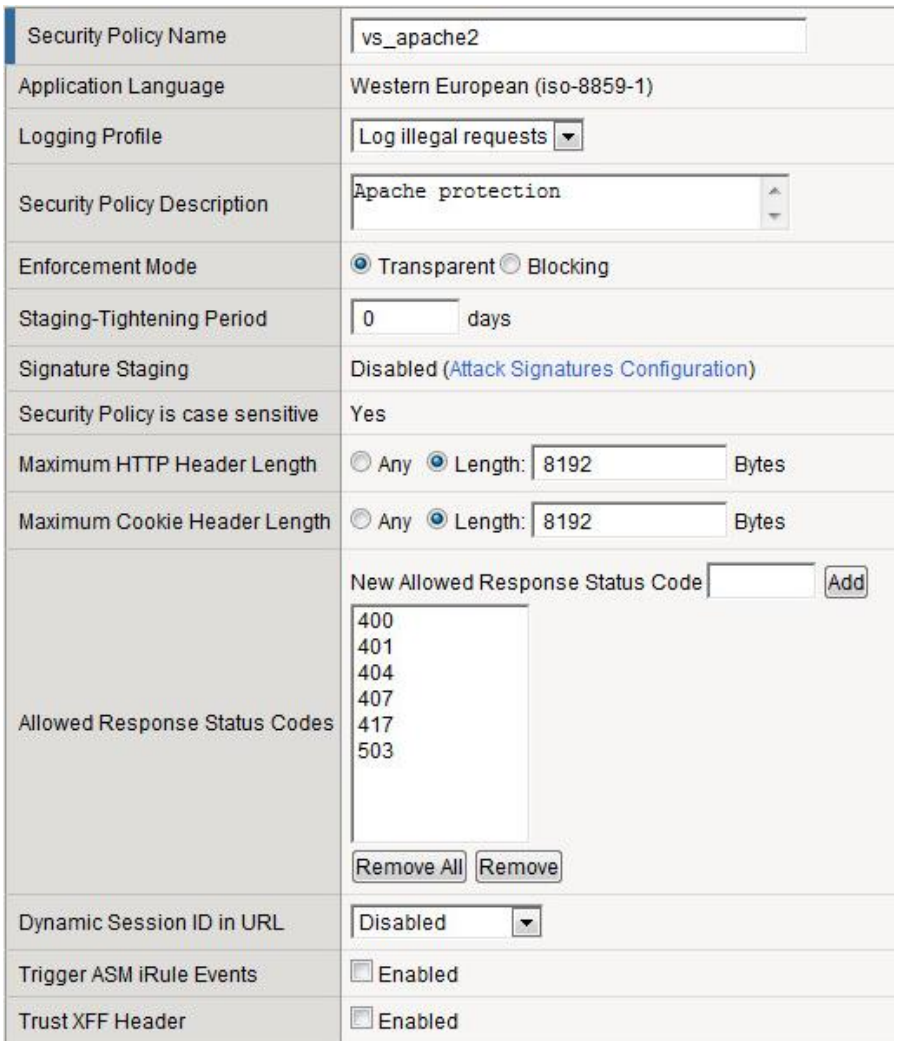
Systems	General Database, Various systems, System Independent, PHP, Apache, MySQL, Unix/Linux
Signature Sets	**Automatically assigned set(s):** Generic Detection Signatures; Systems: PHP, Apache, MySQL...
Signature Staging	Disabled
Apply Signatures to Responses	No

Wildcards Tightening Configuration

File Types Tightening	No
URLs Tightening	No
Parameters Tightening	No

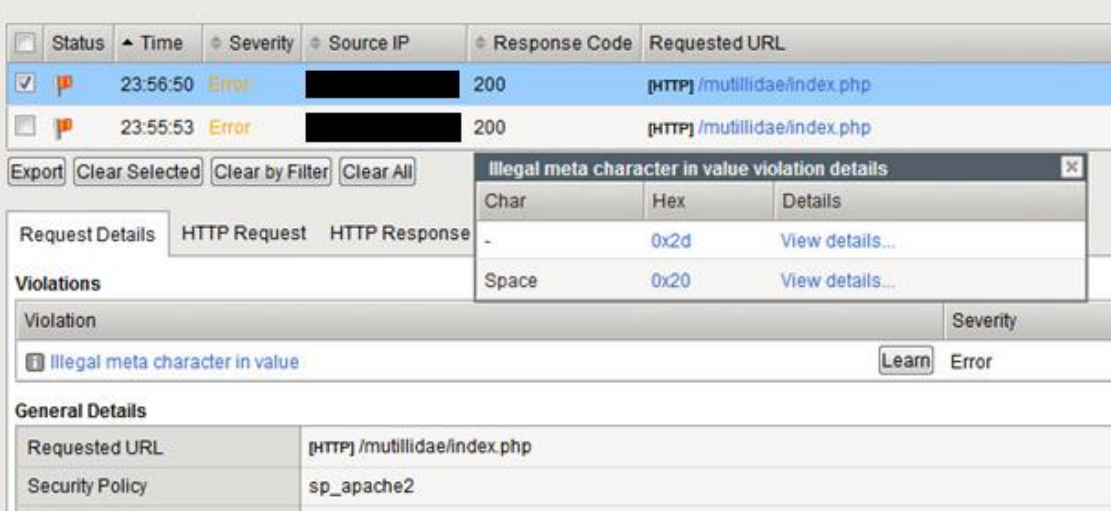
KUVIO 20. Tietoturvakäytännön yhteenveto

Tällöin muutokset tulevat käyttöön ilman odotteluaikaa. Käyttöön otettiin myös erilaisia allekirjoituspaketteja web-palvelimen alustan mukaisesti. Ylimääräiset allekirjoitukset lisäävät vain kuormitusta, joten on suotavaa pitää allekirjoitukset minimimäärässä ainakin suuremmissa kokonaisuuksissa. Allekirjoituksien osalta odotteluaika otettiin myös pois päältä. Kuviossa 21 on esitetty vielä muita asetuksia tietoturvakäytäntöön liittyen. Vain haitalliseksi havaitut pyynnöt kirjataan lokiin, mikä on suotavaa varsinkin suurilla liikennemäärillä. Tilaksi jätetään ”Transparent”, jotta mitään liikennettä ei vielä estetä.



KUVIO 21. ASM:n lisäasetuksia

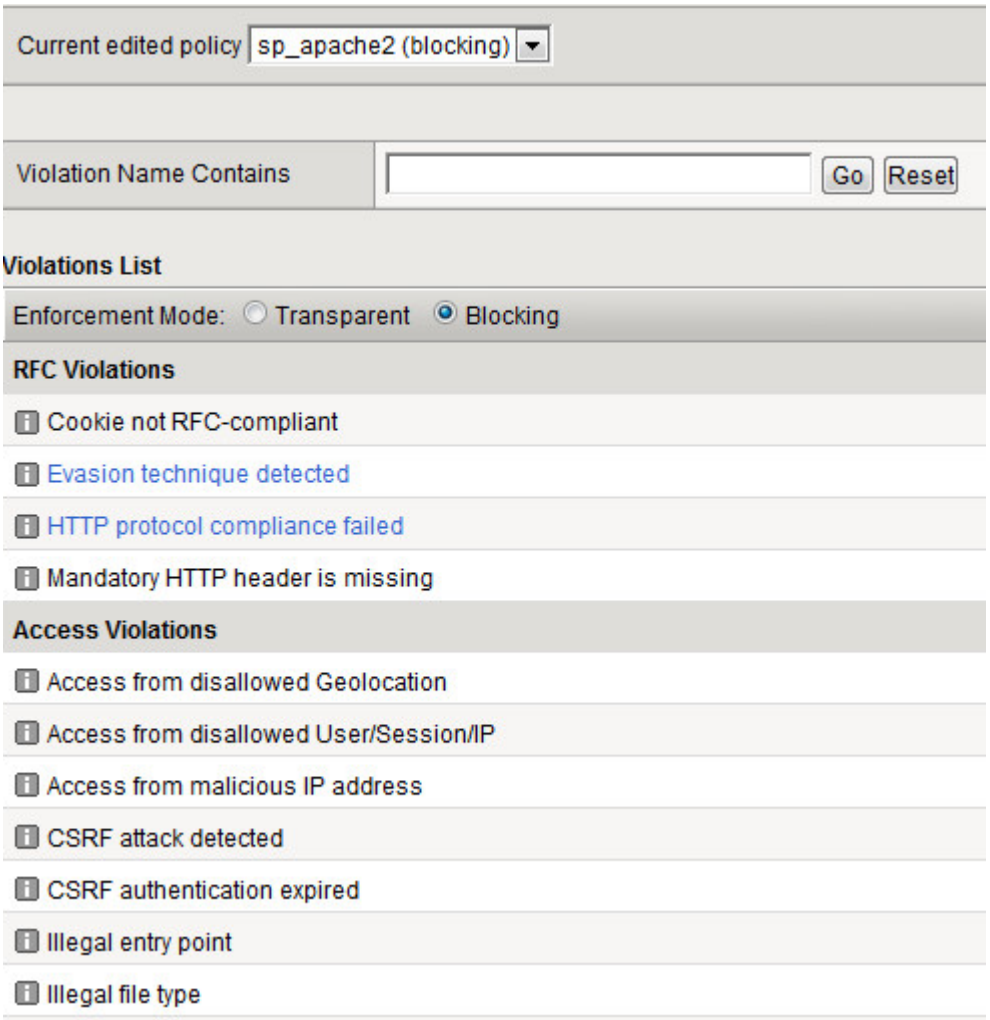
Edellä mainittujen asetusten jälkeen vahvistetaan parametrien, URL-osoitteiden sekä tiedostopäätteiden osalta niiden käyttöönotto. Aluksi on taas olennaista tutkia, että web-palvelimella oleva testausalusta ei aiheuta vääriä hälytyksiä jo valmiiksi. Kirjauduttaessa normaalisti testialustan user-info-sivulta, tunnistetaan se saman tien haitalliseksi, kuten kuviosta 22 voidaan havaita.



KUVIO 22. Haitalliseksi havaittu toimenpide

Syynä hälytykseen ovat kielletyt merkit page-parametrissä, joka pitää sisällään muun muassa web-palvelimelle lähetyt lomakkeen tiedot kuten käyttäjätunnus ja salasana. Koska kyseessä on väärä hälytys, annetaan ASM:n oppia se itse. Klikkaamalla ”Learn”, vahvistamalla haluamansa arvot ja hyväksymällä uuden tietoturvakäytännön, astuvat muutokset voimaan välittömästi. Aina muutoksia tehdessä tietoturvakäytäntö tulee hyväksyä uudelleen. Muutoksen jälkeen väärää hälytystä ei enää esiinny.
Väärä hälytys esiintyy myös ”change-log.html”-sivulla. Tähän on syynä page-parametrissä oleva vinoviiva-merkki (/). Kyseisen merkin käyttö vahvistetaan ja tietoturvakäytäntö hyväksytään.

Tietoturvakäytännön rakentaminen suoritettiin käsin. Automaattinen toiminto olisi vaatinut jatkuvampaa liikennettä ja sen valmistuminen olisi vienyt pitkään. Tarkoitus on hyödyntää vain F5 ASM:n allekirjoitustietokantaa eli toimia negatiivisen tietoturvamallin tavoin. Tällöin asetuksia tulee muuttaa seuraavasta polusta: Policy – Blocking – Settings. Kaikki valinnat otetaan pois lukuun ottamatta ”Negative Security Violations” –kohdan valintoja.



KUVIO 23. Blocking settings

10 HYÖKKÄYKSIEN SUORITTAMINEN

10.1 Alustus

Hyökkäyksiä suoritetaan web-palvelimelle hyödyntäen Burp Suite ja W3af -ohjelmia. Poikkeuksena on SQL-injektiohyökkäys, johon hyödynnetään myös Sqlmap-ohjelmaa.

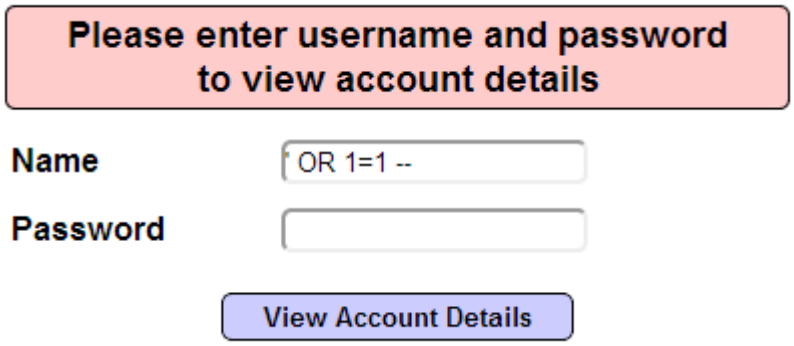
W3af on avoimen lähdekoodin haavoittuvuusskanneri web-ohjelmistoille, joka on saatavilla aina Windows-käyttöjärjestelmästä Mac OS X:ään ja Linuxiin. Haavoittuvuuksien etsimisen lisäksi, W3af mahdollistaa niiden hyväksikäytön. (W3af.org. n.d) Burp Suite on myös web-ohjelmistojen tietoturvan testaukseen tarkoitettu ohjelma, joka helppokäyttöinen aloittelijoita varten, mutta myös hyvin pitkälle konfiguroitavissa ammattilaisten käyttöä ajatellen. (Burp Suite n.d.) Sqlmap on tämän hetken vaikuttavin SQL-injektiohaavoittuvuuksiin keskittyvä skannausohjelma. Se sisältää tuen muun muassa lukuisille eri tietokantatyypeille, kuusi erilaista hyökkäystyyppiä, hash-muodossa olevien salasanojen murtamisen sanakirja-hyökkäyksellä ja on kaiken lisäksi vielä avointa lähdekoodia. (Sqlmap.org. n.d.)

Havainnollistamisen vuoksi hyökkäykset ajetaan aluksi niin, että WAF:t ovat tarkkailevassa tilassa, eli ne eivät estä mitään hyökkäyksiä, mutta kirjoittavat havaintonsa lokitiedostoon. Tämän jälkeen WAF:t laitetaan estämään havaitsemansa hyökkäykset. Vertailut käydään läpi WAF:ien kesken ja tutkitaan, onko niissä merkittäviä eroavaisuuksia. Koska käytetyt hyökkäykset on jo määritelty ja tiedossa haavoittuvasta web-ohjelmistosta, ei niitä etsitä erikseen, vaan kohdistetaan hyökkäykset suoraan web-ohjelmiston haavoittuviin osiin. SQL-injektiossa suoritetaan hieman pidemmät ja intensiteettisemmät hyökkäykset kuin muiden haavoittuvuuksien osalta.

10.2 Haavoittuvuudet ilman suojausta

10.2.1 SQL-injektio

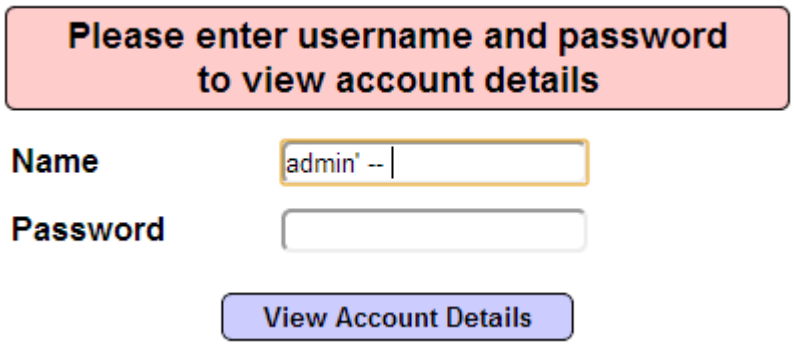
SQL-injektiossa käytetään Mutillidaesta löytyvää lomaketta, jolla voi tarkastella omia rekisteröintitietojaan antamalla ensin käyttäjätunnuksen ja salasanan. Haavoittuvuus on tässä tapauksessa helppo todentaa syöttämällä tietty erikoismerkki lomakkeen syötekenttiin, jolloin sivusto ilmaisee sivulatauksen jälkeen, että annettu SQL-lauseke on virheellinen. Itse haavoittuvuuteen käytämme kuvion 23 nimi-kentässä olevaa lauseketta.



KUVIO 24. SQLi-lauseke

Annettu SQL-injektio on aina tosi kaikkien tietojen kohdalla ja se myös sulkee loput pyynnöt pois, jolloin Password-kenttää ei oteta huomioon suorittaessa. Täten tuloksena on kaikkien käyttäjätietojen listaus, mitä SQL-taulu sisältää. Kuviossa 25 on osa käyttäjätunnuksista, jotka listattiin hyödyntäen SQL-injektiota.

Muitakin vaihtoehtoisia SQL-injektioita voi käyttää. Esimerkiksi hyökkääjä voi olettaa, että admin-käyttäjätunnus on käytössä ja ohittaa autentikoinnin kuvion 24 Name-kentässä olevan lausekkeen mukaisesti. Tällöin tuloksena saadaan kuitenkin vain admin-käyttäjätunnuksen tiedot.



KUVIO 25. Vaihtoehtoinen SQL-injektio

Username=admin
Password=adminpass
Signature=Monkey!

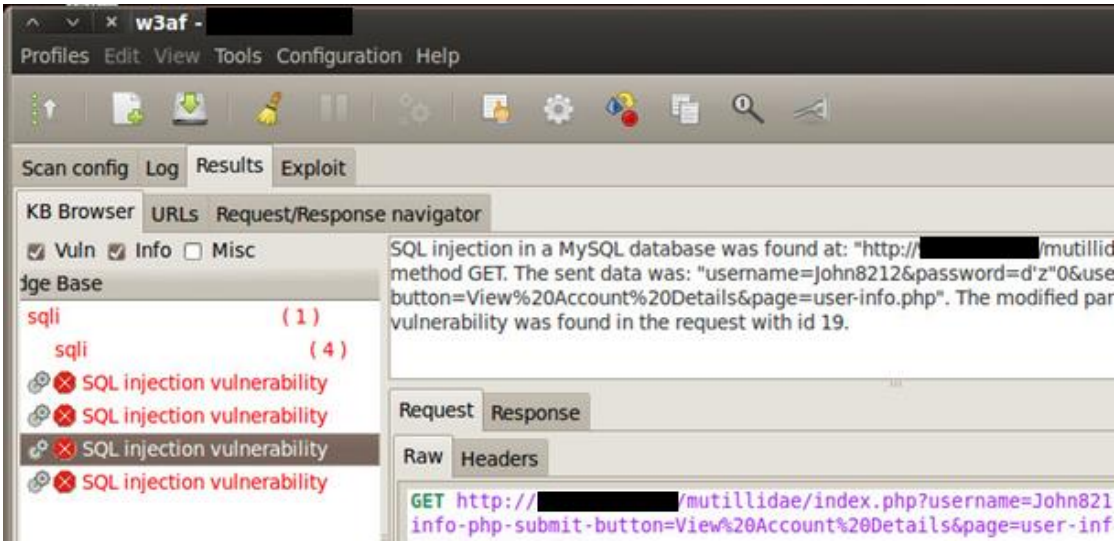
Username=adrian
Password=somepassword
Signature=Zombie Films Rock!

Username=john
Password=monkey
Signature=I like the smell of confunk

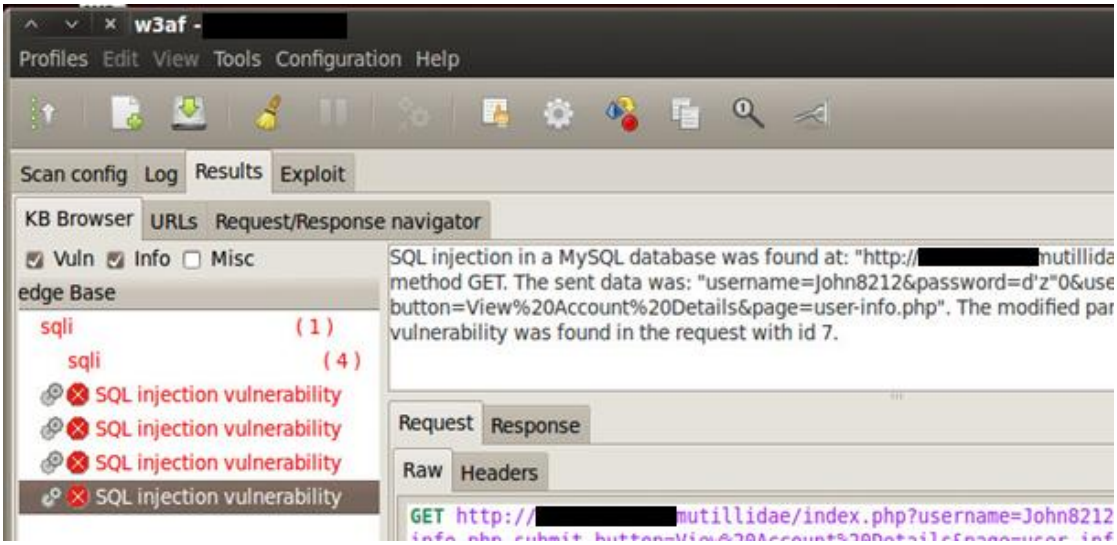
Username=jeremy
Password=password
Signature=d1373 1337 speak

KUVIO 26. Tunnuslistaus

Skannaus SQL-injektioita paikantaessa suoritetaan aluksi vain W3af-ohjelmistolla. Kuviossa 26 on ModSecurityn suojaama web-palvelin ja kuviossa 27 F5 ASM:n suojaama web-palvelin. Molemmissa skannauksissa W3af havaitsee kohteet haavoittuviksi.



KUVIO 27. ModSecurity tarkkailevassa tilassa

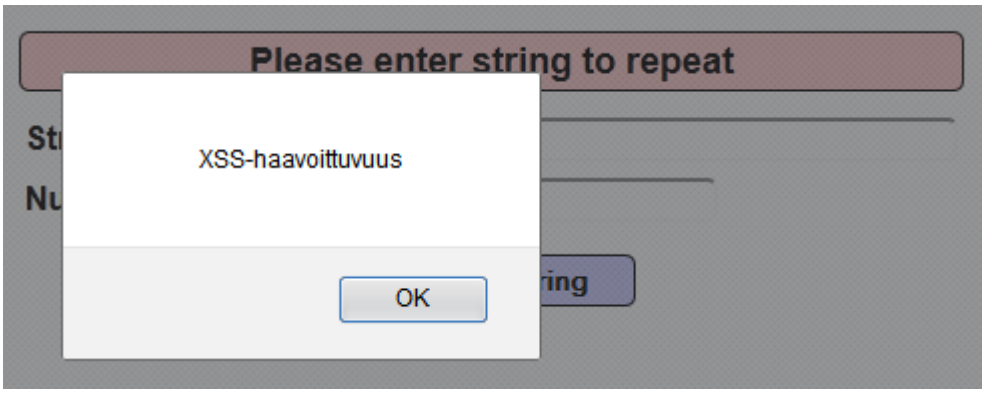


KUVIO 28. F5 ASM tarkkailevassa tilassa

10.2.2 Cross-Site Scripting (XSS)

Repeater-lomakkeella suoritamme XSS-hyökkäyksen, jossa serveri palauttaa käyttäjälle vastauksen, joka sisältää käyttäjän syöttämää koodia. Tällaista hyökkäystä kutsutaan ”XSS Reflecter” –hyökkäykseksi (OWASP - Cross-site Scripting (XSS) 2011). Koska lomakkeen syötekentän sisältöä ei tulkita oikein, suorittaa selain sen normaalisti, eikä käsittele sitä tekstinä kuten pitäisi. Kuviossa 28 on lomakkeelle syötetty esimerkki, jonka tulos on esitetty suorituksen jälkeen kuviossa 29.

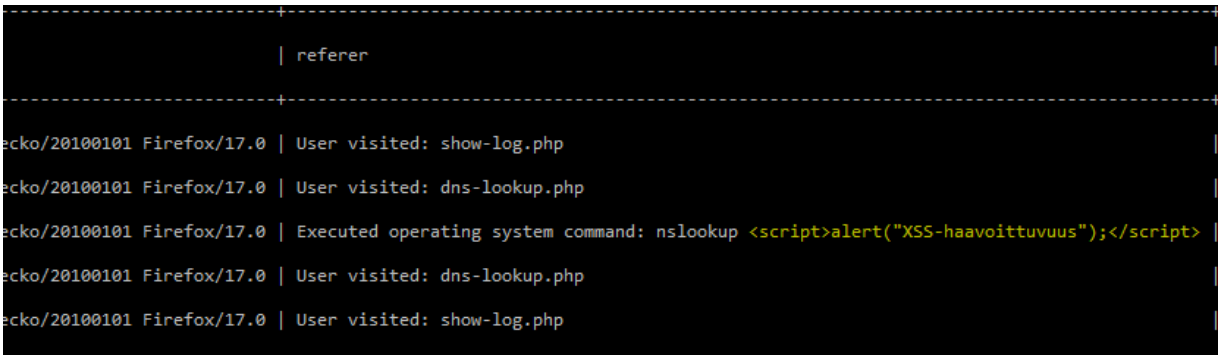
KUVIO 29. XSS-lauseke



KUVIO 30. XSS-haavoittuvuus

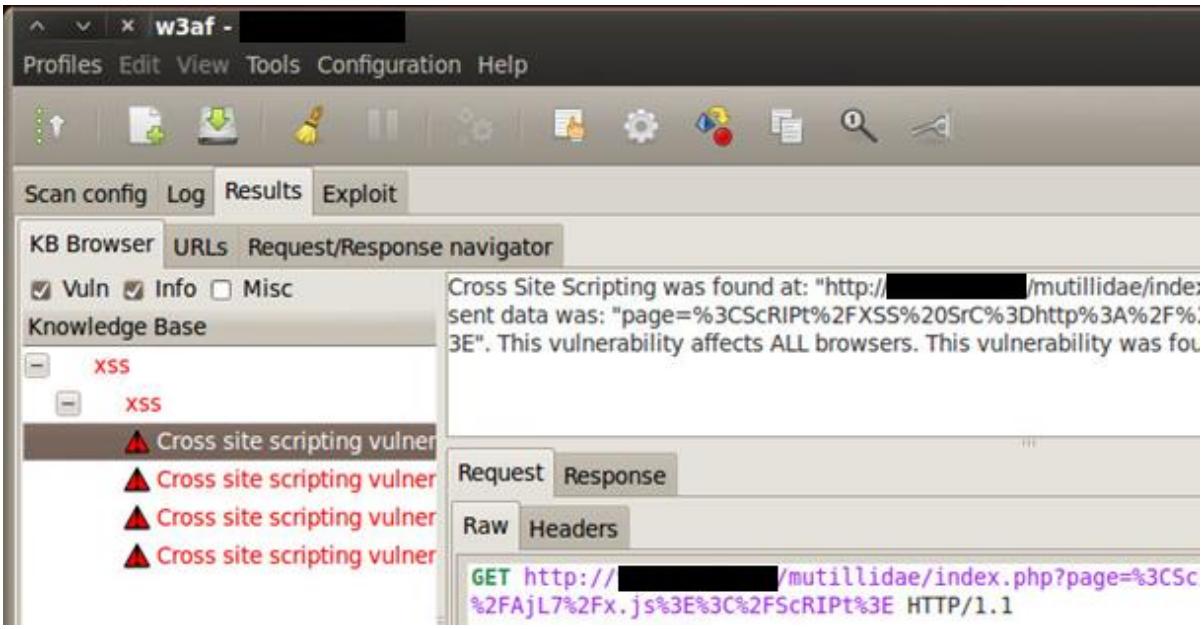
Mutillidaessa on loki, johon kirjataan kaikki sivuston tapahtumat. Kun XSS-hyökkäys suoritettiin, tehtiin siitäkin merkintä lokiin, joka sisältää myös käyttäjän syöttämän datan. Lokisivun ollessa myös puutteellinen tietoturvan osalta, ilmenee aikaisemmin

tehty XSS-hyökkäys toistamiseen lokisivulle mentäessä, kun tallennettu XSS-lauseke suoritetaan automaattisesti uudelleen. Tätä kutsutaan ”XSS Persistent” tai ”XSS Stored” –hyökkäykseksi (OWASP - Cross-site Scripting (XSS) 2011). Kuviossa 30 on esitetty lokiin tallennetut tiedot suoraan MySQL-tietokannasta.

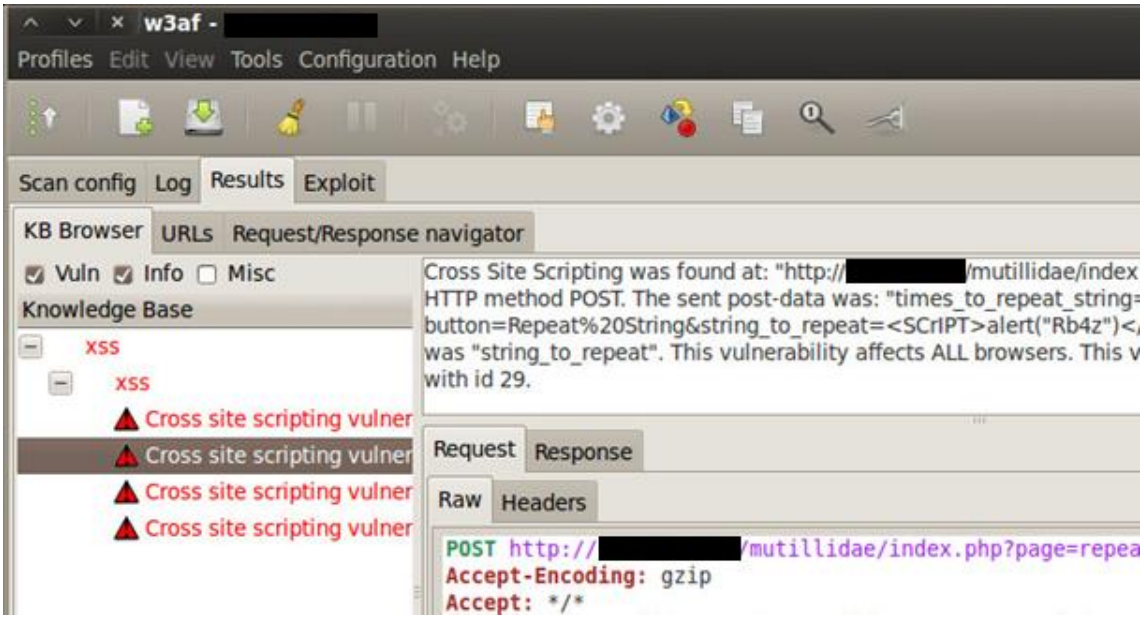


KUVIO 31. XSS-lauseke MYSQL-tietokannassa

Kuviossa 31 ja 32 on esitettynä W3af-työkalun skannaustulos edellä mainitusta XSS-haavoittuvuudesta. XSS-haavoittuvuus löytyi syötekenttien lisäksi myös web-sivuston page-parametristä, jota on testattu kahdella eri XSS-lausekkeella.



KUVIO 32. XSS-hyökkäys ModSecurityn ollessa tarkkailevassa tilassa



KUVIO 33. XSS-hyökkäys F5 ASM:n ollessa tarkkailevassa tilassa

10.2.3 Local File Inclusion

Mutillidaen sivusto on suunniteltu siten, että siirryttävä sivu valitaan page-parametriä käyttäen. Kyseinen parametri on haavoittuvainen, koska se käyttää $_REQUEST-muuttujaa ilman muita tarkistuksia.

```
$lPage = $_REQUEST["page"];
```

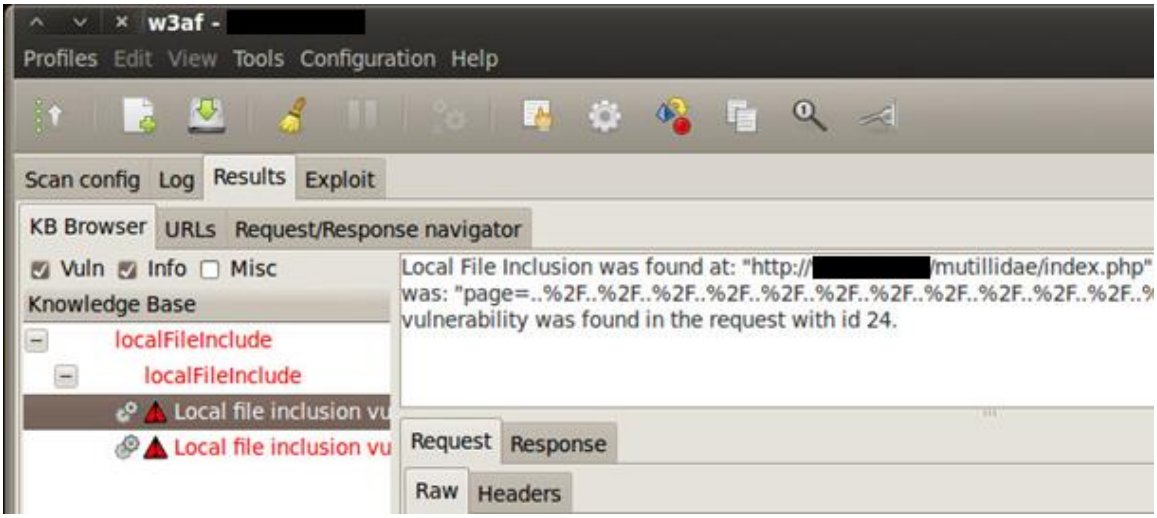
Tällöin hyökkääjä voi itse valita haluamansa syötteen suoritettavaksi. Page-parametri muutetaan osoittamaan palvelimella olevaan tiedostoon seuraavasti:

*/mutillidae/index.php?**page**=/etc/passwd*

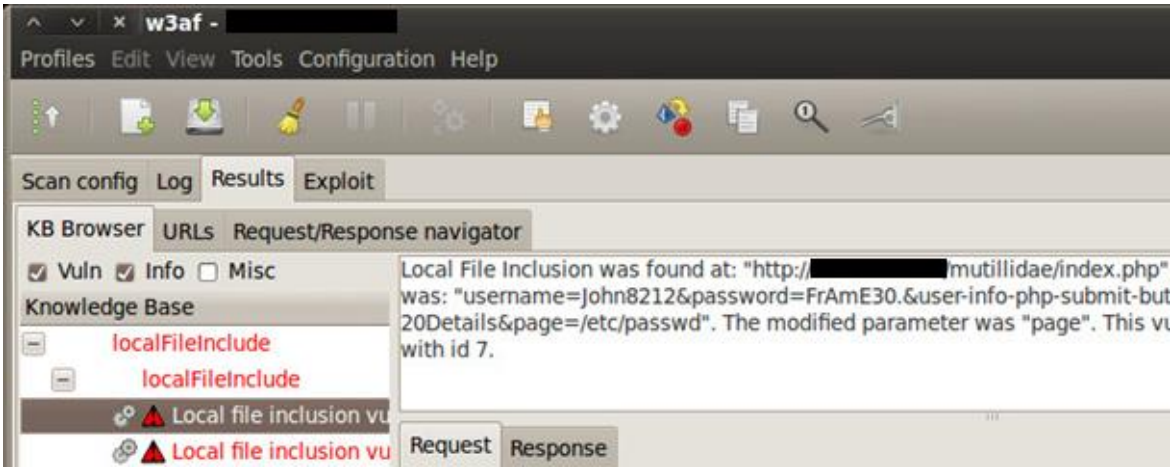
Tuloksena saadaan palvelimen käyttöjärjestelmässä olevan passwd-tiedoston sisältö, joka on esiteltynä kuviossa 33. Koska lähdekoodissa ei ole määritelty edes haluttua tiedostopäätettä, onnistuu eri järjestelmätiedostojen luku riippuen käyttöoikeuksista.

KUVIO 34. /etc/passwd-tiedoston sisältö

Kuvioissa 34 ja 35 ovat W3af-työkalun skannaustulokset LFI-haavoittuvuudesta. Kaikista skannaustuloksista on nähtävissä yhdenmukaisuus. Koska käytettävät web-palvelimet ovat identtiset ja WAF:it eivät estä liikennettä, on tulos odotettu.



KUVIO 35. LFI-hyökkäys F5 ASM:n ollessa tarkkailevassa tilassa



KUVIO 36. LFI-hyökkäys ModSecurityn ollessa tarkkailevassa tilassa

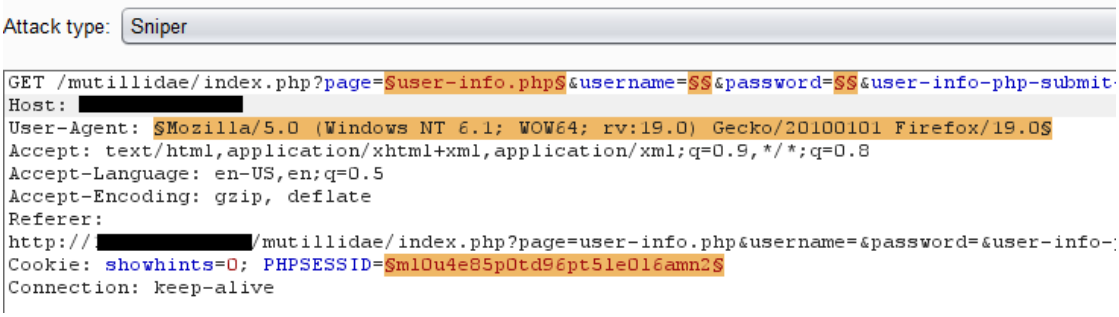
10.3 Haavoittuvuudet suojauksen kanssa

10.3.1 Alustus

Seuraavaksi ModSecurity asetetaan estämään kaikki haitalliseksi havaitsemansa toimenpiteet. Tämä onnistuu ModSecurityn asetustiedostosta laittamalla SecRuleEngine-kohdan tilaan ”On”. Asetuksien muutoksien voimaantulo vaatii Apache-web-palvelimen uudelleenkäynnistyksen. F5 ASM:n tietoturvakäytännöstä laitetaan hyökkäyksien esto myös päälle. Kaikki hyökkäyssivut ovat vastaavat kaikilla tekniikoilla, mitä aikaisemmin on suoritettu ilman suojausta. Osaa tekniikoista kokeillaan ensin Burp Suitella ja sitten automatisoidummin muulla työkalulla.

10.3.2 SQL-injektio

SQL-injektiota etsittäessä hyödynnetään Burp Suite –ohjelmistoa. Burp Suiten intruder-toiminnon avulla haluttu tietosisältö on helppo lähettää eteenpäin hyökkäyksen kohteena olevalle sivustolle. Lähetettävät hyökkäyslausekkeet ovat sotkettu muun muassa kommenttien ja heksamerkkien avulla. Myös hyökkäyslausekkeiden sijoitukset on helppo määrittää Burp Suitessa, kuten kuviosta 36 on nähtävissä. Tarkoitus on lähinnä testata, miten WAF:it reagoivat erityylisiin, SQL-injektioilta näyttäviin tietosisältöihin. Käytettävät parametrit ovat page, username, password ja user-agent. Lisäksi injektiolausekkeita sijoitetaan evästeisiin.



KUVIO 37. Hyökkäyslausekkeiden sijoitus Burp Suite -ohjelmassa

Kuviossa 37 on esitetty osa tuloksista hyökkäyslausekkeiden lähetyksen jälkeen, kun vastassa on ModSecurity. Kaikki hyökkäyslausekkeet estetään. Ensimmäinen pyyntö (nolla) on ei-haitallinen pyyntö, jotta saadaan selville palautettavan sivun koko. Täten on helppo verrata haitallisia pyyntöjä ja pysyykö koko samana, sillä tilakoodi ei vältämättä kerro aina koko totuutta.

Request ▲	Position	Payload	Status	Error	Timeout	Length	Comment
0			200	☐	☐	28807	baseline request
1	1	UNION SELECT 1,2,3,4	403	☐	☐	503	
2	1	/*!UNION*/ /*!SELECT*/ 1,2...	403	☐	☐	503	
3	1	uNIoN sELecT 1,2,3,4	403	☐	☐	503	
4	1	/*!uNIOn*/ /*!SelECt*/ 1,2,3,4	403	☐	☐	503	
5	1	UNIunionON SELselectECT ...	403	☐	☐	503	
6	1	%55nION/**/%53ElecT 1,2,...	403	☐	☐	503	
7	1	/*!u%6eion*/ /*!se%6cect*/...	403	☐	☐	503	
8	1	+(uNioN)+(sElECt)	403	☐	☐	503	
9	1	+(UnI)(oN)+(SeL)(ecT)+	403	☐	☐	503	
10	1	+(uNioN+SeleCT)+	403	☐	☐	503	
11	1	-3333' AND 1=1	403	☐	☐	503	
12	2	UNION SELECT 1,2,3,4	403	☐	☐	503	
13	2	/*!UNION*/ /*!SELECT*/ 1,2...	403	☐	☐	503	
14	2	uNIoN sELecT 1,2,3,4	403	☐	☐	503	
15	2	/*!uNIOn*/ /*!SelECt*/ 1,2,3,4	403	☐	☐	503	
16	2	UNIunionON SELselectECT ...	403	☐	☐	503	

KUVIO 38. ModSecurityä kohti lähetetyt hyökkäyslausekkeet

Kuviossa 38 on esitetty osa tuloksista hyökkäyslausekkeiden lähetyksen jälkeen, kun vastassa on F5 ASM. Pyynnöt estetään lukuun ottamatta user-agent-otsaketta ja evästeitä. F5 ASM ilmoittaa tilakoodiksi estämisestä huolimatta ”200”, joka on ”OK”. On kuitenkin helppo todeta, että web-palvelin ei lähetä alkuperäistä sivua, koska palautetun sivun koko on hyvin pieni. Asian voi vielä varmistaa HTTP-vastauksen sisällöstä.

F5 ASM sallii estämisestä kertovan sivun muokkaamisen aina ulkoasusta tilakoodin muuttamiseen. Ei-estettyjen vastauksien koko vaihtelee user-agent-otsaketta testattaessa johtuen siitä, että web-sivulla esitetään aina käytettävä user-agent tekstimuodossa.

Request ▲	Position	Payload	Status	Error	Timeout	Length
35	4	/*!UNION*/ /*!SELECT*/ 1,2,3,4	200	☐	☐	336
36	4	uNIoN sELecT 1,2,3,4	200	☐	☐	336
37	4	/*!uNIOn*/ /*!SelECt*/ 1,2,3,4	200	☐	☐	336
38	4	UNIunionON SELselectECT 1,2,3,4	200	☐	☐	28608
39	4	%55nION/**/%53ElecT 1,2,3,4	200	☐	☐	336
40	4	/*!u%6eion*/ /*!se%6cect*/ 1,2,3,4	200	☐	☐	336
41	4	+(uNioN)+(sElECt)	200	☐	☐	336
42	4	+(UnI)(oN)+(SeL)(ecT)+	200	☐	☐	336
43	4	+(uNioN+SeleCT)+	200	☐	☐	336
44	4	-3333' AND 1=1	200	☐	☐	336
45	5	UNION SELECT 1,2,3,4	200	☐	☐	28564
46	5	/*!UNION*/ /*!SELECT*/ 1,2,3,4	200	☐	☐	28590
47	5	uNIoN sELecT 1,2,3,4	200	☐	☐	28564
48	5	/*!uNIOn*/ /*!SelECt*/ 1,2,3,4	200	☐	☐	28590

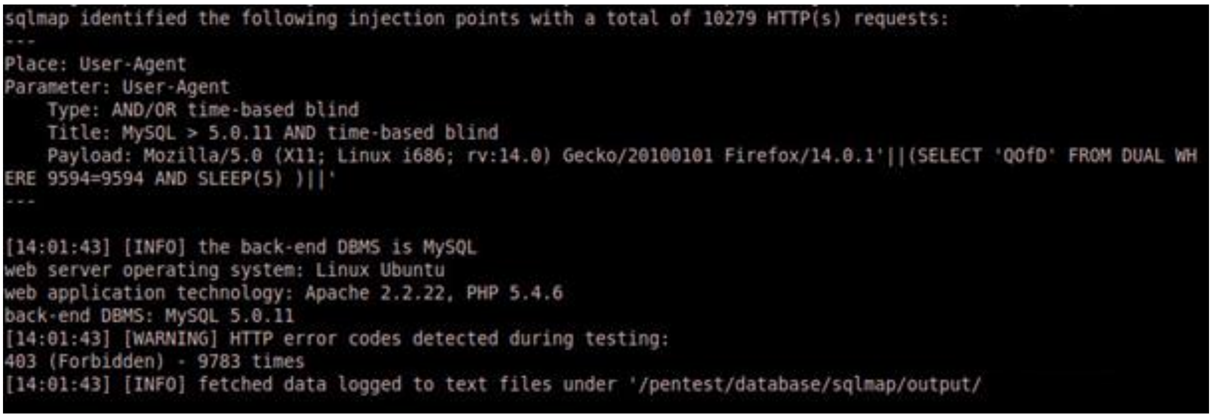
KUVIO 39. F5 ASM:ia kohti lähetetyt hyökkäyslausekkeet

Seuraavat skannaukset suoritetaan Sqlmap
-ohjelmistolla. Ohjelman käyttö ei vaadi käyttäjältä kuin tarvittavien parametrien annon hyökkäystä varten. Hyökkäykset suoritetaan seuraavasti sekä ModSecurityä että F5 ASM:ia kohti:

/sqlmap.py -u http://xxx.xxx.xxx.xxx/mutillidae/index.php?page=user-info.php –dbms mysql --data="username=&password=&user-info-php-submit-button= View+Account+Details" --threads 10 --user-agent="Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20100101 Firefox/14.0.1" --level 5 --risk 5 --cookie=" PHPSESSID=paeq4qalkf17sbqi53id73vse6"

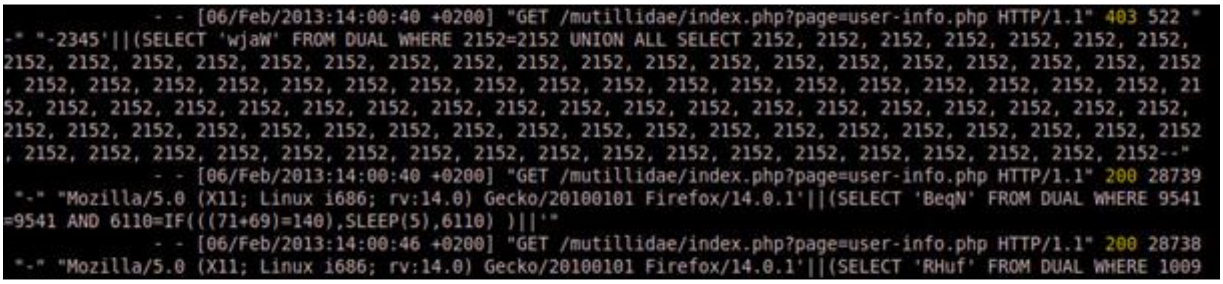
Ensimmäinen parametri on kohteena oleva sivu (-u). Tietokantatyyppi (--dbms) määritetään jo valmiiksi tässä tapauksessa, jotta Sqlmap ei käy läpi kaikkia sen tukemia injektiolausekkeita eri tietokannoille. Dataksi (--data) määritetään lomakkeen parametrit. Hyökkäyksen nopeuttamiseksi käytetään kymmentä säiettä (--threads). User-agent (--user-agent) määritetään näyttämään selaimelta, jotta WAF:ien tunnistetietokannat eivät estä Sqlmapia suoraan tämän perusteella. Tasolla (--level) kolme Sqlmap testaa myös evästeet ja user-agent-otsakkeen SQL-injektioiden varalta.

Sen ollessa täydet viisi, yritetään sql-injektioita löytää vielä suuremmilla kuormilla. Riskitaso (--risk) nostetaan myös mahdollisimman korkeaksi, jolloin SQL-injektioiden skaala on laajin, mutta myös vaarallisin. Injektiokohdan löytyessä tietokantaan saattaa tulla muutoksia, joka on ei-toivottua. Tämän vuoksi riskitasoa kannattaa käyttää harkiten. Evästeet (--cookies) lisätään tarkistukseen myös.



KUVIO 40. Blind SQL -injektio

Kuviossa 39 on haavoittuvuusskannauksen tulos, kun web-palvelinta suojaa ModSecurity. Tuhansien HTTP-pyyntöjen jälkeen, paikannettiin Blind SQL -injektio. Tämän seurauksena on mahdollista ladata palvelimella olevat tietokannat. Syynä tähän on, että ModSecurity ei oletuksena tutki user-agent-otsaketta. Hyökkäyslausekkeita saadaan kuitenkin estettyä hyvinkin paljon muiden sääntöjen toimesta, koska lähetetyssä datassa esiintyy muita poikkeavuuksia, joihin ModSecurity osaa reagoida. Kuviossa 40 on Apache-web-palvelimen lokia hyökkäyksestä.



KUVIO 41. Apache-web-palvelimen lokia hyökkäyksestä

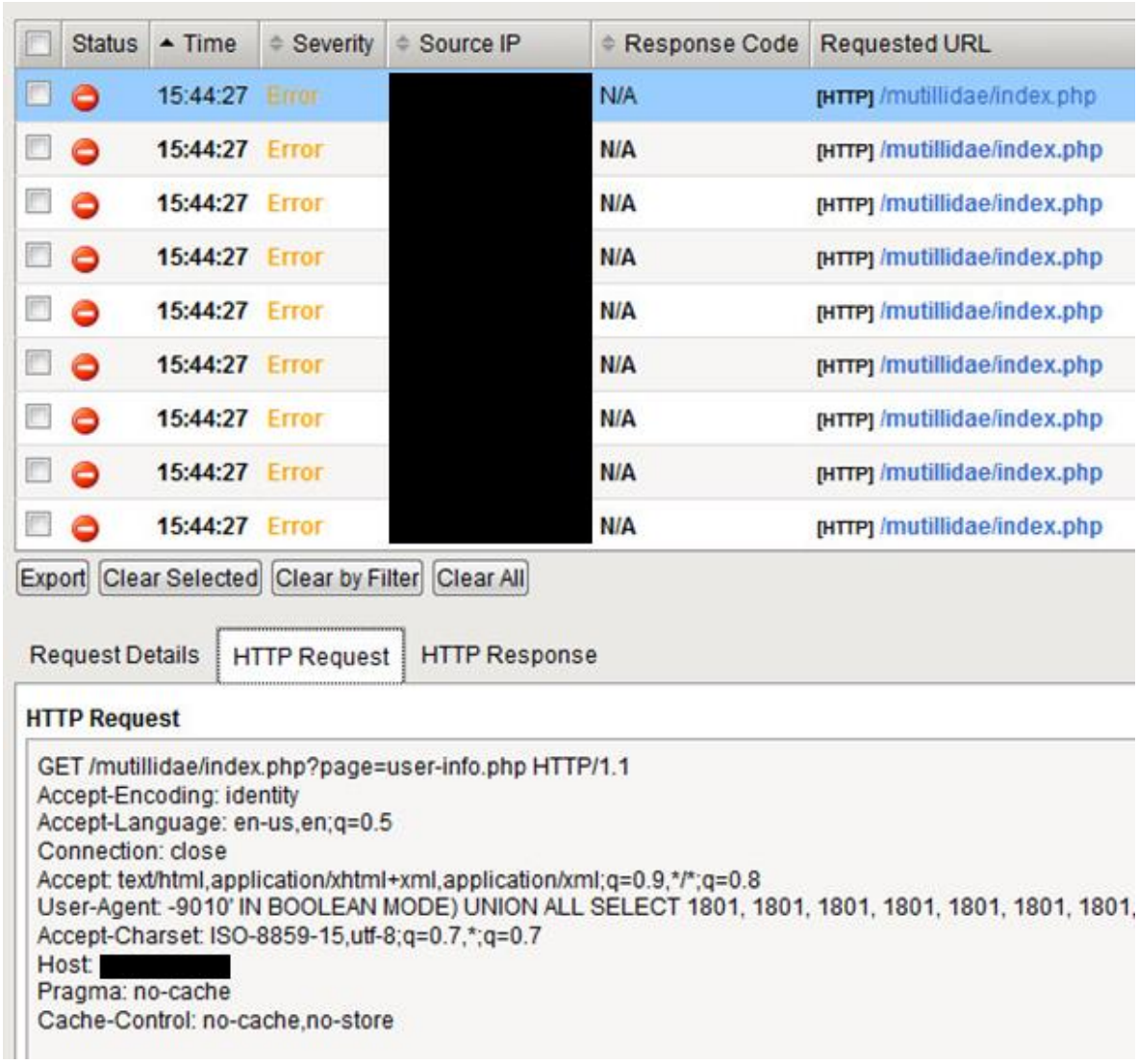
Kuviossa 41 on esitetty tulokset sen jälkeen, kun user-agent-otsakkeen tarkistus on laitettu päälle ModSecurityn asetuksista. User-agent-otsaketta ei kyetä hyödyntää SQL-injektiossa enää. Hyökkäysloki on luettavissa liitteessä 4.

```
[09:53:02] [WARNING] User-Agent parameter 'User-Agent' is not injectable
[09:53:02] [CRITICAL] all parameters appear to be not injectable. Also, you can try to
ring or a valid --regexp, refer to the user's manual for details
[09:53:02] [WARNING] HTTP error codes detected during testing:
403 (Forbidden) - 19334 times

[*] shutting down at 09:53:02
```

KUVIO 42. User-agent-otsakkeen tarkistus on päällä

Sqlmap ei onnistu löytämään SQL-injektiohaavoittuvuuksia, kun F5 ASM suojaa web-palvelinta. Kuviossa 41 on esitetty F5 ASM:n lokia tapahtumista. HTTP-pyyntöjä kohdistui F5 ASM:lle kymmeniä tuhansia. Sqlmap-ohjelman hyökkäysloki on luettavissa liitteessä 5.



KUVIO 43. F5 ASM:n lokia SQL-injektiohyökkäyksestä

10.3.3 Cross-Site Scripting (XSS)

XSS-hyökkäystä alustetaan käsin, lähettämällä erilaisia XSS-hyökkäyksessä käytettäviä merkkijonoja web-palvelimelle. Tarkoitus on katsoa, kuinka WAF:it niihin reagoivat. Aluksi lähdetään liikkeelle erikoismerkeistä, mitä XSS-hyökkäyksissä käytetään. Sitten jatketaan HTML-tageihin. Mukaan sotketaan myös heksajärjestelmää. Kuviossa 43 nähtävissä oleva pyyntö 11 kokeilee, miten WAF reagoi isojen ja pienien kirjaimien sekoitukseen samassa sanassa. Ja pyynnössä 12 nähdään, poistaako WAF kielletyn HTML-tagin merkkijonon keskeltä.

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	29527	baseline request
1	<	200	☐	☐	29527	
2	(	200	☐	☐	29527	
3	<img	200	☐	☐	29533	
4	<img src	403	☐	☐	503	
5	<script>	403	☐	☐	503	
6	alert(	403	☐	☐	503	
7	<body	200	☐	☐	29535	
8	<body onload	403	☐	☐	503	
9	<test>	200	☐	☐	29537	
10	%3cscript%3	403	☐	☐	503	
11	<ScRiPt>	403	☐	☐	503	
12	<scr<script>ipt>	403	☐	☐	503	

KUVIO 44. ModSecuritylle kohdistuvat XSS-hyökkäyslausekkeet

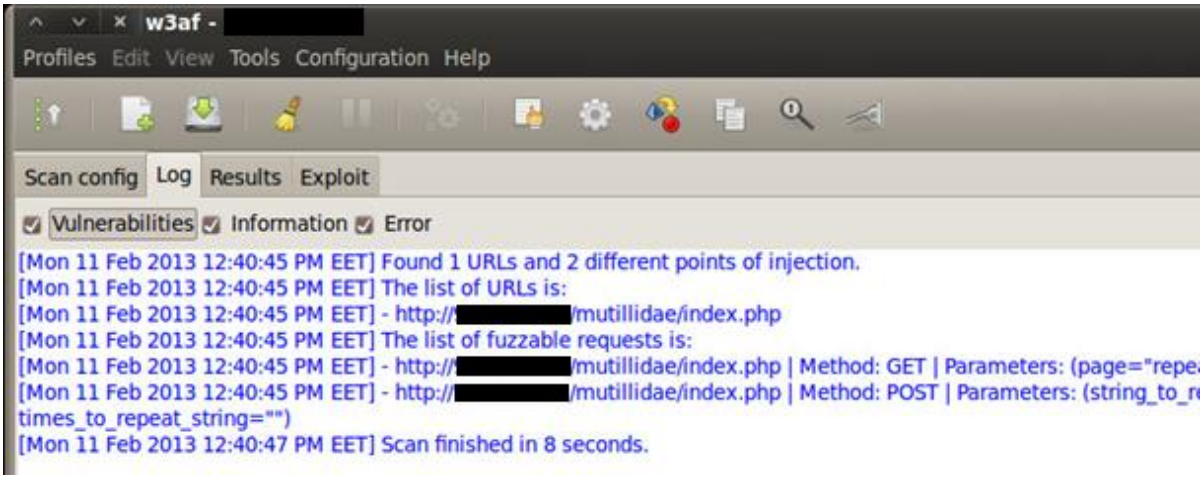
Kuvio 43 on samalla ModSecurityn tulos, jonka perusteella on havaittavissa, että erikoismerkkejä ei pelkästään estetä. Tässä tapauksessa tilakoodi 200 merkitsee ModSecurityn hyväksyntää lähetetylle datalle ja 403 estämistä. On nähtävissä, että kun kyseessä on HTML-tagi, se on suljettava tai sen on jatkuttava. Muuten ModSecurity reagoi siihen. Vaikka pyynnön yhdeksän (9) tagi on suljettu, ei siihen reagoida, koska sellaista ei ole HTML-kuvauskielessä käytössä (HTML Reference n.d). Myöskään sekoittaessa sanoja isoilla ja pienillä kirjaimilla, ei ole merkitystä havaitsemisen kannalta. Heksajärjestelmän käytön ModSecurity tunnistaa, kuten myös pyynnön 12 HTML-tagin, joka estetään kokonaan.

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	29933	baseline request
1	<	200	☐	☐	336	
2	(	200	☐	☐	336	
3	<img	200	☐	☐	336	
4	<img src	200	☐	☐	336	
5	<script>	200	☐	☐	336	
6	alert(	200	☐	☐	336	
7	<body	200	☐	☐	336	
8	<body onload	200	☐	☐	336	
9	<test>	200	☐	☐	336	
10	%3cscript%3	200	☐	☐	336	
11	<ScRiPt>	200	☐	☐	336	
12	<scr<script>ipt>	200	☐	☐	336	

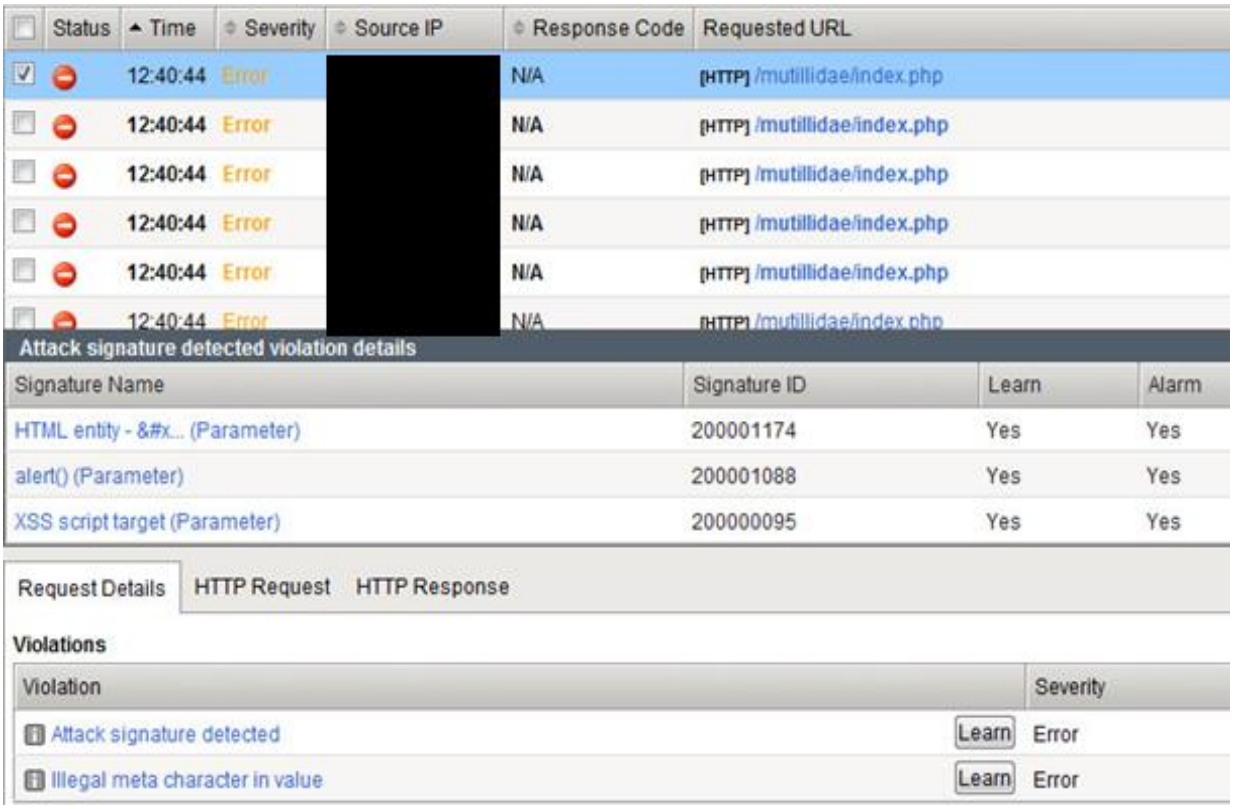
KUVIO 45. F5 ASM:lle kohdistuvat XSS-hyökkäyslausekkeet

F5 ASM:n tulos on esitetty kuviossa 44. Kaikki web-palvelinta kohti lähetetyt merkkijonot estetään. F5 ASM suodattaa kaikki lähetetyt pyynnöt suoraan jo lauseissa esiintyvien erikoismerkkien pohjalta.

Ajettaessa XSS-haavoittuvuusskannausta W3af-ohjelmistolla, kohdistetaan XSS-hyökkäys page-, times_to_repeat_string- ja string_to_repeat-parametreihin. Kuviossa 45 on esitetty W3af:n skannaustulos F5 ASM:n suojaamaa web-palvelinta kohti ja kuviossa 46 on F5 ASM:n lokia hyökkäyksistä. F5 ASM tunnistaa hyökkäyksen tunnistetietokannan pohjalta XSS-hyökkäykseksi, mutta suodattaisi niitä myös kiellettyjen erikoismerkkien vuoksi.

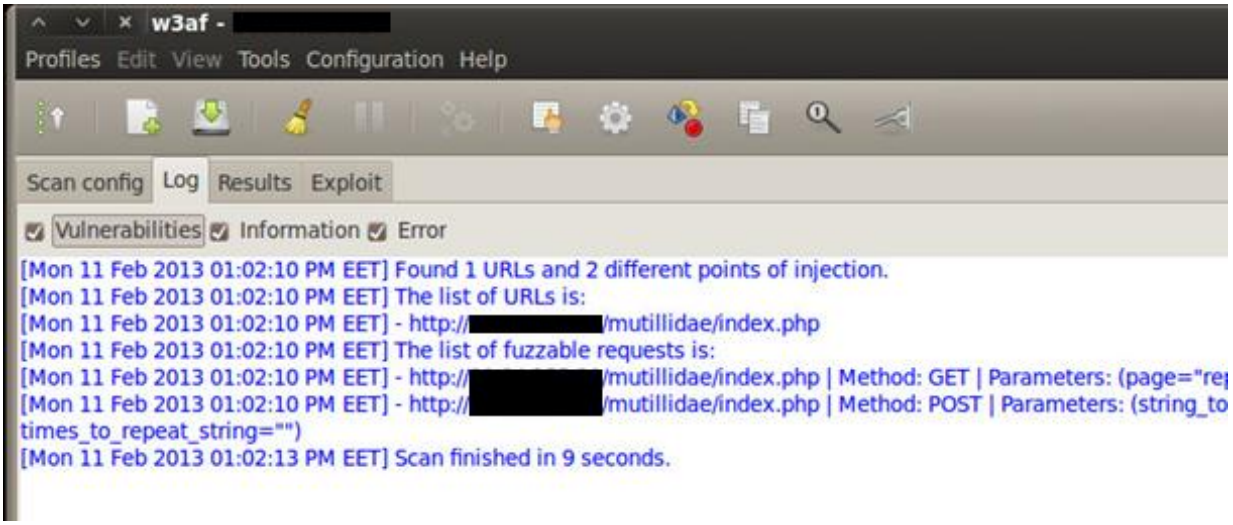


KUVIO 46. W3af ei löydä XSS-haavoittuvuuksia (F5 ASM)

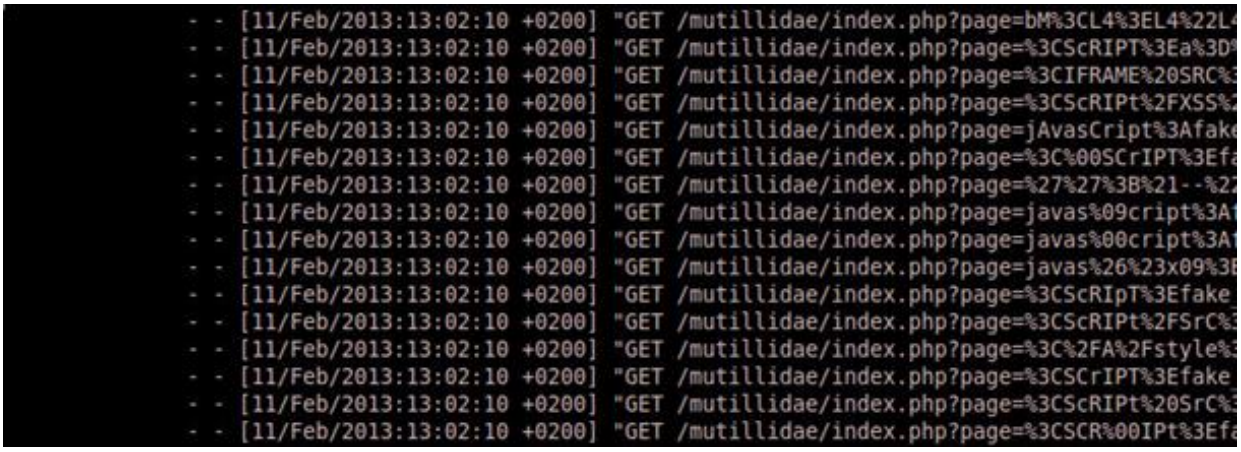


KUVIO 47. F5 ASM:n lokia XSS-hyökkäyksestä

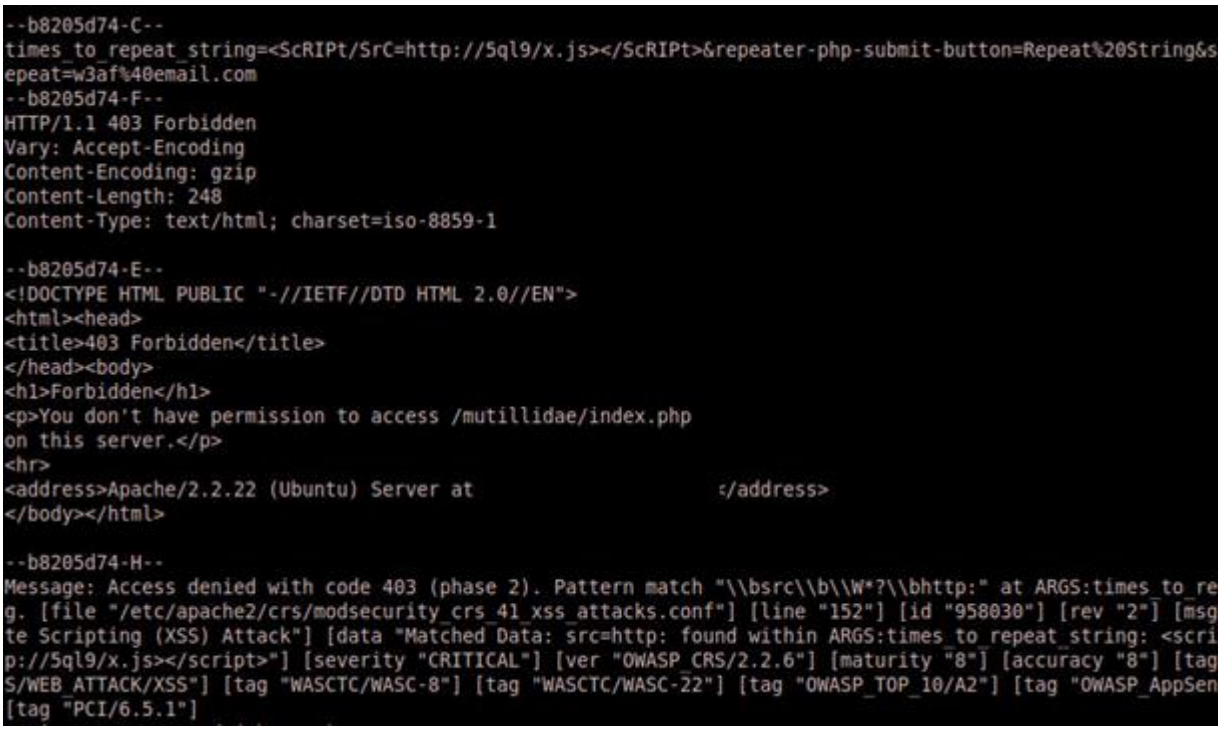
Kohdistettaessa hyökkäys ModSecurityn suojaamalle web-palvelimelle, ovat tulokset hyvin samankaltaiset, kuten kuviosta 47 on nähtävissä. Kuviossa 48 on Apache-web-palvelimen lokia, jolloin hyökkäys kohdistuu page-parametriin. Kuviossa 49 on Mod-Securityn audit-lokia, jossa hyökkäys on tunnistettu XSS-pohjaiseksi.



KUVIO 48. W3af ei löydä XSS-haavoittuvuuksia (ModSecurity)



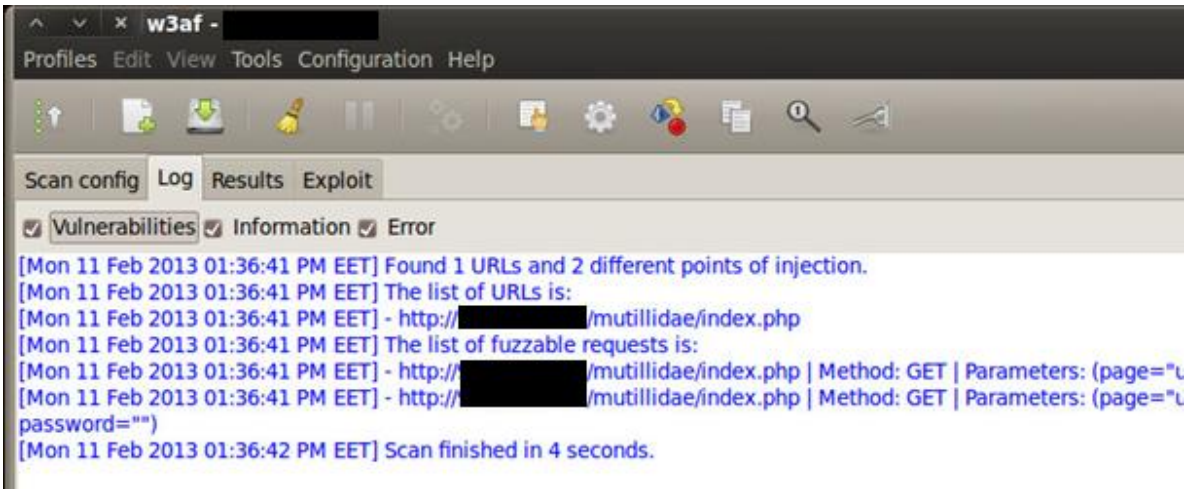
KUVIO 49. Apache-web-palvelimen lokia hyökkäyksestä



KUVIO 50. ModSecurityn lokia XSS-hyökkäyksestä

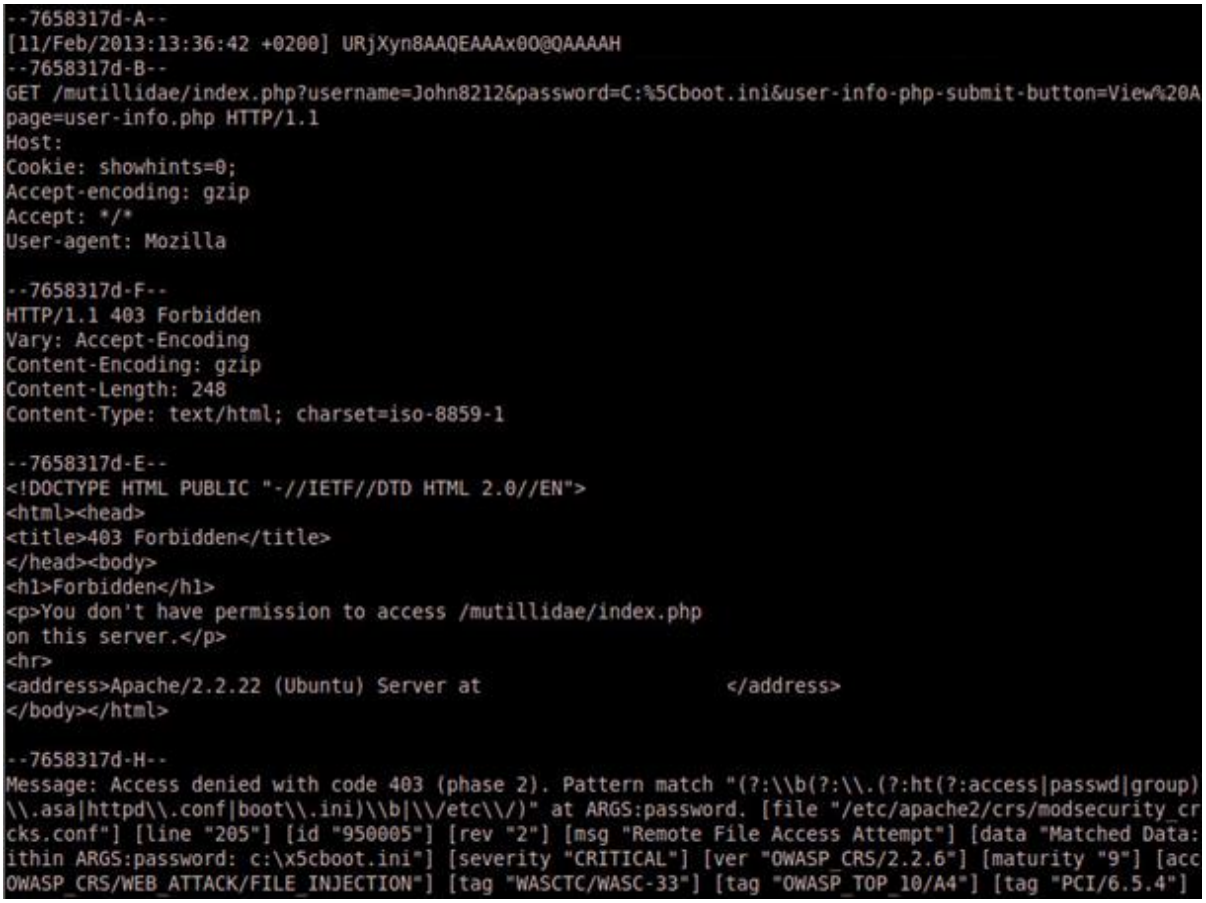
10.3.4 Local File Inclusion

Local File Inclusion –haavoittuvuutta koetetaan myös löytää käyttäen W3af-ohjelmaa. Skannaus kohdistuu username-, password-, ja page-parametreihin. W3af ei kykene löytämään LFI-haavoittuvuutta ModSecurityn suojaamalta web-palvelimelta, kuten kuvio 50 havainnollistaa. Kuviossa 51 on nähtävissä page-parametriin kohdistuvia hyökkäyksiä Apachen lokista ja kuviossa 52 ModSecurityn lokia estetystä hyökkäyksestä.



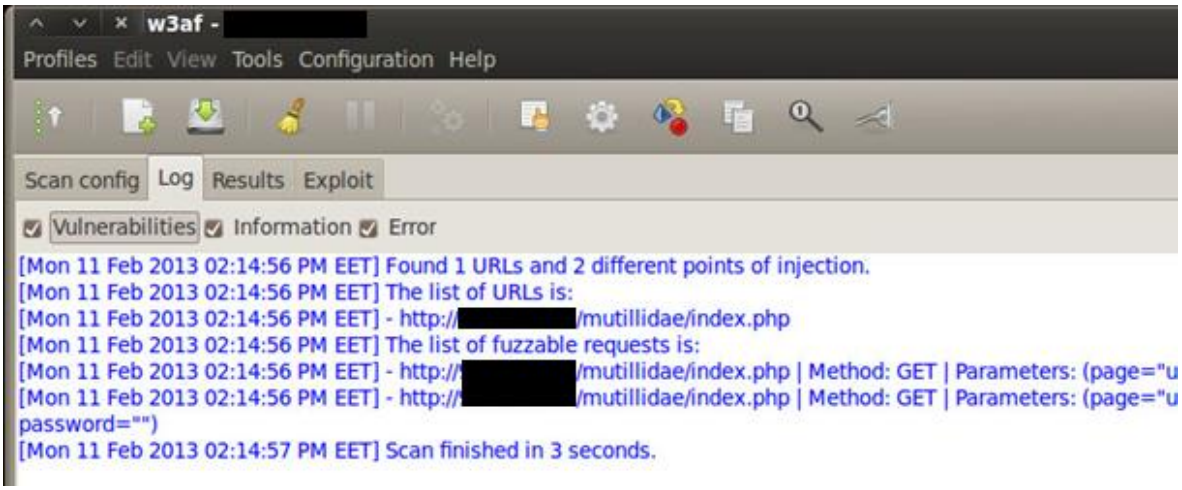
KUVIO 51. W3af ei löydä LFI-haavoittuvuuksia (ModSecurity)

KUVIO 52. Apache-web-palvelimen lokia LFI-hyökkäyksestä

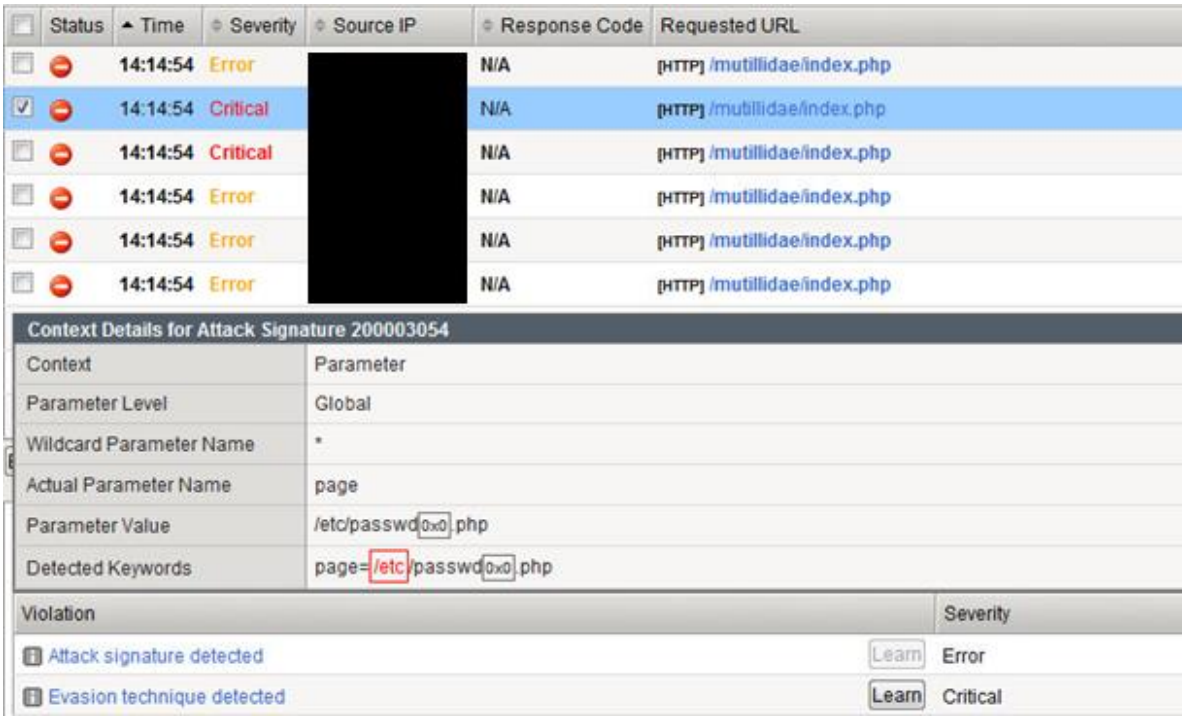


KUVIO 53. ModSecurityn lokia estetystä LFI-hyökkäyksestä

Tulokset ovat F5 ASM:n suojaaman web-palvelimen osalta hyvin samankaltaiset. Hyökkäys kohdistuu samoihin parametreihin ja tässäkin tapauksessa – tuloksetta. Kuviossa 53 on nähtävissä W3af:n tulokseton hyökkäys ja F5 ASM:n lokia hyökkäyksestä kuviossa 54.



KUVIO 54. W3af ei löydä LFI-haavoittuvuuksia (F5 ASM)



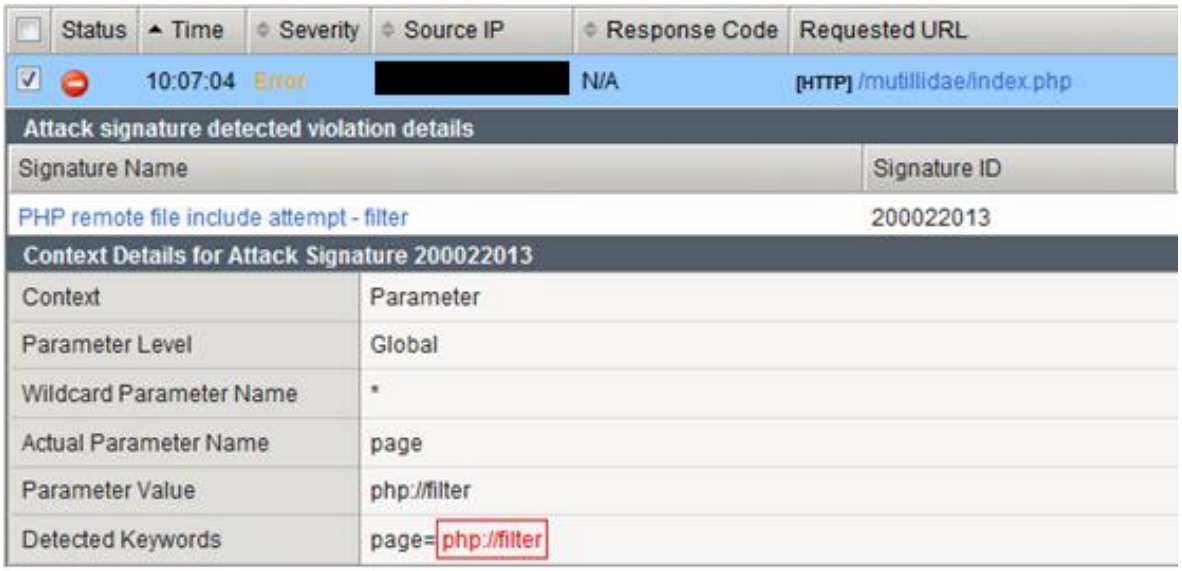
KUVIO 55. F5 ASM:n lokia LFI-hyökkäyksestä

W3af-ohjelman lähettämät hyökkäyslausekkeet olivat hyvin samankaltaisia. Tämän vuoksi testataan vielä hieman erilaista tekniikkaa hyökkäyksen toteuttamiseksi.

/mutillidae/index.php?page=php://filter/convert.base64-encode/resource=index.php

Yllä oleva lauseke hyödyntää PHP:stä löytyvää ”meta wrapper” –toimintoa. Sen avulla dataa voidaan käsitellä, ennen kuin se siirretään seuraavalle funktiolle. (PHP input/output streams 2009.) Haluttu web-sivu asetetaan resource-parametriin. Lausekkeen suorituksen jälkeen index.php-sivu muunnetaan base64-muotoon. Hyökkäys suoritetaan Burp Suite –ohjelmalla.

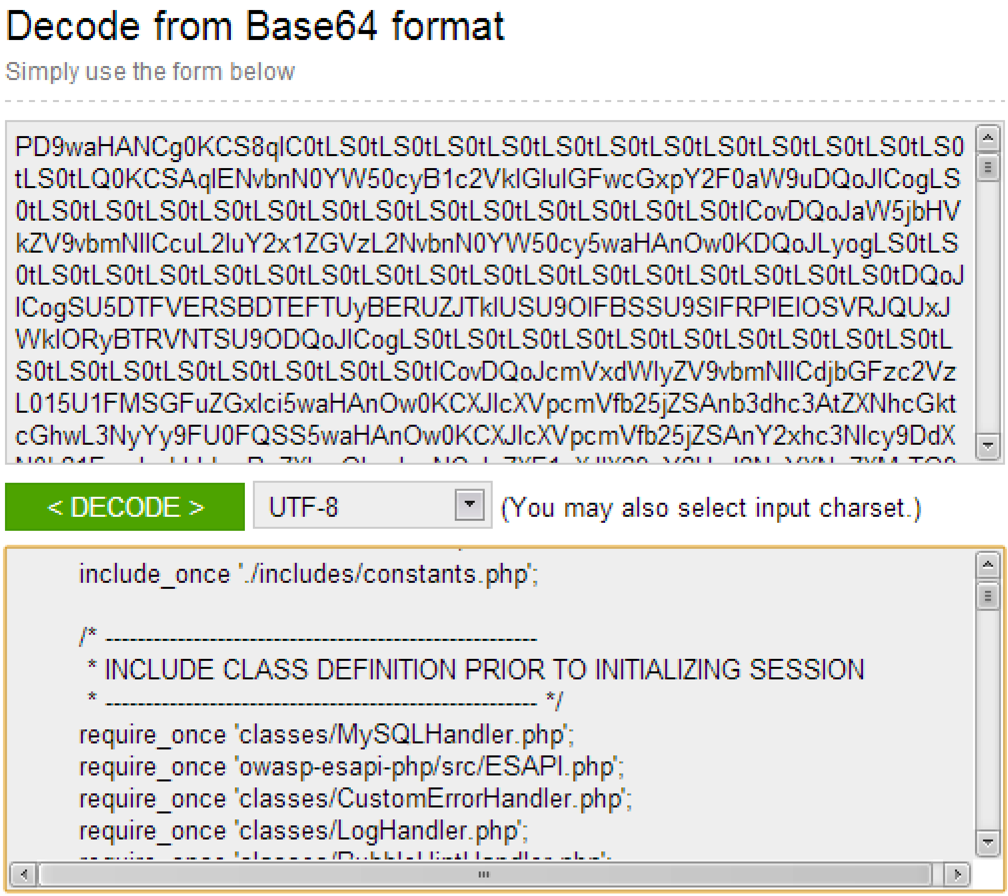
F5 ASM paikantaa hyökkäyksen suoraan tunnistetietokannan perusteella. F5 ASM:n lokia hyökkäyksestä on nähtävissä kuviossa 55.



KUVIO 56. F5 ASM estää muunnellun LFI-hyökkäyksen

ModSecurityn osalta tilanne on erilainen. Kuten kuviosta 56 on nähtävissä, lähetetty hyökkäyslauseke on hyväksytty ja vastauksena saadaan index.php-sivuston lähdekoodi base64-koodattuna. Kun aikaisemmin koodattu base64-teksti puretaan, saadaan halutun sivuston lähdekoodi selkokielisenä. Lähdekoodin sisältö saattaa paljastaa lisää haavoittuvuuksia web-sivustolla, jotka voivat johtaa koko palvelimen haltuunottoon. Lähdekoodi saattaa sisältää suoraan myös käyttäjätunnuksen ja salasanan tietokantaan. Kuviossa 57 on suoritettu base64-koodauksen purku index.php-tiedostolle.

KUVIO 57. ModSecurity ei tunnista LFI-hyökkäyslauseketta



KUVIO 58. Lähdekoodin purku base64-koodauksesta

ModSecurityn rules.conf-tiedostoon lisätään seuraava sääntö:

SecRule ARGS php:// "id:'3',phase:1,t:lowercase,log,deny,status:503"

Sääntö estää jatkossa “meta wrapper” –toiminnon käytön, eikä haavoittuvuuden hyödyntäminen enää onnistu. On mahdollista, että CRS-paketissa olevat experimental-säännöt olisivat estäneet hyökkäyksen. Kuviossa 58 on esitettynä tulos säännön lisäämisen jälkeen, josta on nähtävissä, että hyökkäys estetään.

KUVIO 59. ModSecurity tunnistaa mukautetun LFI-hyökkäyksen

--6fcbc924-A--
[13/Feb/2013:12:21:17 +0200] URtpHX8AAQEAACB9AkcAAAAC xxx.xxx.xxx.xxx 59368 xxx.xxx.xxx.xxx 80
--6fcbc924-B--
GET /mutillidae/index.php?page=php://filter/convert.base64-encode/resource=index.php HTTP/1.1
Host: xxx.xxx.xxx.xxx
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20100101 Firefox/14.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/*;q=0.8*
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Proxy-Connection: keep-alive
Cookie: showhints=0; PHPSESSID=p10o3nhd3hsvhmfnbv63lqfat0

--6fcbc924-H--
Message: Access denied with code 503 (phase 1). Pattern match "php://" at ARGS:page. [file "/etc/apache2/rules.conf"] [line "12"] [id "3"]
Action: Intercepted (phase 1)
Stopwatch: 1360750877874857 3756 (- - -)

Yllä oleva lokiteksti on tiivistetty ModSecurityn audit-lokista hyökkäyksen aikana. Siitäkin on selkeästi nähtävissä, että uudelleenlähetetty hyökkäyslauseke estetään. Käyttäjälle palautetaan vastauksena ennaltamäärätty, HTTP-tilakoodin 503 (Service Temporarily Unavailable) kertova web-sivu.

11 VERTAILU JA TULOKSET

11.1 Alustus

WAF-vertailuun valittiin kaksi tuotetta, jotka olivat ModSecurity sekä F5 ASM. Molempien tuotteiden kohdalla haluttiin tarkastella muun muassa käyttöönottoa, konfigurointia ja toimintaa.

Hyökkäyksiä varten valittiin kolme erilaista haavoittuvuutta: SQL-injektio, Cross-site Scripting ja Local file Inclusion. Valinnat perustuivat pääasiassa siihen, että ne ovat hyvin yleisiä ja käytettyjä haavoittuvuuksia, joita kohdistuu web-ohjelmistoihin internetissä.

Web-palvelimien oltua suojaamattomia, olivat hyökkäykset helppo kohdistaa haavoittuvuuksiin. SQL-injektiohaavoittuvuuden kohdalla tämä merkitsi tietokantojen vuotamista, joka voi johtaa myös muun tiedon menetykseen. XSS-haavoittuvuutta olisi voinut hyödyntää muun muassa evästevarkauksiin. LFI-haavoittuvuus paljasti, että kaikki tarkastukset eivät ole kunnossa tiedon käsittelyn osalta ja tätä virhettä olisi voinut hyödyntää esimerkiksi takaoven asennuksessa palvelimeen.

11.2 ModSecurity

ModSecurity edusti avoimen lähdekoodin tuotetta. Sen käyttöönotto osoittautui aluksi hieman haasteelliseksi, mutta selkeentyi siihen perehdyttäessä. ModSecurity ei tarjoa mitään turvaa ilman sääntöjä, jotka ovat saatavilla ilmaiseksi. Tarjolla on myös maksullinen sääntöpaketti, joka tarjoaa turvaa vielä yksityiskohtaisemmin ja tarkemmin. CRS-paketin päivitykseen on saatavilla Perl-skripti. Riippuen turvattavasta kohteesta ja sen toiminnoista, tulee ModSecurityn asetuksiin tehdä tarvittavat muutokset.

ModSecurityn dokumentointi aiheutti ongelmia. Internetistä löytyi tietoa, joka osoittautui jatkuvasti vanhaksi, kun käytänteet olivat muuttuneet. Ivan Risticin kirjoittama, ilmainen opaskirja kattoi jonkin verran toimintoja. Maksullinen versio kirjasta olisi tarjonnut yksityiskohtaisempaa tietoa. Ainoa ilmainen ja ajankohtainen tuki ohjelmalle löytyi postituslistoilta.

Lokitiedostot ovat hyvin kattavat ja suhteellisen helppolukuiset. ModSecurity ei tarjoa graafista käyttöliittymää lokitiedostoiden lukemiseen. Informaatioltaan lokit ovat hyvät, mutta niiden tulkitseminen ainakin tuotantoympäristössä vaatisi ehdottomasti muutakin kuin konsoli-ikkunan. Kolmannen osapuolen graafisia käyttöliittymiä on saatavilla lokien hallintaa varten.

Suorittaessa hyökkäystä ModSecurityn suojaamaan web-palvelinta kohti, onnistui ModSecurity estämään hyökkäykset, kunnes injektoitava kohta oli user-agent-otsake. Kyseisen otsakkeen kautta SQL-injektio onnistui ja tietovuoto oli mahdollinen. Pitkän selvittelyn jälkeen selvisi, että ModSecurity ei tarkastele user-agent-otsaketta oletuksena. Syynä on ilmeisesti false-positive-hälytysten vähentäminen. Kun user-agent-otsakkeen tarkistus laitettiin päälle, kyseinen SQL-injektiohaavoittuvuus saatiin paikattua. Muitakaan SQL-injektiohaavoittuvuuksia ei enää löytynyt. LFI-hyökkäyksien osalta haavoittuvuus paikannettiin myös, mutta saatiin paikattua. Kaikki XSS-hyökkäysyritykset ModSecurity sai estettyä.

ModSecurity vaatii pitkää perehtymistä, jotta sen käyttö on luontevaa ja varmaa. Asennustoimet vaativat ammattitaitoa, jotta virheitä ei synny heti alkuvaiheessa, eikä myöskään jatkossa. Sääntöjen muokkaus ja lisäys aiheuttivat päänvaivaa, kun tarkoitus on pitää säännöt tiukkoina false-positive-hälytysten hyväksymisen jälkeenkin.

11.3 F5 ASM

Testattavaksi saatu kaupallinen tuote oli F5 ASM. Lukuun ottamatta virtuaalialustalle tehtäviä määrittelyjä, kaikki tarvittavat konfiguraatiot oli mahdollista tehdä graafisen käyttöliittymän kautta, kun IP-osoite hallintaa varten oli määritelty käyttöön. Konfigurointi onnistuu myös SSH-yhteyden avulla.

Dokumentointi F5 ASM:n kohdalla oli oikein hyvä. Jokaiseen mietityttävään yksityiskohtaan löytyi ajan tasalla olevaa opastusta suoraan yleisillä hakukoneilla. Vaikka dokumentointi oli tasokasta, paljastui siitä asioihin perehdyttäessä haittapuoliakin. Tuntui, että dokumentointi oli hiukan levällään, eikä täysin keskitetty. Samasta asiasta saattoi löytyä useita eritasoisia dokumentteja, joissa kaikissa oli eri määrä aiheeseen liittyvää tietoa. Turhautuminen ohjeiden hajanaisuuteen oli havaittavissa internetin keskustelupalstoilla myös muiden käyttäjien toimesta. Kaupallisena tuotteena tuki tietenkin löytyy tuotteen valmistajalta, mutta myös aktiiviselta keskustelupalstalta.

Tuotteen käyttöönotto oli sulavaa. Tähän vaikutti suurelta osin hyvä dokumentointi ja web-käyttöliittymä. Kaikki säännöt ja toiminnot ovat asennuksessa mukana. Kun suojattavasta kohteesta aiheutuvat false-positive-hälytykset saatiin karsittua pois, aloitettiin testaus lähes välittömästi. Päivitykset koskien itse ohjelmistoa ja tunnistetietokantaa, voi määrittää latautumaan automaattisesti tai ladata ne itse F5:n websivuilta. F5 ASM oli tarkoitus asentaa toimimaan vain negatiivisen tietoturvamallin tavoin. Asetusvirheestä johtuen ASM toimi myös positiivisen tietoturvamallin tavoin ja tämä huomattiin vasta valitettavan myöhään. Sivulla 42 on kuvattu asetukset, jotka jäivät tekemättä.

F5 ASM onnistui suojaamaan web-palvelimen ilman, että haavoittuvuuksia havaittiin millään kolmesta eri hyökkäystyypistä.

11.4 Tulosten tarkastelu

Molemmat WAF:it osoittautuivat mielenkiintoisiksi tuotteiksi. Ne onnistuivat estämään web-palvelimille kohdistettuja hyökkäyksiä hyvin laajalti. On otettava kuitenkin huomioon hyökkäysten toteutus, joka ei ole kokonaisvaltainen tai millään lailla täydellinen WAF:ien testaukseen. Hyökkäyksiä oli muokattu jossain määrin WAF:ien ohittamiseksi, mutta ne eivät tuottaneet tulosta. Myöskään kaikkia eri toimintoja tuotteista ei testattu, kun hyökkäykset kohdistuivat vain tietyn tyyppisiin haavoittuvuuksiin.

Ilmaisen ja kaupallisen tuotteen erot kuitenkin olivat hyvin huomattavissa muun muassa käytettävyyden ja dokumentoinnin osalta. WAF vaatii aina jatkuvaa ylläpitoa, jotta se on ajan tasalla uusia, paikkaamattomia haavoittuvuuksia vastaan. Jatkuvan ylläpidon vuoksi graafinen käyttöliittymä olisi hyvin suotava olla käytettävyyttä ajatellen. ModSecurityn osalta tuotteen käyttöön perehtyminen vei huomattavasti enemmän aikaa ja käyttö vaati unix-tuntemusta jonkin verran.

Tuotteiden tehokkuus hyökkäysten estämisen suhteen oli hieman eritasoinen ja tämä johtui ModSecurityn osalta konfiguraatiosta, kun kovin selkeää kuvaa siitä ei saatu. ModSecurity vaatikin enemmän toimenpiteitä konfiguraation suhteen, jotta kaikki hyökkäykset saatiin estettyä. F5 ASM osoittautui helppokäyttöisemmäksi ollen myös paremmin dokumentoitu, sekä tehokkaampi perus asetuksillaan. Sen onnistui estää mukautetutkin hyökkäykset ilman, että erityisiä asetuksia olisi tarvinnut tehdä. Etua saatiin kuitenkin positiivisen tietoturvamallin hyödyntämisestä hyökkäyksiä havaittaessa. Valintakysymykseksi muodostuukin rahan lisäksi muun muassa erityisten ominaisuuksien tarve, suojattavat kohteet ja ylläpitokysymykset. Molemmille tuotteille löytyy omat käyttökohteensa aina tapauskohtaisesti, joissa toinen on toista parempi ratkaisu jollain lailla. Olkoon se sitten vaikka taloudellisesti tai teknisesti.

11.5 Ulkopuolista tutkimustietoa

Viimeisiä testejä WAF-tuotteille tehtiin 13. päivä helmikuuta. 20. päivä helmikuuta Zero Science Lab julkaisi oman raporttinsa kolmen eri WAF-tuotteen vertailusta. Zero Science Lab on makedonialainen tietoturvallisuuden tutkimus- ja kehittämislaboratorio (Zero Science Lab n.d.).

Tutkimuksen kohteena olivat ilmainen ModSecurity ja kaksi maksullista Cloud WAF –tuotetta: CloudFlare ja Incapsula. Tutkimuksen tarkoituksena oli ohittaa kaikkien WAF:ien suojaus keinolla millä hyvänsä. Käytetyt haavoittuvuudet olivat SQL-injektio, Cross Site Scripting (XSS), Local File Inclusion (LFI) ja Remote File Inclusion (RFI). Kohteina käytettiin muun muassa joitain web-ohjelmistoja ja niiden tunnettuja haavoittuvuuksia. (Cabrera, Krstic & Petrushevski 2013)

Tutkimuksen tulokset osoittivat, että ModSecurity vei voiton selkeästi. Vain kaksi (2) LFI/RFI-lauseketta läpäisi suojauksen. Tutkijat kuitenkin korostivat, että false-positive-hälytyksiä ei otettu huomioon, joka voi aiheuttaa jossain määrin ongelmia. Incapsulan suorituskykyä pidettiin hyvänä. Se kykeni estämään suurimman osan hyökkäyksistä. SQL-injektiolausekkeita jäi estämättä yksi (1), XSS-lausekkeita (3) ja LFI/RFI-lausekkeita neljä (4). Incapsula ei estänyt ollenkaan haitallisia XSS-lausekkeita, jotka sijoitettiin HTTP-kehyksen otsakkeisiin. CloudFlare pärjäsi tutkimuksessa huonoiten. Se ei kyennyt estämään yhtäkään testauksessa käytetyistä hyökkäyksistä. CloudFlarea mainostetaan kuitenkin WAF:ina. Tämä vaikuttaisi olevankin vain markkinointikikka. (Cabrera, Krstic & Petrushevski 2013)

Tutkimus osoitti hyvin WAF:ien heikkouksia. Oman työni pohjalta jäin kaipaamaan tutkimukselta yksityiskohtaisempaa tietoa SQL-injektiohyökkäyksestä. Varsinkin, kun sitä testattiin ModSecurityn kanssa. Tutkimuksen perusteella SQL-injektiolausekkeita ei sijoitettu HTTP-kehyksen otsakkeisiin lainkaan. Incapsulan kehittäjät ovat kommentoineet tutkimusta korjauksien kanssa (Incapsula Pentested - Results and Afterthoughts 2013). Myös ModSecuritylle on korjaukset suoritettu (Barnett 2013). CloudFlare ei ole kommentoinut tutkimusta.

12 YHTEENVETO

Web Application Firewall pyrkii estämään OSI-mallin 7. kerroksen kautta tapahtuvat hyökkäykset web-ohjelmistoja vastaan. Kuten hyökkäyksistä oli nähtävissä, WAF:in käyttöönotto vaatii aina huolellisen perehtymisen ja konfiguraation määrittämisen, tai web-palvelin on edelleen haavoittuvainen.

Hyökkäykset osoittivat, kuinka haavoittuvainen web-ohjelmisto voi johtaa web-palvelimella olevan tiedon vuotamiseen, web-palvelimen haltuunottoon tai web-sivuston käyttäjien tunnusvarkauksiin. Kun kyse on esimerkiksi yrityksen tai julkisen hallinnon palveluista, voi tietovuoto olla kriittinen. Ja se ei välttämättä vaadi kuin yhden haavoittuvuuden. Vaikka ohjelmoijat saisivat kattavampaa koulutusta tietoturvan osalta web-ohjelmoinnissa, ei se ole riittävä toimenpide haavoittuvuuksien kitkemiseksi jo ohjelmointivaiheessa. Sisältöä kehittäessä ja päivittäessä virheitä tapahtuu jatkuvasti. Sen yhden ja ainoan, ehkä hyvinkin yksinkertaisen, lähdekoodissa olevan haavoittuvuuden hyväksikäytön WAF saattaisi estää.

Voi olla, että yrityksissä ei ole tietoturvaa mietitty kokonaisuutena ja tietoturvan ajan tasalla oleminen huomioidaan vasta, kun ikäviä asioita on jo sattunut. Sanaan ”palomuuri” luotetaan edelleen paljon, oli sitten kyseessä mikä kyber-uhka tahansa. Ylläpitämällä tietoturvaa yrityksessä jatkuvasti ja ymmärtämällä uhkatekijät, on mahdollista turvata liiketoimintaa hyvin pitkälle. Yrityksen maine ei kärsisi WAF:in hankkimisesta toisin kuin tietovuodon tapahtuessa.

Uskon, että tulevaisuudessa WAF:ien suodatustapa muuttuu pääsääntöisesti positiivisen tietoturvamallin mukaiseksi, jolloin vain ei-haitalliseksi tunnistettu liikenne läpäisee suodatuksen. Ohessa hyödynnettäisiin negatiivista tietoturvamalliakin. Positiivinen suodatusmalli ehkä tuo omat ongelmansa mukanaan, mutta takaa turvallisemman liikennesisällön. Oli suodatusmalli sitten mikä tahansa, ei WAF voi ikinä tarjota kuitenkaan varmaa suojaa.

LÄHTEET

About F5. n.d. Viitattu 02.10.2012.
http://www.f5.com/about/history/

Apache httpd 2.2 vulnerabilities. Viitattu 22.09.2012.
http://httpd.apache.org/security/vulnerabilities_22.html

Barnett, R. 2013. CloudFlare vs Incapsula vs ModSecurity. Viitattu 17.03.2013.
http://permalink.gmane.org/gmane.comp.apache.mod-security.user/10035

Beechey, J. 2009. Web Application Firewalls: Defense in Depth for Your Web Infrastructure. Viitattu 07.11.2012. http://www.sans.edu/student-files/projects/200904_01.doc

BIG-IP Application Security Manager – DATASHEET. 2013. Viitattu 13.02.2013.
http://www.f5.com/pdf/products/big-ip-application-security-manager-ds.pdf

Burp Suite. n.d. Ohjelmiston verkkosivut. Viitattu 16.02.2013.
http://www.portswigger.net/burp/

Cabrera, H., Krstic, G. & Petrushevski, S. 2013. CloudFlare vs Incapsula vs ModSecurity. Viitattu 17.03.2013. http://zeroscience.mk/files/wafreport2013.pdf

Cert-fi. 2010. Tietoturvakatsaus 1/2010. Viitattu 02.10.2012.
http://www.cert.fi/katsaukset/2010/tietoturvakatsaus_1_2010.html

Forms in HTML documents. n.d. Viitattu 11.03.2013.
http://www.w3.org/TR/html401/interact/forms.html

Hock-Chuan, C. 2009. HTTP (HyperText Transfer Protocol). Viitattu 28.09.2012.
http://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/http_basics.html

HTML Reference. n.d. Viitattu 12.02.2013.
http://www.w3schools.com/tags/default.asp

HTTP Caching. n.d. Viitattu 28.09.2012.
http://www.httpwatch.com/httpgallery/caching/

Hypertext Transfer Protocol – HTTP/1.1 – Method Definitions. 1999. Viitattu 16.02.2013. http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html

Hypertext Transfer Protocol – HTTP/1.1 – Overall Operation. 1999.Viitattu 28.09.2012. www.w3.org/Protocols/rfc2616/rfc2616-sec1.html

Hypertext Transfer Protocol – HTTP/1.1 – Request. 1999.Viitattu 28.09.2012. http://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html

Hypertext Transfer Protocol – HTTP/1.1 – Response. 1999.Viitattu 28.09.2012. http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html

Imperva – Hacker Intelligence Initiative, Monthly Trend Report #1. 2011. Viitattu 27.09.2012. http://www.imperva.com/docs/HI_Remote_File_Inclusion.pdf

Imperva Releases Cloud-based Web Application Firewall Service for Mid-Sized Businesses. 2011. Viitattu 12.02.2013. http://www.securityweek.com/imperva-releases-cloud-based-web-application-firewall-service-mid-sized-businesses

Imperva – Remote and Local File Inclusion Vulnerabilities 101. 2012. Viitattu 27.09.2012. http://www.imperva.com/docs/HII_Remote_and_Local_File_Inclusion_Vulnerabilities.pdf

Imperva – Web Application Attack Report Edition #2. 2012. Viitattu 27.09.2012. http://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed2.pdf

Imperva – SQL Injection. n.d. SQL Injection. Viitattu 27.09.2012. http://www.imperva.com/resources/glossary/sql_injection.html

Incapsula Pentested - Results and Afterthoughts. 2013. Viitattu 17.03.2013. http://www.incapsula.com/the-incapsula-blog/item/699-incapsula-pentested-review

JYVSECTEC. 2012. JYVSECTEC-hankkeen kotisivut. Viitattu 24.9.2012. http://jyvsectec.fi/

McMillan, J. 2009. Intrusion Detection FAQ: What is the difference between an IPS and a Web Application Firewall. Viitattu 25.09.2012. http://www.sans.org/security-resources/idfaq/ips-web-app-firewall.php

ModSecurity. n.d. Overview. Viitattu 25.09.2012. http://modsecurity.org/projects/modsecurity/

ModSecurity Rules and Support Services. n.d. Viitattu 22.09.2012. https://www.trustwave.com/modsecurity-rules-support.php

Murphy, A. & Salchow, K. 2007. Applied Application Security—Positive & Negative Efficiency. Viitattu 13.03.2013. http://www.f5.com/pdf/white-papers/applied-app-security-wp.pdf

Mutillidae. n.d. Ohjelmiston verkkosivut. Viitattu 1.2.2012. http://www.irongeek.com/i.php?page=mutillidae/mutillidae-deliberately-vulnerable-php-owasp-top-10

OWASP - About The Open Web Application Security Project. 2012. Viitattu 25.09.2012. https://www.owasp.org/index.php/About_OWASP

OWASP – Brute force attack. 2009. Viitattu 12.03.2013. https://www.owasp.org/index.php/Brute_force_attack

OWASP - Cross-site Scripting (XSS). 2011. Viitattu 11.02.2013. https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

OWASP ModSecurity Core Rule Set Project. 2012. Viitattu 22.09.2012. https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

OWASP – OWASP Top 10. 2012. Viitattu 25.09.2012. https://www.owasp.org/index.php/Top_10_2010

OWASP - Web Application Firewall. 2012. Viitattu 25.09.2012. https://www.owasp.org/index.php/Web_Application_Firewall

Php.net. PHP input/output streams. 2009. Viitattu 12.02.2013. http://php.net/manual/bg/wrappers.php.php

Ristic, I. 2011. IronBee, a new Apache-licensed web application firewall. Viitattu 28.09.2012. http://blog.ironbee.com/2011/02/ironbee-a-new-apache-licensed-web-application-firewall.html

Ristic, I. 2012. ModSecurity Handbook. The Complete Guide to the Popular Open Source Web Application Firewall. Viitattu 22.09.2012. https://www.feistyduck.com/books/modsecurity-handbook/modsecurity-handbook-getting-started-may-2012.pdf

Nortio, J. 2012. Salainen Selvitys: Intialainen koodaus on suomalaista kalliimpaa. Viitattu 15.03.2013. http://www.3t.fi/artikkeli/uutiset/talous/salainen_selvitys_intialainen_koodaus_on_suomalaista_kalliimpaa

Sqlmap.org. n.d. Ohjelmiston verkkosivut. Viitattu 16.02.2013. http://sqlmap.org/

W3af.org. n.d. Ohjelmiston verkkosivut. Viitattu 11.02.2013. http://W3af.org

Zero Science Lab. n.d. Viitattu 16.03.2013. http://zeroscience.mk/en/about

LIITTEET

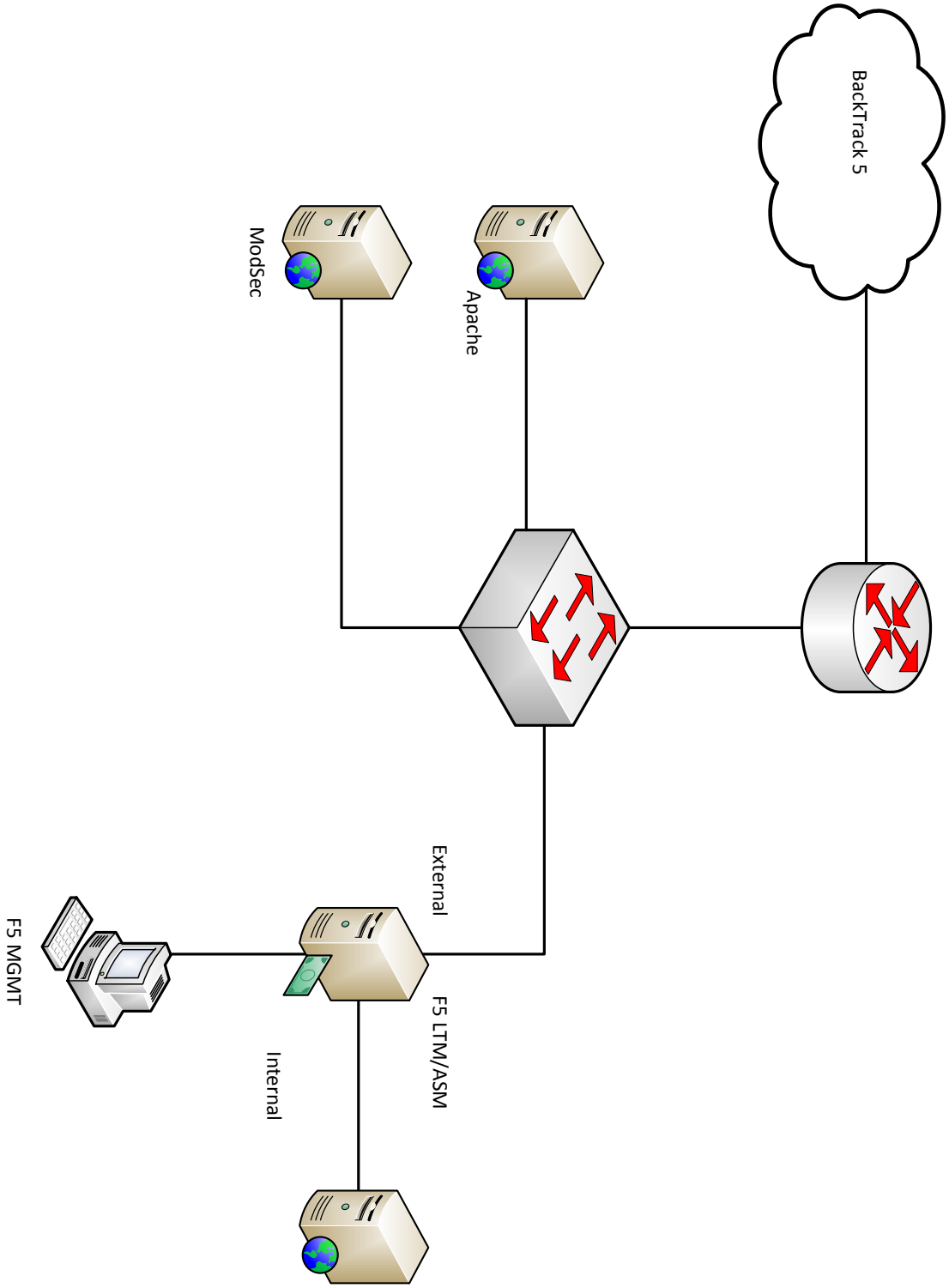
Liite 1 OWASP Top 10

A1 – Injection	Injection flaws, such as SQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the into executing unintended commands or accessing unauthorized data interpreter into executing unintended commands or accessing unauthorized data.
A2 – Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation and escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A3 – Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities.
A4 – Insecure Direct Object References	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5 – Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
A6 – Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date, including all code libraries used by the application.
A7 – Insecure Cryptographic Storage	Many web applications do not properly protect sensitive data, such as credit cards, SSNs, and authentication credentials, with appropriate encryption or hashing. Attackers may steal or modify such weakly protected data to conduct identity theft, credit card fraud, or other crimes.
A8 - Failure to Restrict URL Access	Many web applications check URL access rights before rendering protected links and buttons. However, applications need to perform similar access control checks each time these pages are accessed, or attackers will be able to forge URLs to access these hidden pages anyway.
A9 - Insufficient Transport Layer Protection	Applications frequently fail to authenticate, encrypt, and protect the confidentiality and integrity of sensitive network traffic. When they do, they sometimes support weak algorithms, use expired or invalid certificates, or do not use them correctly.
A10 – Unvalidated Redirects and Forwards	Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Liite 2 OWASP Top 10 –muutokset

OWASP Top 10 – 2007 (Previous)	OWASP Top 10 – 2010 (New)
A2 – Injection Flaws	A1 – Injection
A1 – Cross Site Scripting (XSS)	A2 – Cross-Site Scripting (XSS)
A7 – Broken Authentication and Session Management	A3 – Broken Authentication and Session Management
A4 – Insecure Direct Object Reference	A4 – Insecure Direct Object References
A5 – Cross Site Request Forgery (CSRF)	A5 – Cross-Site Request Forgery (CSRF)
<was T10 2004 A10 – Insecure Configuration Management>	A6 – Security Misconfiguration (NEW)
A8 – Insecure Cryptographic Storage	A7 – Insecure Cryptographic Storage
A10 – Failure to Restrict URL Access	A8 – Failure to Restrict URL Access
A9 – Insecure Communications	A9 – Insufficient Transport Layer Protection
<not in T10 2007>	A10 – Unvalidated Redirects and Forwards (NEW)
A3 – Malicious File Execution	<dropped from T10 2010>
A6 – Information Leakage and Improper Error Handling	<dropped from T10 2010>

Liite 3 Verkon topologia



Liite 4 Sqlmap-hyökkäysloki ModSecurity

root@bt:/pentest/database/sqlmap# ./sqlmap.py -u http://xxx.xxx.xxx.xxx/mutillidae/index.php?page=user-info.php --dbms mysql --data="username=&password=&user-info-php-submit-button=View+Account+Details" --threads 10 --user-agent="Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20100101 Firefox/14.0.1" --level 5 --risk 5 --cookie="PHPSESSID=paeq4qalkf17sbqi53id73vse6"

sqlmap/1.0-dev-25eca9d - automatic SQL injection and database takeover tool
http://sqlmap.org

[!] legal disclaimer: usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Authors assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 09:35:37

[09:35:38] [INFO] testing connection to the target url
[09:35:39] [INFO] heuristics detected web page charset 'ascii'
[09:35:39] [INFO] testing if the url is stable, wait a few seconds
[09:35:40] [INFO] url is stable
[09:35:40] [INFO] testing if POST parameter 'username' is dynamic
[09:35:41] [WARNING] POST parameter 'username' appears to be not dynamic
[09:35:41] [WARNING] reflective value(s) found and filtering out
[09:35:41] [WARNING] heuristic test shows that POST parameter 'username' might not be injectable
[09:35:41] [INFO] testing for SQL injection on POST parameter 'username'
[09:35:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:35:42] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:35:44] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:35:46] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[09:35:54] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:36:01] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:36:08] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[09:36:10] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[09:36:10] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'

[09:36:10] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[09:36:10] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[09:36:10] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[09:36:10] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[09:36:10] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[09:36:11] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[09:36:11] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:36:11] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:36:11] [INFO] testing 'MySQL stacked conditional-error blind queries'
[09:36:13] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[09:36:15] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:36:16] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[09:36:17] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[09:36:17] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[09:36:23] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:36:24] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[09:36:25] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[09:36:30] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[09:36:36] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[09:36:36] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[09:36:36] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[09:36:36] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[09:36:36] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[09:36:36] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[09:36:36] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[09:36:37] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[09:36:38] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[09:36:39] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[09:36:39] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[09:36:40] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'

[09:36:41] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[09:36:47] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[09:36:53] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[09:37:07] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[09:37:19] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[09:37:29] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[09:37:40] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[09:37:49] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[09:37:58] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[09:38:11] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[09:38:24] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[09:38:33] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[09:38:44] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[09:38:55] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[09:39:10] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[09:39:23] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[09:39:35] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[09:39:45] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[09:39:54] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[09:40:06] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[09:40:19] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[09:40:31] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[09:40:44] [WARNING] POST parameter 'username' is not injectable
[09:40:44] [INFO] testing if POST parameter 'password' is dynamic
[09:40:45] [WARNING] POST parameter 'password' appears to be not dynamic
[09:40:45] [WARNING] heuristic test shows that POST parameter 'password' might not be injectable
[09:40:45] [INFO] testing for SQL injection on POST parameter 'password'
[09:40:45] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:40:47] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:40:48] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:40:51] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[09:40:58] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:41:06] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:41:13] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[09:41:15] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[09:41:15] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'

[09:41:15] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[09:41:15] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[09:41:15] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[09:41:15] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[09:41:15] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[09:41:15] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[09:41:15] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:41:15] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:41:15] [INFO] testing 'MySQL stacked conditional-error blind queries'
[09:41:17] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[09:41:18] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:41:19] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[09:41:19] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[09:41:20] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[09:41:26] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:41:27] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[09:41:28] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[09:41:33] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[09:41:39] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[09:41:39] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[09:41:39] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[09:41:39] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[09:41:39] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[09:41:39] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[09:41:39] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[09:41:40] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[09:41:40] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[09:41:41] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[09:41:42] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[09:41:43] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'

[09:41:43] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[09:41:49] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[09:41:55] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[09:42:08] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[09:42:23] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[09:42:33] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[09:42:44] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[09:42:54] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[09:43:05] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[09:43:17] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[09:43:27] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[09:43:37] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[09:43:47] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[09:43:57] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[09:44:09] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[09:44:20] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[09:44:30] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[09:44:41] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[09:44:50] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[09:44:59] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[09:45:13] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[09:45:27] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[09:45:46] [WARNING] POST parameter 'password' is not injectable
[09:45:46] [INFO] testing if POST parameter 'user-info-php-submit-button' is dynamic
[09:45:47] [WARNING] POST parameter 'user-info-php-submit-button' appears to be not dynamic
[09:45:47] [WARNING] heuristic test shows that POST parameter 'user-info-php-submit-button' might not be injectable
[09:45:47] [INFO] testing for SQL injection on POST parameter 'user-info-php-submit-button'
[09:45:47] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:45:49] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:45:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:45:54] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[09:46:02] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:46:09] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:46:17] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[09:46:19] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'

[09:46:19] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[09:46:19] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[09:46:19] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[09:46:19] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[09:46:19] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[09:46:20] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[09:46:20] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[09:46:20] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:46:20] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:46:20] [INFO] testing 'MySQL stacked conditional-error blind queries'
[09:46:22] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[09:46:22] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:46:23] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[09:46:24] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[09:46:25] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[09:46:31] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:46:32] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[09:46:33] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[09:46:39] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[09:46:45] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[09:46:45] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[09:46:45] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[09:46:45] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[09:46:45] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[09:46:45] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[09:46:45] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[09:46:46] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[09:46:46] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[09:46:47] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[09:46:48] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'

[09:46:49] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[09:46:50] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[09:46:57] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[09:47:04] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[09:47:17] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[09:47:29] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[09:47:40] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[09:47:50] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[09:48:00] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[09:48:14] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[09:48:27] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[09:48:38] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[09:48:48] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[09:48:58] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[09:49:09] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[09:49:19] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[09:49:29] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[09:49:38] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[09:49:47] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[09:49:57] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[09:50:09] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[09:50:24] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[09:50:36] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[09:50:47] [WARNING] POST parameter 'user-info-php-submit-button' is not injectable
[09:50:48] [INFO] testing if GET parameter 'page' is dynamic
[09:50:48] [INFO] confirming that GET parameter 'page' is dynamic
[09:50:48] [INFO] GET parameter 'page' is dynamic
[09:50:48] [WARNING] heuristic test shows that GET parameter 'page' might not be injectable
[09:50:48] [INFO] testing for SQL injection on GET parameter 'page'
[09:50:48] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:50:50] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:50:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:50:53] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[09:50:58] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:51:06] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:51:15] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'

[09:51:17] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[09:51:17] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[09:51:17] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[09:51:17] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[09:51:17] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[09:51:17] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[09:51:17] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[09:51:17] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[09:51:18] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:51:18] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:51:18] [INFO] testing 'MySQL stacked conditional-error blind queries'
[09:51:20] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[09:51:20] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:51:21] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[09:51:22] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[09:51:23] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[09:51:28] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:51:29] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[09:51:30] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[09:51:35] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[09:51:40] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[09:51:40] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[09:51:40] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[09:51:40] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[09:51:40] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[09:51:40] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[09:51:40] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[09:51:40] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[09:51:41] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'

[09:51:42] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[09:51:43] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[09:51:43] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[09:51:44] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[09:51:50] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[09:51:56] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[09:52:07] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[09:52:18] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[09:52:28] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[09:52:36] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[09:52:45] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[09:52:55] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[09:53:05] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[09:53:16] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[09:53:26] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[09:53:36] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[09:53:47] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[09:53:57] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[09:54:09] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[09:54:25] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[09:54:35] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[09:54:45] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[09:54:54] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[09:55:06] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[09:55:20] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[09:55:35] [WARNING] GET parameter 'page' is not injectable
[09:55:35] [INFO] testing if User-Agent parameter 'User-Agent' is dynamic
[09:55:35] [INFO] confirming that User-Agent parameter 'User-Agent' is dynamic
[09:55:35] [WARNING] User-Agent parameter 'User-Agent' appears to be not dynamic
[09:55:35] [WARNING] heuristic test shows that User-Agent parameter 'User-Agent' might not be injectable
[09:55:35] [INFO] testing for SQL injection on User-Agent parameter 'User-Agent'
[09:55:35] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:55:39] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:55:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[09:55:43] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[09:55:51] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[09:55:59] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'

[09:56:06] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[09:56:08] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[09:56:08] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[09:56:08] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[09:56:08] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[09:56:08] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[09:56:08] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[09:56:09] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[09:56:09] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[09:56:09] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:56:09] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[09:56:09] [INFO] testing 'MySQL stacked conditional-error blind queries'
[09:56:11] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[09:56:12] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:56:12] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[09:56:14] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[09:56:15] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[09:56:24] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[09:56:25] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[09:56:26] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[09:56:32] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[09:56:38] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[09:56:38] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[09:56:38] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[09:56:38] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[09:56:38] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[09:56:38] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[09:56:38] [INFO] testing 'MySQL > 5.0.11 stacked queries'

[09:56:39] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[09:56:40] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[09:56:41] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[09:56:42] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[09:56:43] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[09:56:45] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[09:56:53] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[09:57:00] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[09:57:16] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[09:57:30] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[09:57:42] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[09:57:55] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[09:58:08] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[09:58:21] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[09:58:33] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[09:58:43] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[09:58:53] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[09:59:06] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[09:59:21] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[09:59:33] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[09:59:43] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[09:59:53] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[10:00:06] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[10:00:22] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[10:00:39] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[10:00:56] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[10:01:12] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[10:01:24] [WARNING] User-Agent parameter 'User-Agent' is not injectable
[10:01:24] [INFO] testing if Cookie parameter 'PHPSESSID' is dynamic
[10:01:25] [WARNING] Cookie parameter 'PHPSESSID' appears to be not dynamic
[10:01:25] [WARNING] heuristic test shows that Cookie parameter 'PHPSESSID' might not be injectable
[10:01:25] [INFO] testing for SQL injection on Cookie parameter 'PHPSESSID'
[10:01:25] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:01:26] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[10:01:28] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[10:01:30] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[10:01:36] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[10:01:42] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'

[10:01:49] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[10:01:51] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[10:01:51] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[10:01:51] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[10:01:51] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[10:01:51] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[10:01:51] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[10:01:51] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[10:01:51] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[10:01:51] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[10:01:51] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[10:01:51] [INFO] testing 'MySQL stacked conditional-error blind queries'
[10:01:53] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[10:01:54] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[10:01:54] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[10:01:55] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[10:01:56] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[10:02:02] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[10:02:03] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[10:02:04] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[10:02:11] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[10:02:17] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[10:02:17] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[10:02:17] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[10:02:17] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[10:02:17] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[10:02:17] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[10:02:17] [INFO] testing 'MySQL > 5.0.11 stacked queries'

[10:02:18] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[10:02:19] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[10:02:20] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[10:02:21] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[10:02:22] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[10:02:23] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[10:02:30] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[10:02:37] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[10:02:48] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[10:02:59] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[10:03:13] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[10:03:26] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[10:03:37] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[10:03:47] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[10:03:58] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[10:04:12] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[10:04:25] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[10:04:36] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[10:04:46] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[10:04:56] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[10:05:07] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[10:05:23] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[10:05:38] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[10:05:49] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[10:06:00] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[10:06:16] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[10:06:29] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[10:06:40] [WARNING] Cookie parameter 'PHPSESSID' is not injectable
[10:06:40] [CRITICAL] all parameters appear to be not injectable. Also, you can try to rerun by providing either a valid --string or a valid --regexp, refer to the user's manual for details
[10:06:40] [WARNING] HTTP error codes detected during testing:
403 (Forbidden) - 58001 times

[*] shutting down at 10:06:40

Liite 5 Sqlmap-hyökkäysloki F5 ASM

root@bt:/pentest/database/sqlmap# ./sqlmap.py -u http://xxx.xxx.xxx.xxx/mutillidae/index.php?page=user-info.php --dbms mysql --data="username=&password=&user-info-php-submit-button=View+Account+Details" --threads 10 --user-agent="Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20100101 Firefox/14.0.1" --level 5 --risk 5 --cookie="PHPSESSID=paeq4qalkf17sbqi53id73vse6"

sqlmap/1.0-dev-25eca9d - automatic SQL injection and database takeover tool
http://sqlmap.org

[!] legal disclaimer: usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Authors assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 15:06:37

[15:06:38] [INFO] testing connection to the target url
[15:06:38] [INFO] heuristics detected web page charset 'ascii'
[15:06:38] [INFO] testing if the url is stable, wait a few seconds
[15:06:39] [INFO] url is stable
[15:06:39] [INFO] testing if POST parameter 'username' is dynamic
[15:06:40] [WARNING] POST parameter 'username' appears to be not dynamic
[15:06:40] [WARNING] reflective value(s) found and filtering out
[15:06:40] [WARNING] heuristic test shows that POST parameter 'username' might not be injectable
[15:06:40] [INFO] testing for SQL injection on POST parameter 'username'
[15:06:40] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:06:42] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:06:44] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:06:46] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[15:06:52] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:06:58] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:07:04] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[15:07:05] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[15:07:05] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'

[15:07:06] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[15:07:06] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[15:07:06] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[15:07:06] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[15:07:06] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[15:07:06] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[15:07:06] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:07:06] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:07:06] [INFO] testing 'MySQL stacked conditional-error blind queries'
[15:07:08] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[15:07:09] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:07:09] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[15:07:10] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[15:07:11] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[15:07:16] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:07:16] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[15:07:17] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[15:07:22] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[15:07:27] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[15:07:27] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[15:07:27] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[15:07:27] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[15:07:27] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[15:07:27] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[15:07:27] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[15:07:28] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[15:07:28] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[15:07:29] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[15:07:30] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[15:07:30] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'

[15:07:31] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[15:07:36] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[15:07:40] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[15:07:50] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[15:08:00] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[15:08:09] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[15:08:18] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[15:08:28] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[15:08:37] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[15:08:47] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[15:08:56] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[15:09:05] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[15:09:15] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:09:25] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[15:09:35] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[15:09:44] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[15:09:53] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[15:10:03] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[15:10:13] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[15:10:23] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[15:10:32] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[15:10:41] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[15:10:50] [WARNING] POST parameter 'username' is not injectable
[15:10:50] [INFO] testing if POST parameter 'password' is dynamic
[15:10:50] [WARNING] POST parameter 'password' appears to be not dynamic
[15:10:50] [WARNING] heuristic test shows that POST parameter 'password' might not be injectable
[15:10:50] [INFO] testing for SQL injection on POST parameter 'password'
[15:10:50] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:10:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:10:54] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:10:56] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[15:11:02] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:11:07] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:11:13] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[15:11:15] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[15:11:15] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'

[15:11:15] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[15:11:15] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[15:11:15] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[15:11:15] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[15:11:15] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[15:11:16] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[15:11:16] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:11:16] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:11:16] [INFO] testing 'MySQL stacked conditional-error blind queries'
[15:11:18] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[15:11:18] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:11:19] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[15:11:20] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[15:11:20] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[15:11:25] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:11:26] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[15:11:27] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[15:11:32] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[15:11:36] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[15:11:36] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[15:11:36] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[15:11:36] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[15:11:36] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[15:11:36] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[15:11:36] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[15:11:37] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[15:11:38] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[15:11:39] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[15:11:39] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[15:11:40] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'

[15:11:41] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[15:11:46] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[15:11:51] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[15:12:01] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[15:12:11] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[15:12:21] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[15:12:30] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[15:12:39] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[15:12:48] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[15:12:58] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[15:13:07] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[15:13:16] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[15:13:25] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:13:35] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[15:13:45] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[15:13:55] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[15:14:04] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[15:14:13] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[15:14:22] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[15:14:31] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[15:14:41] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[15:14:50] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[15:14:59] [WARNING] POST parameter 'password' is not injectable
[15:14:59] [INFO] testing if POST parameter 'user-info-php-submit-button' is dynamic
[15:14:59] [WARNING] POST parameter 'user-info-php-submit-button' appears to be not dynamic
[15:14:59] [WARNING] heuristic test shows that POST parameter 'user-info-php-submit-button' might not be injectable
[15:14:59] [INFO] testing for SQL injection on POST parameter 'user-info-php-submit-button'
[15:14:59] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:15:01] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:15:03] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:15:06] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[15:15:13] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:15:19] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:15:25] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[15:15:27] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'

[15:15:27] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[15:15:27] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[15:15:28] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[15:15:28] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[15:15:28] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[15:15:28] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[15:15:28] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[15:15:28] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:15:28] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:15:28] [INFO] testing 'MySQL stacked conditional-error blind queries'
[15:15:30] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[15:15:31] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:15:31] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[15:15:32] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[15:15:33] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[15:15:38] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:15:38] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[15:15:39] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[15:15:44] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[15:15:49] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[15:15:49] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[15:15:49] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[15:15:49] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[15:15:49] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[15:15:49] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[15:15:49] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[15:15:50] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[15:15:50] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[15:15:51] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[15:15:52] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'

[15:15:53] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[15:15:53] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[15:15:59] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[15:16:04] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[15:16:14] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[15:16:24] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[15:16:34] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[15:16:43] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[15:16:53] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[15:17:02] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[15:17:11] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[15:17:21] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[15:17:30] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[15:17:40] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:17:50] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[15:18:00] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[15:18:09] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[15:18:19] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[15:18:28] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[15:18:37] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[15:18:47] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[15:18:56] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[15:19:05] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[15:19:15] [WARNING] POST parameter 'user-info-php-submit-button' is not injectable
[15:19:15] [INFO] testing if GET parameter 'page' is dynamic
[15:19:15] [INFO] confirming that GET parameter 'page' is dynamic
[15:19:15] [INFO] GET parameter 'page' is dynamic
[15:19:15] [WARNING] heuristic test shows that GET parameter 'page' might not be injectable
[15:19:15] [INFO] testing for SQL injection on GET parameter 'page'
[15:19:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:19:17] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:19:19] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:19:21] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[15:19:26] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:19:32] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:19:37] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'

[15:19:39] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[15:19:39] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[15:19:39] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[15:19:39] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[15:19:39] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[15:19:39] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[15:19:39] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'
[15:19:39] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[15:19:39] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:19:39] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:19:39] [INFO] testing 'MySQL stacked conditional-error blind queries'
[15:19:42] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[15:19:42] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:19:43] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[15:19:44] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[15:19:44] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[15:19:49] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:19:50] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[15:19:51] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[15:19:55] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[15:20:00] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[15:20:00] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[15:20:00] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[15:20:00] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[15:20:00] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[15:20:00] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[15:20:00] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[15:20:01] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[15:20:02] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'

[15:20:03] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[15:20:04] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[15:20:04] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[15:20:05] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[15:20:11] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[15:20:18] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[15:20:29] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[15:20:39] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[15:20:48] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[15:20:58] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[15:21:08] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[15:21:17] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[15:21:27] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[15:21:36] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[15:21:46] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[15:21:55] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:22:06] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[15:22:16] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[15:22:25] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[15:22:34] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[15:22:43] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[15:22:53] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[15:23:02] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[15:23:12] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[15:23:21] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[15:23:31] [WARNING] GET parameter 'page' is not injectable
[15:23:31] [INFO] testing if User-Agent parameter 'User-Agent' is dynamic
[15:23:31] [WARNING] User-Agent parameter 'User-Agent' appears to be not dynamic
[15:23:31] [INFO] heuristic test shows that User-Agent parameter 'User-Agent' might be injectable (possible DBMS: MySQL)
[15:23:31] [INFO] testing for SQL injection on User-Agent parameter 'User-Agent'
[15:23:31] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:23:33] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:23:35] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[15:23:41] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:23:47] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[15:23:47] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'

[15:23:47] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[15:23:47] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:23:49] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:23:55] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[15:23:57] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[15:23:57] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[15:23:57] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[15:23:58] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[15:23:58] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[15:23:58] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:23:58] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:23:58] [INFO] testing 'MySQL stacked conditional-error blind queries'
[15:24:00] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[15:24:00] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:24:01] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[15:24:02] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[15:24:02] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[15:24:07] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:24:08] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[15:24:09] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[15:24:15] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[15:24:20] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[15:24:20] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[15:24:20] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[15:24:20] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[15:24:20] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[15:24:20] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[15:24:20] [INFO] testing 'MySQL > 5.0.11 stacked queries'

[15:24:21] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[15:24:21] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[15:24:24] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[15:24:27] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[15:24:28] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[15:24:28] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[15:24:35] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[15:24:41] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[15:24:51] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[15:25:13] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[15:25:23] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[15:25:43] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[15:25:52] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[15:26:14] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[15:26:23] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[15:26:43] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[15:26:53] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[15:27:15] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:27:25] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[15:27:45] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[15:27:54] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[15:28:14] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[15:28:23] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[15:28:44] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[15:28:53] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[15:29:14] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[15:29:23] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[15:29:43] [WARNING] User-Agent parameter 'User-Agent' is not injectable
[15:29:44] [INFO] testing if Cookie parameter 'PHPSESSID' is dynamic
[15:29:44] [WARNING] Cookie parameter 'PHPSESSID' appears to be not dynamic
[15:29:44] [WARNING] heuristic test shows that Cookie parameter 'PHPSESSID' might not be injectable
[15:29:44] [INFO] testing for SQL injection on Cookie parameter 'PHPSESSID'
[15:29:44] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:35:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:36:10] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[15:36:21] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Generic comment)'
[15:36:32] [INFO] testing 'Generic boolean-based blind - Parameter replace (original value)'
[15:36:33] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses'

[15:36:33] [INFO] testing 'Generic boolean-based blind - GROUP BY and ORDER BY clauses (original value)'
[15:36:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:36:52] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[15:37:02] [INFO] testing 'MySQL boolean-based blind - WHERE or HAVING clause (RLIKE)'
[15:37:20] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[15:37:21] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[15:37:21] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[15:37:22] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace (original value)'
[15:37:22] [INFO] testing 'MySQL < 5.0 boolean-based blind - Parameter replace (original value)'
[15:37:22] [INFO] testing 'MySQL >= 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:37:23] [INFO] testing 'MySQL < 5.0 boolean-based blind - GROUP BY and ORDER BY clauses'
[15:37:23] [INFO] testing 'MySQL stacked conditional-error blind queries'
[15:37:42] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'
[15:37:47] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:37:51] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'
[15:37:56] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'
[15:38:01] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE or HAVING clause'
[15:38:05] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[15:38:10] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE or HAVING clause (UPDATEXML)'
[15:38:15] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause'
[15:38:19] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause'
[15:38:23] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[15:38:23] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[15:38:23] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[15:38:23] [INFO] testing 'MySQL >= 5.0 error-based - GROUP BY and ORDER BY clauses'
[15:38:23] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (EXTRACTVALUE)'
[15:38:23] [INFO] testing 'MySQL >= 5.1 error-based - GROUP BY and ORDER BY clauses (UPDATEXML)'
[15:38:23] [INFO] testing 'MySQL > 5.0.11 stacked queries'

[15:38:28] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[15:38:33] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'
[15:38:37] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (comment)'
[15:38:42] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
[15:38:47] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[15:38:51] [INFO] testing 'MySQL > 5.0.11 OR time-based blind'
[15:38:55] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
[15:38:59] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[15:39:58] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[15:40:57] [INFO] testing 'MySQL UNION query (NULL) - 11 to 20 columns'
[15:41:50] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[15:42:44] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[15:43:37] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[15:44:31] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[15:45:25] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[15:46:19] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[15:47:12] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[15:48:06] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:49:04] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[15:50:03] [INFO] testing 'Generic UNION query (NULL) - 11 to 20 columns'
[15:50:57] [INFO] testing 'Generic UNION query (random number) - 11 to 20 columns'
[15:51:50] [INFO] testing 'Generic UNION query (NULL) - 21 to 30 columns'
[15:52:43] [INFO] testing 'Generic UNION query (random number) - 21 to 30 columns'
[15:53:37] [INFO] testing 'Generic UNION query (NULL) - 31 to 40 columns'
[15:54:31] [INFO] testing 'Generic UNION query (random number) - 31 to 40 columns'
[15:55:25] [INFO] testing 'Generic UNION query (NULL) - 41 to 50 columns'
[15:56:18] [INFO] testing 'Generic UNION query (random number) - 41 to 50 columns'
[15:57:12] [WARNING] Cookie parameter 'PHPSESSID' is not injectable
[15:57:12] [CRITICAL] all parameters appear to be not injectable. Also, you can try to rerun by providing either a valid --string or a valid --regexp, refer to the user's manual for details

[*] shutting down at 15:57:1