IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Skripsi

untuk memenuhi sebagian persyaratan

mencapai derajat Sarjana S-1

Program Studi Teknik Informatika

Disusun oleh :

Feri Setiyawan

10650030

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2014

IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Skripsi

untuk memenuhi sebagian persyaratan

mencapai derajat Sarjana S-1

Program Studi Teknik Informatika

Disusun oleh :

Feri Setiyawan

10650030

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2014

Universitas Islam Negeri Sunan Kalijaga FM-UINSK-BM-05-07/R0

PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/2850/2014

Skripsi/Tugas Akhir dengan judul : Implementasi Firewall Aplikasi Web Untuk Mencegah SQL Injection Menggunakan Naxsi

Yang dipersiapkan dan disusun oleh :
Nama : Feri Setiyawan
NIM : 10650030
Telah dimunaqasyahkan pada : Selasa, 26 Agustus 2014
Nilai Munaqasyah : A

Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Dr. Imam Riadi, M.Kom
NIY. 60020397

Penguji I

M. Mustakim, M.T
NIP.19790331 200501 1 004

Penguji II

Arief Ikhwan W, M.Cs
NIP.

Yogyakarta, 25 September 2014
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan

Prof. Dr. H. Akh. Minhaji, M.A, Ph.D
NIP. 19580919 198603 1 002

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Permohonan
Lamp : -

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama	: Feri Setiyawan
NIM	: 10650030
Judul Skripsi	: Implementasi Firewall Aplikasi Web Untuk Mencegah SQL Injection Menggunakan Naxsi

sudah dapat diajukan kembali kepada Program Studi Tekni Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Teknik Informatika

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 17 Agustus 2014
Pembimbing

Dr. Imam Riadi, M.Kom
NIY: 60020397

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama	: Feri Setiyawan
Nim	: 10650030
Program Studi	: Teknik Informatika
Fakultas	: Sains dan Teknologi

Menyatakan bahwa skripsi dengan judul **Implementasi *Firewall* Aplikasi Web Untuk Mencegah *SQL Injection* Menggunakan *Naxsi*** tidak terdapat pada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi, dan sepengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 17 Agustus 2014

Yang Menyatakan,

Feri Setiyawan
NIM : 10650030

MOTTO

“Ketakutan yang menenggelamkan dan keberanian yang menyesatkan”

“Tetap melaju kencang di rute yang tak selalu nyaman”

“Apapun yang pernah tergenggam pasti akan memudar lalu hilang”

“Sudahilah sedihmu yang belum sudah segera mulailah syukurmu yang pasti indah”

KATA PENGANTAR

Alhamdulillahirabbil'alamiin, Puji syukur penulis panjatkan kepada Allah SWT karena dengan restu-Nya pelaksanaan dan penyusunan skripsi yang berjudul "Implementasi Firewall Aplikasi Web Untuk Mencegah SQL Injection Menggunakan Naxsi" dapat diselesaikan sebagai persyaratan menyelesaikan Sarjana Strata Satu (S1) Jurusan Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

Penulisan skripsi ini tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, ucapan terimakasih penulis sampaikan kepada :

1. Kepada orangtuaku, Bapak Muhdi dan Ibu Sri Sudarti yang selalu mendoakan agar penulis selalu dalam lindungan-Nya dan dimudahkan dalam segala urusannya, juga mendukung dalam segala kebaikan penulis.
2. Bapak Prof. Drs. H. Akh Minhaji, M.A., Ph.D., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
3. Bapak Agus Mulyanto, M.Kom, selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
4. Bapak Dr. H. Imam Riadi, M.Kom, selaku Dosen Pembimbing yang selalu meluangkan waktunya untuk mengarahkan penulis dalam menyelesaikan tugas akhir ini.

5. Bapak Nurrochman, M.Kom, selaku Dosen Pembimbing Akademik yang selalu memberikan dorongan terus belajar dan berkarya dalam proses belajar.
6. Bapak M. Mustakim, M.T dan Bapak Arief Ikhwan W, M.Cs selaku Dosen Penguji Munaqosyah yang telah memberikan masukan-masukan agar tugas akhir saya menjadi lebih baik
7. Bapak/Ibu Dosen Teknik Informatika yang sangat luar biasa dalam menularkan ilmunya selama penulis belajar di UIN Sunan Kalijaga.
8. Kakakku Bambang Arditya Fitri Sulistyanto A.Md yang selalu memotivasi penulis untuk selalu lebih baik dari kakaknya.
9. Bapak Mohammad Wahdan, selaku staff TU yang selalu membantu penulis dalam menyelesaikan administrasi guna terselesaikannya tugas akhir ini.
10. Sahabat saya Rahmat Nur Faizin yang telah memberikan banyak bantuan dan sharing ilmunya kepada saya untuk dapat menyelesaikan tugas akhir ini.
11. Sahabat saya Rasyid Yeni Saputra yang telah memberikan sharing ilmunya dan sahabat saya Hana Soffa yang telah memberikan fasilitas untuk mencetak dan menyelesaikan tugas akhir saya.
12. Sahabat-sahabat seperbimbingan Bapak Imam Riadi Angkatan 2010, Fafa, Hanan, Arya, Fajar, Faizal yang telah berjuang bersama-sama, sampai setiap hari menginap di Lab. Terpadu bersama-sama. Semoga kebersamaan ini senantiasa dieratkan.

13. Sahabat-sahabat seperjuangan Teknik Informatika 2010 baik kelas regular ataupun mandiri yang telah menemani penulis baik susah maupun senang.
14. Bapak Nur, Bapak Suwono, Bapak Didit selaku security UIN Sunan Kalijaga yang selalu membuka pintu Laboraturium Terpadu setiap pagi.
15. Semua pihak yang tidak bisa disebutkan satu per satu, terima kasih atas segala bantuannya.

Penulis berharap semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang sudah banyak membantu penulis dalam menyelesaikan tugas akhir ini dengan melimpahkan rahmat dan karunia-Nya, Amin....

Semoga karya penelitian tugas akhir ini dapat memberikan manfaat dan kebaikan bagi banyak pihak demi kemajuan bersama serta bernilai ibadah dihadapan Allah SWT. Amin..

Yogyakarta, 29 Oktober 2014

Penulis,

Feri Setiyawan
10650030

HALAMAN PERSEMBAHAN

Allah SWT, yang selalu melimpahkan banyak rahmat dan kenikmatan-Nya sehingga skripsi ini dapat terselesaikan dengan lancar

Nabi akhiruzzaman Muhammad SAW,
semoga shalawat senantiasa terhatur kepadamu.

Ayahanda Muhdi dan Ibunda Sri Sudarti, terimakasih atas bimbingan moral dan spiritualnya selama ini.
Semoga kalian berdua selalu dijunjung tinggi haknya di dunia maupun di akhirat.

Kakak ku , Adit yang senantiasa
menyemangatiku ketika down untuk terus maju.

Almamater tercinta Teknik Informatika
UIN Sunan Kalijaga Yogyakarta.
Inilah yang bisa aku torehkan sebagai
cinderamata hasil pembelajaranku.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LIST

IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Feri Setiyawan

10650030

INTISARI

Keamanan website sangat diperlukan bagi suatu organisasi ataupun perusahaan karena untuk menjaga integritas data dan informasi website tersebut. Website yang tidak mempunyai sistem keamanan akan sangat berpotensi kehilangan data dan informasi yang dimiliki. Namun kebanyakan pengelola website mengabaikan sistem keamanan pada suatu website. Padahal saat ini banyak *cracker* yang tidak bertanggung jawab yang dapat mencari kelemahan sistem dan merusak sistem yang ada pada website tersebut. Salah satu serangan yang sering digunakan oleh para *cracker* adalah *SQL Injection*.

Penelitian ini lebih menekankan implementasi *firewall aplikasi web* dengan menggunakan naxsi yang akan dikonfigurasikan pada *web server nginx* untuk dapat mencegah serangan yang dilakukan dengan menggunakan teknik *SQL Injection*. Penelitian ini menggunakan metode pengumpulan data berupa studi literatur. Sedangkan tahapan penelitian terbagi dalam beberapa langkah yakni analisa kebutuhan sistem, perancangan sistem dan jaringan, implementasi rancangan sistem dan pengujian sistem.

Berdasarkan hasil penelitian, menunjukan bahwa implementasi *firewall aplikasi web naxsi* berhasil mencegah serangan dengan menggunakan teknik *SQL Injection* berdasarkan pengujian *penetration testing*. Performa *web server* tetap optimal dan tidak terpengaruh dengan digunakannya naxsi dengan hasil persentase 63.3% sangat setuju dan 36.7% setuju saat dilakukan pengujian *baseline performance measurement* menggunakan *apache benchmark*. Selanjutnya dapat disimpulkan bahwa implementasi *firewall aplikasi web naxsi* dapat berjalan dengan baik dan lancar.

Kata Kunci : *SQL Injection, Naxsi, Web Server, Nginx, Baseline Performance Measurement*

IMPLEMENTATION OF WEB APPLICATION FIREWALL FOR PREVENTING SQL INJECTION USING NAXSI

Feri Setiyawan

10650030

ABSTRACT

Website security is necessary for an organization or company as to maintain the integrity of the data and information on the organization or company. A website that does not have a security system would potentially lose data and information held. However, most managers websites ignore the security system on a website. While now many crackers that are not responsible for the weakness of the system that can locate and wreck the existing system on the website. One of attacking techniques that are often to used by the cracker is SQL Injection

This study emphasize the implementation of a web application firewall uses naxsi to be configured on the web server nginx to be able to prevent the attacks carried out by using SQL Injection techniques. This study uses the method of data collection in the form of literature. While the study is divided into several steps that analysis of system requirements, the design of systems and networks, implementation of the system design and system testing.

Based on the results of study, indicate that the implementation of a web application firewall naxsi successfully prevent attacks using SQL Injection Techniques, based on penetration testing. Performance of the web server still optimum and not affected by use of naxsi, with the percentage 63,3% strongly agree and 36,7% agree when testing baseline performance measurement using apache benchmark. It can be concluded that the implementation of a web application firewall naxsi can run well and smoothly.

Keywords : *SQL Injection, Naxsi, Web Server, Nginx, Baseline Performance Measurement*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan akan informasi yang semakin berkembang di dunia maya membuat para *developer website* berlomba-lomba menyajikan berbagai layanan untuk para pengguna. Dari masa ke masa teknologi *website* mengalami perkembangan yang begitu pesat. Keamanan *website* sangat diperlukan bagi suatu organisasi ataupun perusahaan karena untuk menjaga integritas data dan informasi pada organisasi atau perusahaan tersebut. *Website* yang tidak menawarkan keamanan akan sangat berpotensi hilangnya integritas data dan kepercayaan konsumen akan perusahaan tersebut. Namun, kebanyakan pemilik *website* mengabaikan *security system* pada *website* tersebut. Padahal banyak sekali *cracker-cracker* yang tidak bertanggung jawab yang dapat merusak dan mencari kelemahan sistem dalam *website* tersebut. Salah satu serangan yang paling sering digunakan oleh para *cracker* tersebut adalah *SQL Injection*.

SQL Injection merupakan suatu teknik mengeksploitasi web aplikasi yang didalamnya menggunakan database sebagai penyimpanan datanya. *SQL Injection* mengijinkan user tidak sah untuk mengakses database, *SQL Injection* juga memungkinkan seorang *attacker* merubah, menghapus maupun menambahkan data-data pada *website* tanpa harus memiliki *account* sebagai admin.

Website CV Ganesha Muda Arcitech masih menggunakan *php* versi lama, dimana masih terdapat banyak *bug SQL Injection*. Belum adanya *firewall* juga menambah kerentanan terhadap serangan ke website ini. *Web server* yang digunakan di CV Ganesha Muda Arcitech adalah *nginx.*

Nginx merupakan salah satu Web Server berbasis *Free and Open Source Software*. *Nginx* memiliki keunggulan dengan hanya membutuhkan sedikit memori tetapi berkemampuan tinggi, lebih stabil dan mempunyai banyak fitur. Hal ini menyebabkan banyak perusaahan mulai beralih menggunakan *Nginx* sebagai Web Server andalan mereka. Namun, dari sisi keamanan *nginx* juga mempunyai celah - celah yang dapat disalahgunakan oleh orang yang tidak bertanggungjawab.

Naxsi merupakan singkatan dari *Nginx Anti XSS & SQL Injection* dan memiliki pendekatan *negatif* untuk pemeriksaan lalu lintas *website* dengan metode *blacklisting.* Ini berarti lalu lintas diblokir secara *default. Naxsi* menggunakan dua file berbeda yang berbeda aturan. Pertama pada tingkat konfigurasi server. Kedua, pada tingkat konfigurasi lokasi HTTP (Pellikaan, 2013).

Berdasarkan latar belakang diatas, maka peneliti akan mengimplementasikan cara untuk mencegah serangan *SQL Injection* dengan menggunakan *firewall aplikasi web naxsi* dan akan mengukur sejauh mana *naxsi* mampu mengatasi serangan *SQL Injection* tersebut.

1.2 Rumusan Masalah

Dengan didasari oleh latar belakang diatas, maka pemasalahan penelitian dapat dirumuskan sebagai berikut :

1. Bagaimana implementasi *firewall aplikasi web naxsi* pada *web server nginx* ?
2. Bagaimana mencegah serangan *SQL Injection* dengan *firewall aplikasi web naxsi* ?
3. Apakah implementasi *firewall aplikasi web naxsi* dapat mempengaruhi kinerja web server ?

1.3 Batasan Masalah

Agar penelitian dapat mencapai sasaran dan tujuan yang diharapkan, maka permasalahan yang ada dibatasi sebagai berikut:

1. Aplikasi yang digunakan untuk menangani *SQL Injection* ini adalah *Naxsi versi 0.49*
2. *Web server* menggunakan *nginx versi 1.4.4*
3. *CMS* yang digunakan *website* CV Ganesha Muda Arcitech adalah *CMS* Joomla versi 3.2.1
4. Penelitian ini terbatas tentang pencegahan *SQL Injection*.
5. Pengujian performa yang dilakukan terbatas pada kecepatan waktu respon (*respon time*) dan banyaknya permintaan per detik (*request per second*) sebelum dan sesudah pengimplementasian *naxsi*.

6. Pengujian performa dilakukan dengan menggunakan kecepatan koneksi 512 kb/s.
7. Pengujian performa dilakukan pada web server nginx yang mempunyai 20 website didalamnya.
8. Algoritma *firewall* menggunakan metode *Blacklisting.*

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini, antara lain :

1. Mengimplementasikan *firewall aplikasi web naxsi* pada *web server nginx*.
2. Mencegah serangan *SQL Injection* dengan *firewall aplikasi web naxsi*.
3. Mengetahui pengaruh *firewall aplikasi web naxsi* terhadap kinerja web server.

1.5 Keaslian penelitian

Penelitian umum terkait pencegahan *SQL Injection* sudah pernah dilakukan sebelumnya. Akan tetapi penelitian terdahulu masih belum ada yang pernah menggunakan *Naxsi* sebagai solusi penanganan *SQL Injection.* Pada penelitian ini penulis akan melakukan implementasi *Firewall Aplikasi Web Naxsi* untuk melakukan pencegahan terhadap serangan *SQL Injection,* yang mana sejauh ini topik serupa belum pernah ada dilakukan sebelumnya, khususnya di UIN Sunan Kalijaga Yogyakarta.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. *Firewall aplikasi web* naxsi berhasil dibangun pada web server nginx dengan menggunakan metode *blacklisting* untuk memblock sintaks-sintaks sql yang dapat berpotensi membahayakan untuk keamanan website.

2. *Firewall Aplikasi Web Naxsi* dapat digunakan untuk mencegah SQL Injection baik yang dilakukan secara manual ataupun dilakukan dengan menggunakan SQL Injection Tools. Firewall aplikasi web naxsi menjadi perlindungan yang baik untuk website dari serangan SQL Injection karena langsung menutup semua celah-celah vulneravility yang terdapat dari suatu website.

3. *Firewall aplikasi web naxsi* yang telah terpasang pada web server nginx tidak terlalu berpengaruh terhadap kinerja dari web server nginx, berdasarkan pada pengujian *baseline performance measurement* yang menggunakan parameter *response time* dan *request per second* performa dari *web server nginx* tetap optimal dengan adanya *firewall aplikasi web naxsi*.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, masih membutuhkan saran-saran untuk mendukung kesempurnaan dalam penelitian ini, saran tersebut diantaranya sebagai berikut :

1. Penelitian kedepan, peneliti bisa menggunakan *SQL Injection tools* yang lebih canggih, agar pengujian dari keamanan *website* dapat dilakukan dengan lebih optimal.
2. Penelitian kedepan, peneliti selanjutnya bisa menggunakan beberapa metode algoritma pengujian dalam satu penelitian, misalnya selain menggunakan metode *blacklisting* bisa menggunakan juga metode *whitelisting* dan metode *greylisting*. Peneliti ke depan juga bisa melakukan perbandingan dengan menggunakan ketiga metode tersebut untuk mengetahui metode mana yang lebih baik digunakan dalam hal mencegah serangan yang menggunakan teknik *SQL Injection.*

DAFTAR PUSTAKA

Alisherov, Farkhod., Feruza Sattarova (2009) *Methology For Penetration Testing*. Multimedia Engineering Department, Hannam University, South Korea

Cartealy, Imam (2013) *Linux Networking*, Jasakom.

Clarke, Justin (2012) *Sql Injection Attacks And Defense*, Elsevier.

Dahlan, Mohammad (2014) *Pengujian Dan Analisa Keamanan Website Terhadap Serangan SQL Injection*, Kudus : Universitas Muria

Dharmawan, Eka Adhitya., Erni Yudaningtyas., M.Sarosa (2013) *Perlindungan Web pada login sistem Menggunakan Algoritma Rijndael*, sumber : http://jurnaleeccis.ub.ac.id/index.php/eeccis/article/view/207

Handaya, Wilfridus Bambang Triadi., Bernard Renaldy Suteja., Ahmad Ashari (2010) *Linux System Administrator*, Bandung: Penerbit Informatika.

Komputer, Wahana. (2003). *Konsep jaringan komputer dan pengembangannya***,** Jakarta: Salemba Infotek

Nedelcu, Clement (2010) *Nginx HTTP Server : Adopt Nginx for Your Web Application to Make the Most of Your Insfrastructure and Server Pages Faster Than Ever***,** Packt Publishing Ltd.

Nugroho, Bunafit (2004) *PHP & MySQL Dreamweaver MX*, Yogyakarta : Penerbit Andi.

Nystrom, Martin (2007) *Sql Injection Defenses***,** O'Reilly Media.

Oetomo, Budi Sutedjo Dharma (2003) *Konsep dan Perancangan Jaringan Komputer***,** Yogyakarta: Penerbit Andi.

Pellikaan, Denis (2013) *Naxsi Performance Measurement*, University of Amsterdam.

Puspita, Oky Ristyarani (2011). *Implementasi Pencegahan Serangan Sql Injection Menggunakan Greensql Berbasis Php*. Politeknik Telkom : Bandung.

Rahardjo, Budi (2002) *Keamanan Sistem Informasi Berbasis Internet***,** Jakarta : PT Insan Indonesia & PT INDOCISC.

Ramadhani, Graifhan. (2003). *Modul Pengenalan internet* sumber: http://directory.umm.ac.id/tik/pengenalan_*internet*.pdf

Setiasih, Meita., Deborah Estefanus., Michael Andinata (2012) *Perancangan Keamanan Halaman Web Berbasis Platform PHP Terhadap Serangan SQL Injection*, Jakarta : Universitas Bina Nusantara

Syafrizal, Melwin. (2005) *Pengantar Jaringan Komputer, Yogyakarta* : Penerbit Andi.

Yudistira, Alifandi. (2012***)*** *Analisis Keamanan Otentikasi dan Basis Data Pada Web Simple-O Menggunakan SQL Injection*, Depok : Universitas Indonesia

Wagito. (2007). *Jaringan Komputer Teori dan Implementasi Berbasis Linux*. Yogyakarta: GAVA media.

http://centrin.net.id/~leonardl/linux/gnuplot.html diakses pada 23 Juni 2014

http://laser.cs.umass.edu/manual/programs/ab.html diakses pada 21 Juni 2014

http://www.exploit-db.com/exploits/31459/ diakses pada 27 Februari 2014

http://www.itsecteam.com/products/havij-advanced-sql-injection/ diakses pada 29 Mei 2014

http://www.proteansec.com/linux/naxsi/ diakses pada 12 Januari 2014

LAMPIRAN A

KODE SUMBER (SOURCE CODE) KONFIGURASI WEB SERVER NGINX

Nama file : interfaces
Lokasi : web server nginx

```
This file describes the network interfaces available on your
system
# and how to activate them. For more information, see
interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
        address 192.168.100.2
        netmask 255.255.255.0
        network 192.168.100.0
        broadcast 192.168.100.255
        gateway 192.168.100.1
        # dns-* options are implemented by the resolvconf package,
if installed
        dns-nameservers 192.168.100.1
        dns-search lab.jarkom.uin
```

Nama file : resolv.conf
Lokasi : web server nginx

```
search lab.jarkom.uin
nameserver 8.8.8.8
```

Nama file : nginx.conf
Lokasi : Web Server Nginx

```
user www-data;
worker_processes 4;
pid /var/run/nginx.pid;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 2;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ##
        # Logging Settings
        ##

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        ##
        # Gzip Settings
        ##
        gzip on;
        gzip_disable "msie6";

        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json
        application/x-javascript text/xml application/xml
        application/xml+rss text/javascript;

        ##
        # nginx-naxsi config
        ##
```

```
        # Uncomment it if you installed nginx-naxsi
        ##

        include /etc/nginx/naxsi_core.rules;

        ##
        # nginx-passenger config
        ##
        # Uncomment it if you installed nginx-passenger
        ##

        #passenger_root /usr;
        #passenger_ruby /usr/bin/ruby;

        ##
        # Virtual Host Configs
        ##

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
        client_max_body_size 8M;
}


#mail {
#       # See sample authentication script at:
#       #
http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#       # auth_http localhost/auth.php;
#       # pop3_capabilities "TOP" "USER";
#       # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#       server {
#               listen     localhost:110;
#               protocol   pop3;
#               proxy      on;
#       }
#
#       server {
#               listen     localhost:143;
#               protocol   imap;
#               proxy      on;
#       }
#}
```

Nama file : Sites-available
Lokasi : Web Server Nginx

```
# You may add here your
# server {
#       ...
# }
# statements for each of your virtual hosts to this file

##
# You should look at the following URL's in order to grasp a solid
understanding
# of Nginx configuration files in order to fully unleash the power
of Nginx.
# http://wiki.nginx.org/Pitfalls
# http://wiki.nginx.org/QuickStart
# http://wiki.nginx.org/Configuration
#
# Generally, you will want to move this file somewhere, and start
with a clean
# file but keep this around for reference. Or just disable in
sites-enabled.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed
examples.
##

server {
#       proxy_set_header Proxy-Connection "";
        listen   80; ## listen for ipv4; this line is default and
implied
        listen   [::]:80 default ipv6only=on; ## listen for ipv6

        root /usr/share/nginx/www/joomla;
        index index.php index.html index.htm;

        # Make site accessible from http://localhost/
        server_name localhost;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a
404.
                try_files $uri $uri/ /index.php?$args;
                # Uncomment to enable naxsi on this location
                include /etc/nginx/naxsi.rules;
#               proxy_pass http://192.168.100.2/;
      }
        location /doc/ {
                alias /usr/share/doc/;
                autoindex on;
                allow 127.0.0.1;
                allow ::1;
                deny all;
        }
```

```
        # Only for nginx-naxsi used with nginx-naxsi-ui : process
          denied requests
        location /RequestDenied {
        #       proxy_pass http://127.0.0.1:8080;
                return 500;
        }

        #error_page 404 /404.html;
        error_page 500 /naxsi.html;
        location = /naxsi.html {
                root /usr/share/nginx/www;
        }
        # redirect server error pages to the static page /50x.html
        #
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
                root /usr/share/nginx/www;
        }

        # pass the PHP scripts to FastCGI server listening on
          127.0.0.1:9000
        #
        location ~ \.php$ {
                try_files $uri =404;
                include /etc/nginx/naxsi.rules;
                fastcgi_split_path_info ^(.+\.php)(/.+)$;
                # NOTE: You should have "cgi.fix_pathinfo = 0;" in
                  php.ini

                # With php5-cgi alone:
                # fastcgi_pass 127.0.0.1:9000;
                # With php5-fpm:
                fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_param SCRIPT_FILENAME
        $document_root$fastcgi_script_name;
                fastcgi_index index.php;
                include fastcgi_params;
        }

        # deny access to .htaccess files, if Apache's document
           root
        # concurs with nginx's one
        #
        location ~ /\.ht {
                deny all;
        }


}
# another virtual host using mix of IP-, name-, and port-based
configuration
#
#server {
#       listen 8000;
#       listen somename:8080;
#       server_name somename alias another.alias;
```

```
#       root html;
#       index index.html index.htm;
#
#       location / {
#               try_files $uri $uri/ =404;
#       }
#}

# HTTPS server
#
#server {
#       listen 443;
#       server_name localhost;
#
#       root html;
#       index index.html index.htm;
#
#       ssl on;
#       ssl_certificate cert.pem;
#       ssl_certificate_key cert.key;
#
#       ssl_session_timeout 5m;
#
#       ssl_protocols SSLv3 TLSv1;
#       ssl_ciphers
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv3:+EXP;
#       ssl_prefer_server_ciphers on;
#
#       location / {
#               try_files $uri $uri/ =404;
#       }
#}
```

LAMPIRAN B

KODE SUMBER (SOURCE CODE) KONFIGURASI FIREWALL APLIKASI WEB NAXSI

Nama file : Naxsi.rules
Lokasi : Web Server Nginx

```
# Sample rules file for default vhost.

LearningMode;
SecRulesEnabled;
#SecRulesDisabled;
DeniedUrl "/RequestDenied";

## check rules
CheckRule "$SQL >= 2" BLOCK;
CheckRule "$RFI >= 8" BLOCK;
CheckRule "$TRAVERSAL >= 4" BLOCK;
CheckRule "$EVADE >= 4" BLOCK;
CheckRule "$XSS >= 8" BLOCK;
```

Nama file : Naxsi_core.rules
Lokasi : Web Server Nginx

```
##################################
## INTERNAL RULES IDS:1-10      ##
##################################
#weird_request : 1
#big_body : 2
#no_content_type : 3

#MainRule "str:yesone" "msg:foobar test pattern" "mz:ARGS"
"s:$SQL:42" id:1999;

##################################
## SQL Injections IDs:1000-1099 ##
##################################
MainRule
"rx:select|union|update|delete|insert|table|from|ascii|hex|unhex"
"msg:sql keywords" "mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie"
"s:$SQL:4" id:1000;
MainRule "str:\"" "msg:double quote"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1001;
MainRule "str:0x" "msg:0x, possible hex encoding"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:2" id:1002;
## Hardcore rules
MainRule "str:/*" "msg:mysql comment (/*)"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8" id:1003;
MainRule "str:*/" "msg:mysql comment (*/)"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8" id:1004;
MainRule "str:|" "msg:mysql keyword (|)"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8" id:1005;
```

```
MainRule "rx:&&" "msg:mysql keyword (&&)"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8" id:1006;
## end of hardcore rules
MainRule "str:--" "msg:mysql comment (--)"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1007;
MainRule "str:;" "msg:; in stuff" "mz:BODY|URL|ARGS" "s:$SQL:4"
id:1008;
MainRule "str:=" "msg:equal in var, probable sql/xss"
"mz:ARGS|BODY" "s:$SQL:2" id:1009;
MainRule "str:(" "msg:parenthesis, probable sql/xss"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1010;
MainRule "str:)" "msg:parenthesis, probable sql/xss"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1011;
MainRule "str:'" "msg:simple quote"
"mz:ARGS|BODY|URL|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1013;
MainRule "str:\"" "msg:double quote"
"mz:ARGS|BODY|URL|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1014;
MainRule "str:," "msg:, in stuff"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1015;
#MainRule "str:#" "msg:mysql comment (#)"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1016;

################################
## OBVIOUS RFI IDs:1100-1199 ##
################################
MainRule "str:http://" "msg:html comment tag"
"mz:ARGS|BODY|$HEADERS_VAR:Cookie" "s:$RFI:8" id:1100;
#MainRule "str:https://" "msg:html comment tag"
"mz:ARGS|BODY|$HEADERS_VAR:Cookie" "s:$RFI:8" id:1101;
MainRule "str:ftp://" "msg:html comment tag"
"mz:ARGS|BODY|$HEADERS_VAR:Cookie" "s:$RFI:8" id:1102;
MainRule "str:php://" "msg:html comment tag"
"mz:ARGS|BODY|$HEADERS_VAR:Cookie" "s:$RFI:8" id:1103;

########################################
## Directory traversal IDs:1200-1299 ##
########################################
MainRule "str:.." "msg:html comment tag"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$TRAVERSAL:4" id:1200;
MainRule "str:/etc/passwd" "msg:html comment tag"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$TRAVERSAL:4" id:1202;
MainRule "str:c:\\" "msg:html comment tag"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$TRAVERSAL:4" id:1203;
MainRule "str:cmd.exe" "msg:html comment tag"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$TRAVERSAL:4" id:1204;
MainRule "str:\\" "msg:html comment tag"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$TRAVERSAL:4" id:1205;
#MainRule "str:/" "msg:slash in args"
"mz:ARGS|BODY|$HEADERS_VAR:Cookie" "s:$TRAVERSAL:2" id:1206;
########################################
## Cross Site Scripting IDs:1300-1399 ##
########################################
MainRule "str:<" "msg:html open tag"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$XSS:8" id:1302;
MainRule "str:>" "msg:html close tag"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$XSS:8" id:1303;
```

```
MainRule "str:'" "msg:simple quote"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$XSS:8" id:1306;
MainRule "str:\"" "msg:double quote"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$XSS:8" id:1307;
MainRule "str:(" "msg:parenthesis"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$XSS:8" id:1308;
MainRule "str:)" "msg:parenthesis"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$XSS:8" id:1309;
MainRule "str:[" "msg:html close comment tag"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$XSS:4" id:1310;
MainRule "str:]" "msg:html close comment tag"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$XSS:4" id:1311;
MainRule "str:~" "msg:html close comment tag"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$XSS:4" id:1312;
MainRule "str:;" "msg:semi coma" "mz:ARGS|URL|BODY" "s:$XSS:8"
id:1313;
MainRule "str:`"  "msg:grave accent !"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$XSS:8" id:1314;
MainRule "rx:%[2|3]."  "msg:double encoding !"
"mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$XSS:8" id:1315;

####################################
## Evading tricks IDs: 1400-1500 ##
####################################
MainRule "str:&#" "msg: utf7/8 encoding"
"mz:ARGS|BODY|URL|$HEADERS_VAR:Cookie" "s:$EVADE:4" id:1400;
MainRule "str:%U" "msg: M$ encoding"
"mz:ARGS|BODY|URL|$HEADERS_VAR:Cookie" "s:$EVADE:4" id:1401;
MainRule negative "rx:multipart/form-data|application/x-www-form-
urlencoded" "msg:Content is neither mulipart/x-www-form.."
"mz:$HEADERS_VAR:Content-type" "s:$EVADE:4"$

##############################
## File uploads: 1500-1600 ##
##############################
MainRule "rx:.ph*|.asp*" "msg:asp/php file upload!" "mz:FILE_EXT"
"s:$UPLOAD:8" id:1500;
```

LAMPIRAN C

KODE SUMBER (SOURCE CODE) MODUL NAXSI

Nama file : Index.tpl
Lokasi : Web Server Nginx

```
<html lang="en">
<head>
<title>Naxsi Graphs</title>
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<link href="bootstrap/css/bootstrap.css" rel="stylesheet">
<link href="bootstrap/css/bootstrap-responsive.css"
rel="stylesheet">
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"
type="text/javascript"></script>
<script type="text/javascript"
src="bootstrap/js/bootstrap.js"></script>
<style>
      body {
        padding-top: 60px;
      }
    </style>
<script type="text/javascript" src="js/highcharts.js"></script>

<script type="text/javascript">
function DisplayHome() {
document.getElementById('home').setAttribute("class", "active");
document.getElementById('help').setAttribute("class", "inactive")
document.getElementById('stats').setAttribute("class",
"inactive");
document.getElementById('display').innerHTML = "Welcome to the
NAXSI Web Interface !";
}

function DisplayHelp() {
document.getElementById('home').setAttribute("class", "inactive");
document.getElementById('help').setAttribute("class", "active");
document.getElementById('stats').setAttribute("class",
"inactive");
document.getElementById('display').innerHTML = "<p style='text-
align:center'><b>Naxsi Rules Extractor</b></p>\n    <h3>How to
extract generated rules from the database$
}

function DisplayStats()
{
document.getElementById('home').setAttribute("class", "inactive");
document.getElementById('help').setAttribute("class", "inactive");
document.getElementById('stats').setAttribute("class", "active");
document.getElementById('display').innerHTML = "__STATS__";
}
```

```
function changestatus()
{
  $.get( "/changestatus", function( data ) {
    alert( "Status Change" );
    location.reload(true);
  });
}

function restartnginx()
{
  $.get("/restartnginx", function(data) {
    alert("Nginx Restarted");
    location.reload(true);
    });
}
</script>

</head>
<body>

 <div class="navbar navbar-fixed-top">
      <div class="navbar-inner">
        <div class="container">
          <a class="btn btn-navbar" data-toggle="collapse" data-
target=".nav-collapse">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </a>
          <a class="brand" href="#">Naxsi Web Interface</a>
          <div class="nav-collapse">
            <ul class="nav">
              <li class="active" id="home"><a href="#"
onclick="javascript:DisplayHome()">Home</a></li>
              <li class="inactive" id="Graphics"><a
href="/graphs">Graphics</a></li>
              <li class="inactive" id="rules"><a
href="/get_rules">Generate Whitelist</a></li>
              <li class="inactive" id="stats"><a href="#"
onclick="javascript:DisplayStats()">Statitics</a></li>
              <li class="inactive" id="help"><a href="#"
onclick="javascript:DisplayHelp()">Help</a></li>
            </ul>
          </div>
        </div>
      </div>
    </div>
<div class="container">
  <div id="display">
  <div class="container">
        <section id="thumbnails">

          <div class="row-fluid">
            <ul class="thumbnails">
              <li class="span4">
                <div class="thumbnail">
```

```
        <!-- <img src="gambar/2.jpg" alt=""> -->
        <div class="caption">
          <h2>Naxsi Status<br /></h2>
          %learningmode%
        </div>
      </div>
    </li>
    <li class="span4">
      <div class="thumbnail">
        <!-- <img src="gambar/2.jpg" alt=""> -->
        <div class="caption">
          <table><tr>
          <td>
          <div width="100%" style="text-align:left;">
          <img src ="http://1.bp.blogspot.com/-
          6QWMc4PHAq8/UIeOkUujg7I/AAAAAAAAACo/yfvLa3Zw7R
          U/s200/logo-uin-suka-baru-warna.jpg">
          </div>
          </td>
          <td>
          <div width="100%" style="text-align:center;">
          <h2>Naxsi Modul<br /></h2>
          </div>
          </td>
          <td>
          <div width="100%" style="text-align:right;">
          <img src ="http://1.bp.blogspot.com/-
          6QWMc4PHAq8/UIeOkUujg7I/AAAAAAAAACo/yfvLa3Zw7R
          U/s200/logo-uin-suka-baru-warna.jpg">
          </div>
          </td>
          </tr>
          </table>
      </div>
    <li class="span4">
      <div class="thumbnail">
        <!-- <img src="gambar/2.jpg" alt=""> -->
        <div class="caption">
          <h2>Nginx Status</h2>
          %nginxmode%
        </div>
      </div>
    </li>
  </ul>
</div>
<div class="row-fluid">
  <ul class="thumbnails">
    <li class="span12">
      <div class="thumbnail">
        <!-- <img src="gambar/1.jpg" alt=""> -->
        <div class="caption">
          <h3><center>RULES</center></h3>
            <div class="bs-docs-example">
              %tablerule%
</div>
        </div>
```

```
              </div>
            </li>
          </ul>
        </div>
      </section>
      </div>
    </div> <!-- /container -->
  </div>
</div>
</body>
</html>
```

Nama file : nx_extract.py
Lokasi : Web Server Nginx

```
from ConfigParser import ConfigParser
from twisted.web import http
from twisted.internet import protocol
from twisted.internet import reactor, threads
from ordereddict import OrderedDict # don't lose compatibility
with python < 2.7

import MySQLdb
import MySQLConnector
import pprint
import re
import getopt
import sys
import datetime
import time
import cgi
import os


glob_allow=True
glob_rules_file="/etc/nginx/naxsi_core.rules"
glob_conf_file = ''
glob_username = ''
glob_pass = ''
glob_fileList = []

class rules_extractor(object):
   def __init__(self, page_hit, rules_hit, rules_file,
conf_file='naxsi-ui.conf'):
      self.db =
MySQLConnector.MySQLConnector(glob_conf_file).connect()
      self.cursor = self.db.cursor(MySQLdb.cursors.DictCursor)
      self.rules_list = []
      self.final_rules = []
      self.base_rules = []
      self.page_hit = page_hit
      self.rules_hit = rules_hit
      self.core_msg = {}
      self.extract_core(glob_rules_file)
   def extract_core(self, rules_file):
```

```
      try:
         fd = open(glob_rules_file, 'r')
         for i in fd:
            if i.startswith('MainRule'):
               pos = i.find('id:')
               pos_msg = i.find('msg:')
               self.core_msg[i[pos + 3:i[pos + 3].find(';') - 1]]
= i[pos_msg + 4:][:i[pos_msg + 4:].find('"')]
         fd.close()
      except:
         pass

   def gen_basic_rules(self,url=None, srcip=None, dsthost=None,
                rule_id=None, exception_md5=None,
                exception_id=None):
      tmp_rules = []
      self.cursor.execute("""select exception.exception_id as id,
exception.md5 as md5, exception.url as url, exception.count as
count, srcpeer.peer_ip as src, count(distinct srcpeer.peer_ip) as
cnt_peer, dstpeer.peer_host as dst, GROUP_CONCAT(distinct "mz:",
match_zone.rule_id, ":", "$", match_zone.zone, "_VAR:",
match_zone.arg_name)  as match_zones from exception LEFT JOIN
(peer as srcpeer, peer as dstpeer, connections, match_zone)  on
(connections.src_peer_id = srcpeer.peer_id and
connections.dst_peer_id = dstpeer.peer_id and
connections.exception_id = exception.exception_id and
match_zone.exception_id = exception.exception_id) GROUP BY id;""")
      data = self.cursor.fetchall()
      for row in data:
         if (url is not None and not re.search(url, row.get("url",
"""))):
            continue
         if (srcip is not None and not re.search(srcip,
row.get("src", ""))):
            continue
         if (dsthost is not None and not re.search(dsthost,
row.get("dst", ""))):
            continue
         if (exception_md5 is not None and not
re.search(exception_md5, row.get("md5", ""))):
            continue
         tmp_rules.append(row)
      for i in tmp_rules:
         if i['match_zones'] is None:
            continue
         for j in i['match_zones'].split(','):
            if len(j.split(':')) < 2:
               continue
            da_dict = {}
            da_dict['url'] = i['url']
            da_dict['arg'] = ':'.join(j.split(':')[2:])
            # fix exception of URL
            da_dict['arg'] = da_dict['arg'].replace("$URL_VAR:",
"URL")
            da_dict['id'] = j.split(':')[1]
            da_dict['count'] = i['count']
```

```
            da_dict['cnt_peer'] = i['cnt_peer']
            if da_dict not in self.rules_list:
               self.rules_list.append(da_dict)
      self.base_rules = self.rules_list[:]

   def opti_rules_back(self):
      lr = len(self.rules_list)
      i = 0
      while i < lr:
         matching = []
         if (len(self.rules_list[i]['arg'].split(':')) > 1):
            arg_type, arg_name =
tuple(self.rules_list[i]['arg'].split(':'))
         else:
            # Rules targeting URL zone
            if self.rules_list[i]['arg'] == "URL":
               arg_name = ""
               arg_type = "URL"
            # Internal rules have small IDs
            elif self.rules_list[i]['id'] < 10:
               arg_name = ""
               arg_type = ""
         id = self.rules_list[i]['id']
         url = self.rules_list[i]['url']
         matching = filter(lambda l: (l['arg'] == arg_type + ':' +
arg_name) and id == l['id'] , self.rules_list)
         if len(matching) >= self.page_hit:
            #whitelist the ids on every url with arg_name and
arg_type -> BasicRule wl:id "mz:argtype:argname"
            self.final_rules.append({'url': None, 'id': id, 'arg':
arg_type + ':' + arg_name})
            for bla in matching:
               self.rules_list.remove(bla)
            lr -= len(matching)
            i = 0
            print "*) "+str(len(matching))+" hits for same
mz:"+arg_type+':'+arg_name+" and id:"+str(id)
            print "removed "+str(len(matching))+" items from
biglist, now :"+str(len(self.rules_list))
            continue
         matching = filter(lambda l: url == l['url'] and l['arg']
== arg_type + ':' + arg_name, self.rules_list)
         if len(matching) >= self.rules_hit:
            #whitelist all id on url with arg_name and arg_type ->
BasicRule wl:0 "mz:$url:xxx|argtype:argname"
            self.final_rules.append({'url': url, 'id': str(0),
'arg': arg_type + ':' + arg_name})
            print "about to del "+str(len(matching))+" items from
biglist, now :"+str(len(self.rules_list))
            for bla in matching:
               self.rules_list.remove(bla)
            lr -= len(matching)
            i = 0
            print "*) "+str(len(matching))+" hits for same
mz:"+str(url)+'|'+str(arg_type)+':'+str(arg_name)+" and
id:"+str(id)
```

```
            print "removed "+str(len(matching))+" items from
biglist, now :"+str(len(self.rules_list))
            print " current LR:"+str(lr)
            continue
         i += 1
      if self.rules_list == self.final_rules:
         return self.base_rules, self.final_rules
      #append rules that cant be optimized
      self.final_rules += self.rules_list
      #remove duplicate
      tmp_list = []
      for i in self.final_rules:
         if i not in tmp_list:
            tmp_list.append(i)
      self.final_rules = tmp_list
     #try to reoptimize
      self.rules_list = self.final_rules
      self.opti_rules_back()
      return self.base_rules, self.final_rules

   def generate_stats(self):
      stats = ""
      self.cursor.execute("select count(distinct md5) as
uniq_exception from exception")
      uniq_ex = self.cursor.fetchall()[0]['uniq_exception']
      self.cursor.execute("select count(distinct peer_ip) as
uniq_peer from peer where peer_ip is not NULL")
      uniq_peer = self.cursor.fetchall()[0]['uniq_peer']
      self.cursor.execute("select count(distinct peer_ip) as
uniq_peer_mon from http_monitor where peer_ip is not NULL")
      uniq_peer_mon = self.cursor.fetchall()[0]['uniq_peer_mon']
      self.cursor.execute("select count(distinct md5) as
uniq_exception_mon from http_monitor where md5 is not NULL")
      uniq_exception =
self.cursor.fetchall()[0]['uniq_exception_mon']
      return "<ul><li>There is currently %s unique
exceptions.</li></ul><ul><li>There is currently %s different peers
that triggered rules.</li></ul><ul><li>There is currently %s peers
being monitored</li></ul><ul><li>There is currently %s exceptions
being monitored</li></ul>" % (uniq_ex, uniq_peer, uniq_peer_mon,
uniq_exception)

   def parserule(self):
      file = open('/etc/nginx/naxsi_core.rules','r')
      rules = {}
      rule_table = ''
      key = -1

      for i in file:
         if i.startswith("MainRule"):
            rule = {}
            pos = i.find('id:')
            pos_msg = i.find('msg:')
            pos_mz = i.find('mz:')
            pos_s = i.find('s:')
            pos_str = i.find('str:')
```

```
            pos_rx = i.find('rx:')
            id = i[pos + 3:i[pos + 3].find(';') - 1]
            rule['id'] = id
            msg = i[pos_msg + 4:][:i[pos_msg + 4:].find('"')]
            rule['msg'] = msg
            mz = i[pos_mz + 3:][:i[pos_mz + 3:].find('"')]
            rule['mz'] = mz
            s = i[pos_s + 3:][:i[pos_s + 3:].find('"')]
            s.split(':')
            type = s.split(':')[0]
            rule['type'] = type
            value = s.split(':')[-1]
            rule['value'] = value
            if pos_str >= 0:
               str = i[pos_str + 4:][:i[pos_str + 4:].find('"')]
               rule['str'] = str
            elif pos_rx >= 0:
               rx = i[pos_rx + 3:][:i[pos_rx + 3:].find('"')]
               rule['str'] = rx
            key += 1
            rules[key] = rule

      rule_table += '<table class="table table-bordered">'
      rule_table += '<thead>'
      rule_table += '<tr>'
      rule_table += '<th>ID</th>'
      rule_table += '<th>Type</th>'
      rule_table += '<th>Value</th>'
      rule_table += '<th>String</th>'
      rule_table += '<th>Comment</th>'
      rule_table += '<th>Location</th>'
      rule_table += '</tr>'
      rule_table += '</thead>'
      rule_table += '<tbody>'
      for i in rules:
         rule_table += '<tr><td> %s </td> <td> %s </td> <td> %s
</td> <td> %s </td> <td> %s </td> <td> %s </td></tr>' %
(rules[i]['id'], rules[i]['type'], rules[i]['value'],
rules[i]['str'], rules[i]['msg'], rules[i]['mz'])
      rule_table += '</tbody>'
      rule_table += '</table>'

      return rule_table
      file.close()

   def viewstatus(self):
      file = open('/etc/nginx/naxsi.rules','r')
      status = ''

      for i in file:
         if i.find('LearningMode') >= 0:
            if i.startswith("#"):
               status += '<div class="alert alert-success">Naxsi
On !! <br /></div> <a onClick = "changestatus()" class="btn btn-
danger">Disable</a>'
            else:
```

```
               status += '<div class="alert alert-error">Naxsi Off
!! <br /></div> <a onClick = "changestatus()" class="btn btn-
success">Enable</a>'
      return status
      file.close()

   def changestatus(self):
      file = open('/etc/nginx/naxsi.rules','r')
      lines = []
      for i in file:
         if i.find('LearningMode') >= 0:
            if i.startswith("#"):
               i = i.replace("#LearningMode", "LearningMode")
            else:
               i = i.replace("LearningMode", "#LearningMode")
         lines.append(i)
      file.close()
      file = open('/etc/nginx/naxsi.rules','w')
      file.writelines(lines)
      file.close()

   def nginxstatus(self):
      import commands
      status = ''
      output = commands.getoutput('ps -A')
      if 'nginx' in output:
         status += '<div class="alert alert-success">Nginx is
running! <br /></div><a onClick = "restartnginx()" class="btn btn-
primary">Restart</a>'
      else:
         status += '<div class="alert alert-error">Nginx is not
running! <br /></div>'
      return status

   def restartnginx(self):
      import subprocess

      command = ['service', 'nginx', 'restart'];
      subprocess.call(command, shell=False)


class InterceptHandler(http.Request):
   def create_js_array(self, res):
      array = '['
      for i in res:
         date_begin = str(i).split('-')
         date_begin[1] = str(int(date_begin[1]) - 1)
         date_begin = ','.join(date_begin)
         array += '[Date.UTC(' + date_begin  + '),' + str(res[i])
+ '],'
      if array != '[':
         array = array[:-1] + ']'
      else:
         array += ']'
      return array
```

```
   def build_dict(self, res):
      d = OrderedDict()
      for i in res:
         if i['d'] not in d.keys():
            d[i['d']] = i['ex']
      return d


   def build_js_array(self, id_beg = None, id_end = None):
      if id_beg is None or id_end is None:
         self.ex.cursor.execute('select date(date) as d,
count(exception_id) as ex from connections group by date(date)')
      else:
         self.ex.cursor.execute('select date(date) as d,
count(co.exception_id) as ex from connections as co join
match_zone as m on (co.match_id = m.match_id) where m.rule_id >=
%s and m.rule_id <= %s group by date(date);', (str(id_beg),
str(id_end)))
      count = self.ex.cursor.fetchall()
      mydict = self.build_dict(count)
      total_hit = 0
      for i in count:
         total_hit += i['ex']
      myarray = self.create_js_array(mydict)
      return myarray, total_hit

   def check_auth(self):
      user = self.getUser()
      passwd = self.getPassword()

      if user != glob_user or passwd != glob_pass:
         self.setResponseCode(401)
         self.setHeader('WWW-Authenticate', 'Basic realm="NAXSI
Web Interface"')
         self.setHeader('content-type', 'text/html')
         self.write('<h1>Unauthorized User</h1>')
         self.finish()
         return -1

      return 42

   def handle_request(self):

      if self.check_auth() == -1:
         return

      self.ex = rules_extractor(0,0, None)

      if self.path == '/get_rules':
         self.setHeader('content-type', 'text/plain')
         ex = rules_extractor(int(self.args.get('page_hit',
['10'])[0]),
                              int(self.args.get('rules_hit',
['10'])[0]),
                              glob_rules_file)
         ex.gen_basic_rules()
```

```
         base_rules, opti_rules = ex.opti_rules_back()
         r = '############ Rules Before Optimisation
###################\n'

         for i in base_rules:
            r += '#%s hits on rule %s (%s) on url %s from %s
different peers\n' % (i['count'], i['id'],

ex.core_msg.get(i['id'],

'Unknown id. Check the path to the core rules file and/or the
content.'),

i['url'], i['cnt_peer'])
            r += '#BasicRule wl:' + i['id'] + ' "mz:$URL:' +
i['url']
            if '|NAME' in i['arg']:
               i['arg'] = i['arg'].split('|')[0] + '_VAR|NAME'
            if i['arg'] is not None and len(i['arg']) > 0:
               r += '|' + i['arg']
            r +=  '";\n'
         r += '############ End Of Rules Before Optimisation
############\n'

         for i in opti_rules:
            r += 'BasicRule wl:' + i['id'] + ' "mz:'
            if i['url'] is not None and len(i['url']) > 0:
               r += '$URL:' + i['url']
            if i['arg'] is not None and len(i['arg']) > 0:
               if i['url'] is not None and len(i['url']):
                  r += '|'+i['arg']
               else:
                  r += i['arg']
            r += '";\n'

         self.write(r)
         self.finish()

      elif self.path == '/':
         fd = open('index.tpl', 'r')
         helpmsg = ''
         for i in fd:
            helpmsg += i
         fd.close()
         helpmsg = helpmsg.replace('__STATS__',
self.ex.generate_stats())
         helpmsg = helpmsg.replace('__HOSTNAME__',
self.getHeader('Host'))
         helpmsg = helpmsg.replace('%tablerule%',
self.ex.parserule())
         helpmsg = helpmsg.replace('%learningmode%',
self.ex.viewstatus())
         helpmsg = helpmsg.replace('%nginxmode%',
self.ex.nginxstatus())
         self.setHeader('content-type', 'text/html')
         self.write(helpmsg)
```

```
            self.finish()

        elif self.path == '/changestatus':
            self.ex.changestatus()
            self.finish()

        elif self.path == '/restartnginx':
            self.ex.restartnginx()
            self.finish()

        elif self.path == '/graphs':
            fd = open('graphs.tpl')
            html = ''
            for i in fd:
                html += i
            fd.close()

            array_excep, _ = self.build_js_array()
            sqli_array, sql_count = self.build_js_array(1000, 1099)
            xss_array, xss_count = self.build_js_array(1300, 1399)
            rfi_array, rfi_count = self.build_js_array(1100, 1199)
            upload_array, upload_count = self.build_js_array(1500,
1599)
            dt_array, dt_count = self.build_js_array(1200, 1299)
            evade_array, evade_count = self.build_js_array(1400,
1499)
            intern_array, intern_count = self.build_js_array(0, 10)

            self.ex.cursor.execute('select p.peer_ip as ip,
count(exception_id) as c from connections join peer as p on
(src_peer_id = p.peer_id) group by p.peer_ip order by
count(distinct exception_id) DESC limit 10;')
            top_ten = self.ex.cursor.fetchall()
            top_ten_html = '<table class="table table-bordered"
border="1" ><thead><tr><th>IP</th><th>Rule
Hits</th></tr></thead><tbody>'
            for i in top_ten:
                top_ten_html += '<tr><td>' + cgi.escape(i['ip']) + '
</td><td> ' + str(i['c']) + '</td></tr>'
            top_ten_html += '</tbody></table>'

            self.ex.cursor.execute('select distinct url,
count(exception_id) as c from exception  group by url order by
count(exception_id) DESC limit 10;')
            top_ten_page = self.ex.cursor.fetchall()
            top_ten_page_html = '<table class="table table-bordered"
border="1" ><thead><tr><th>URI</th><th>Exceptions
Count</th></tr></thead><tbody>'

            for i in top_ten_page:
                top_ten_page_html += '<tr><td>' + cgi.escape(i['url'])
+ ' </td><td> ' + str(i['c']) + '</td></tr>'
            top_ten_page_html += '</tbody></table>'

            dict_replace = {'__TOPTEN__': top_ten_html,
'__TOPTENPAGE__': top_ten_page_html, '__TOTALEXCEP__':
```

```
array_excep, '__SQLCOUNT__': str(sql_count),  '__XSSCOUNT__':
str(xss_count), '__DTCOUNT__': str(dt_count), '__RFICOUNT__':
str(rfi_count), '__EVCOUNT__': str(evade_count), '__UPCOUNT__':
str(upload_count), '__INTCOUNT__': str(intern_count),
'__SQLIEXCEP__': sqli_array, '__XSSEXCEP__': xss_array,
'__RFIEXCEP__': rfi_array, '__DTEXCEP__': dt_array,
'__UPLOADEXCEP__': upload_array, '__EVADEEXCEP__': evade_array,
'__INTERNEXCEP__': intern_array}

         html = reduce(lambda html,(b, c): html.replace(b, c),
dict_replace.items(), html)
         self.write(html)
         self.finish()

      else:
         try:
            if self.path.endswith('.js'):
               self.setHeader('content-type', 'text/javascript')

            if '.' + self.path not in glob_fileList:
               self.setResponseCode(403)
               self.finish()
               return

            fd = open(self.path[1:], 'rb')
            for i in fd:
               self.write(i)
            fd.close()
         except IOError, e:
            pass
         self.finish()

   def process(self):
      threads.deferToThread(self.handle_request)

class InterceptProtocol(http.HTTPChannel):
   requestFactory = InterceptHandler

class InterceptFactory(http.HTTPFactory):
   protocol = InterceptProtocol

def usage():
   print 'Usage : python nx_extract /path/to/conf/file'

def build_file_list(path):
   rootdir = path
   for root, subFolders, files in os.walk(rootdir):
      for file in files:
         glob_fileList.append(os.path.join(root,file))


if __name__  == '__main__':
   if len(sys.argv) != 2:
      usage()
      exit(42)
   glob_conf_file = sys.argv[1]
```

```
   fd = open(sys.argv[1], 'r')
   conf = ConfigParser()
   conf.readfp(fd)
   try:
      port = int(conf.get('nx_extract', 'port'))
   except:
      print "No port in conf file ! Using default port (8081)"
      port = 8081
   try:
      glob_rules_file = conf.get('nx_extract', 'rules_path')
   except:
      print "No rules path in conf file ! Using default
(/etc/nginx/sec-rules/core.rules)"

   try:
      glob_user = conf.get('nx_extract', 'username')
   except:
      print 'No username for web access ! Nx_extract will exit.'
      exit(-1)

   try:
      glob_pass = conf.get('nx_extract', 'password')
   except:
      print 'No password for web access ! Nx_extract will exit.'
      exit(-1)
   fd.close()


   build_file_list('.')

   reactor.listenTCP(port, InterceptFactory())
   reactor.run()
```

LAMPIRAN D

LEMBAR KUISIONER HASIL PENGUJIAN SISTEM

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Bambang Arditya fitri Sulistyanto A.md.

Keterangan : Karyawan CV Ganesha Muda Architect

Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	✓	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	✓	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	✓	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	✓	

Saran :

Yogyakarta , 14 Juli 2019

Penguji

Bambang Arditya. FS

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Bambang Aditya Fitri Sulistyanto A.md

Keterangan : Karyawan CV Ganesha Muda Architect

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?		✓		
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			

Saran :

Yogyakarta , 14 Juli 2014

Penguji

Bambang Aditya F-S

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Setiya Budi

Keterangan : Mahasiswa UIN'07 / Alumni

Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	√	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	√	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	√	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	√	

Saran :

Yogyakarta , 10 Juli 2014

Penguji

Setiya Budi

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Setiya Budi

Keterangan : Mahasiswa UIN '07 / Alumni

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?		✓		
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?		✓		
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?		✓		

Saran :

Yogyakarta , 10 Juli 2014

Penguji

Setiya Budi

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : M. Husna Mubarok

Keterangan : Praktisi Jaringan

Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	V	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	V	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	V	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	V	

Saran :

Yogyakarta , 13 Juli 2014

Penguji

Mubarok

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : M Husna Mubarok

Keterangan : Praktisi Jaringan

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?		V		
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?		V		
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?		V		

Saran :

Yogyakarta, 13 Juli 2019

Penguji

Mubarok

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Rasyid YS

Keterangan : Mahasiswa

Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	✓	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	✓	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	✓	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	✓	

Saran :

Yogyakarta, 10 Juli 2014

Penguji

Rasyid

_____

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Rayyd Ys

Keterangan : Mahasiswa

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?	✓			
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			

Saran :

Yogyakarta, 10 Juli 2019

Penguji

Rayyd

_____

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Indras Woro Widodo

Keterangan : Member IBT Chapter Jogja

Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	V	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	V	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	V	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	V	

Saran :

Yogyakarta , 11 Juli 2014

Penguji

Indras Woro Widodo

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Indras Woro Widodo

Keterangan : Member IBT Chapter Jogja

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?		✓		
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			

Saran :

Yogyakarta, 11 Juli 2014

Penguji

Indras Woro Widodo

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Rahmat Nur Faizin

Keterangan : Mahasiswa

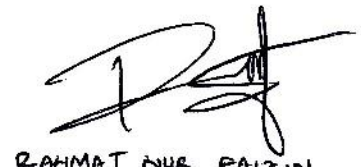
Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	V	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	V	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	V	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	V	

Saran :

Yogyakarta , 10 JULI 2019

Penguji



RAHMAT NUR FAIZIN

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Rahmat Nur Fazin

Keterangan : Mahasiswa

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?		✓		
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			

Saran :

Yogyakarta , 10 Juli 2019

Penguji

RAHMAT NUR FA IZ IN

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : ARYA ERVAN L

Keterangan : MAHASISWA

Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	✓	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	✓	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	✓	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	✓	

Saran :

Yogyakarta , 10 JULI 2014

Penguji

ARYA ERVAN L

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : ARYA ERVAN LEORESTA

Keterangan : MAHASISWA

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?	✓			
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			

Saran :

Yogyakarta , 10 JULI 2014

Penguji

ARYA ERVANL

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Hana Saffa

Keterangan : Mahasiswa

Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	✓	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	✓	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	✓	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	✓	

Saran :

Yogyakarta , 10 Juli 2019

Penguji

Hana Saffa

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Irma Soffa

Keterangan : Mahasiswa

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?	✓			
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			

Saran :

Yogyakarta , 10 Juli 2019

Penguji

Irma Soffa

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Syud Arifin

Keterangan : Mahasiswa

Tabel Pengujian Penetration Testing

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian firewall aplikasi web naxsi berhasil memblok syntax yang telah masuk ke dalam daftar blacklist ?	V	
2	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan yang menggunakan tool SQL Injection ?	V	
3	Apakah skenario pengujian firewall aplikasi web naxsi dapat mengatasi serangan SQL Injection yang dilakukan secara manual ?	V	
4	Apakah tombol yang berada pada modul naxsi dapat berfungsi dengan lancar ?	V	

Saran :

Yogyakarta, 11 Juli 2019

Penguji

Syud Arifin

LEMBARAN ANGKET PENGUJIAN IMPLEMENTASI FIREWALL APLIKASI WEB UNTUK MENCEGAH SQL INJECTION MENGGUNAKAN NAXSI

Nama : Syjud Arifin

Keterangan : Mahasiswa

Tabel Pengujian Baseline Performance Measurement

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah implementasi firewall aplikasi web naxsi tidak mempengaruhi performa web server nginx ?		✓		
2	Apakah pengujian request per second bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			
3	Apakah pengujian response time bisa sebagai tolak ukur dalam mengetahui perbandingan performa web server nginx sebelum dan sesudah implementasi firewall naxsi ?	✓			

Saran :

Yogyakarta , 11 Juli 2014

Penguji

Syjud Arifin

DAFTAR RIWAYAT HIDUP

Nama	: Feri Setiyawan
Tempat, tanggal lahir	: Yogyakarta, 21 Mei 1992
Jenis Kelamin	: Laki-laki
Agama	: Islam
Alamat Asal	: Ringinsari RT01 RW49 Maguwoharjo, Depok, Sleman
No. HP	: 083869422092
Email	: ferise92@gmail.com

Riwayat Pendidikan :

1. SD Negeri Maguwoharjo 1 (1998-2004)
2. SMP Negeri 3 Depok (2004-2007)
3. SMA Negeri 2 Ngaglik (2007-2010)
4. S1 Teknik Informatika UIN Sunan Kalijaga Yogyakarta (2010-2014)