



## Evaluation d'un reverse proxy en sécurité par défaut Nginx + Naxsi

Mathieu Tham

Université Toulouse 1 Capitole

Mardi 1<sup>er</sup> Juillet 2014



## Contexte

- Stage de deuxième année de DUT informatique
- Pendant 2 mois et demi
- Université Toulouse 1 Capitole DSI : Service Système
- Etude d'un reverse proxy





## C'est quoi NAXSI ?

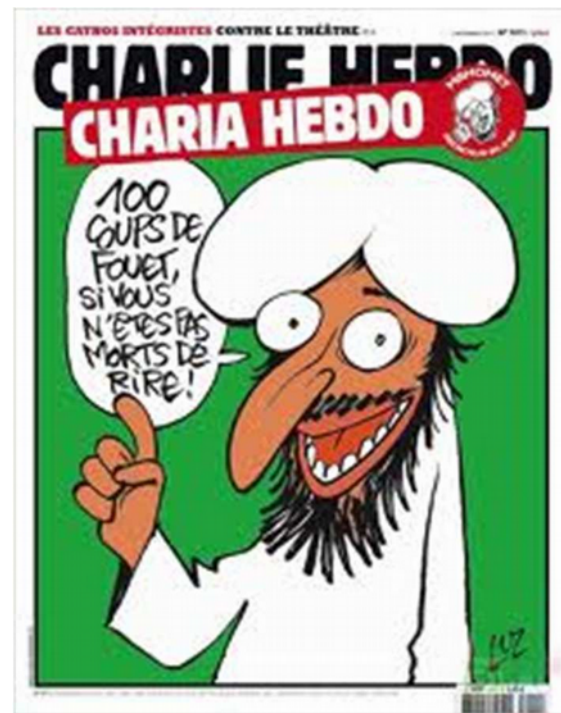
- WAF Open Source
- Idée de Thibault Koechlin alias buixor en avril 2011
- Projet officiel OWASP
- Financé par NBS-System
- Module du serveur web/reverse proxy Nginx





## NAXSI : Le cas Charlie-Hebdo

- Site défacé plusieurs fois
- Intervention de NBS-System
- 1 jour plus tard le site est remis sur pied
- Baptême du feu pour Naxsi
- A cette époque, 80% du trafic était malveillant





## Serveur Web Apache + Dokuwiki

- Machine UNIX Fedora
- Packages Httpd et dokuwiki
- Objectif : Protéger cette machine en mettant en place Nginx et Naxsi en reverse proxy





## Reverse proxy Nginx

- Un package existe sous Fedora mais impossible d'installer Naxsi ensuite
- Installation par les sources sur la même machine que Apache
- Puis Installation sur machine différente
- Sous Debian et Ubuntu, un paquet nginx-naxsi existe



## Reverse proxy Nginx

- Commande de compilation de Nginx avec Naxsi :

```
./configure -conf-path=/etc/nginx/nginx.conf  
-add-module=../naxsi-x.xx/naxsi_src/  
-error-log-path=/var/log/nginx/error.log  
-http-client-body-temp-path=/var/lib/nginx/body  
-http-fastcgi-temp-path=/var/lib/nginx/fastcgi  
-http-log-path=/var/log/nginx/access.log  
-http-proxy-temp-path=/var/lib/nginx/proxy  
-lock-path=/var/lock/nginx.lock  
-pid-path=/var/run/nginx.pid  
-with-http_ssl_module  
-without-mail_pop3_module  
-without-mail_smtp_module  
-without-mail_imap_module  
-without-http_uwsgi_module  
-without-http_scgi_module  
-with-ipv6 -prefix=/usr
```



## NAXSI : WAF open source

- Nginx Anti XSS and SQL Injection
- La plupart des WAF tiennent plus de l'antivirus que du pare-feu
- Base importante de signatures qui doivent être mises à jour régulièrement
- NAXSI est un WAF qui ne requiert pas de mise à jour de signatures, seulement de la reconfiguration





## NAXSI : sans Whitelist



### Installeur DokuWiki

Nom du wiki

Activer les ACL (recommandé)

Super-utilisateur

Nom

Adresse de courriel

Mot de passe

Répéter nouveau mot de passe

Politique d'ACL initiale

Wiki ouvert (lecture, écriture, envoi de fichiers pour tout le monde)

Veillez choisir la licence sous laquelle placer votre contenu :

- None
- CC0 1.0 Universal <sup>[?]</sup>
- Public Domain <sup>[?]</sup>
- CC Attribution 3.0 Unported <sup>[?]</sup>
- CC Attribution-Share Alike 3.0 Unported <sup>[?]</sup>
- GNU Free Documentation License 1.3 <sup>[?]</sup>
- CC Attribution-Noncommercial 3.0 Unported <sup>[?]</sup>
- CC Attribution-Noncommercial-Share Alike 3.0 Unported <sup>[?]</sup>

Enregistrer



Choisissez votre langue:

Cette page vous assiste dans la première installation et la configuration de [DokuWiki](#). Pour plus d'information sur cet installeur, reportez-vous à sa [page de documentation](#).

DokuWiki utilise des fichiers textes ordinaires pour stocker les pages du wiki et les autres informations associées à ces pages (tel que images, index de recherche, anciennes révisions, etc.). Pour fonctionner correctement, DokuWiki **doit** avoir accès en écriture aux différents répertoires qui contiennent ces fichiers. L'installeur n'est pas capable de modifier les permissions sur les répertoires. Ceci doit être effectué directement sur la ligne de commande de votre shell, ou, si vous êtes hébergé, via FTP ou votre panneau de contrôle (tel que cPanel).

Cet installeur va paramétrer votre configuration de DokuWiki pour des ACL, qui permettront l'accès à un identifiant administrateur et l'accès au menu d'administration de DokuWiki pour l'ajout de modules externes (greffons), la gestion d'utilisateurs, la gestion de l'accès aux pages du wiki et les modifications des paramètres de configuration. Il n'est pas nécessaire au fonctionnement de DokuWiki, néanmoins il facilite l'administration de DokuWiki.

Les utilisateurs expérimentés ou ceux nécessitant des paramétrages particuliers devraient se reporter aux liens suivants pour les détails concernant les [instructions d'installation](#) et les [paramètres de configuration](#).



# NAXSI : Avec Whitelist

10.26.101.1/dokuwiki/doku.php?id=nginx

S'enregistrer Connexion

Dokuwiki Naxsi

Derniers changements Gestionnaire de médias Index

Piste: • start • injection\_sql • serveur\_apache • tests\_unitaires • naxsi • nginx

## Serveur Nginx

Nginx est un type de serveur http comme Apache. On peut aussi l'utiliser, comme dans notre cas, en tant que reverse proxy. Nous allons voir comment le mettre en place et comment le configurer.

### Installation

**commande d'installation :**

```
$ yum install nginx
```

### Configuration

Modifier le fichier /etc/nginx/nginx.conf :

```
http {
    proxy_pass http://nginx; //(Faux)
}
```

rajouter du code ici

Modifier ensuite le fichier /etc/nginx/sites-enabled/default

#### Table des matières

- Serveur Nginx
- Installation
- Configuration
- Difficultés
- Conclusion



## NAXSI : Règles de bases

- NAXSI repose sur 35 règles ciblant SQLi, XSS, RFI/LFI, file uploads...
- Ressemblant à ceci :

```
MainRule "rx:select|union|update|delete|insert|table|from|ascii|hex|unhex|drop" "msg:sql  
keywords" "mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1000;
```

- Quand une requête atteint un score limite, une action est prise sur la requête
- C'est le fichier `naxsi_core.rules`
- Approche simple et rapide mais on a besoin d'une liste blanche



## NAXSI : Learning Mode

- Apprentissage par un script
- Naxsi ne bloque rien mais les exceptions sont stockées dans les logs
- Permet la génération de la whitelist et des rapports
- 3 versions depuis la création de Naxsi



## NAXSI : Learning Mode

- Fichier de configuration mysite.rules

```
(#)LearningMode; #Enables learning mode '#' pour désactiver  
SecRulesEnabled;  
DeniedUrl "/RequestDenied"; #Là où on rejette les requêtes bloquées  
## check rules  
CheckRule "$SQL >= 8" BLOCK; # Si le score dépasse 8 alors l'exception sera levée.  
CheckRule "$RFI >= 8" BLOCK;  
CheckRule "$TRAVERSAL >= 4" BLOCK;  
CheckRule "$EVADE >= 4" BLOCK;  
CheckRule "$XSS >= 8" BLOCK;
```



## NAXSI : Learning Tool

- Naxsi\_ui (obsolète)
  - composé de nx\_intercept.py et nx\_extract.py
  - Base de données SQLite
- Nx\_util
  - Plus besoin de lancer nx\_intercept.py
  - nx\_extract s'appelle maintenant nx\_util.py
  - Options de filtrage
  - Générations de rapports
- Nxapi
  - Le dernier Learning tool sorti récemment
  - Base de donnée Elasticsearch
  - Génération de whitelist par template



## NAXSI : Syntaxe des règles

- MainRule "rx:select|union|update|delete|insert|table|from|ascii|hex|unhex|drop" "msg:sql keywords" "mz:BODY|URL|ARGS|\$HEADERS\_VAR:Cookie" "s:\$SQL:4" id:1000;
- rx ou str : regex ou string
- msg : Message aidant à la compréhension de la règle
- s : la section du score, si le score dépasse celui autorisé alors la requête se retrouve bloquée
- mz : Match zone : la zone de la requête qui a été inspectée et détectée
- id : ID de la règle de Naxsi. Les ID inférieurs à 1000 sont les règles internes de Naxsi



## NAXSI : Exceptions

- Situées dans votre error.log
- 2013/11/10 07:36:19 [error] 8278#0: \*5932 **NAXSI\_FMT:**  
ip=X.X.X.X&server=Y.Y.Y.Y&uri=/phpMyAdmin-  
2.8.2/scripts/setup.php&learning=0&vers=0.52&total\_processed=472&total\_blocked=2  
04&block=0&cscore0=\$UWA&**score0=8**&zone0=HEADERS&id0=42000227&var\_name0=  
user-agent, client: X.X.X.X, server: blog.memze.ro, request: "GET /phpMyAdmin-  
2.8.2/scripts/setup.php HTTP/1.1", host: "X.X.X.X"





## NAXSI : Exceptions

- Extensives logs :
  - 2013/05/30 20:47:05 [debug] 10804#0: \* 1 **NAXSI\_EXLOG**:  
ip=127.0.0.1&server=127.0.0.1&uri=/&id=1302&zone=ARGS&var\_name=a&content=a<>bcd
  - 2013/05/30 20:47:05 [error] 10804#0: \* 1 **NAXSI\_FMT**:  
ip=127.0.0.1&server=127.0.0.1&uri=/&learning=0&vers=0.50&total\_processed=1&total\_blocked=1&cscore0=\$UWA&score0=8&zone0=ARGS&id0=1302&var\_name0=a, client: 127.0.0.1, server: , request: "GET /?a=a<>bcd HTTP/1.0", host: "127.0.0.1"
- Rajouter « set \$naxsi\_extensive\_log 1; » dans le fichier /etc/nginx/site-enabled/mysite



## NAXSI : Whitelist

- Lancement du script nx\_util.py
- `./nx_util.py -l /répertoire_des_error_logs -o`
- `o` pour sortir la whitelist de la base SQLite net l'afficher en sortie et à copier dans `mysite.rules` ou bien dans un autre fichier.rules
- Exemple de whitelist :

```
# total_count:26 (0.41%), peer_count:1 (33.33%) | parenthesis, probable sql/xss  
BasicRule wl:1011 "mz:$ARGS_VAR:id";
```



## NAXSI : Fichiers

- Les fichiers principaux de Naxsi et de Nginx
  - nginx.conf
  - site-enabled/default
  - conf.d/proxy.conf
  - mysites.rules
  - naxsi\_core.rules
  - votre\_whitelist.rules



## NAXSI : HttpSubModule

- Pour tester Naxsi sur [www.ut-capitole.fr](http://www.ut-capitole.fr) sans gêner les utilisateurs
- Substitution d'URL avec un module de Nginx
- Permet à notre reverse proxy de rediriger sur [wwwp.ut-capitole.fr](http://wwwp.ut-capitole.fr)
- Ajouter « `subs_filter 'www.ut-capitole.fr' 'wwwp.ut-capitole.fr'`  
-g»

dans la configuration de Nginx

- Module disponible également sur Apache



Grrr ! Pourquoi je n'ai pas vu ça avant !!!  
Les heures de travail que j'aurai pu gagner...



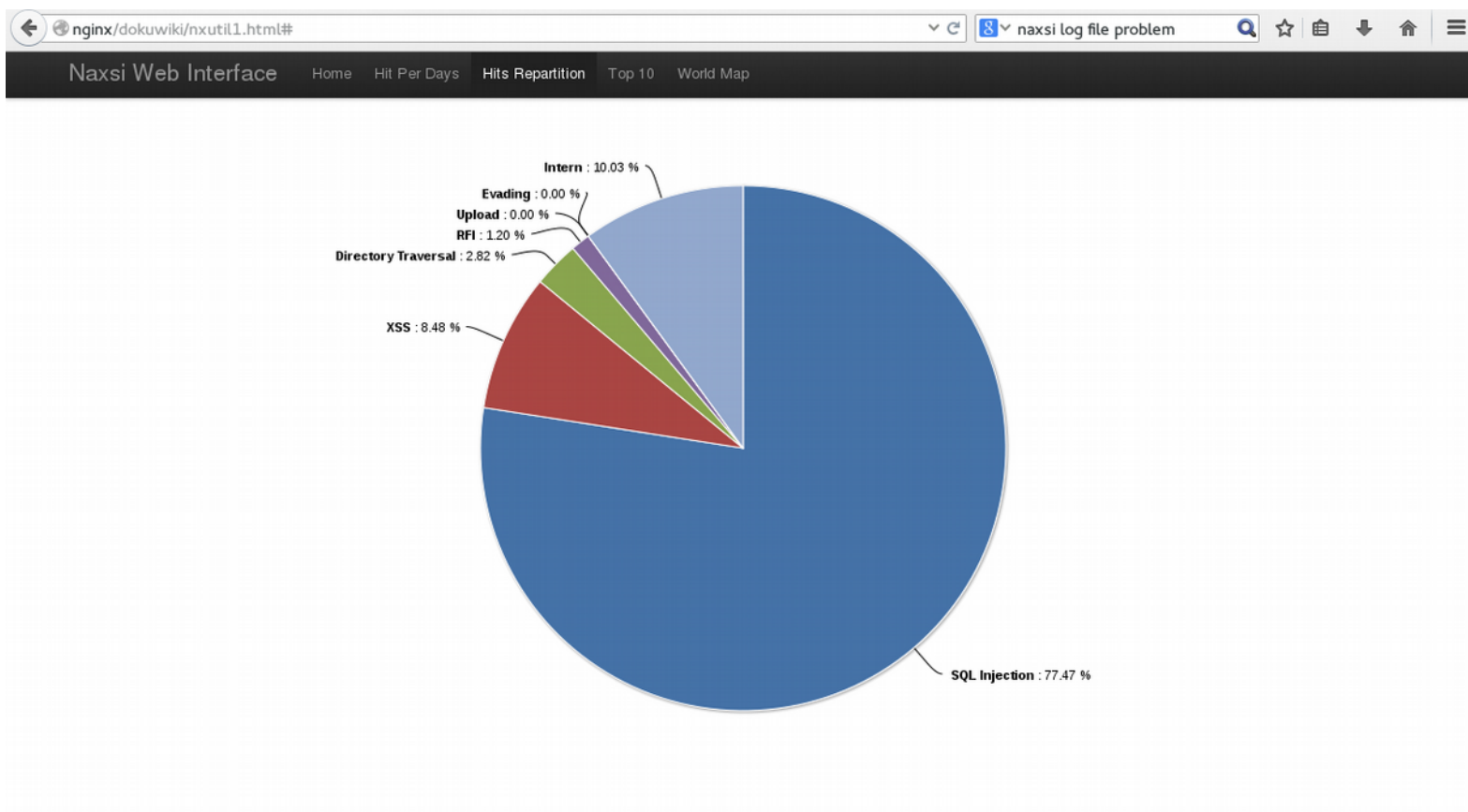
## NAXSI : Rapports

- Lancement du script nx\_util.py
- `./nx_util.py -l /répertoire_des_logs -H /répertoire_du_fichier_html`
- Diagramme, graphique et mapemonde si GeoIP est activé
- Plusieurs filtres disponibles : par date, ip, serveur, uri, zone de la requête, id de l'exception, var\_name, country, contenu, etc...
- Mais le mécanisme de filtre est extrêmement primitif d'après les développeurs donc prudence avec les filtres trop complexes.



## NAXSI : Rapports

### ■ Diagramme :





## NAXSI : Rapports

### ■ Graphique :

Naxsi Web Interface

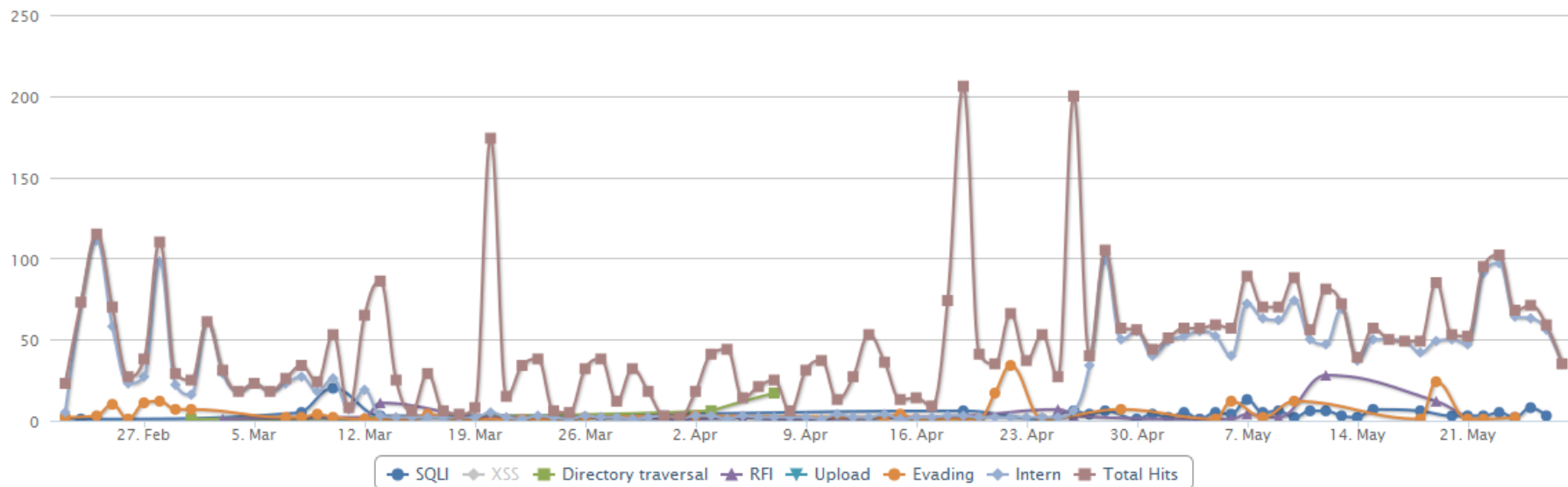
Home

Hit Per Days

Hits Repartition

Top 10

Rules hit per day





## Outils de test

- Nikto
  - Un web scanner qui va tester le plus rapidement possible le serveur pour trouver de simples informations, les versions qui ne sont pas à jour, etc...
- Wapiti
  - Se charge de tester les vulnérabilités d'une applications web en testant plusieurs attaques différentes





## Résultats Dokuwiki

- Nikto
  - il trouve certaines informations. Pas de trace de Naxsi.
- Wapiti
  - Avec Naxsi : Wapiti ne trouve rien
  - Sans Naxsi : Wapiti ne trouve rien également
- La sécurité interne de Dokuwiki ne nous permet pas de voir véritablement Naxsi



# Badstore

**BADSTORE.NET**  
Quick Item Search [View Cart](#)

Welcome to BadStore.net!

[Home](#)  
[What's New](#)  
[Sign Our Guestbook](#)  
[View Previous Orders](#)  
[About Us](#)  
[My Account](#)  
[Login / Register](#)  
[- Suppliers Only -](#)  
[Supplier Login](#)  
[- Reference -](#)  
[BadStore.net Manual v1.2](#)

BadStore v1.2.3s - Copyright © 2004-2006



## Résultats Badstore

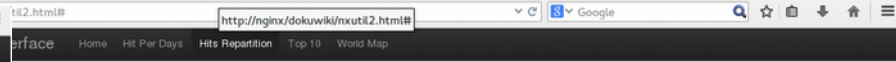
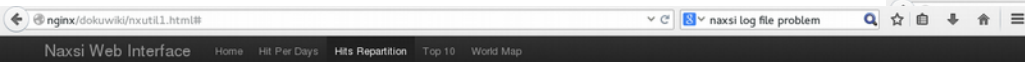
- Virtualbox de Badstore
- Wapiti
  - Sans Naxsi : 19 failles
  - Avec Naxsi : 1 faille de type file upload non exploitable heureusement.



# Résultats Rapports

## Badstore

## Dokuwiki





## Bilan : NAXSI

- Va être mis en place à l'université
- Petit test de performance sur 10000 requêtes
  - Sans Naxsi : 3700 requêtes/s
  - AvecNaxsi : 2700 requêtes/s
  - Sans Modsecurity : 2100 requêtes/s
  - Avec Modsecurity : 1100 requêtes/s
- Syntaxe facile à apprendre
- Protection assurée
- Possibilité de LearningMode continu
- Une étude comparative Naxsi / ModSecurity serait un plus