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

Mathieu Tham

Université Toulouse 1 Capitole

Mardi 1er Juillet 2014

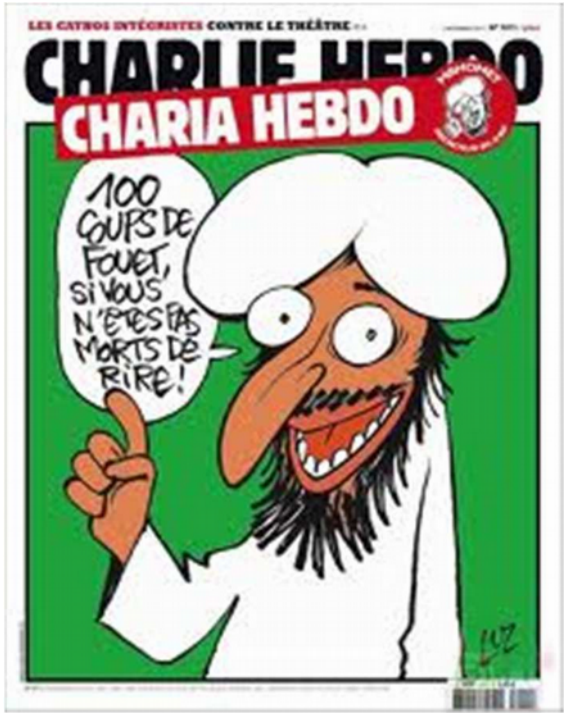
Contexte

- 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
- Baptème 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

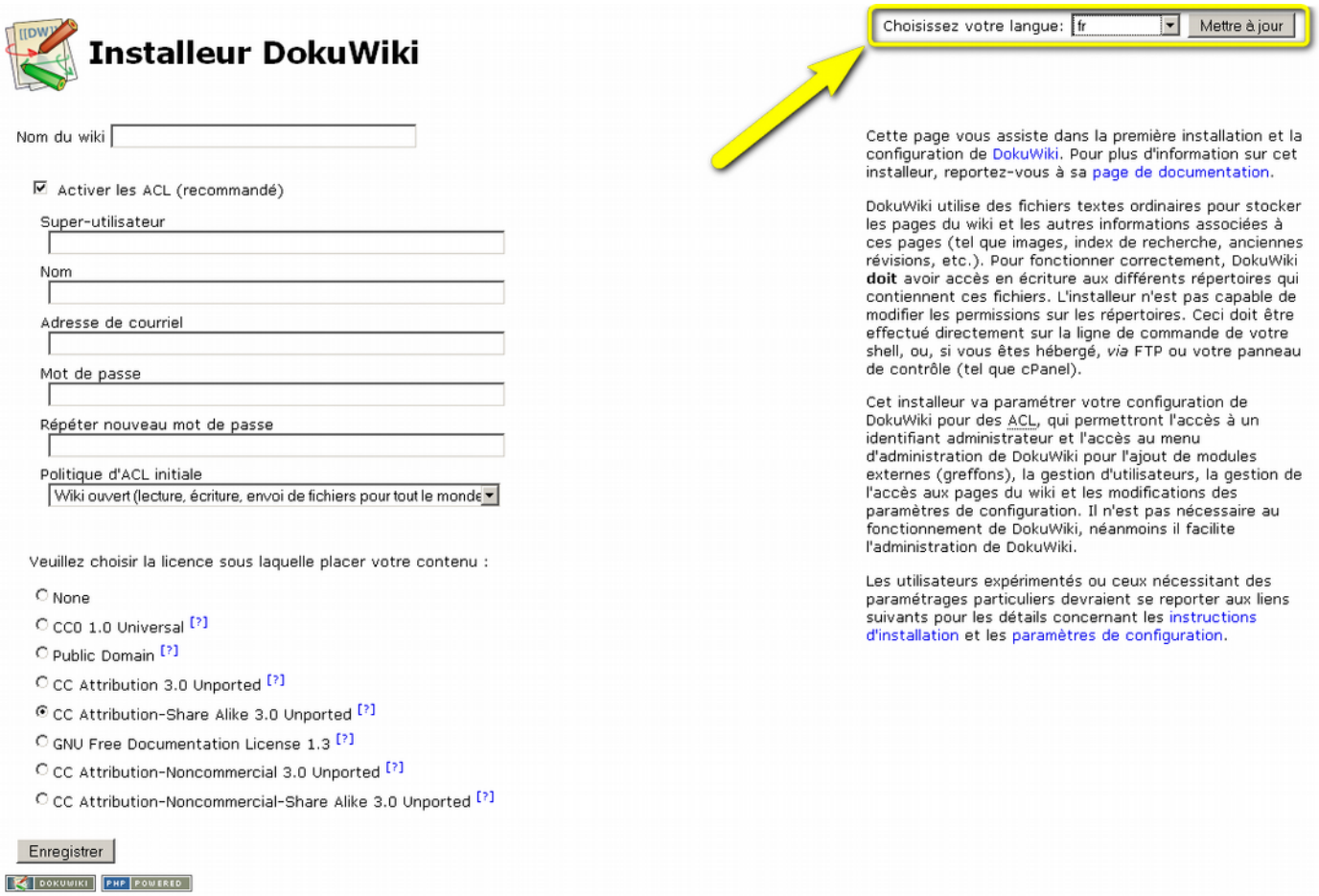
- Commande de compilation de Nginx avec Naxsi :

./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

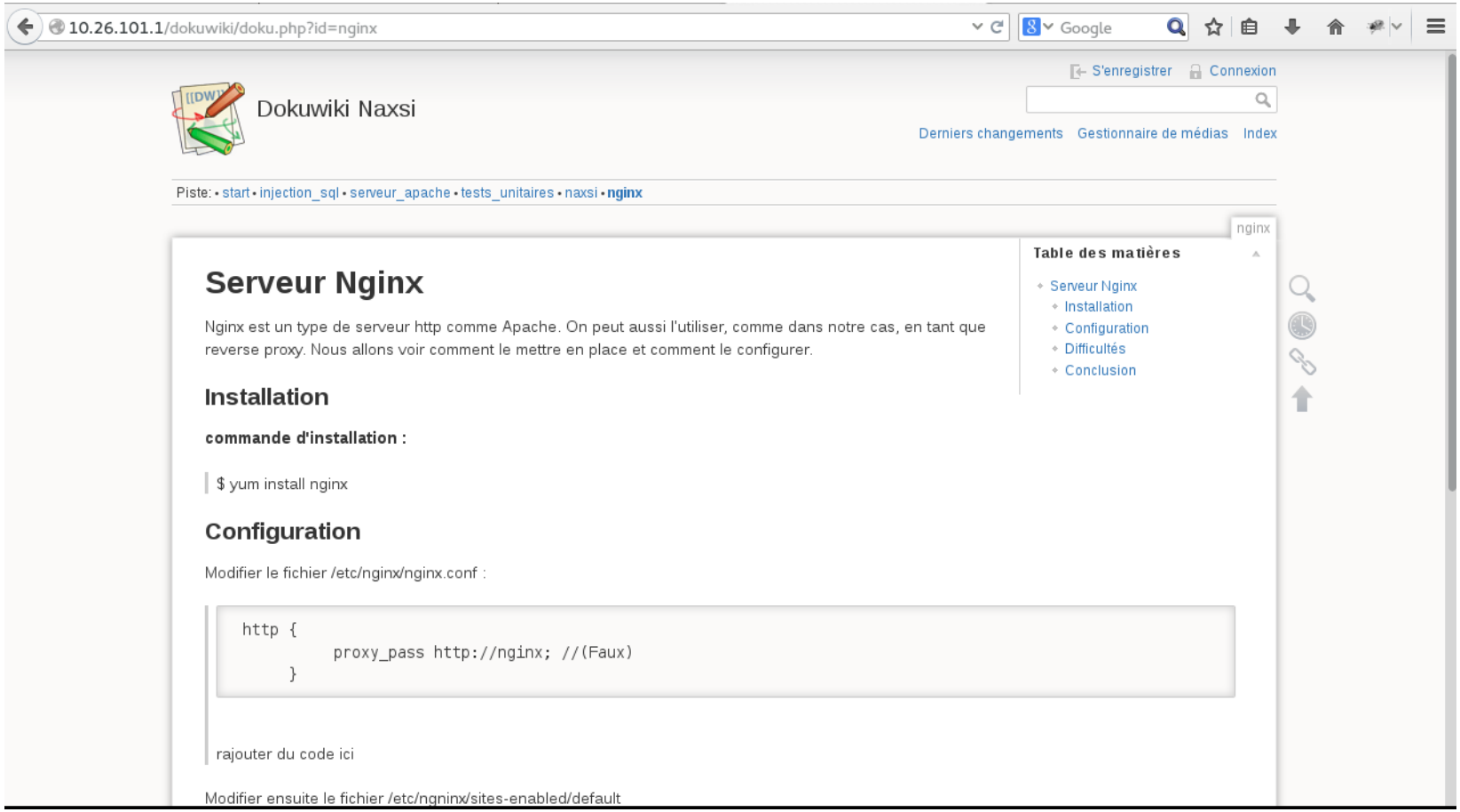
NAXSI : WAF open source

- 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



NAXSI : Avec Whitelist



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 récement
  - 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=204&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 sans gêner les utilisateurs
- Substitution d'URL avec un module de Nginx
- Permet à notre reverse proxy de rediriger sur www**p**.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

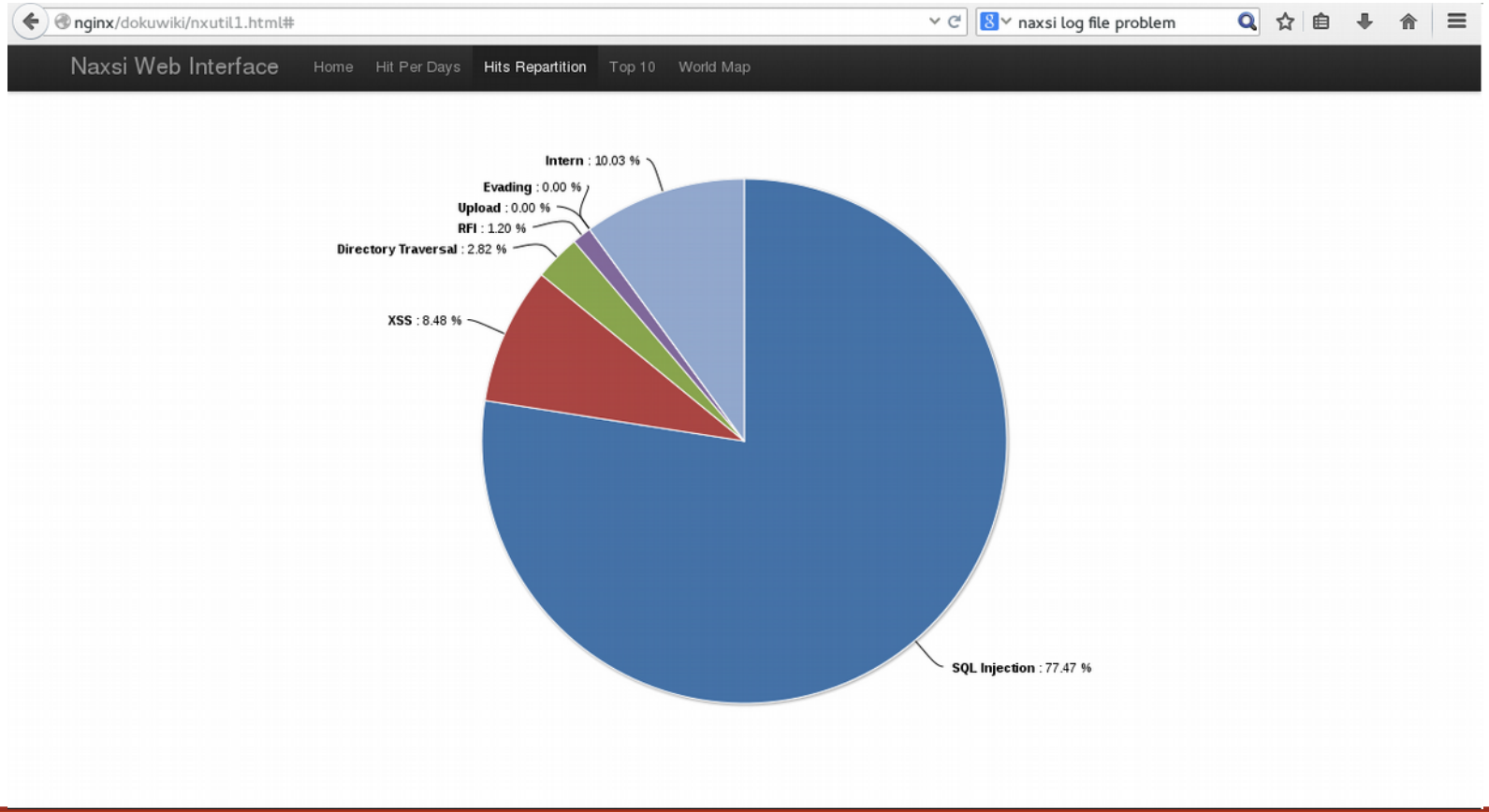
Grrr ! Pourquoi je n'ai pas vu ça avant !!!
Les heures de travail que j'aurai pu gagner…

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 méchanisme de filtre est extrêmement primitif d'après les développeurs donc prudence avec les filtres trop complexes.

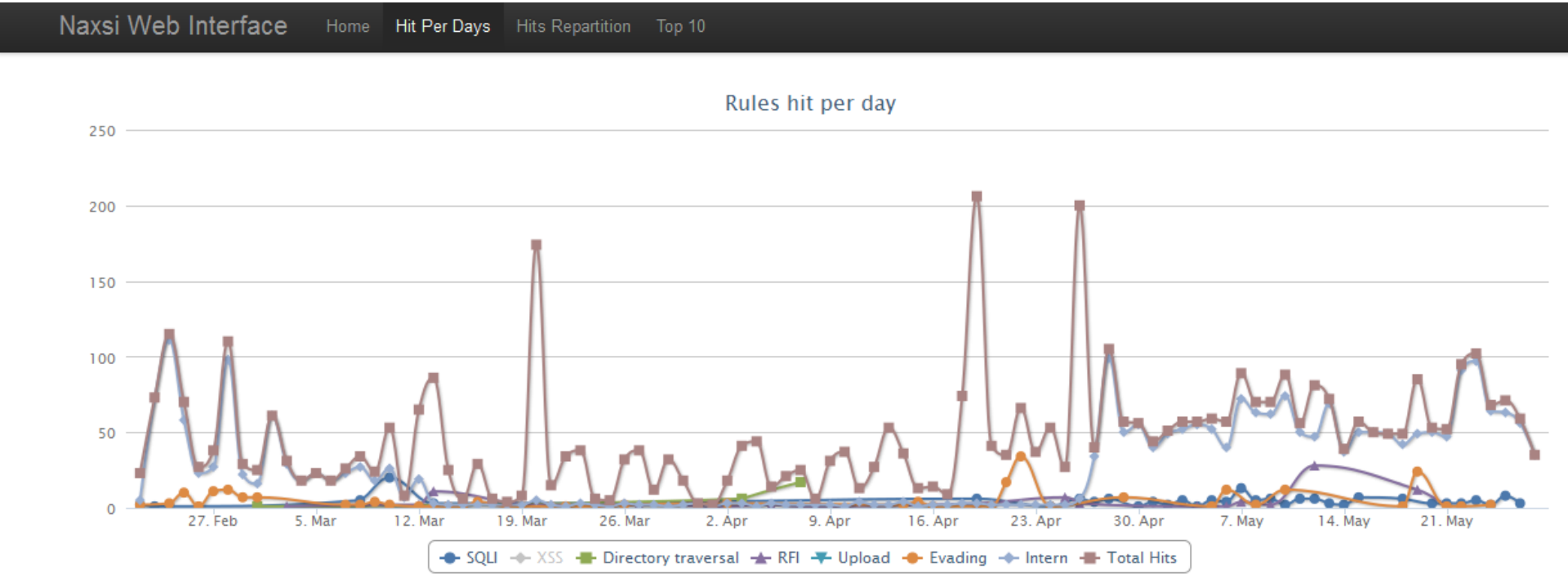
NAXSI : Rapports

- Diagramme :



NAXSI : Rapports

- Graphique :



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 : Waptiti ne trouve rien également
- La sécurité interne de Dokuwiki ne nous permet pas de voir véritablement Naxsi

Badstore

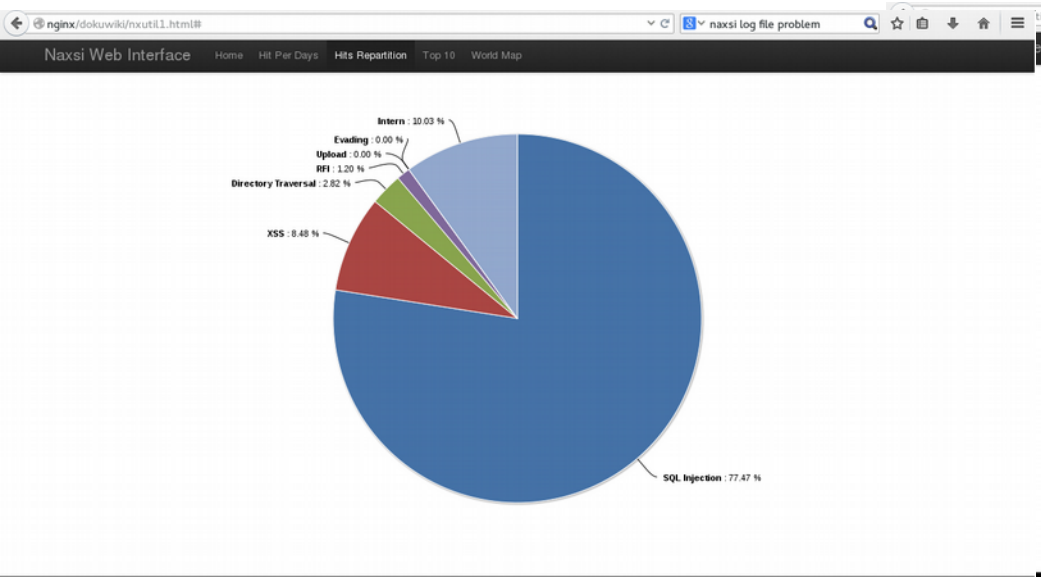

BADSTORE.NET
Quick Item Search
View Cart
Welcome to BadStore.net!
Home
What's New
Sign Our Guestbook
View Previous Orders
About Us
My Account
Login / Register
- Suppliers Only -
Supplier Login
- Reference -
BadStore.net Manual v1.2
GASKILL BROS
BadStore v1.2.3s - Copyright © 2004-2006

Résultats Badstore

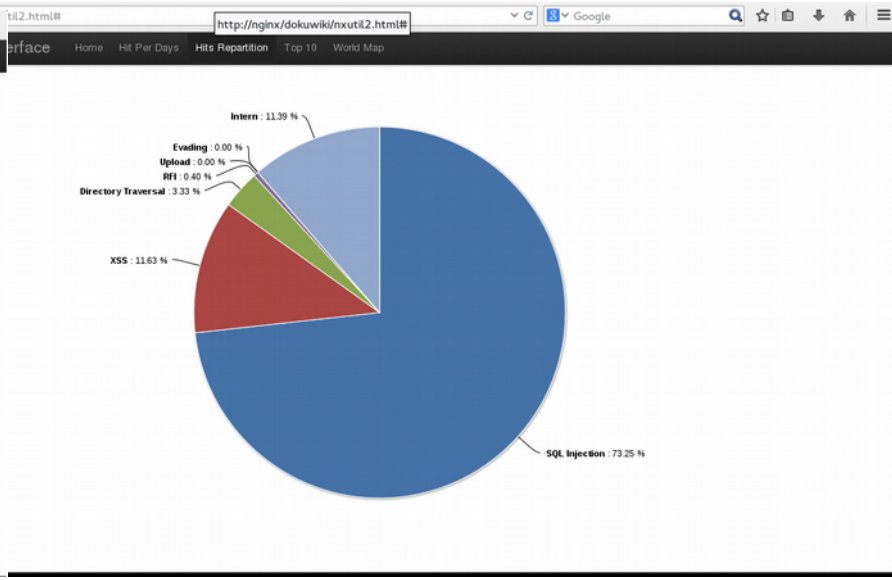
- Virtualbox de Badstore
- Wapiti
  - Sans Naxsi : 19 failles
  - Avec Naxsi : 1 faille de type file upload non exploitable heureusement.

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