

SECURINETS



CLUB DE LA SECURITE INFORMATIQUE
INSAT

SECURILIGHT 2011

[Skipfish]

Chef Atelier : Emna Ben Achour (RT5)

Chaima Ben Haha (RT4)

Soumaya Haouari (RT4)

Aïcha Lamia Thabet (RT5)

Mohamed Fekih (RT5)

30/11/2011

I. Etude théorique :

a. Présentation de Skipfish :

Skipfish est un scanner Open Source destiné à auditer la sécurité des sites et des applications Web. Il a été mis en oeuvre par google et est destiné à aider les développeurs et les professionnels de la sécurité dans leurs tâches d'audit des sites et des applications Web. Développé entièrement en C et spécifiquement optimisé pour le protocole HTTP, Skipfish devrait avant tout se caractériser par ses hautes performances. En effet, il arrive à envoyer près de 2000 requêtes http sur un serveur distant, si bien sûr le serveur le supporte, et arrive à 7000 requêtes sur un serveur local.

Skipfish utilise des méthodes heuristiques automatisées, ce qui pourrait ainsi réduire le taux de faux positifs. En effet, les résultats se basent sur une longue liste de vulnérabilités connues qui aurait été établie sur la seule identification d'une bannière applicative, éventuellement usurpée.

Parmi les tests critiques (compromission du système), on notera les injections SQL, de commandes Shell et XPATH, ainsi que l'exploitation des vulnérabilités liées aux chaînes de format et aux dépassements d'entiers. Les attaques XSS, Directory Traversal et CSS sont quant à elles classées dans les impacts modérés (compromission de données).

Skipfish est compatible avec les systèmes d'exploitation GNU/Linux, FreeBSD et Mac OS X. Il peut également être utilisé avec les systèmes de type Microsoft Windows dès lors que ces derniers présentent un environnement Cygwin fonctionnel.

b. Pourquoi utiliser Skipfish?

Parmi les avantages de Skipfish, on peut lister :

- une haute performance : nombre élevé de requêtes par seconde sur des hôtes réagissant bien,
- la simplicité d'utilisation : Skipfish s'utilise par une simple exécution en ligne de commande.
- des vérifications de sécurité bien pensées.
 - scan rapide : Le scan complet peut durer jusqu'à 24h mais on peut interrompre le scan au bout de quelques minutes et avoir déjà un rapport bien fourni
- Un rapport en HTML simple et bien organisé.
- Une faible consommation de ressources.

II. Etude pratique

a. Installation de Skipfish

Skipfish n'étant pas disponible dans les dépôts Linux (ubuntu ou autres), on ne pourra pas l'installer par un simple apt-get install (ou rpm -i), il faut pour cela suivre les étapes suivantes :

Tout d'abord le téléchargement : Deux étapes : le téléchargement du dossier compressé des dépôts de google et la décompression

```
emna@ubuntu:~$ sudo wget http://skipfish.googlecode.com/files/skipfish-1.26b.tgz
--2011-11-24 07:09:20-- http://skipfish.googlecode.com/files/skipfish-1.26b.tgz
Resolving skipfish.googlecode.com... 209.85.147.82
Connecting to skipfish.googlecode.com|209.85.147.82|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 174487 (170K) [application/x-gzip]
Saving to: `skipfish-1.26b.tgz'

100%[=====] 174,487 121K/s in 1.4s

emna@ubuntu:~$ sudo tar zxvf skipfish-1.26b.tgz
skipfish/
skipfish/README
skipfish/types.h
skipfish/string-inl.h
skipfish/config.h
skipfish/debug.h
skipfish/Makefile
skipfish/database.c
skipfish/database.h
skipfish/alloc-inl.h
skipfish/http_client.h
skipfish/http_client.c
skipfish/crawler.c
skipfish/crawler.h
skipfish/dictionaries/
skipfish/dictionaries/extensions-only.wl
```

Puis l'installation. Il faut penser à installer les bibliothèques dont aurait besoin Skipfish pour son exécution, puis faire un make et install pour Skipfish. Et l'installation sera terminée!!

```
emna@ubuntu:~$ sudo apt-get install libssl-dev libidn1-dev
root@ubuntu:/home/emna# cd skipfish
root@ubuntu:/home/emna/skipfish# make
cc -L/usr/local/lib/ -L/opt/local/lib skipfish.c -o skipfish -O3 -Wno-format -Wall -funsigned-char -g -ggdb -D_FORTIFY_SOURCE=0 -I/usr/local/include/ -I/opt/local/include/ \
    http_client.c database.c crawler.c analysis.c report.c -lcrypto -lssl -lidn -lz

See dictionaries/README-FIRST to pick a dictionary for the tool.

Having problems with your scans? Be sure to visit:
http://code.google.com/p/skipfish/wiki/KnownIssues
```

b. Utilisation de Skipfish:

b.i. Les commandes Skipfish:

Il est nécessaire de copier et renommer le fichier dictionnaire à utiliser (minimal, default, complete, etc.) dans le répertoire skipfish.

Regardons un exemple avec le dictionnaire default :

```
#cp dictionaries/default.wl skipfish.wl
```

Pour lancer le scan, il suffit tout simplement de se rendre dans le dossier skipfish en mode superutilisateur (root) et d'exécuter la commande suivante :

```
./skipfish -o /home/user/scan http://www.example.com
```

L'option -o indique le dossier dans lequel le fichier de rapport sera enregistré.

Les autres options que nous pouvons utiliser dans skipfish sont :

- m : nombre maximal de connexions établies
- B : ne va pas dans les sous domaines
- L : interdire un auto_aprentissage de nouveaux mots clés
- V : ne pas mettre à jour la liste de mot extraite du scan
- Y : inclure les extensions dans le dossier de brute-force

Skipfish offre également la possibilité de faire l'attaque du brute-force . On peut là choisir de la faire sur un ou plusieurs répertoires, et définir la durée de temps durant laquelle l'attaque se déroulera.

Pour faire l'attaque de Brute Force uniquement sur un seul répertoire et avec un timeout de 5 secondes :

```
./skipfish -P -I http://www.example.com/dir1/ -o output_dir -t 5 -I  
http://www.example.com/dir1/
```

Pour faire l'attaque BruteForce, on doit tout d'abord déterminer un dictionnaire des noms de fichiers possibles en format dictionnaire de Skipfish. Cela peut être fait avec un script écrit avec perl et qui sera de la forme suivante:

```
perl -e 'for $h (0..23)  
{  
  for $m (0..59)  
  { for $s (0..59) \  
{ $w = sprintf("something_110202%02d%02d%02d.txt",$h,$m,$s) ; \ print "w 1 1 1  
$w\n"; } }' > mywordlist.wl
```

Skipfish utilisera la liste de mot sans trainer le site web et de sans rien dissimiler

```
./skipfish -W mywordlist.wl -I http://www.example.com/data/ -o tt -O -P -L -V -Y -d 5  
-c 86400 http://www.example.com/data/
```

b.ii. Les vulnérabilités détectées

Les vulnérabilités sont classés selon un degré de dangerosité pour le site web : les couleurs devant les failles détectées expliquent cela et on a bien sûr dans le rapport une légende pour expliquer à quel type de vulnérabilité appartiennent les vulnérabilités détectés.

Dans le classement des vulnérabilités, on trouve :

Issue type overview - click to expand:

- Integer overflow vector (5)
- SQL injection vector (1)
- HTML form with no apparent XSRF protection (7)
- External content embedded on a page (lower risk) (1)
- Response varies randomly, skipping injection checks (20)
- IPS filtering enabled (4)
- Limits exceeded, fetch suppressed (40)
- Resource fetch failed (60)
- Numerical filename - consider enumerating (15)
- HTML form (not classified otherwise) (1)
- Server error triggered (23)
- Resource not directly accessible (20)
- New 404 signature seen (13)
- New 'X-*' header value seen (25)
- New 'Server' header value seen (2)
- New HTTP cookie added (3)

On commencera donc par expliquer les vulnérabilité haut risque (High risk flaws (potentially leading to system compromise)):

- Integer overflow :L'affectation d'un entier non-signé 32 bits à un entier non-signé 16 bits permettait de modifier l'index d'une table dans laquelle des valeurs étaient écrites pour cibler des endroits non-autorisés dans la mémoire.
- SQL injection:Un programme permettant de soumettre à une base de données des requêtes SQL malformées ou trop génériques.

Nous avons ensuite les failles de bas risque (Low risk issues (limited impact or low specificity)), c'est à dire qui ont un impact limité ou spécifique sur le serveur:

- HTML forms with no XSRF protection:Les attaques de type XSRF utilisent l'utilisateur comme déclencheur, celui-ci devient complice sans en être conscient. L'attaque étant actionnée par l'utilisateur, un grand nombre de systèmes d'authentification sont contournés.
- External content embedded on a page : Un contenu externe est intégré dans la page

Nous avons également des failles qui concernent des données entrées comme par exemple à la base de données (Non-specific informational entries):

- Server error triggered: nous pouvons là avoir deux types d'erreurs :
 - Error establishing a database connection : problème sur la base de données ou problème de configuration
 - 500 Internal Server Error : mauvaise configuration du site
- Resource not directly accessible :Les erreurs de la Base de Données
- New 'X-*' header value seen : ici aussi nous avons relevé deux types de failles qui sont

X-Powered-By : informe sur les failles du serveur (affiche la version de php utilisée)

X-Pingback : notifie l'auteur lorsque quelqu'un accède à l'article

- New HTTP cookie added
PHPSESSID :En cas d'ouverture de session , si les cookies ne sont pas disponibles, alors le paramètre PHPSESSID est ajouté à l'url.

III. Conclusion

Skipfish a permis de rendre accessible et facile d'utilisation les scanner de vulnérabilités web. Son installation et son fonctionnement font en sorte que quelconque personne qui s'intéresserait d'une façon ou d'une autre à la sécurité informatique peut juger la vulnérabilité de son site web. Par ailleurs, nous remarquons quelques défauts , surtout en ce qui concerne la signalisation de la faille exactement. Mais ceci se comprend, skipfish étant exécuté coté client, on ne peut voir le code PHP. En plus, montrer exactement les failles rendrait la tâche facile non seulement aux experts en sécurité, mais surtout les pirates qui auront des facultés à s'introduire dans les sites à la portée.