CHRISTOPHE ALLADOUM

Google Hacking

Degré de difficulté

Les fonctionnalités offertes par le moteur de Google vont largement au-delà de la simple recherche de mot. Avec une utilisation adaptée, il devient un outil redoutable pour trouver des failles de configuration, et cela de manière quasi transparente. Nous essayerons de comprendre comment un Pentester peut utiliser Google pour récupérer furtivement des informations sur une cible.

Google est le moteur de recherche le plus fréquenté en France et dans le monde. Né sur l'idée originale de Larry Page et Sergey Brin en 1998, il représente aujourd'hui plus de 82% de parts de marché dans les moteurs de recherche, et précède de loin ses concurrents (Microsoft, Yahoo, AOL...). Ce chiffre impressionnant est en partie lié à la bonne image que possède Google en France auprès du public. Ceci vient entre autres du fait que sa prise en main est immédiate: pas de page ultra chargée, pleine de détails inutiles, longue à s'afficher, retournant des résultats pas forcément très convaincants, etc... Le moteur de recherche Google séduit par sa sobriété, sa légèreté, sa rapidité de traitement et sa pertinence de résultat.

Il est donc par la même un acteur de choix dès lors qu'un webmaster souhaite populariser le contenu de son site Web. Celui-ci va donc tenter de mettre à disposition un maximum d'informations pour que Google analyse le contenu de son site le plus intelligemment possible, et donc que l'utilisateur effectuant une recherche sur Google soit amené rapidement sur son site. Mais, ce que ce webmaster peut avoir oublié, est qu'il dispose également des données qui ne doivent surtout pas être mis à disposition du grand public, et donc doit chercher par tout moyen à prévenir l'accès à ces informations confidentielles par un moteur de recherche (pas forcément Google par ailleurs).

Nous verrons tout au long de cet article que le potentiel de cet outil va largement plus loin que la recherche basique de mots-clés, et que Google peut tout à fait être considéré comme un outil de Pentest.

Utilisation Basique de Google

Le Google Hacking est une technique consistant à détourner la puissance de récupération d'informations par Google afin de retrouver des données en jouant avec les différents paramètres fournis par le moteur de recherche. Concrètement, il s'agit de connaître ses paramètres pour forger des requêtes beaucoup plus précises et ciblées. Nous reviendrons donc dans cette partie, sur l'utilisation usuelle du moteur de Google après avoir expliqué le fonctionnement de la récupération d'informations sur le Web par un moteur de recherche.

Rappel sur le fonctionnement d'un moteur de recherche

Tous les moteurs de recherche actuels utilisent de petits agents chargés d'explorer les sites Web. Nommés *Web Crawler*, *Web Robots* ou encore *Web Spider*, ces outils scrutent les pages du Web et basculent d'une page à l'autre grâce aux liens hypertextes contenus dans cette page, en mémorise le contenu. S'il trouve un ou plusieurs liens identifiés par la balise `<a href='lien _ vers _ une _ autre _ page'>Mon Lien</a>`, il bondit vers le lien, en envoie le contenu au *Crawler Manager*, et ainsi de suite. Le contenu des pages (textes, images, documents, metadata, etc...) est indexé et stocké: il servira de base au moteur

CET ARTICLE EXPLIQUE...

Le fonctionnement de Google et des moteurs de recherche

Les outils offerts par le moteur de recherche et les API

Comment limiter la diffusion des informations de ses sites Web.

CE QU'IL FAUT SAVOIR...

Etre familier avec les différents explorateurs (Internet Explorer, Firefox, Opera...)

Avoir des bases concernant le protocole HTTP

Connaître le langage HTML

Quelques connaissances du langage Python

de recherche. La Figure 1 nous montre que le Web Crawler se décompose en un agent qui analyse une page sur une URL donnée et le Crawler Manager chargé de gérer, trier, stocker en base de données les informations récupérées par l'agent. Ceci va nous permettre de mieux comprendre le fonctionnement du moteur de Google.

Principe de fonctionnement de Google

À partir d'un Web Crawler, chaque moteur de recherche (Google, Yahoo, Live Search) possède toute liberté pour se perfectionner. De son côté, Google a misé sur la sobriété, la rapidité, et un concept nouveau: le PageRank. Il s'agit d'un système de notation dont le principe est simple: plus une page possède de liens vers elle, plus elle est populaire, plus elle doit apparaître haute dans le classement. Dans le passé, le système de PageRank a pu être exploité pour forcer certaines correspondances abusives et sans pertinence (par exemple, en tapant *escroc*, on retrouvait en résultat une page de présentation de… Jacques Chirac). Cette technique, nommée *Google Bombing*, consistait simplement à créer de faux sites dans lesquels on mettait des liens indiquant *Jacques Chirac est un escroc*. Le GoogleBot explorant ces sites faisaient grimper au classement PageRank l'association {*escroc, Jacques Chirac*}, d'où le résultat de la recherche.

Il s'agissait des premières exploitations abusives de Google, et cette faille a été résolue par un système de confiance dans les liens des pages.

Pour que la recherche soit la plus pertinente possible, Google utilise les préceptes suivants :

- *l'ordre des mots a son importance :* taper `pizza+café` ne retournera pas forcément la même chose que `café+pizza`,
- *la recherche par bloc* : on peut demander à Google de chercher un ensemble de mots très précis. Cela est très pratique pour retrouver des citations, paroles de musique et autres. Il suffit de mettre le bloc à chercher entre guillemets,
- *la casse importe peu* : les requêtes `pizza`, `pIzZA`, et `PiZZa` sont équivalentes,
- le *stemming*: ce procédé consiste à supposer qu'un mot-clé peut être la souche (stem en anglais) d'un autre mot, que Google cherchera aussi. Ainsi, entrer 'hack' nous montrera des résultats contenant `hacker`, `hackers`, `hacking`…,
- *un jeu d'options* permettant d'affiner une recherche, ce que nous détaillerons en profondeur dans le paragraphe suivant,
- *certains mots peuvent être ignorés par Google* car ils sont trop communs et ne constituent pas un motif pertinent pour une recherche. Par exemple, les mots 'qui', 'quoi', 'et', 'ou', 'donc', etc. ne seront pas considérés dans le traitement de la requête par le moteur Google sauf s'ils appartiennent à un bloc (délimité par des guillemets).

La construction de réponses par Google se fait en deux temps: analyser la requête en utilisant chacun des mots-clés pour établir une base solide de recherche puis se servir d'autres paramètres, tels que le PageRank ou bien les filtres de recherches pour affiner le résultat avec le plus de pertinence.

On peut manipuler le moteur à deux niveaux :

- directement dans la barre d'URL de l'explorateur. On forge sa requête en utilisant les paramètres d'une requête `GET` de HTTP, paramètres dont on trouvera une liste en Tableau 1. Il existe de nombreux autres paramètres `GET` acceptés par Google,
- Manipulation du champ de recherche principale (le paramètre que nous avons

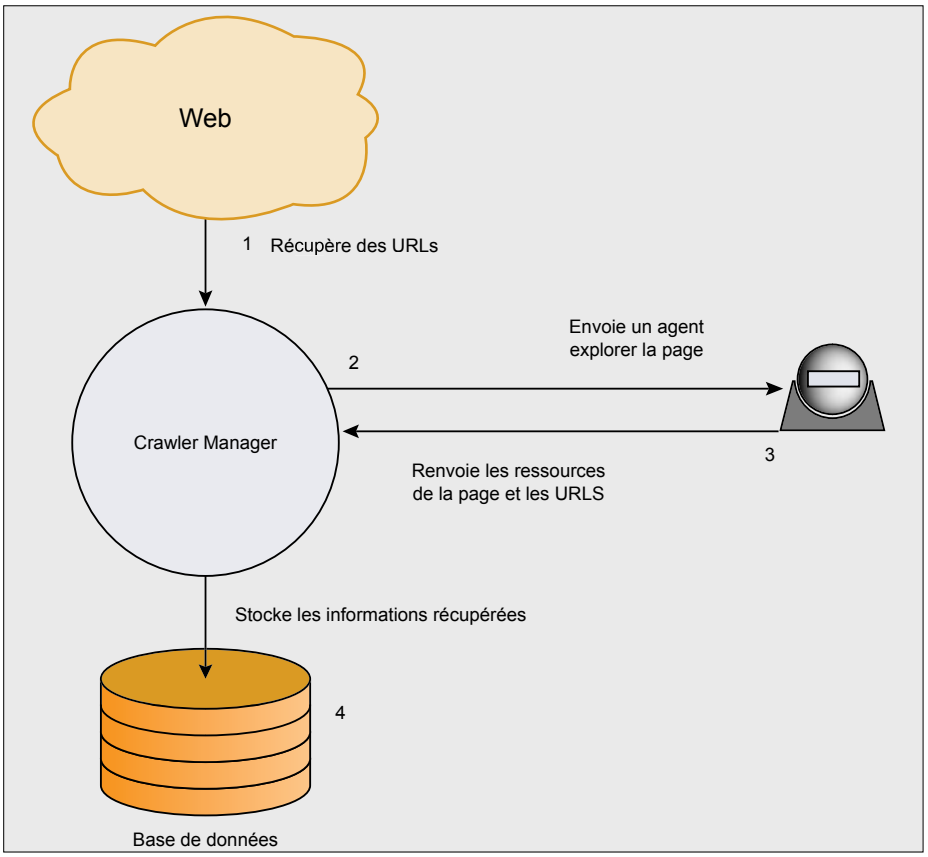

Figure 1. *Principe d'un Web Crawler*

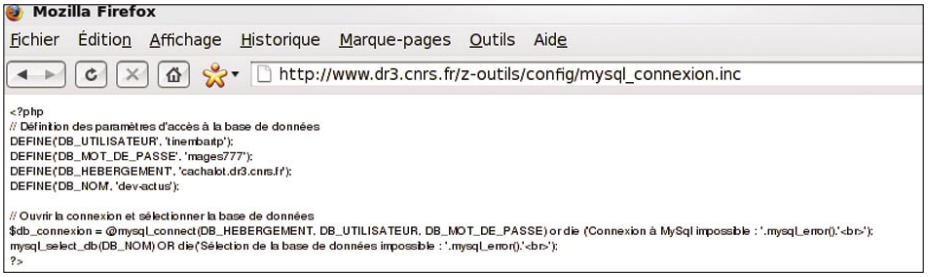

Figure 2. *Code PHP en clair avec les identifiants de la base de données*

vu plus haut), par certains mots-clés mentionnés dans le Tableau 2, peut modifier et affiner le résultat d'une recherche.

Récupération d'informations avancées via Google

Par le Google Hacking, un pirate peut utiliser à son avantage la confiance accordée au fameux moteur de recherche pour extraire des informations non voulues par le webmaster.

Même si elles diffèrent par leur forme (*Smurf, Ping Of Death…*), les attaques informatiques sont relativement semblables dans leur fond. Elles suivent un schéma assez récurrent qui permet d'optimiser les chances de succès et l'ampleur de l'attaque. Ces différentes étapes sont dans l'ordre, la Collecte d'information, Mapping du Réseau, Recherche de Vulnérabilité, Accès à la machine/au réseau, Augmentation des Privilèges, et Nettoyage des Traces. L'utilisation approfondie de Google nous place dans l'étape 1, soit la Récupération d'information, ou Information Gathering. Cette étape est importante et nous verrons un peu plus loin que bien utilisé peut faire gagner beaucoup de temps à un attaquant car cela peut lui permettre de connaître des logins/mots de passe sans laisser de traces sur le serveur ciblé. Sans plus attendre, voici un ensemble des filtres, en Tableau 2 qui sont supportés par le moteur de rechercher Google mais il existe d'autres filtres applicables aux autres services de la société (email, groupe, news…). Nous ne nous attarderons que sur certains de ces filtres mais l'ensemble de ces options a été très bien décrite dans l'excellent livre de Johnny Long, *Google Hacking For Penetration Testers* (voir l'encadré sur Johnny Long). Ce dernier est expert en sécurité informatique et a mené un grand nombre de recherche sur l'utilisation détournée de Google (on notera qu'on trouve sur son site *http://johnny.ihackstuff.com* une base de données référençant de nombreux Google Hacks).

Listing 1. *Collecteur d'email en Python*

```
#!/usr/bin/env python
import httplib, urllib, re
nb_reponses='100'
params=''
headers = {"User-Agent": "Mozilla/5.0 Firefox/3.0.0",
   "Content-type": "application/x-www-form-urlencoded",
   "Accept": "text/plain"}
query = "@hotmail.com+@gmail.com+@yahoo.com"
email_pattern="[a-zA-Z0-9_.]+\@[a-zA-Z]+\.[a-zA-Z]+"
conn = httplib.HTTPConnection("www.google.com")
conn.request("GET", "/search?hl=en&num="+nb_
      reponses+"&q="+query, params, headers)
response = conn.getresponse()
print response.status, response.reason
data = response.read()
email = re.compile(email_pattern)
list = re.findall (email, data)
print list
conn.close()
```

Tableau 1. *Liste de quelques paramètres GET de Google*

Paramètre	Description	Exemple
q=requete	Champ principal, il possède la chaîne de caractère de la recherche(variable requete)	*http://www.google.fr/search?q='to be or not to be'*
start=xxx	Fait commencer la liste des résultats à partir du numéro xxx	*http://www.google.fr/search?q='to be or not to be'&start=5*
num=yyy	yyy indique le nombre de résultat de la page	*http://www.google.fr/search?q='to be or not to be'&start=5&num=10*
filter={0|1}	Filtre ou non les duplications	*http://www.google.fr/search?q='to be or not to be'&start=5&num=10&filter=0*
safe={active|off}	Active le filtrage par SafeSearch sur le contenu des résultats.	*http://www.google.fr/search?q='to be or not to be'&safe=off*
lr={lang_fr|lang_en|lang_de|lang_ru}	Permet de restreindre le résultat à un langage (ici français, anglais, allemand, russe).	*http://www.google.fr/search?q='to be or not to be'&lr=ru*
hl={fr|en|ru}	Indique à Google en quel langue doivent apparaître ses propres messages (à ne pas confondre avec le résultat des recherches)	*http://www.google.fr/search?q='to be or not to be'&hl=fr*

Présentation des différents filtres supportés par Google

Google offre un panel impressionnant d'options pour cibler sa recherche. Parmi les plus basiques, on retrouve les `AND` (opérateur: plus +), `NOT` (opérateur: moins -) et `OR` (opérateur: pipe |).

L'opérateur `AND` sert de jointure entre différents termes ou les blocs d'une requête. De plus, il est équivalent à l'opérateur +. On peut donc taper la requête de façon strictement équivalente : `browser AND firefox` et `browser+firefox`. Cet opérateur devient de plus en plus obsolète, car Google a pour comportement par défaut d'analyser tous les mots (sauf ceux contenus dans un bloc, qu'il considère dans son intégralité).

L'opérateur `NOT` permet, à l'inverse de `AND`, d'exclure du résultat de recherche les pages contenant un mot, ou un motif. Il est plus souvent utilisé dans les requêtes par son équivalent, l'opérateur – (moins). Par exemple, en tapant la recherche: `informatique -windows`, Google me rendra les pages contenant le terme `informatique` mais ne contenant pas `windows`. Enfin, l'opérateur `OR` , équivalent de `| (pipe)`, permet de considérer deux ou plus motifs dans la recherche. La requête `admin+(password | passwd)+(username | login)` recherchera toutes les pages

contenant le terme admin, le terme password ou passwd, et le terme username ou login.

Ces opérateurs élémentaires permettent déjà de faire des requêtes plus pointus. Mais ils peuvent être couplés aux filtres suivants, pour aller plus loin dans le filtrage. Leur syntaxe est la suivante:

```
{ + | - } opérateur:motif
   '+' : affichera dans les résultats
les requêtes validant ce filtre
   '-' : affichera les résultats ne
validant pas ce filtre
```

Il ne s'agit en fait qu'une partie des filtres de Google mais sont ceux les plus utilisés pour affiner une recherche. Lançons donc quelques recherches dans Google en combinant efficacement les filtres fournis par la Tableau 2.

`inurl:.inc +intext: mysql_connect -inurl: (.php|.html|.htm|.asp)` va retourner des pages de configuration PHP contenant des informations comme le login et mot de passe de leur base de données en clair comme le montre la Figure 2; les '.inc' étant une extension souvent utilisée par des webmasters pour les fichiers à inclure dans du code PHP,

`site:microsoft.com +inurl: microsoft.com -www.microsoft.com` : cette technique se nomme *Site Mapping* et consiste à récupérer tous les sous-domaines d'un site. On peut ainsi connaître une partie de la topologie DNS du nom de domaine ciblé, comme le montre la Figure 3.

`-inurl:(.php|.html|.asp|.htm) intitle:'Index Of' +'last modified' +'parent directory'` permet de faire du site listing et de voir tout le contenu d'un répertoire du site. Cette technique peut donc par exemple servir à chercher des fichiers audio ou vidéo stockés sur des sites, en forgeant simplement une requête du type `-inurl:(.php|.html|.asp|.htm) intitle:'Index Of' +'last modified' +'parent directory' +inurl(.mp3|.ogg|.wma) + 'mon artiste'` . Vous serez impressionné de voir la quantité de données audio/vidéo accessibles en *direct download* sur le Web !

`ext:htpasswd -inurl: (.html|.htm|.asp|.php) intext:admin'` : dans Apache (le serveur Web le plus utilisé au monde), le couple de fichiers htaccess/htpasswd permet de restreindre l'accès à un répertoire, un fichier à un utilisateur authentifié. Le fichier htpasswd contient une syntaxe similaire à celui du passwd sous Linux. Ainsi ici on obtiendra des réponses de la forme `admin:mot_de_passe_hashé` (la Figure 4 montre un exemple de fichier de configuration récupérée). Connaissant cette information, il ne vous reste qu'à lancer un John The Ripper ou Abel&Cain pour bruteforcer ce condensat.

`inurl:view/view.shtml` : cette simple requête permet d'explorer parfois avec seulement un léger différé, de caméras de surveillance.

Les combinaisons de filtres sont extrêmement nombreuses et peuvent être facilement adaptées à une application ciblée (Apache, IIS, MySQL…). Par exemple, un serveur LAMP laissera passer des données importantes dans le cas d'une mauvaise ou d'une absence de configuration, comme l'arborescence du script de connexion, ou bien sur le serveur HTTP/SQL, son nom, sa version, le login d'accès comme on peut le voir en Figure 5.

Il serait beaucoup trop long et inutile de dresser une liste complète de toutes les actions qui vous sont offertes par ces filtres, mais les quelques requêtes citées peuvent déjà vous servir de base pour explorer de nouveaux horizons via

Listing 2. *Exemple de /robots.txt*

```
# le diese indique un commentaire
# propriete pour le robot Google
User-Agent:GoogleBot
# on interdit l'acces au répertoire /foo1/
Disallow:/foo1/
# et a tous les fichiers .doc du repertoire /foo2/
Disallow:/foo2/*.doc
# on interdit tous les autres robotos
User-Agent:*
Disallow:/
```

Tableau 2. *Liste des filtres de Google*

Syntaxe du filtre	Description
inurl	Retourne les pages contenant un lien vers un fichier du type mentionné en argument – exemple : `inurl:admin`
filetype	Retourne les pages contenant un lien vers un fichier du type mentionné en argument – exemple : `filetype:pdf`
intext	Recherche un motif dans le contenu – exemple : `intext:mysql_connect`
site	Permet de filtrer la recherche sur un site passé en argument.
link	Retourne l'ensemble des pages contenant un lien vers le motif passé en argument
cache	Permet d'accéder à la version mis en cache par Google. Cette option est surtout utilisée pour visiter discrètement un site.
define	Fournit une définition au terme passé en argument
intitle	Recherche dans le champ ‹title›‹/title› d'une page HTML
ext	Recherche dans les pages dont l'extension (html, php, etc…) est le motif passé.
[X]…[Y]	Effectue une recherche dans l'intervalle [X, Y]. Par exemple: `page+1...100`
info	Récupère des informations sur le site passé en paramètres. Par exemple, `info:www.kernel.org`
related	Retrouve les sites sémantiquement liés au paramètre. Par exemple, `related:www.kernel.org`

Terminologie

- *HTML* : standard défini par l'organisme W3C (pour *World Wide Web Consortium*). Cette norme est également homologué à l'ISO comme la norme ISO8879. La version la plus répandue aujourd'hui est la 4, la 5 étant en cours de création. Il s'agit du langage le plus répandu sur le Web, car il est simple, textuel, extrêment modulable,
- *Directory Listing* : aperçu du répertoire courant. Cette méthode par certains serveurs HTTP lorsqu'il ne trouve pas de page HTML par défaut à afficher (*index.php*, *index.html*...),
- *DoS* : Denial of Service: attaque réseau consistant à empêcher l'accès à un service ou une machine,
- *Smurf* : inondation d'une cible avec des messages réseau. Le résultat peut en être un DoS,
- *Ping Of Death* : envoi de paquets de ping avec une taille non supportée par le protocole, Cette attaque peut aboutir sur un DoS,
- *Web Crawler* : aussi connu comme Web Spider ou Web Robot, c'est un automate dont le but est d'explorer les sites du net, utilisant les balises de liens (`<a></a>`) pour passer d'un site à l'autre,

Google, et vous permettre de commencer à collecter des informations sur des sites potentiellement vulnérables, car mal configuré.

Vous avez surement remarqué que nous n'avons pas encore discuté de l'option *cache*, or il se trouve que ce mot-clé est très utile car il permet de visiter une page d'un site Web tel qu'elle a été enregistrée dans le cache du moteur Google. On n'accède donc pas directement au site, mais on lit la version en cache de Google. Cette manipulation possède l'énorme avantage que vous ne serez jamais identifié dans le journal d'événement du site, donc intraçable par ce dernier qui ne verra que les entrées de Google, mais l'inconvénient majeur est que vous ne verrez peut-être pas la version à jour du site.

Outils d'automatisation de manipulation de Google

Voici quelques outils crées qui permettent d'automatiser la recherche via les hacks de Google.

Google Hacks

Crée par Jason Stalling, Google Hacks se présente sous la forme d'une GUI facilitant la recherche avancée dans Google. Elle forge des requêtes pour le moteur de recherche, en filtrant par type de fichiers recherchés: mp3, wma pour de l'audio; avi, mpg pour de la vidéo; etc. Il ne fait que faciliter la recherche avec les filtres mentionnés dans la partie précédente.

Gooscan

Dans ces recherches de hacks sur Google, Johnny Long a crée un scanneur de vulnérabilités de sites web. Codé en C, il se comporte comme un CGI Scanner mais possède l'avantage de lancer ces requêtes via Google et non directement (vous êtes donc caché par Google). L'outil est inclus dans le Live CD Backtrack.

Goolag

Goolag est un projet du CdC (Cult Of The Dead Cow) en deux parties: la partie interface Web vise à parodier le célèbre moteur de recherche, en réponse aux ententes entre ce dernier et des régimes répressifs pour contrôler l'information diffusée. D'autre part, on trouve un outil développé également par le CdC, le Goolag Scanner qui est un scanneur de vulnérabilités capable d'automatiser des audits sur un serveur Web avec près de 1500 requêtes de recherches prédéfinies. Il se présente comme une GUI de Windows configurable via un fichier XML (attention : lancer de façon abusive un Goolag scan sur un serveur peut engendrer un blocage de votre adresse IP sur le serveur).

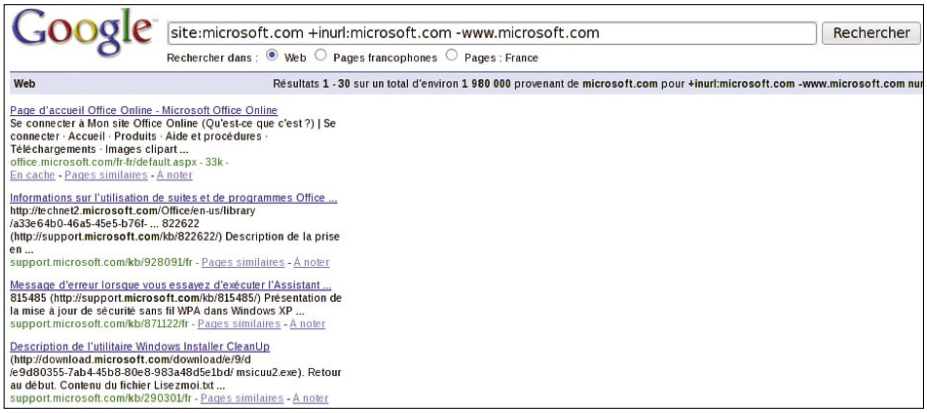

Figure 3. *Site Mapping*

Exemple: création d'un collecteur d'email à partir de Google

Nous avons vu précédemment quels étaient les paramètres HTTP que l'on pouvait passer à Google pour en cibler le contenu (langage, nombre de réponses, etc). Ainsi, à partir des tableaux 1 et 2, nous allons voir dans cette partie comment fabriquer très vite un outil envoyant des requêtes valides à Google pour en extraire des informations précises, en l'occurrence des emails.

Ce script assez simple (en Listing 1) permet de récupérer facilement une liste d'adresses email. Il affichera sous la forme d'un tableau les adresses collectées via une requête sur Google recherchant les emails des domaines Gmail, Hotmail et Yahoo. Le nombre de réponses demandées étant contenu dans une variable, on peut modifier augmenter ou diminuer le nombre de résultats.

Google autorise selon sa charte d'utilisation à créer des outils automatisés utilisant l'usage de son moteur. Cependant, on notera qu'un outil générant des requêtes de manière abusive se verra vite bloqué par Google (cf *http://www.google.com/terms_of_service.html pour plus de renseignements*). Pour pallier ce problème, nous inclurons un header HTTP valide mentionnant un navigateur existant et une de ces versions.

On notera qu'il existe de nombreuses API facilitant l'interaction avec le moteur de recherche Google ainsi que les autres services Google (GMail, GCalendar, Google Docs...) et de plugins Firefox pour optimiser le comportement de Google (Googlepedia, GooglePreview, GoogleEnhancer).

Quelques Mesures de Protection

Nous utiliserons les techniques décrites ci-dessus pour proposer quelques solutions de protection contre une exploration un peu trop intrusive d'un moteur de recherche.

Bannir le directory listing

Nous avons vu qu'il était possible d'accéder au contenu d'un site par la technique du *directory listing* : cette technique consistant simplement à faire une requête avec les mots clés typiques (*parent directory, index of*, description). Il faut donc prévenir cela soit en interdisant le listing dans la configuration de votre serveur HTTP, en restreignant l'accès. Le moyen le plus propre consisterait à mettre contrôle d'accès via un *.htaccess* , mais ceci n'est pas toujours faisable. Un autre moyen plus simple consisterait simplement à mettre une page *index.html* – ou *index.htm*, *index.php*... – dont le contenu est vide. Ainsi, le directory listing sera plus difficilement faisable par un moteur de recherche. Pour aller un peu loin pour retarder un visiteur un peu trop curieux, on veillera également à utiliser des noms de fichiers et de répertoires non conventionnels : on évitera donc d'avoir un *user.html*, *db/*, *sql/* etc.

Interdire les 'Crawling Robots' avec /robots.txt

Comme nous l'avons vu plus haut, le parcours des sites web se fait par des robots qui explorent la toile. Pour limiter les abus un consensus a été créé : *RobotsTxt.org*. Il définit avec précision le comportement des robots ainsi que le moyen d'empêcher qu'un robot ne parcourt un site. Il suffit de créer un fichier nommé *robots.txt* accessible en lecture par tous, à la racine du site Web. On y mettra une liste d'instructions indiquant la portée que l'on donne à un robot pour ce site (cf. Listing 2). L'ensemble des robots respectant ce consensus est listé dans une base de donnée accessible librement sur le site de *RobotsTxt.org*.

Testez la visibilité de son propre site

On peut enfin avoir un aperçu de ce que Google fournit comme informations au monde entier en testant soi-même ou par une société professionnelle (idéalement les deux) le niveau de visibilité de son site. Cette manipulation est simple et peut être effectuée fréquemment, ce qui est recommandé dans le cas de site dont le contenu varie beaucoup (blog, journal, etc). On n'est jamais mieux servi que par soi-même.

Sur Internet

- *http://www.comscore.com/press/release.asp?press=2327* – Classement des moteurs de recherche en France par l'institut ComScore,
- *http://www.google.com/intl/fr/corporate/* – informations générale sur la société Google,
- *http://fr.wikipedia.org/wiki/Crawler* – définition d'un web crawler,
- *http://fr.wikipedia.org/wiki/PageRank* – définition du mécanisme de PageRanking,
- *http://www.w3.org/TR/html401/* – Spécification du HTML 4.01,
- *http://johnny.ihackstuff.com/* – Site de Johnny Long, sur lequel on retrouve la GHDB, (*Google Hacking DataBase*) référençant quantité de hacks de Google,
- *http://code.google.com/p/googlehacks/* – site des Google Hacks,
- *http://www.robotstxt.org/robotstxt.html* – site descriptif du RobotsTxt,
- *http://www.googleguide.com/advanced_operators.html* – site tutorial pour trouver des astuces et apprendre encore plus sur les Google Hacks
- *http://www.goolag.org* – site du CdC, parodie de Google
- *http://www.goolag.org/privacy.txt* – Pour plus d'informations, lire la Privacy Policy sur.

Figure 4. *Identifiant administrateur d'un Trac*

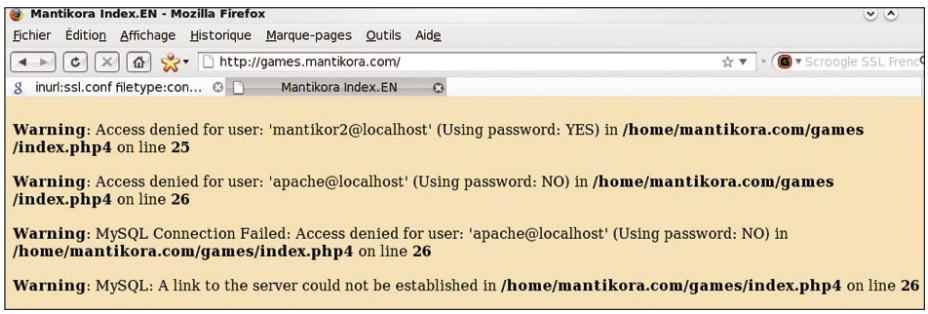

Figure 5. *Informations sur l'utilisateur, le SGBD utilisé, le chemin d'exécution*

Conclusion

Même si concevoir un site est relativement aisé, empêcher de laisser filtrer des informations non voulues l'est beaucoup moins, surtout lorsque Google s'en mêle. Au cours de cet article, nous nous sommes efforcés de comprendre les mécanismes de récupération d'informations par un moteur de recherche en général, et Google en particulier; et comment utiliser ces informations à notre avantage contre une cible. Nous avons enfin proposé de façon non exhaustive quelques solutions faciles à mettre en place pour prévenir une récupération abusive de données par Google mais aussi d'autres moteurs de recherche. Car sur Internet, il est difficile de retirer une information dévoilée ne serait-ce que quelques heures (vive les caches). Ces solutions peuvent être complétées, modifiées, mais permettent à un propriétaire de site de connaître un peu mieux ce qu'il laisse en diffusion sur le net. Alors, testez, et testez encore, et souvenez-vous toujours que *Google is your friend* (Google est votre ami)!

À Propos de l'Auteur

Christophe Alladoum vit en France et est étudiant à l'Université Pierre et Marie Curie à Paris (UPMC). Il achève actuellement un cursus de Master Informatique spécialité Réseaux et Télécommunications en cohabilitation avec l'Ecole Normale Supérieure des Télécommunications (Télécom ParisTech). Il se passionne pour la sécurité informatique et les réseaux de communication depuis de nombreuses années. Il participe activement à la vie de la communauté Open Source, particulièrement autour de la distribution Linux Fedora. Vous pouvez le contacter sur l'adresse mail: *christophe.alladoum@gmail.com*.