

Universidade do Minho
Conselho de Cursos de Engenharia
Licenciatura em Engenharia de Sistemas e Informática

Disciplina de Opção III - Projecto

Ano Lectivo de 2007/08



escola de engenharia



departamento de
informática

Estudo Comportamental dos Web Crawlers

Ricardo Coelho(nº47072),Rui Azevedo(nº47065)

Julho, 2008

Data de Recepção	
Responsável	
Avaliação	
Observações	

Estudo Comportamental dos Web Crawlers

Ricardo Coelho(nº47072)

Rui Azevedo(nº47065)

Julho,2008

Resumo

Este projecto teve, como finalidade, efectuar um estudo acerca da actividade dos *Web Crawlers*. Um *Web Crawler* (também conhecido como *web spider* ou *web robot*) é um programa que pesquisa e percorre a *World Wide Web* de uma maneira automática sendo, por exemplo aplicado aos motores de busca. A sua aplicabilidade nos motores de busca é bastante importante uma vez que são os *Web Crawlers* que permitem indexar as páginas que percorram. Para além desta, existem outras utilidades dos *Web Crawlers*, desde a recolha de informações específicas (tipo endereços de correio electrónico) até a manutenção das páginas (validando *links* ou código *html*). Um dos problemas dos *Web Crawlers* prende-se com o facto destes, para além de gastarem largura de banda, invadirem a privacidade e consequentemente guardarem informação sobre o utilizador (um exemplo são os endereços de correios electrónicos que mais tarde poderão ser utilizados para fazer spam). Para se estudar estes programas, foi criado um sítio *Web* que funcionasse como chamariz, atraindo os *Web Crawlers*. Um exemplo de um bom chamariz é o da pornografia, uma vez que é dos temas mais pesquisados na *Web*. Deste modo, o sítio contém falso material pornográfico para um utilizador, mas que para o *Web Crawler* é visto como conteúdo realmente pornográfico. Foram criadas várias secções no site como imagens, vídeos, cotação das bolsas, etc. Com vista a aumentar a probabilidade de o sítio ser visitado por um *Web Crawler*, palavras como “EURO2008” e outras relacionadas com temas bastante importantes foram espalhadas pelo sítio, sendo estas invisíveis a um utilizador humano.

Área de Aplicação: Funcionamento Dos Web Crawlers na Rede

Palavras-Chave: Web Crawler, Crawling, Internet, Indexação, Algoritmos,

Políticas, Servidor, HTTP REQUEST, User-Agent.

Conteúdo

Resumo	i
Conteúdo	iv
Lista de Figuras	v
1 Introdução	1
1.1 Contextualização	1
1.2 Apresentação do Caso de Estudo	1
1.3 Motivação e Objectivos	2
1.4 Estrutura do Relatório	2
2 Web Crawlers em detalhe	3
2.1 Funcionamento de um Web Crawler	3
2.1.1 Política de selecção	4
2.1.2 Política que define quando voltar a visitar um sítio	4
2.1.3 Política de educação	4
2.1.4 Política de Paralelismo	5
2.2 Arquitectura base de um Web Crawler	5
2.3 Identificação de um Web Crawler	5
2.4 Métodos de restrição de acessos de Web Crawlers	6
3 Métodos: Como estudar a actividade dos Web Crawlers	7

3.1 Criação de uma página Web e sua evolução	7
4 Resultados	10
5 Conclusões e Trabalho Futuro	15
Bibliografia	16
Referências WWW	17
Lista de Acrónimos	18

Lista de Figuras

2.1	Arquitectura base de um Web Crawler.	5
3.1	Imagem da página de teste.	9

1 Introdução

1.1 Contextualização

Um *Web Crawler* é um pequeno programa que pesquisa e percorre a *World Wide Web* de uma maneira rápida e automática. Estes programas são muito mais utilizados do que o normalmente se *Web* pensa. Utilidades tais como guardar cópias das páginas visitadas para mais tarde serem processadas, providenciar uma manutenção automática de páginas verificando *links* ou o próprio código html, guardar informação presente no site, etc. Uma aplicabilidade são os motores de pesquisa, como por exemplo o Google com o Google Bot. As páginas *Web* ao serem visitadas, são copiadas e mais tarde processadas pelo motor de busca, indexando-as. Outro exemplo do uso destes programas são os *news feed* muitas vezes utilizados nos sítios contendo informação dinâmica, tais como sítios com informação sobre a Bolsa de Valores, sítios com notícias desportivas, etc. Estes sítios muitas vezes utilizam *Web Crawlers* para obterem informação actualizada sobre variadíssimos temas. Uma outra utilização também bastante comum dos *Web Crawler* é a de guardar informação específica como por exemplo os endereços de correio electrónico, permitindo fazer *spam* das caixas de correio. Em suma, os *Web Crawlers* são absolutamente essenciais ao funcionamento da *Web*

1.2 Apresentação do Caso de Estudo

Após perceber a verdadeira importância dos *Web Crawlers*, vamos estudar a actividade destas ferramentas de modo a compreender melhor o modo como

actuar, o impacto que têm sobre a rede e verificar se respeitam as regras impostas pelos utilizadores. Para tal, foi construído um sítio recheado de conteúdos bastante procurados na *Internet*. O sítio deverá possuir conteúdos estáticos e dinâmicos para assim melhor estudar o comportamento dos *Web Crawlers*.

1.3 Motivação e Objectivos

Os *Web Crawlers* são ferramentas muito poderosas, com a possibilidade de criar problemas, não só nível da rede, como também problemas relacionados com a privacidade do utilizador. Problemas como o acesso repetido e desmedido de um *Web Crawler* a um sítio provocando assim um gasto de largura de banda adicional, a quebra de privacidade de um utilizador quando um *Web Crawler* guarda dados e informação, o mau uso da informação guardada, são apenas exemplos do que os *Web Crawlers* são capazes de fazer quando são indevidamente utilizados. Temos então por objectivo, compreender o funcionamento dos *Web Crawlers*, estudando as maneiras como percorrem a *Web* e os métodos que permitem controlar a actividade dos mesmos.

1.4 Estrutura do Relatório

No capítulo seguinte iremos aprofundar o modo como os *Web Crawlers* funcionam, algoritmos usados por estes, o esqueleto básico deste tipo de ferramenta e algumas técnicas que evitam o acesso exagerado e não autorizado de *Web Crawlers*. No capítulo quatro, iremos mostrar e explicar mais detalhadamente o modo como se estudou a actividade dos *Crawlers*, mostrando tudo o que foi feito para ser possível estudar e compreender os *Crawlers*. O capítulo cinco contém os resultados obtidos e a interpretação dos mesmos. Finalmente, o último capítulo contém as conclusões sobre este projecto.

2 Web Crawlers em detalhe

Neste capítulo, vamos estudar em detalhe o modo de funcionamento dos *Web Crawlers*, assim como os métodos para diminuir os problemas causados por estes.

2.1 Funcionamento de um Web Crawler

Um *Web Crawler* contém uma lista de *URLs* a visitar, chamadas de *seeds*. À medida que o *Web Crawler* visita estes *URLs*, ele identifica todos os *hyperlinks* do sítio e adiciona-os à lista de *URLs* a visitar chamada de *crawl frontier*. Os *URLs* pertencentes a esta *frontier* são visitados recurvivamente de acordo com um conjunto de políticas. Existem 3 características muito importantes da *Web* que dificultam a tarefa dos *Web Crawlers*: o seu grande volume, a rápida modificação de conteúdos e a geração dinâmica de páginas. Por ter um grande volume, um *Web Crawler* apenas pode fazer descarregar uma pequena fracção das páginas *Web*, por isso necessita de estabelecer prioridades relativamente às descargas. A rápida modificação de conteúdos aumenta as probabilidades de um *Web Crawler* descarregar conteúdo desactualizado. A geração dinâmica de páginas aumenta o número de combinações possíveis do parametro *HTTP GET* (método disponível no *Http* que permite obter uma representação de um pedido), sendo que apenas uma pequena porção destas combinações devolvem um resultado único. Um exemplo disto é uma galeria de fotos. Existem várias maneiras e tamanhos de ordenar as fotos. Assim, surgem várias combinações para obter o mesmo resultado. Os *Web Crawlers* são então obrigados a percorrerem várias com-

binações para obterem apenas um resultado único. Um *Web Crawler* deve então escolher cuidadosamente que site visitar a seguir. Surgem várias políticas que determinam o comportamento dos Crawlers.

2.1.1 Política de selecção

Um *Web Crawler* apenas consegue guardar uma pequena fracção das páginas *Web* e por isso convém que essa fracção contenha as páginas mais relevantes e não uma página à sorte da *Web*. É necessário definir uma política de selecção de páginas. Foram feitos vários estudos utilizando vários algoritmos desde o algoritmo *Depth First*, o algoritmo *Bread First*, entre outros. Por exemplo, ficou provado que, para sítios de um único domínio, o melhor algoritmo para obter as páginas mais relevantes no início do processo de *crawling* é o algoritmo PageRank. Este é um algoritmo que mede a relevância do site.

2.1.2 Política que define quando voltar a visitar um sítio

Percorrer uma fracção da *Web* pode demorar semanas ou até mesmo meses. Quando um *Web Crawler* acabar de percorrer essa fracção, muitos dos conteúdos poderão estar desactualizados. São definidas duas novas medidas: *Freshness* e *Age*. *Freshness* indica se a cópia que o *Web Crawler* possui está actualizada. *Age* indica o quão desactualizada está a cópia que o *Web Crawler* possui. Existem duas políticas. Uma política uniforme, que revisita todas as páginas com a mesma frequência desprezando a taxa de modificação, e uma política proporcional que revisita as páginas que são alteradas mais frequentemente.

2.1.3 Política de educação

Especifica se um *Web Crawler* respeita os métodos usados para restringir o acesso destes a alguns conteúdos do sítio.

2.1.4 Política de Paralelismo

Um *Web Crawler* paralelo é um *Web Crawler* que corre múltiplos processos em paralelo. O objectivo é maximizar a taxa de descargas enquanto se evita repetir descargas da mesma página. Para evitar descarregar a mesma página mais do que uma vez, o *Web Crawler* requer uma política para atribuição de novos *URLs* descobertos durante o processo de *crawling*, uma vez que o mesmo *URL* pode ser descoberto por dois ou mais processos.

2.2 Arquitectura base de um Web Crawler

Se um *Web Crawler* for criado com o objectivo de descarregar um enorme quantidade de páginas durante um longo período de tempo, necessita de ter uma arquitectura bastante optimizada. Segue-se uma imagem que representa uma arquitectura de alto nível de um crawler.

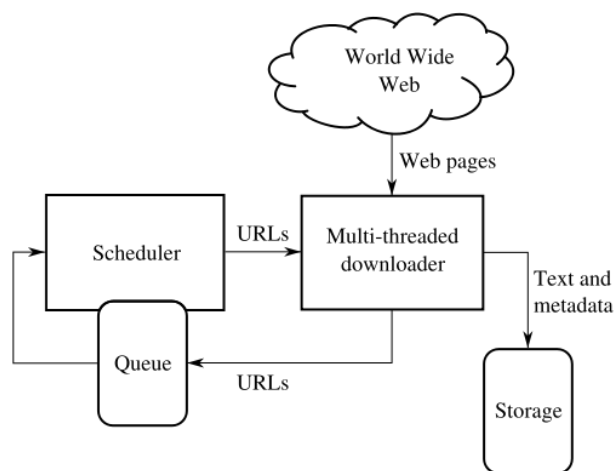


Figura 2.1: Arquitectura base de um Web Crawler.

2.3 Identificação de um Web Crawler

Um *Web Crawler* identifica-se ao servidor *Web* usando o campo *User-Agent* de um *HTTP request*. No entanto, caso o *Web Crawler* não deseje identificar-

se, este pode não enviar a identificação ou então mandar uma identificação falsa.

2.4 Métodos de restrição de acessos de Web Crawlers

Existem vários mecanismos para tentar impedir o acesso de *Web Crawlers*. O mais comum, e aquele que seria perfeito se os *Web Crawlers* respeitassem os limites impostos pelos utilizadores, é o *robots.txt*. O ficheiro *robots.txt* é onde se definem os conteúdos que serão acessíveis aos *Web Crawlers*, e os conteúdos que serão privados, ou seja, conteúdos de acesso vedado aos *Web Crawlers*. Especifica-se também para que *Web Crawlers* se aplicam essas regras. No entanto, se as especificações do *Web Crawler* estiverem definidas de modo a não respeitar essas regras, estes não irão ter em conta as restrições impostas pelo *robots.txt* e irão aceder aos conteúdos desejados. Outro método implementável, é o de verificar o *User-Agent* que acedeu ao sítio e caso corresponda a um *Web Crawler* que se deseje bloquear, bloqueia-se a ligação com o endereço *Ip* correspondente. No entanto este método também é facilmente contornado, uma vez que é possível alterar o endereço *Ip* aplicando uma máscara.

3 Métodos: Como estudar a actividade dos Web Crawlers

Para estudar a actividade dos *Web Crawlers*, criou-se uma página *Web* com conteúdo muito procurado na *Internet* de modo a que os *Web Crawlers* visitassem a página. Assim, estudando os registos de pedidos efectuados à pagina, registos esses efectuado pelo servidor *Apache*, é possível efectuar um estudo comportamental dos *Web Crawlers*. Os *logs* de acesso permitem descobrir o endereço *Ip*,o dia e hora do acesso, o tipo de pedido(*GET,POST..*),o que foi pedido e o *User-Agent*. A construção do sítio foi gradual, sendo todas as alterações efectuadas de modo a obter mais acessos por parte de *Web Crawlers* para assim se produzir um estudo mais rigoroso e completo. Explicam-se agora os passos que foram dados, desde a criação do sítio, até à implementação de um mecanismo de restrição de acessos, permitindo assim o estudo da actividade dos *Web Crawlers*.

3.1 Criação de uma página Web e sua evolução

O primeiro passo foi o de criar a página de raiz. Criaram-se ligações para páginas diferentes, criou-se a secção de imagens onde se colocaram imagens variadas e criou-se também uma secção destinada a videos. Com o objectivo de atrair *Web Crawlers* cujos interesses são endereços de correio electrónico, colocaram-se também vários endereços de correio electrónico na página. Numa segunda fase, acrescentaram-se ainda mais imagens, criou-se uma nova secção contendo informação sobre a bolsa de valores e uma nova secção dedicada unica e exclusivamente à pornografia. A pornografia

e os conteúdos eróticos são os pontos chave do nosso sítio, uma vez que são estes temas que lideram o volume de pesquisas na *Web*. Assim, a probabilidade de o site ser visitado por *Web Crawlers*. Nessa secção colocaram-se imagens com nomes pornográficos, sem no entanto haver alguma relação entre a imagem em si e o seu nome. Escreveu-se também um pequeno disclaimer para alertar os utilizadores humanos para o facto de a página ser apenas para efectuar um estudo, sendo o conteúdo da mesma falso. De modo a aumentar as probabilidades de um *Web Crawler* percorrer a página, decidiu-se traduzir tudo para Inglês. Tendo em conta o funcionamento dos *Web Crawlers*, especialmente as suas políticas que definem quando voltar a visitar um sítio, implementaram-se conteúdos dinâmicos na página. Assim, por cada pedido efectuado para a secção “Porno”, as imagens que são carregadas são sempre diferentes. Para conseguir imprimir este dinamismo à página, utilizamos a função *random()*. Assim, por cada acesso à página, novos valores são gerados, sendo estes os valores que ditam quais imagens serão carregadas. Deste modo, os *Web Crawlers* terão quase sempre conteúdos novos para percorrer.

Numa fase mais avançada, escreveram-se palavras, relacionadas com assuntos actuais e muito pesquisados, ao longo de todo o sítio. Todos estes passos e alterações à página têm como único objectivo aumentar a probabilidade esta ser visitada por *Web Crawlers*.

Para finalizar a página, introduziu-se o *robots.txt* o que nos permite estudar o comportamento dos *Web Crawlers* ao depararem-se com as limitações impostas pelo *robots.txt*

Segue-se uma pequena imagem da nossa página.

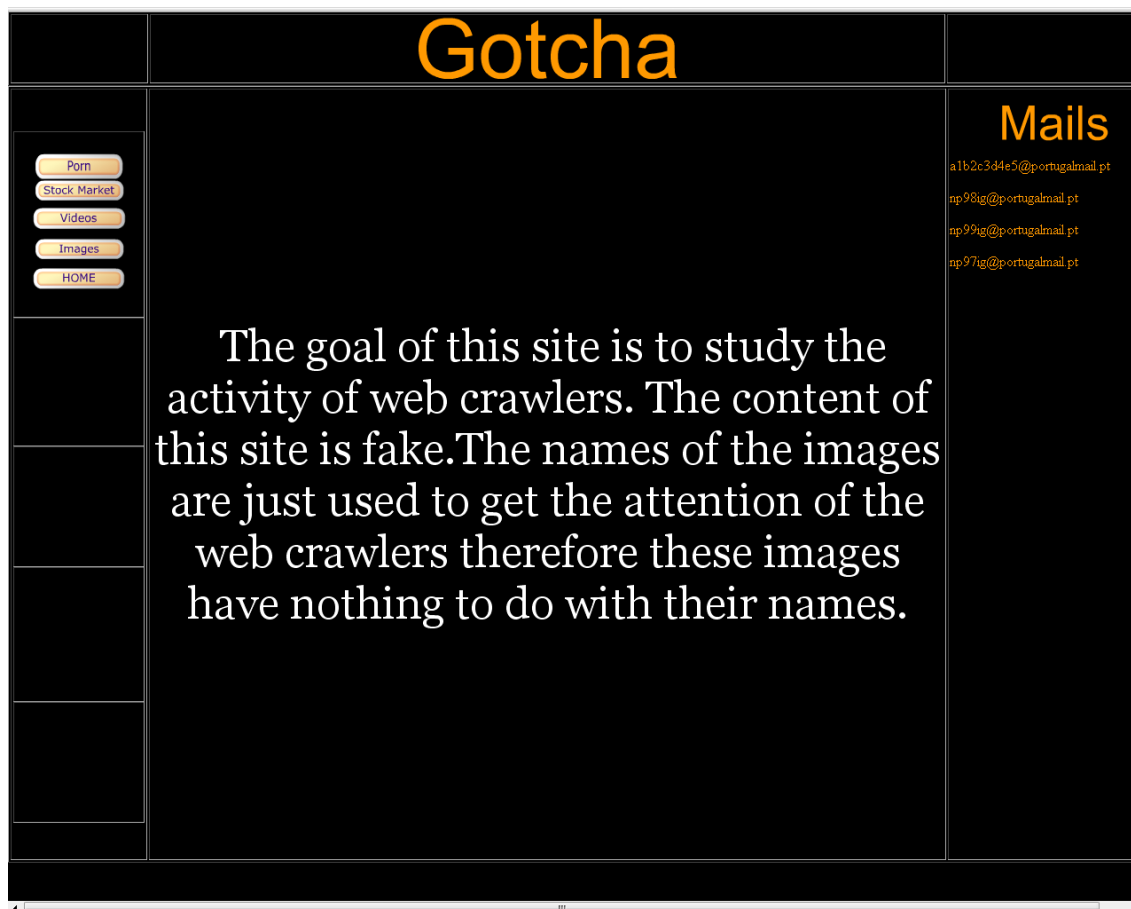


Figura 3.1: Imagem da página de teste.

4 Resultados

<<Sendo o principal objectivo deste projecto o estudo comportamental dos *web crawlers* precisavamos de algo que nos dissesse que a nossa pagina tinha sido visitada por um *web crawler* e de que maneira se tinha comportado. Tal é feito, ou pelo menos pode ser deduzido, pelo registo de acessos criado pelo Apache, o *access.log*, que não é mais que um ficheiro de texto que lista todos os pedidos feitos ao servidor. Embora o registo contenha informação sobre todos os pedidos, feitos por *crawlers* ou humanos, pode-se concluir muitas vezes quem realmente tentou aceder aos conteúdos do *site*. Por exemplo, se varios pedidos foram feitos a partir da mesma maquina num curto período de tempo, se não houver um *user-agent* especificado ou se houver tentativas de acesso a conteúdos que não fazem parte do *site*, tudo isto são bons indicadores de que foi um *crawler* a varrer a nossa pagina e não uma pessoa.

Apos esta pequena introdução vejamos entao alguns casos interessantes que foram obtidos pelo nosso *access.log*. Para cada um deles serão apresentadas as entradas correspondentes no referido ficheiro seguido da nossa interpretação do sucedido, tentando concluir o padrão de comportamento do *crawler*.

```
89.106.8.2 - - [12/Jun/2008:21:10:31 +0100] "GET /phpmyadmin/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:31 +0100] "GET /phpMyAdmin/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:31 +0100] "GET /db/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:32 +0100] "GET /web/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:32 +0100] "GET /PMA/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:32 +0100] "GET /PMA2006/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:32 +0100]
```

+0100] "GET /admin/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:
+0100] "GET /dbadmin/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:
+0100] "GET /pma2006/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:
+0100] "GET /sqlmanager/main.php HTTP/1.0"404 1123 --"89.106.8.2 - -
[12/Jun/2008:21:10:34 +0100] "GET /mysqlmanager/main.php HTTP/1.0"404
1123 --"[...] 89.106.8.2 - - [12/Jun/2008:21:10:44 +0100] "GET /phpMyAdmin-
2.5.6-rc2/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:44
+0100] "GET /phpMyAdmin-2.5.6/main.php HTTP/1.0"404 1123 --"89.106.8.2
- - [12/Jun/2008:21:10:44 +0100] "GET /phpMyAdmin-2.5.7/main.php HTTP/1.0"404
1123 --"89.106.8.2 - - [12/Jun/2008:21:10:45 +0100] "GET /phpMyAdmin-
2.5.7-pl1/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:45
+0100] "GET /phpMyAdmin-2.6.0-alpha/main.php HTTP/1.0"404 1123 --"89.106.8.2
- - [12/Jun/2008:21:10:45 +0100] "GET /phpMyAdmin-2.6.0-alpha2/main.php
HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:46 +0100] "GET
/phpMyAdmin-2.6.0-beta1/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:
+0100] "GET /phpMyAdmin-2.6.0-beta2/main.php HTTP/1.0"404 1123 --"89.106.8.2
- - [12/Jun/2008:21:10:46 +0100] "GET /phpMyAdmin-2.6.0-rc1/main.php HTTP/1.0"404
1123 --"89.106.8.2 - - [12/Jun/2008:21:10:47 +0100] "GET /phpMyAdmin-
2.6.0-rc2/main.php HTTP/1.0"404 1123 --"89.106.8.2 - - [12/Jun/2008:21:10:47
+0100] "GET /phpMyAdmin-2.6.0-rc3/main.php HTTP/1.0"404 1123 --"

Esta é apenas uma parte dos acessos relativos a este *crawler*. É possível concluir que é um *crawler* e não uma pessoa a efectuar os pedidos tendo em conta registos temporais dos mesmos. São feitos vários por segundo, o que indica uma pesquisa automatizada.

Este *crawler* tem um comportamento que à partida poderá parecer um pouco estranho. Faz numerosos pedidos de algo que nem se encontra no servidor e o facto de omitir o seu *user-agent* dá a entender que as suas intenções poderão não ser as melhores. Com certeza anda à procura de algo, possivelmente uma brecha na segurança do sistema. De facto, após pesquisar na *Web*, descobrimos que este comportamento é típico de um *crawler* chamado *pmafind* cujo objectivo é procurar vulnerabilidades nas configurações do *phpMyAdmin* e *mysql*. Se bem sucedido, o ataque daria acesso total à *shell*. Fomos vítimas de vários ataques como este. No entanto, nenhum deles teve êxito.

80.197.69.240 - - [22/Jun/2008:20:10:34 +0100] "GET /w00tw00t.at.ISC.SANS.DFind:) HTTP/1.1"400 364 --"

Esta é mais uma entrada estranha no access.log. O valor do campo a seguir ao pedido dá-nos informação sobre a resposta ao pedido, se foi feito com sucesso, redireccionado etc. Neste caso, o valor 400 indica-nos que o pedido foi mal feito. Consultando o error.log verifica-se que o cliente enviou um pedido HTTP 1.1 sem *hostname*. No entanto, ao contrário do que poderia parecer à primeira vista, esta entrada não é tão inocente quanto isso. Mais uma vez depois de pesquisar na *Web* informação relativa a esta entrada descobrimos que corresponde a um *bot* que procura vulnerabilidades na *Web* e surge normalmente associado a ataques como o mencionado anteriormente.

66.249.70.56 - - [26/Jun/2008:06:46:50 +0100] "GET / HTTP/1.1"200 12356 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:47:02 +0100] "GET /button4.swf HTTP/1.1"200 2833 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:47:03 +0100] "GET /button1.swf HTTP/1.1"200 2575 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:50:57 +0100] "GET /text78.swf HTTP/1.1"200 2031 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:55:17 +0100] "GET /button6.swf HTTP/1.1"200 3060 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:59:38 +0100] "GET /button8.swf HTTP/1.1"200 2594 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:10:03:59 +0100] "GET /text62134.swf HTTP/1.1"200 847 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:10:08:20 +0100] "GET /text62341.swf HTTP/1.1"200 777 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:10:12:41 +0100] "GET /button5.swf HTTP/1.1"200 2777 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:06:46:50 +0100] "GET /robots.txt HTTP/1.1"404 1128 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:06:46:50 +0100] "GET / HTTP/1.1"200 12356
-Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:47:02 +0100] "GET /button4.swf HTTP/1.1"200
2833 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:47:03 +0100] "GET /button1.swf HTTP/1.1"200
2575 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:50:57 +0100] "GET /text78.swf HTTP/1.1"200
2031 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:55:17 +0100] "GET /button6.swf HTTP/1.1"200
3060 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:09:59:38 +0100] "GET /button8.swf HTTP/1.1"200
2594 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:10:03:59 +0100] "GET /text62134.swf HTTP/1.1"200
847 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:10:08:20 +0100] "GET /text62341.swf HTTP/1.1"200
777 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

66.249.70.56 - - [26/Jun/2008:10:12:41 +0100] "GET /button5.swf HTTP/1.1"200
2777 -Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

Estas entradas são bastante interessantes pois correspondem ao *GoogleBot*, o *web crawler* que o *google* usa nos seus varrimentos. Verifica-se que é um *bot* bastante correcto pois a primeira coisa que faz é pedir o *robots.txt*. No entanto, nesta altura ainda não tínhamos escrito o *robots.txt* pelo que o servidor devolveu o valor 404, sinal de que o ficheiro não foi encontrado, e o *bot* continuou o seu trabalho sem entraves.

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /horde//README HTTP/1.1"404 1123
-Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /horde2//README HTTP/1.1"404
1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /horde3//README HTTP/1.1"404
1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /horde-3.0.9//README HTTP/1.1"404
1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /Horde//README HTTP/1.1"404 1123
-Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET //README HTTP/1.1"404 1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /horde//README HTTP/1.1"404 1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /horde2//README HTTP/1.1"404 1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /horde3//README HTTP/1.1"404 1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /horde-3.0.9//README HTTP/1.1"404 1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

83.145.78.243 - - [29/Jun/2008:14:37:25 +0100] "GET /Horde//README HTTP/1.1"404 1123 -Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

Este bot exhibe um comportamento bastante semelhante ao pmafind, visto que tenta aceder a um dado ficheiro pelos mais variados caminhos. Sendo o comportamento similar assumimos que o seu objectivo seja semelhante, ou seja, explorar uma qualquer falha de segurança. O facto de nos indicar que usa o Windows98 apenas faz aumentar as nossas suspeitas, tendo em conta que já não deve haver muita gente a usa-lo.

131.162.135.111 - - [27/Jun/2008:18:41:48 +0100] "GET / HTTP/1.0"200 12356 -AcadiaUniversityWebCensusClient"

Esta entrada é interessante no sentido que indica que, supostamente, fomos indexados pelo web crawler de uma outra universidade. No entanto, o crawler apenas pediu a pagina inicial do nosso site pelo que não há muito a concluir acerca do seu comportamento.

Estes resultados são um pouco parcós, no entanto, há que ter em conta que apenas são considerados logs a partir do dia 12 de Junho, ou seja, têm sensivelmente três semanas.>>

5 Conclusões e Trabalho Futuro

<<Olhando para trás para o trabalho que foi feito, acreditámos que atingimos os objectivos que nos foram propostos. Colocámos todo o tipo de conteúdos que achámos mais atractivos para os crawlers e conseguimos, de facto, "caçar"alguns e observar o seu comportamento. Deste modo, pensamos que os resultados obtidos são satisfatórios, se bem que, com mais tempo, pudessem ter maior volume e ser mais interessantes, de maneira a poder ser feito um estudo mais aprofundado. Na verdade, para trabalho futuro haveriam bastantes melhorias que poderiam ser implementadas. Uma opção interessante e, de certa forma, irónica, seria a de usarmos nos proprios um web crawler para descarregar conteúdos para o nosso site; poderíamos varrer alguns sites de notícias ou de informação sobre a bolsa diariamente e descarregaríamos para a nossa pagina o conteúdo que achássemos mais pertinente. Esta seria uma maneira de acrescentar dinamismo à nossa pagina. Ao mesmo tempo, estaríamos sempre a acrescentar e diversificar os conteúdos da página, de maneira a torna-la cada vez mais apetecível aos bots.

Para terminar, ao longo de todo este projecto, à medida que íamos mergulhando mais fundo no mundo dos crawlers, fomos-nos apercebendo da sua extrema importância. Mundo esse que, curiosamente, à partida nem sabíamos que existia. De facto, hoje em dia ninguém pensa em fazer pesquisa na net sem usar um motor de busca como o Google ou o Yahoo, mas certamente poucos serão aqueles que sabem que tais motores assentam, fundamentalmente, em web crawlers. Tendo em conta que motores de busca como os referidos estão entre os sites mais visitados em todo o mundo (com milhões de visitas diarias), não será demais dizer que sem web crawlers a Web de hoje estaria ainda na idade das trevas.>>

Bibliografia

Mike Thelwall, David Stuart (2005), Web Crawling Ethics Revisited: Cost, Privacy and Denial of Service

Referências WWW

01 ***www.wikipedia.org***

Aqui podemos encontrar variadíssima informação sobre o funcionamento dos Web Crawlers, bem como URLs para sítios que permitem descarregar Web Crawlers.

Lista de Acrónimos

URL *Uniform Resource Locator*

IP *Internet Protocol*