



Crie páginas à prova de spam!

Endereços protegidos

Quem envia emails não solicitados normalmente “garimpa” os endereços publicados em páginas na Internet. Este artigo mostra como divulgar seu próprio endereço e, ao mesmo tempo, mantê-lo a salvo dos programas de garimpagem automática. **POR TOBIAS EGGENDORFER**

O spam é um incômodo realmente difícil de combater. Embora os filtros de spam usem com relativo sucesso técnicas de **heurística** para separar o joio do trigo, os spammers estão sempre um passo à frente das armas de defesa e continuamente desenvolvem novos e engenhosos métodos para contorná-las.

A Raiz do Problema

Espantosamente, administradores históricos contribuem para a derrocada de suas próprias defesas ao devolver as mensagens rejeitadas para os spammers, detalhando o motivo pelo

qual o filtro considerou aquela mensagem como spam. Mesmo sem essa ajuda, os spammers buscam continuamente novas formas de contornar os filtros anti-spam. Devemos, portanto, evitar pensar nos filtros como solução completa para o problema.

Outra abordagem muito usada é combater o spam em sua origem, evidenciada pela crescente tendência de se usar servidores de **SMTP** com autenticação. Um usuário que deseje enviar um email deve necessariamente se autenticar no servidor de email usando seu número IP ou uma senha - muitos provedores já implementaram esse método.



Figura 1: A pescaria de endereços é feita principalmente no oceano da Web.

GLOSSÁRIO

Heurística: (Do grego “heuriskein”: descobrir) Procura por padrões baseados em regras empíricas que têm uma probabilidade muito alta de sucesso. Teoricamente um resultado desse tipo não é confiável, mas a operação de busca é muito mais rápida do que o processamento computacional preciso.

SMTP: Acrônimo para “Simple Mail Transfer Protocol” [Protocolo Simples para Transferência de Email]; Um servidor SMTP recebe e retransmite mensagens de email.

robots.txt: Arquivos com esse nome em servidores Web contêm informações sobre as páginas hospedadas. Os robôs de pesquisa

dos mecanismos de busca interpretam esse arquivo para saber quais páginas **não** vasculhar, entre outras coisas.

Entidade HTML: codificação HTML que imprime os caracteres de acordo com seu código ASCII. Para usar essa técnica, basta colocar o prefixo `&#` seguido do código ASCII e um sinal de ponto-e-vírgula. Dessa forma, `u` corresponde à letra “u”.

ASCII: O “American Standard Code for Information Interchange” (Código Americano Padrão para Intercâmbio de Informações) atribui um número, chamado de Código ASCII, para todas as letras, símbolos e números.

JavaScript: Linguagem de scripts para uso em websites. Se um browser possui suporte a JavaScript, é capaz de interpretar e executar os comandos embutidos no código HTML.

XOR: “Exclusive OR” – em português, “OU Exclusivo”. Operação lógica comum em matemática binária. É representada pelo símbolo `^` em JavaScript e pode ser usada para criptografia simétrica.

Flash: Formato proprietário da Macromedia que cria entidades com imagens, animações, vídeo e som em websites. Os navegadores precisam de plugins especiais para mostrar entidades em Flash.

De acordo com uma pesquisa realizada pelo Center for Democracy and Technology (CDT), os spammers fazem a garimpagem de endereços de email a partir de websites abertos ao grande público [1]. Durante o processo de investigação, o CDT deliberadamente publicou endereços de email, especialmente criados, em home pages, grupos de discussão e notícias e vários serviços na Web. Resultado: 97,3% das 8842 mensagens recebidas por esses endereços e classificadas como email não-solicitado foram endereçadas aos emails publicados em páginas Web (figura 1).

Baseado nos resultados da investigação, parece fazer sentido a idéia de não incluir endereços de email em sites para não colaborar com os spammers. De fato, os autores do relatório afirmam que vale a pena retirar os endereços de email das páginas, pois houve uma queda acentuada no número de mensagens por SPAM depois da remoção dos endereços usados no teste (figura 2).

Garimpagem automática de endereços

O método usado pela maioria dos spammers é bem primitivo. A partir de um site qualquer, eles simplesmente procuram e catalogam qualquer link do tipo *mailto:*, representando endereços de email. Depois clicam nos links para outras páginas e repetem o procedimento.

De vez em quando, os spammers chegam a alguma página referenciada por apenas um link. Eles usam técnicas semelhantes às dos mecanismos de busca. Não é difícil escrever um programa que automatize a tarefa – usualmente chamado de “spider” ou “harvester” (em português, poderíamos chamá-lo de “garimpeiro”). Depois de removidos os endereços duplicados, o spammer tem em suas mãos uma lista com centenas de vítimas potenciais.

É possível montar um garimpeiro simples usando as ferramentas que qualquer distribuição Linux possui por padrão, como *wget*, *sed*, *tr*, *sort*, e *uniq*. Os resultados são impressionantes!

O *wget* navega pelos sites; já o *sed* procura por emails em cada uma das páginas. O comando *tr* pode ser usado para colocar os endereços em minúsculas e capitalizar corretamente o pri-

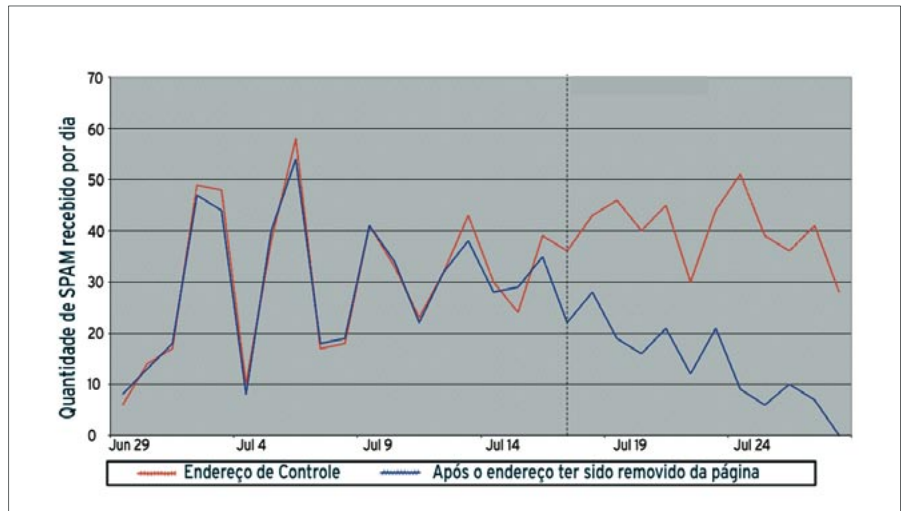


Figura 2: Vale a pena retirar das páginas os endereços de email existentes.

meiro nome da vítima, o *sort* classifica os emails em ordem alfabética e o *uniq* elimina as duplicidades.

Ao testar esse garimpeiro feito em casa em minha própria homepage, consegui mais de 90 emails em apenas oito minutos. Escolher uma página com mais links e não interpretar o arquivo *robots.txt* me trouxe muito mais resultados nos mesmos oito minutos. Além disso, a procura por endereços de email pode ser estendida para as páginas que apontam para a que estamos varrendo (“referred to”), mesmo que não haja um link nela para a página anterior.

Não divulgar endereços de email não é, normalmente, a solução preferida dos donos de sites. Afinal de contas, o obje-

tivo principal de um site na web é oferecer um canal de comunicação adicional. Em alguns países, os proprietários de páginas na Internet são obrigados por lei a fornecer pelo menos um endereço – é o caso da Alemanha, por exemplo.

Técnicas de camuflagem

Há diversas técnicas bem populares de camuflagem de email. Por exemplo, podemos usar a técnica do *spammer_cai_fora*. Nela, um endereço qualquer (por exemplo, *vendas@meudominio.com.br*) seria grafado como *vendas@spammer_cai_fora.meudominio.com.br*. Nem todos os internautas, entretanto, lembram-se de apagar a parte falsa do endereço antes de clicar no botão *Enviar*, e muitos

Listagem 1: Emails no cabeçalho HTML

```
<HTML>
<HEAD>
<TITLE>P&aaacute;gina exemplo</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
mailaddress = 'usuario@exemplo.com.br';
//-->
</SCRIPT>
</HEAD>
<BODY>
[... ]
<SCRIPT LANGUAGE="JavaScript">
<!--
document.write('<A HREF="mailto:'+mailaddress+'"'>'+mailaddress+'</A>');
//-->
</SCRIPT>
[... ]
</BODY>
</HTML>
```

sequer conseguem discernir qual parte do email é falsa. Novamente, restrições legais em alguns países podem proibir essa prática.

Há ainda a tendência atual dos spammers forjarem (no jargão do submundo a técnica é chamada de *spoofing*) o endereço do remetente da mensagem. As mensagens de erro não lotam a caixa postal do spammer. Lotam, isso sim, servidores de terceiros, que não têm qualquer ligação com ele. Muitos desses servidores inocentes chegam a sair do ar por excesso de tráfego.

O relatório do CDT [1] mencionado sugere a codificação dos endereços em páginas web como **entidades HTML**, onde `usuario@exemplo.com.br` se tornaria:

```
&#117;&#115;&#117;&#097;&#114;$$
&#105;&#111;&#064;&#101;&#120;&#101;$$
&#109;&#112;&#108;&#111;&#46;&#099;$$
&#111;&#109;&#46;&#098;&#114;
```

Os navegadores de internet não têm problema algum para interpretar isso, mas os garimpeiros não conseguem encontrar o padrão de texto comum de emails no código da página.

No relatório do CDT, endereços codificados dessa forma não receberam nenhuma mensagem de spam. Nosso garimpeiro improvisado, entretanto, ainda conseguiu encontrar 10 endereços.

Como o uso desse formato está crescendo, espera-se que novas versões de programas de garimpagem de email logo sejam capazes de automaticamente converter essas entidades HTML para endereços reais. A longo prazo, emails cifrados com entidades não terão quaisquer vantagens sobre os não cifrados.

JavaScript ao resgate?

Muitos programas de garimpagem de emails não interpretam código **JavaScript**. Isso permite que os proprietários de websites usem scripts para camuflar endereços de email.

A listagem 1 mostra como se pode especificar um endereço na seção **HEAD** de uma página HTML e usar a função `document.write()` do JavaScript para mostrar o endereço. Essa variante permite que os visitantes normais vejam o endereço em texto puro, enquanto as ferramentas especializadas ficam a ver navios.

Listagem 2: O JavaScript manipulando links

```
<HTML>
<HEAD>
<TITLE>P&acute;gina exemplo</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
mailaddress = 'usuario@exemplo.com,br';
function mailMe()
{
    document.location.href="mailto:"+mailaddress;
}
//-->
</SCRIPT>
</HEAD>
<BODY>
[... ]
<A HREF="javascript:mailMe();">Mail sender</A>
[... ]
</BODY>
</HTML>
```

Há várias maneiras de se atribuir um valor a uma variável. A mais simples – e mais fácil de manter – é usar um arquivo JavaScript externo para armazenar os endereços de email, que só são carregados pelo browser quando necessário. Os programas de garimpagem não conseguem manipular esse tipo de arquivo.

Isso posto, alguns navegadores encontram dificuldades para usar a função `document.write` para imprimir dados armazenados em arquivos externos. Ademais, spammers velhacos sempre podem encontrar métodos para descobrir o conteúdo de arquivos JavaScript referenciados ou mesmo baixá-los.

Há uma maneira de contornar o problema do `document.write()` nesse browsers. Em vez de usar um link HTML (``) que aponte diretamente para um endereço de email, pode-se lançar mão do JavaScript, novamente, para resolver o caso. Em vez de escrever o endereço de email em tags HTML, a listagem 2 mostra como usar a função `document.location.href` do JavaScript.

Mais camuflagem

Podemos usar um método extremamente simplório de criptografia, a operação lógica **XOR**, para evitar que um programa de garimpagem encontre endereços válidos em nosso site. Esses métodos de criptografia não podem ser considerados verdadeiramente seguros

do ponto de vista criptográfico, mas são extremamente eficientes para frustrar spammers que precisam de resultados rápidos. Embora o garimpeiro de emails possa, ainda, conseguir os endereços lendo o código fonte HTML da página, tais endereços estarão criptografados. O spammer deverá, pois, entender os comandos JavaScript – e provavelmente fazer engenharia reversa – para obter os endereços corretos.

A listagem 3 mostra como criptografar apenas o nome do usuário (ou seja, o nome antes do @). A função `document.location.hostname` recupera os elementos que não devem ser criptografados – ou seja, o domínio. Esse exemplo só funciona quando o domínio do email é idêntico ao domínio do servidor que hospeda a página. Se preferir, você pode usar o mesmo algoritmo para criptografar também o domínio.

O algoritmo de criptografia usado é facilmente extensível. Entretanto, você precisa certificar-se de que tanto o procedimento quanto a chave de criptografia estão no próprio script. Isso permite que qualquer um que rode o script possa ver os endereços de email e, obviamente, é necessário para que olhos humanos possam ler esses endereços.

A maior vantagem desse método é não permitir que os clientes que não sabem interpretar JavaScript reconheçam os endereços de email. Até o momento (e rezemos para que continue assim sem-

Listagem 3: Endereços criptografados

```
<HTML>
<HEAD>
<TITLE>P&aacute;gina exemplo</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
local = new Array (194,196,210,197);
local_part = '';
for (i=0;
    i<local.length;
    local_part += String.fromCharCode(local[i] ^ 183), i++) ;
    mailaddress = local_part + String.fromCharCode(64) + document.location.hostname;
//-->
</SCRIPT>
</HEAD>
<BODY>
[... ]
<SCRIPT LANGUAGE="JavaScript">
<!--
document.write('<A HREF="mailto:'+mailaddress+'"'>'+mailaddress+'</A>');
//-->
</SCRIPT>
[... ]
</BODY>
</HTML>
```

pre) essa é uma habilidade que os programas de garimpagem não possuem.

Você pode facilmente combinar os trechos de código JavaScript mostrados aqui – por exemplo, a técnica de criptografia da listagem 3 com os links JavaScript da listagem 2.

Seria muito bom se pudéssemos dizer que apenas os programas de garimpagem tropeçam nas rotinas de JavaScript, mas isso é uma triste verdade. Browsers em modo texto, como o popularíssimo Lynx, têm o mesmo problema. Além disso, muitos usuários desabilitam o suporte a JavaScript no browser gráfico por questões de segurança. Páginas cujos endereços de email estejam em JavaScript impedirão que tais usuários entrem em contato com você.

Os métodos de camuflagem descritos até aqui funcionam com endereços de email e links HTML. Se essas técnicas forem aplicadas em todo o planeta, os garimpeiros terão um trabalho danado para contornar o problema. Por outro lado, mecanismos de busca como o Google e internautas com JavaScript desligado também terão sua navegabilidade prejudicada.

Para não ter que forçar os visitantes a usar JavaScript, é possível usar um truque

bem simples, embora também excludente para os usuários do Lynx e assemelhados. Inclua uma imagem na página com o endereço impresso nela. Como o endereço não é mostrado em texto simples, os programas de garimpagem serão fragorosamente derrotados em sua faina de rastrear os endereços de email do site. Os spammers não podem usar softwares de OCR para obter esses endereços – afinal, qualquer uma das dezenas de imagens do site pode conter um endereço de email. Isso também abre uma brecha jurídica para os donos de sites obrigados por lei a publicar pelo menos um endereço eletrônico para contato.

Você pode atribuir um link a uma imagem sem precisar revelar o endereço: use um formulário de contato, que não precisa ter um endereço público, mas que envia todas as mensagens provenientes do site diretamente à caixa postal de seu proprietário.

Uma animação **Flash** que exiba um endereço clicável é outra variação da solução da imagem com link. Entretanto, o uso de Flash afasta uma parcela muito grande de visitantes do seu site.

Pegando pesado

Se você possui seu próprio domínio, pode usar um site dinâmico para tentar

descobrir a origem dos garimpeiros de email. Para tanto, crie um link “mailto:” que seja atualizado constantemente. Use a hora do dia, a data e o endereço IP do visitante para “montar” o endereço do link, de forma a obrigar o visitante a fornecer sua localização ao carregar a página.

Se esse endereço começar a receber spam em grande quantidade, ele próprio possui o endereço IP do atacante – e isso pode ser usado como prova num tribunal, caso necessário. ■

SOBRE O AUTOR

Tobias Eggendorfer trabalha como instrutor e consultor freelance em Munique, Alemanha. O combate ao spam – com armas tecnológicas ou não – é um dos esteios de sua carreira. Sua página pessoal é <http://www.eggendorfer.de>



INFORMAÇÕES

[1] Estudo do CDT sobre spam: “Why am I getting all this spam?”: <http://www.cdt.org/speech/spam/030319spamreport.html>

[2] robots.txt: <http://www.robotstxt.org/>