

Projektowanie stron internetowych chronionych przed spamem

Adres zastrzeżony

Dostawcy niechcianej poczty (czyli tzw. spamu) zdobywają znaczną część adresów poczty elektronicznej ze stron internetowych. W tym artykule omówimy sposoby zamieszczania własnych adresów pocztowych na stronach internetowych, z jednoczesnym ukryciem ich przed automatycznymi programami zbierającymi dane tego rodzaju.

TOBIAS EGGENDORFER



Hannes Keller, visipix.com

Spam jest utrapieniem niemal wszystkich użytkowników Internetu i trudno z nim walczyć. Filtry anty-spamowe wykorzystują do tego celu mniej lub bardziej skuteczną **heurystykę**, która umożliwia oddzielenie „plew od ziarna”. Niestety, w przeciwieństwie do zagrożenia wirusami, spamerzy są cały czas o krok przed dostępnymi obecnie technikami obronnymi, nieustannie rozwijając i opracowując nowe metody przenikania niechcianej poczty przez filtry pocztowe użytkowników.

Istota problemu

Paradoksalnie najczęściej szkód mogą spowodować nadgorliwi administratorzy – konfiguruje serwery pocztowe tak, że wysyłają one odrzucone przez system wiadomości z powrotem do nadawców spamu wraz ze szczegółowym opisem powodu ich odrzucenia przez system, dzięki czemu nadawca łatwo może sprawdzić, czy dany adres istnieje. Nawet bez takiej pomocy spamerzy po-

szukują nowych sposobów pokonywania filtrów pocztowych – nie jest to już zatem jedynie słuszne rozwiązanie zapewniające ochronę przed niechcianą pocztą.

Jedną z metod jest „zduszenie” niechcianej poczty w zarodku (czyli przy wysyłaniu), czego dowodem jest wzrastający nacisk na korzystanie ze standardu uwierzytelnianego **SMTP**. Użytkownik, który będzie chciał wysłać wiadomość, musi przesłać na serwer pocztowy potwierdzenie zawierające adres IP lub hasło użytkownika – wielu dostawców poczty elektronicznej już korzysta z tej metody obrony.

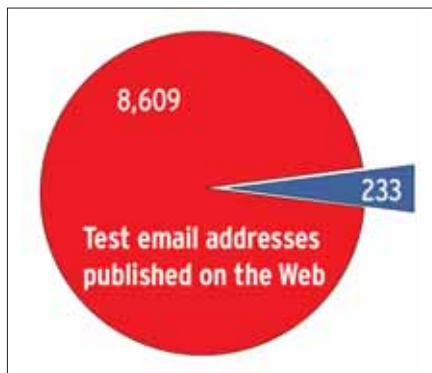
Z wyników ankiety przeprowadzonej przez CDT (ang. Center for Democracy and Technology) wynika, że osoby rozesyłające niechcianą pocztę zdobywają adresy pocztowe głównie dzięki ogólnodostępnym stronom internetowym [1]. Podczas przeprowadzania tych badań centrum CDT celowo ujawniło specjalnie stworzone do tego celu adresy poczty elektronicznej – na stronach domowych, na grupach dyskusyjnych

oraz w różnych usługach sieciowych. Aż 97,3% z 8842 otrzymanych wiadomości, które dotarły na te adresy, zostało sklasyfikowanych jako niechciana poczta reklamowa (Rysunek 1).

Z przeprowadzonych badań wynika, że nie warto ujawniać przy projektowaniu strony internetowej własnego adresu poczty elektronicznej. Autorzy raportu wręcz zalecają ukrycie bądź zakamuflowanie adresów pocztowych na istniejących stronach internetowych, gdyż po usunięciu specjalnie stworzonych do celów badawczych adresów pocztowych ze stron internetowych odnotowano znaczne zmniejszenie ilości otrzymywanych niechcianych wiadomości (Rysunek 2).

Automatyczne zdobywanie adresów

Najczęściej stosowana przez spamerów metoda zdobywania adresów jest trywialnie prosta. Przeglądając kolejne strony internetowe, zachowują oni po prostu wszystkie znalezione odnośniki mailto: (czyli te, któ-



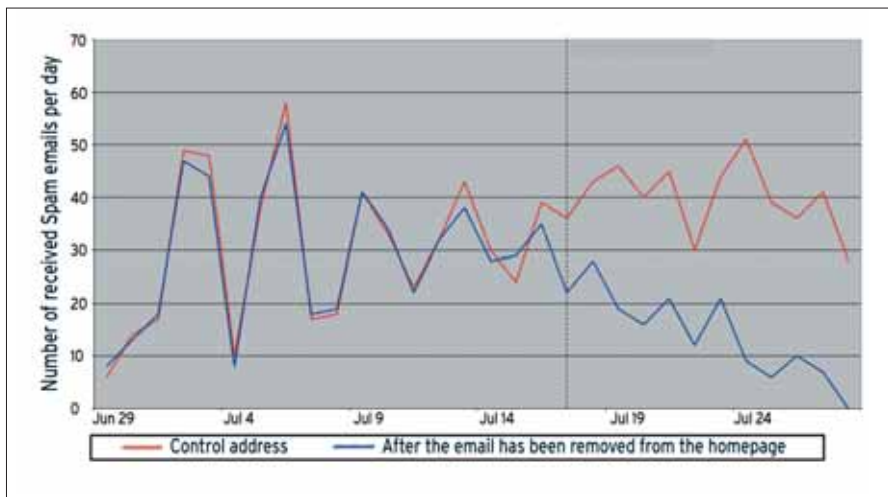
Rysunek 1: Spamerzy zdobywają adresy pocztowe głównie poprzez strony internetowe.

re zawierają adresy pocztowe). Następnie, korzystając z kolejnych odnośników, przechodzą do poszczególnych podstron i powtarzają tę procedurę dla każdej z nich.

W ten sposób nadawcy niechcianej poczty znajdują się w końcu na stronie bez dalszych odnośników. Podobnie wyglądają techniki służące do przeszukiwania całej sieci Internet (przeszukiwanie stron znajdujących w wyszukiwarkach internetowych). Napisanie programu, który wykonywałby automatycznie takie przeszukiwanie stron internetowych, jest dość proste – wystarczy tzw. „pająk” lub „żniwiarka” (ang. „spider” i „harvester”). Po usunięciu powtarzających się adresów poczty elektronicznej osoba rozsyłająca niechcianą pocztę otrzymuje listę potencjalnych ofiar.

Przy pomocy standardowych narzędzi Linuksa (wget, sed, tr, sort i uniq) możemy zaprojektować prostą żniwiarkę. Wyniki mogą być zadziwiające.

Program wget odwiedza strony internetowe w sieci, sed przeszukuje strony pod kątem adresów pocztowych. Z kolei tr ujednolica stosowanie małych i wielkich liter,



Rysunek 2: Warto ze swoich stron WWW usunąć adres poczty elektronicznej.

a sort układa odszukane adresy w kolejności alfabetycznej, aby przy pomocy uniq usunąć powtarzające się pozycje.

Po przetestowaniu tej metody na naszej stronie domowej, zdołaliśmy zebrać ponad 90 różnych adresów pocztowych w czasie ośmiu minut. Gdybyśmy wybrali stronę domową zawierającą większą liczbę odnośników, ignorując jednocześnie przetwarzanie pliku **robots.txt** [2], otrzymalibyśmy znacznie więcej adresów pocztowych w tym samym czasie.

Całkowite zrezygnowanie z umieszczenia adresu poczty elektronicznej na stronie internetowej jest rozwiązaniem skutecznym, ale nie lubianym przez właścicieli serwisów. Bądź co bądź, strona internetowa ma być w końcu dodatkowym kanałem komunikacji pomiędzy autorem strony a użytkownikami, którzy ją odwiedzają. W niektórych krajach (np. w Niemczech) właściciele stron internetowych są zobligowani prawnie do podawania adresu pocztowego na swoich stronach.

Sposoby ukrywania adresu

Obecnie istnieje wiele metod ukrywania adresu pocztowego. Jedną z nich jest metoda `usun_to_aby_do_mnie_napisac`, w której zamiast standardowego adresu `ktos@przyklad.com` używamy np. adresu `ktos@usun_to_aby_do_mnie_napisac.przyklad.com`. Nie wszyscy użytkownicy pamiętają o usunięciu środkowej części adresu przed kliknięciem przycisku Odpowiedz, a czasami zdarza się, że nie wiemy, którą część adresu należy usunąć. Ponownie jednak napotykałyśmy problem wymogu podawania prawidłowych adresów pocztowych na stronach domowych w niektórych krajach, co uniemożliwia stosowanie tego rodzaju ukrywania adresu pocztowego.

Dodatkowo, ostatnio pojawiła się tendencja fałszowania adresów nadawcy przez osoby rozsyłające niechcianą pocztę elektroniczną. Komunikaty o błędach nie docierają do prawdziwego nadawcy wiadomości

SŁOWNICZEK

Heurystyka: (z greckiego „heuriskein”: znajdować, odkrywać) poszukiwanie wzorców w oparciu o zebrane wcześniej doświadczenie. Wyszukiwanie heurystyczne jest znacznie szybsze niż każdorazowe wykonywanie precyzyjnych obliczeń sprawdzających.

SMTP: Protokół służący do przesyłania poczty elektronicznej (ang. Simple Mail Transfer Protocol).

robots.txt: Pliki o takiej nazwie znajdujące się na serwerach sieciowych zawierające informacje o stronach, które m.in. nie są automatycznie wyświetlane w wyszukiwarkach.

Znaczniki HTML: Kodowanie HTML z zapi-

sem znaków w formacie uwzględniającym przedrostek `&#`, po którym występuje kod **ASCII** i znak średnika: tak więc kod `u` oznacza literę „u”.

ASCII: Skrót od „American Standard Code for Information Interchange”) – standard przypisujący wszystkim literom, cyfrom i znakom specjalnym określoną liczbę (tzw. kod ASCII).

JavaScript: Język skryptowy wykorzystywany na stronach internetowych. Jeżeli przeglądarka potrafi obsługiwać język Java, wszystkie polecenia dla tego języka są interpretowane prawidłowo i wykonywane.

XOR: Różnica symetryczna (ang. 'Exclusive OR') to metoda obliczeniowa spotykana głównie w matematycznym systemie dwójkowym, reprezentowana w języku JavaScript znakiem `^`. Może być stosowana do symetrycznego szyfrowania danych.

Flash: Opatentowany format firmy Macromedia, obsługujący animacje, technikę wideo, dźwięk oraz obrazy graficzne i umożliwiający wyświetlanie ich na stronach internetowych. Aby elementy w tym formacie były prawidłowo wyświetlane na stronie internetowej, do przeglądarki należy doinstalować stosowną wtyczkę.

(czyt. spamera), lecz do niczego nie spodziewających się firm i osób prywatnych, co może w rezultacie doprowadzić do przerwania pracy serwerów ze względu na zbyt duże obciążenie.

Raport CDT [1], o którym mówiliśmy wcześniej, sugeruje rozwiązanie tego problemu poprzez szyfrowanie adresów pocztowych umieszczanych na stronach internetowych przy pomocy **znaczników HTML**, dzięki czemu adres `user@example.com` będzie mieć postać:

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

Przeglądarki internetowe bez problemu odczytują adres w takiej postaci, ale programy-żniwiarki nie są w stanie rozpoznać takiego kodu źródłowego.

W raporcie opracowanym przez CDT adresy kodowane tym sposobem nie otrzymywały żadnej niechcianej poczty. Nasza prymitywna żniwiarka odnalazła 10 adresów tego typu.

Jako że jest to coraz częściej stosowane zabezpieczenie przed spamem, należy spodziewać się, że w niedalekiej przyszłości programy-pająki „nauczą się” rozpoznawać i automatycznie zamieniać adresy pocztowe tego typu do zrozumiałej postaci. Na dłuższą metę adresy kodowane tym sposobem nie są wiele lepsze niż adresy pocztowe zapisane prostym tekstem, bez kodowania.

JavaScript przychodzi z odsieczą?

Większość pajaków nie potrafi obsługiwać **JavaScript**. Umożliwia to właścicielom stron internetowych zakamuflowanie adresów pocztowych na ich stronach przy pomocy Java.

Na Listingu 1 pokazano sposób przechowywania adresu pocztowego w nagłówku

strony HTML, a następnie użycie polecenia `JavaScript document.write()` do wyświetlenia tego adresu na stronie. Dzięki takiemu rozwiązaniu adres pocztowy będzie widziany na stronie dla każdego odwiedzającego, ale dla mechanizmów wyszukiwania adresów pocztowych pozostanie niewidoczny.

Istnieje wiele możliwości przypisania wartości do zmiennej. Najprostszą z nich i najłatwiejszą do zarządzania, jest wykorzystanie zewnętrznego pliku JavaScript do przechowywania adresów pocztowych, które będą odczytywane na żądanie przeglądarki. Programy-żniwiarki nie biorą obecnie pod uwagę tego typu plików w swoich poszukiwaniach.

Trzeba jednak pamiętać, że niektóre przeglądarki mają kłopoty z odwołaniami do plików zewnętrznych przy pomocy funkcji `document.write()`. Ponadto przebiegły spamer znajdzie metodę przeglądania wszystkich plików JavaScript znajdujących się na stronie, aby później przeszukać je pod kątem ukrytych w nich adresów pocztowych.

Istnieje pewne obejście problemu z poleceniem `document.write()`, z którym nie radzą sobie niektóre przeglądarki. Odnośnik HTML (``), bezpośrednio wskazujący adres pocztowy, zastępujemy poleceniem JavaScript. Z kolei do wyświetlenia adresu na stronie skorzystamy z języka HTML (a nie Javy, jak w poprzednim przykładzie). Na Listingu 2 pokazaliśmy przykład zastosowania funkcji odnośnika JavaScript `document.location.href`.

Steganografia

Aby zabezpieczyć się przed programem wyszukującym adresy pocztowe, wystarczy skorzystać z wyjątkowo prostego szyfrowania różnicą symetryczną (**XOR**). Metoda ta jest bardzo skuteczna w przypadku osób rozsyłających niechcianą pocztę elektro-

niczną – szczególnie dla oczekujących maksymalnych efektów w jak najkrótszym czasie. Mimo że program-żniwiarka nadal może zdobyć adres wprost ze źródła HTML, sam adres pocztowy znajduje się tam w formie zaszyfrowanej. Aby uzyskać prawidłowy adres pocztowy, należałoby zrozumieć i odwrócić proces przeprowadzony przez polecenia JavaScript.

Na Listingu 3 pokazaliśmy, jak zaszyfrować nazwę użytkownika w adresie pocztowym, czyli element adresu znajdujący się przed znakiem `@`. Funkcja JavaScript o nazwie `document.location.hostname` pobiera brakujące elementy w formie niezasyfrowanego tekstu z paska adresu przeglądarki. Sposób ten działa jedynie w przypadku, gdy strona domowa serwera jest stroną domową dla adresu pocztowego. Przy pomocy tego samego algorytmu możemy zaszyfrować pozostałe składniki adresu pocztowego.

Procedurę szyfrowania można łatwo rozwinać, należy jednak pamiętać, aby zarówno procedura szyfrowania jak i klucz szyfrujący znajdowały się wewnątrz skryptu. W ten sposób każdy użytkownik, który uruchomi skrypt JavaScript, zobaczy adres pocztowy w formie niezasyfrowanej, co jest oczywiście niezbędne dla użytkowników Internetu odwiedzających naszą stronę.

Największą zaletą tej metody jest to, że programy klienta, które nie potrafią interpretować języka JavaScript, nie będą w stanie odczytać adresu pocztowego. Obecnie (i miejmy nadzieję, że będzie tak jeszcze długo) programy-pająki nie są w stanie odczytywać adresów pocztowych.

Możemy teraz w prosty sposób połączyć poszczególne skrawki kodu JavaScript znajdujące się w tym artykule: np. dołączyć szyfrowanie adresu z Listingu 3 do Listingu 2.

Byłoby wspaniale, gdybyśmy mogli powiedzieć, że problemy ze skryptami Java mają wyłącznie żniwiarki, ale niestety tak nie jest: wiele przeglądarek tekstowych, np.

Listing 1: Przechowywanie adresu w nagłówku HTML

```
<HTML>
<HEAD>
<TITLE>Sample Page</TITLE>
<SCRIPT LANGUAGE='JavaScript'>
<!--
mailaddress = &#109;
'uzytkownik@przyklad.com';
//-->
</SCRIPT>
</HEAD>
<BODY>
[... ]
<SCRIPT LANGUAGE='JavaScript'>
JavaScript'&#109;';
<!--
document.write('<A HREF='&#109;
```

Listing 2: JavaScript z odnośnikami adresu

```
<HTML>
<HEAD>
<TITLE>Sample Page</TITLE>
<SCRIPT LANGUAGE='JavaScript'>
<!--
mailaddress = 'uzytkownik@przy-
klad.com';
function mailMe()
{
  document.location.href='mail-
to:'+mailaddress;
}
//-->
</SCRIPT>
</HEAD>
<BODY>
[... ]
<A HREF='javascript:mail-
Me();'>Mail sender</A>
[... ]
</BODY>
</HTML>
```

Lynx, napotyka bowiem na ten sam problem. Ponadto wielu użytkowników wyłącza obsługę Javy w swoich przeglądarkach graficznych ze względów bezpieczeństwa. Pamiętajmy zatem, że strony ze skryptami JavaScript skutecznie uniemożliwią takim użytkownikom kontakt z nami.

Metody steganograficzne, opisane w naszym artykule, sprawdzają się dla adresów pocztowych i odnośników internetowych. Gdyby metody te zastosowano globalnie, programy-żniwiarki miałyby twarde orzech do zgryzienia. Z drugiej jednak strony, utrudnilibyśmy także działanie takim przeglądarkom jak Google, a użytkownicy musieliby odczuć prawdziwy ciężar Internetu na przeglądarkach bez obsługi JavaScript.

Aby nie zmuszać odwiedzających naszą stronę internautów do korzystania z JavaScript, możemy użyć prostej sztuczki, która niestety ponownie nie dotyczy użytkowników przeglądarek tekstowych. Wystarczy, że na naszej stronie znajdzie się obraz graficzny z zapisanym adresem pocztowym. Jako że sam adres nie pojawia się w tekstonej treści strony HTML, żniwiarki nie mają szans na zdobycie takiego adresu. Spamerzy nie mogą polegać na programach do rozpoznawania tekstu (OCR), aby w ten

Listing 3: Zaszyfrowany adres pocztowy

```
<HTML>
<HEAD>
<TITLE>Sample page</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) + 'z'
document.location.hostname;
//-->
</SCRIPT>
</HEAD>
<BODY>
[... ]
<SCRIPT LANGUAGE='JavaScript'>
<!--
document.write('<A HREF='mailto:'+mailaddress+'>'+mailaddress+'</A>');
//-->
</SCRIPT>
[... ]
</BODY>
</HTML>
```

sposób automatycznie zdobyć upragniony adres pocztowy – w końcu nie każdy obrazek umieszczony na stronie zawiera adres pocztowy. Jest to chyba jedyne rozwiązanie dla właścicieli stron internetowych w krajach, gdzie publikacja własnego, pełnego adresu pocztowego jest wymagana prawem.

Możemy także stworzyć odnośnik bez ujawniania adresu pocztowego: wystarczy użyć formularza kontaktowego, bez ujawniania adresu odbiorcy, dzięki czemu będziemy mogli otrzymywać wiadomości pocztowe od osób odwiedzających naszą stronę internetową.

Kolejnym rozwiązaniem może być animacja **Flash**, wyświetlająca aktywny adres pocztowy (reagujący na kliknięcie myszy). Musimy mieć jednak świadomość, że takie rozwiązanie uniemożliwi kontakt jeszcze większej liczbie osób odwiedzających naszą stronę niż w przypadku umieszczenia na

stronie zwykłego obrazu graficznego.

Odzew

Jeżeli posiadamy własną domenę internetową, dzięki stronie dynamicznej możemy znaleźć pochodzenie żniwiarek adresów pocztowych. Tworzymy w tym celu odnośnik do adresów poczty elektronicznej, który jest nieustannie aktualizowany na stronie (możemy użyć do tego celu aktualnej daty i godziny oraz adresu IP aktualnego użytkownika). Zmuszamy w ten sposób użytkownika do podania własnego adresu pocztowego przy logowaniu do naszej strony internetowej.

Jeżeli adres pocztowy jest fałszywy, możemy poznać adres źródłowy dla tej wiadomości – a jest to o tyle ważne, że może to być jeden z głównych dowodów w sądzie przeciwko osobie rozsyłającej niechcianą pocztę reklamową, czyli spam. ■

AUTOR

Tobias Eggendorfer pracuje jako konsultant IT w Monachium (Niemcy). Walka ze spamem – od strony technicznej, ale nie tylko – jest jednym z jego głównych zajęć zawodowych.

INFO

- [1] Raport „Dlaczego dostajemy tyle spamu?": <http://www.cdt.org/speech/spam/030319spamreport.html>
- [2] robots.txt: <http://www.robots.txt.org>