Jak wysyłany jest spam?

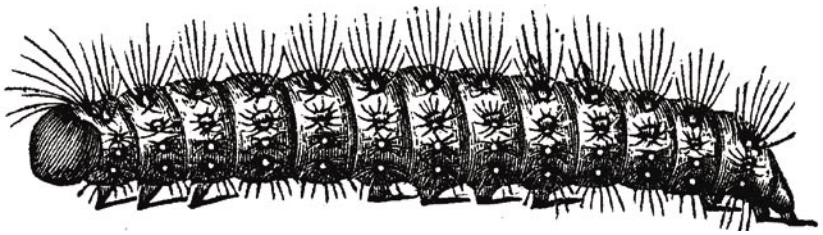
Tomasz Nidecki

Artykuł opublikowany w numerze 2/2004 magazynu „Hakin9”

Magazyn „Hakin9”, Wydawnictwo Software, ul. Lewartowskiego 6, 00-190 Warszawa, piotr@software.com.pl

Jak wysyłany jest spam

Tomasz Nidecki



Spamerzy często wykorzystują słabo zabezpieczone systemy. Kłopoty i koszty związane z wysyłką nierzadko dziesiątek lub setek tysięcy listów przenoszone są na osoby trzecie. Dowiedzmy się, na czym polegają stosowane przez spamerów techniki i jak się przed nimi zabezpieczyć.

Masowa wysyłka poczty elektronicznej pochłania bardzo wiele zasobów. By ją przeprowadzić niezbędne jest szybkie łącze i dedykowany komputer. Jeśli nawet spamer dysponuje takimi zasobami, to wysyłka może zająć wiele godzin. Dostawcy usług internetowych nie są zaś zazwyczaj zachwyceni, jeśli ich łącza wykorzystuje się do spamowania i spamer może stracić połączenie z siecią, zanim uda mu się rozesłać dużą liczbą wiadomości. Jeśli zostanie złapany, może też ponieść poważne konsekwencje prawne lub finansowe.

Aby przyspieszyć i usprawnić wysyłkę spamerzy stosują dwie podstawowe metody. Pierwsza polega na minimalizacji czasu niezbędnego do wysłania listu. Zwana jest *fire and forget* czyli wyślij i zapomnij. Przy stosowaniu tej metody komputer rozsyłający spam nie oczekuje na odpowiedzi od serwerów, z którymi się kontaktuje. Druga ze stosowanych metod polega na kradzieży zasobów należących do osób postronnych, które źle skonfigurowały system lub padły ofiarą wirusa. Większość kosztów, a nierzadko także odpowiedzialność za rozsyłanie spamu spada na nich, a spamer jest bezkarny.

Protokół SMTP

Aby zrozumieć metody stosowane przez spamerów konieczne jest poznanie zasad funkcjonowania najpopularniejszego protokołu do przesyłania poczty elektronicznej – SMTP. Opiera się on, podobnie jak większość protokołów stosowanych w Internecie, na prostych poleceniach tekstowych.

Etapy przesyłania poczty

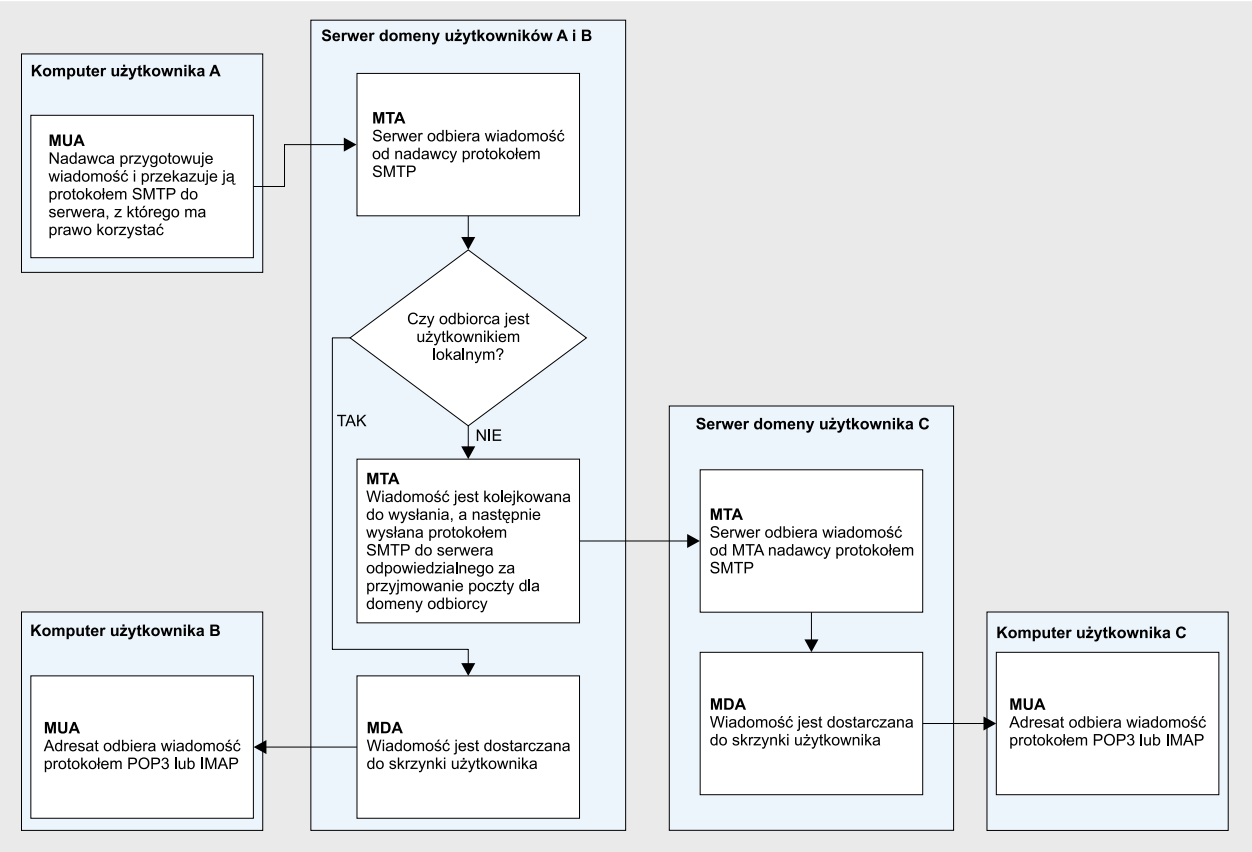
Poczta elektroniczna przesyłana jest w kilku etapach (patrz Rysunek 1). Dla lepszego ich zrozumienia załóżmy, że chcemy przesłać list

Z artykułu dowiesz się...

- w jaki sposób spamerzy rozsyłają spam (korzystając z komputerów niewinnych osób),
- jak zabezpieczyć swój serwer przed wykorzystaniem przez spamerów,
- jak działa protokół SMTP,
- co to jest *open relay*, *open proxy* i *zombie*.

Co powinieneś wiedzieć...

- jak stosować podstawowe narzędzia w systemie Linux.



Rysunek 1. *Etapy przesyłania poczty elektronicznej*

od *hakin9@hakin9.org* do *nobody @example.com*. Użytkownik wysyłający list korzysta z programu *Mozilla Thunderbird* w sieci lokalnej, a odbiorca – z *Microsoft Outlook Express* za pośrednictwem łącza typu dial-up.

W pierwszym etapie program *Mozilla Thunderbird* kontaktuje się z serwerem SMTP podanym w ustawieniach konta użytkownika *hakin9@hakin9.org* – *mail.software.com.pl*. List przesyłany jest do serwera protokołem SMTP. W drugim – *mail.software.com.pl* zagląda do wpisów w serwerach DNS. Dowiaduje się, że za odbiór poczty do domeny *example.com* odpowiedzialny jest *mail.example.com*. Informację tę znajduje w tzw. rekordzie MX (*Mail Exchanger*) publikowanym przez DNS odpowiedzialny za domenę *example.com* (możemy ją uzyskać za pomocą programów *host* lub *dig*: `host -t mx example.com` lub `dig example.com mx`).

W trzecim – *mail.software.com.pl* łączy się z *mail.example.com* i przekazuje mu list. W czwartym – *mail.example.com* dostarcza odebrany list do lokalnej skrzynki pocztowej użytkownika *nobody*. W piątym – użytkownik skrzynki *nobody* łączy się przez łącze dial-up z serwerem *mail.example.com* protokołem POP3 (lub IMAP) za pomocą programu *Outlook Express* i pobiera list.

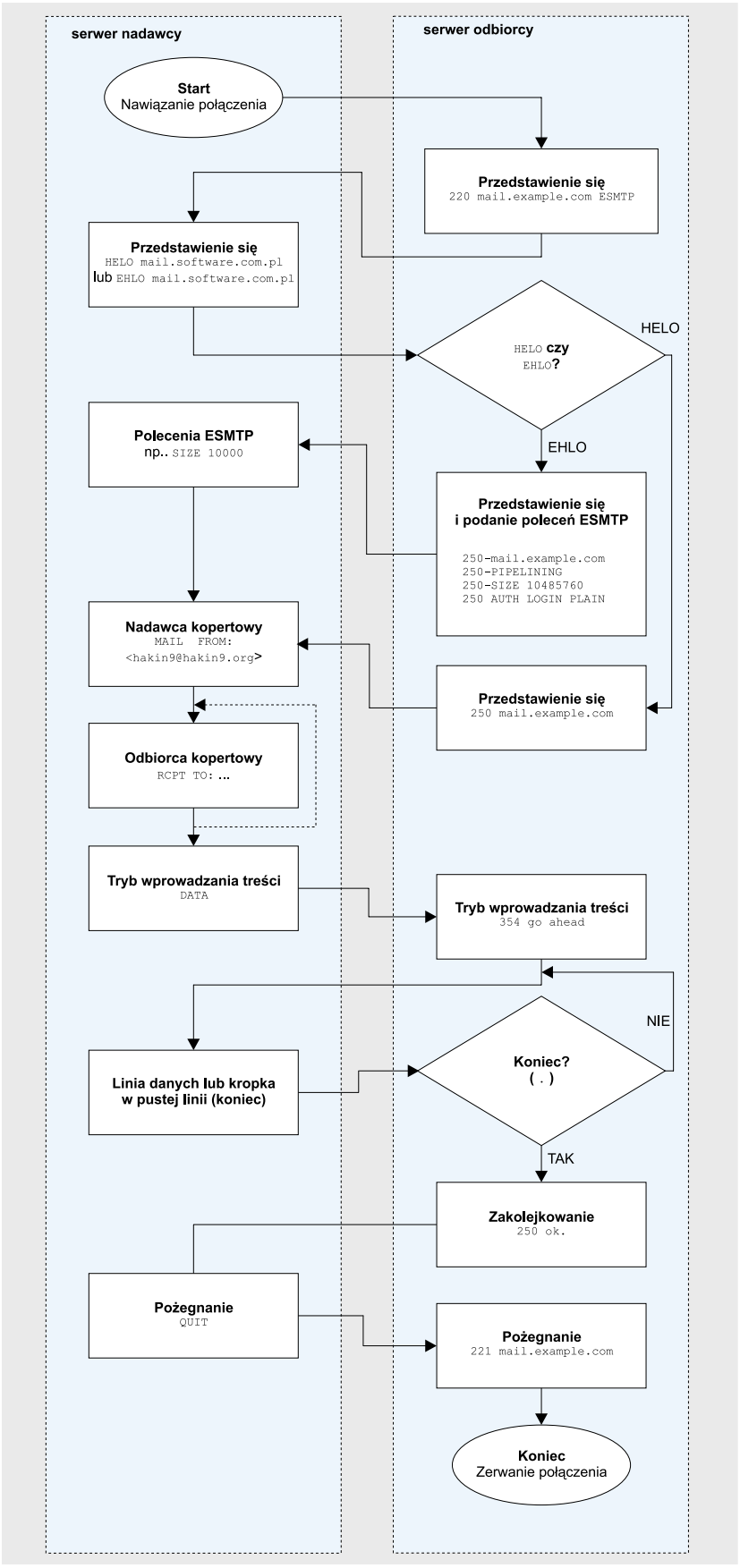
Historia SMTP

Prekursorem SMTP był program *SNDMSG* (*Send Message*), wykorzystany w 1971 roku przez Raya Tomlinsona (w połączeniu z jego własnym projektem – *CYPNET*) do stworzenia aplikacji umożliwiającej przesyłanie poczty elektronicznej w sieci *ARPANET*. Rok później program wykorzystywany w *Arpanecie* do przesyłania plików – *FTP*, został poszerzony o polecenia `MAIL` i `MLFL`. Pocztę przesyłano za pomocą *FTP* do 1980 roku – wtedy powstał pierwszy standard protokołu poczty elektronicznej – MTP (*Mail Transfer Protocol*), opisany w dokumencie RFC 772. MTP kilkukrotnie modyfikowano (RFC 780, 788), a w 1982 roku, w RFC 821 Jonathan B. Postel opisał *Simple Mail Transfer Protocol*.

SMTP w podstawowej formie nie spełnił jednak wszystkich oczekiwań. Powstało więc wiele dokumentów opisujących rozszerzenia protokołu. Do najważniejszych z nich należą:

- RFC 1123 – wymagania dla serwerów internetowych (obejmują też SMTP),
- RFC 1425 – wprowadzenie standardu rozszerzeń protokołu SMTP – ESMTP,
- RFC 2505 – zbiór sugestii dotyczących ochrony antyspamowej serwerów,
- RFC 2554 – autoryzacja połączeń – wprowadzenie polecenia `AUTH`,

Aktualny standard SMTP opisany został w 2001 roku w RFC 2821. Komplet RFC zamieszczamy na naszym CD.



Rysunek 2. *Etapy komunikacji za pomocą SMTP*

Zdarza się, że list przebywa nieco dłuższą drogę. Nadawca mógłby korzystać z oddzielnych serwerów pocztowych (SMTP), np.: *odbior.software.com.pl* i *wysylka.software.com.pl*. Wtedy list odbierany byłby od użytkowników przez *odbior.software.com.pl*, przekazywany do *wysylka.software.com.pl*, a następnie wysyłany do *mail.example.com*. Podobnie w przypadku *mail.example.com* – za odbiór i dostarczenie poczty do użytkownika mogą być odpowiedzialne różne serwery.

Programy biorące udział w przesyłaniu poczty

W przesyłaniu poczty bierze udział kilka programów:

- Program stosowany przez użytkownika końcowego, służący do odbioru i wysyłania, a także czytania i pisania listów, nazywany jest MUA – *Mail User Agent*. Przykładami MUA są: *Mozilla Thunderbird*, *Outlook Express*, *PINE*, *Mutt*.
- Część serwera odpowiedzialna za komunikowanie się z użytkownikami (odbiór poczty) oraz wysyłanie i odbieranie poczty od innych serwerów nazywana jest MTA – *Mail Transfer Agent*. Najpopularniejsze to: *Sendmail*, *qmail*, *Postfix*, *Exim*.
- Część serwera odpowiedzialna za dostarczenie poczty do lokalnego użytkownika nazywana jest MDA – *Mail Delivery Agent*. Przy-

Następca SMTP?

Prof. Dan Bernstein, autor *qmaila*, opracował protokół o nazwie QMTP (*Quick Mail Transfer Protocol*), który miał zastąpić SMTP. Eliminuje on wiele problemów występujących w SMTP, lecz jest niekompatybilny ze swoim poprzednikiem. Niestety nie został zaimplementowany praktycznie nigdzie poza *qmailem*. Jest wykorzystywany w praktyce m.in. przez serwery pocztowe Wirtualnej Polski.

Więcej informacji o QMTP pod adresem: *http://cr.yp.to/proto/qmtp.txt*

kłady samodzielnych MDA to: *Maildrop*, *Procmail*. Większość MTA posiada własne mechanizmy dostarczania poczty lokalnym użytkownikom, więc nie zawsze konieczne jest korzystanie z dodatkowych MDA.

Etapy komunikacji w SMTP

Przesłanie listu za pomocą SMTP jest podzielone na kilka etapów. Oto przykładowa sesja SMTP między serwerami *mail.software.com.pl*, a *mail.example.com*. Dane wysyłane przez *mail.software.com.pl* oznaczono znakiem >, a dane odbierane z *mail.example.com* – <.

Po nawiązaniu połączenia *mail.example.com* przedstawia się:

```
< 220 mail.example.com ESMTP Program
```

informując, iż jego pełna nazwa hosta (FQDN – *Fully Qualified Domain Name*) to *mail.example.com*. Dowiadujemy się też, że możemy korzystać z poleceń ESMTP (rozszerzonego SMTP – patrz Ramka) i że stosowany MTA to *Program*. Nazwa programu jest opcjonalna – niektóre MTA, np. *qmail*, nie podają jej.

Przedstawiamy się:

```
> HELO mail.software.com.pl
```

w odpowiedzi otrzymujemy:

```
< 250 mail.example.com
```

co oznacza, iż *mail.example.com* jest gotów do odbioru poczty. Następnie podajemy tzw. adres nadawcy kopertowego – adres pocztowy, pod który ma być skierowany ewentualny zwrot listu:

```
> MAIL FROM:<hakin9@hakin9.org>
< 250 ok
```

Podajemy adresy, na które chcemy przesłać list:

```
> RCPT TO:<test1@example.com>
< 250 ok
> RCPT TO:<test2@example.com>
< 250 ok
> RCPT TO:<test3@example.com>
< 250 ok
```

Następnie, po poleceniu `DATA` przekazujemy nagłówki i treść listu. Nagłówki oddzielamy od treści pustą linią, a list kończymy kropką w oddzielnej linii:

```
> DATA
< 354 go ahead
> From: nikt@hakin9.org
> To: wszyscy@example.com
> Subject: Nic
>
> To jest test
> .
< 250 ok 1075929516 qp 5423
```

Po wysłaniu listu możemy zakończyć połączenie:

```
> QUIT
< 221 Bye
```

Serwer nie zawsze jest zdolny do wykonania naszych poleceń. Jeśli otrzymamy kod zaczynający się od cyfry 4 (kod z serii 4xx) oznacza to, że serwer odmawia tymczasowo przyjęcia listu. Powinniśmy spróbować wysłać list ponownie, nieco póź-

Tabela 2. *Przegląd najważniejszych kodów zwrotnych SMTP*

Kod	Opis
220	Usługa aktywna – serwer wita się informując, że można mu przesyłać polecenia
250	Polecenie zostało przyjęte
354	Można rozpocząć wprowadzanie treści listu
450	Skrzynka użytkownika chwilowo zajęta (np. zablokowana przez inny proces)
451	Błąd lokalny przy przetwarzaniu poczty
452	Chwilowy brak miejsca na dysku
500	Nie ma takiego polecenia
501	Błąd składniowy w poleceniu lub jego parametrach
502	Polecenie nie zostało zaimplementowane
550	Skrzynka użytkownika niedostępna
552	Przekroczono limit miejsca na dysku

Pełną listę kodów oraz zasady ich tworzenia można znaleźć w RFC 2821 (na naszym CD).

Tabela 1. *Przegląd najczęściej używanych poleceń protokołu SMTP*

Polecenie	Opis
HELO <FQDN>	Przedstawienie się serwerowi
EHLO <FQDN>	Przedstawienie się serwerowi połączone z prośbą o podanie listy dostępnych poleceń ESMTP
MAIL FROM:<adres>	Podanie adresu nadawcy kopertowego – adresu, na który ma być kierowany ewentualny zwrot listu
RCPT TO:<adres>	Podanie adresu odbiorcy wiadomości
DATA	Przejście w tryb odbioru treści wiadomości
AUTH <mechanizm>	Autoryzacja połączenia (ESMTP) – najczęściej stosowane mechanizmy to: LOGIN, PLAIN i CRAM-MD5

Rozszerzoną listę poleceń SMTP i ESMTP można znaleźć pod adresem http://fluffy.codeworks.gen.nz/esmtp.html

Listing 1. *Najprostszy open relay.*

```
$ telnet lenox.designs.pl 25
< 220 ESMTP xenox
> helo hakin9.org
< 250 xenox
> mail from:<hakin9@hakin9.org>
< 250 Ok
> rcpt to:<nobody@example.com>
< 250 Ok
> data
< 354 End data with
  <CR><LF>.<CR><LF>
> Subject: test
>
> To jest test
> .
< 250 Ok: queued as 17C349B22
> quit
< 221 Bye
```

niej. Jeśli otrzymamy kod zaczynający się od 5, oznacza to, że serwer ostatecznie odmawia odebrania od nas poczty i ponowne próby nic nie dadzą. Listę najważniejszych poleceń oraz kodów otrzymywanych od serwera SMTP przedstawiamy w Tabelach 1 i 2.

Serwery *open relay*

Kiedy został stworzony protokół SMTP, nie istniał problem spamu i każdy użytkownik mógł skorzystać z dowolnego serwera do przesłania swojej wiadomości w świat. Teraz, gdy spamerzy szukają okazji, by wykorzystać czyjś serwer do rozesłania tysięcy wiadomości, takie podejście już się nie sprawdza. Serwery umożliwiające wysyłanie poczty w świat bez autoryzacji nazywane są *open relay*.

Każdy serwer, który umożliwia nieautoryzowanym użytkownikom wysyłanie poczty, prędzej czy później zostanie wykorzystany przez spamerów. To z kolei może mieć poważne konsekwencje. Po pierwsze: może spowodować redukcję mocy przerobowych serwera, który zamiast odbierać i dostarczać wiadomości autoryzowanych użytkowników, będzie wysyłał spam. Po drugie: dostawca łącza internetowego może wymówić umowę w związku z faktem, iż serwer jest wykorzystywany do nielegalnego i niemoralnego procederu. Po trzecie: adres IP serwera trafi na czarne listy i wiele innych serwerów nie będzie odbierało od niego żadnej poczty (usunięcie IP z wielu czarnych list jest trudne, a czasem nawet niemożliwe).

Listing 2. *Serwer open relay umożliwiający wysyłanie tylko istniejącym użytkownikom.*

```
$ telnet kogut.o2.pl 25
< 220 o2.pl ESMTP Wita
> helo hakin9.org
< 250 kogut.o2.pl
> mail from:<ania@o2.pl>
< 250 Ok
> rcpt to:<hakin9@hakin9.org>
< 250 Ok
> data
< 354 End data with
  <CR><LF>.<CR><LF>
> Subject: test
>
> To jest test
> .
< 250 Ok: queued as 31B1F2EEA0C
> quit
< 221 Bye
```

Listing 3. *Serwer multistage open relay umożliwiający wysyłanie tylko istniejącym użytkownikom.*

```
$ telnet smtp.poczta.onet.pl 25
< 220 smtp.poczta.onet.pl ESMTP
> helo hakin9.org
< 250 smtp.poczta.onet.pl
> mail from:<ania@buziaczek.pl>
< 250 2.1.0 Sender syntax Ok
> rcpt to:<hakin9@hakin9.org>
< 250 2.1.5 Recipient address
  syntax Ok;
  rcpt=<hakin9@hakin9.org>
> data
< 354 Start mail input;
  end with <CRLF>.<CRLF>
> Subject: test
>
> To jest test
> .
< 250 2.6.0 Message accepted.
> quit
< 221 2.0.0
  smtp.poczta.onet.pl Out
```

Wykorzystanie open relay

Sprawdźmy, jak łatwe jest wykorzystanie *open relay* do przesłania spamu. Przykładowo posłużmy się jednym ze źle skonfigurowanych polskich serwerów wykorzystywanych przez spamerów – *lenox.designs.pl*. Jak widać na Listingu 1 w tym przypadku nie musieliśmy podejmować żadnych specjalnych kroków, by móc wysłać list. Serwer traktuje każdego łączącego się z nim użytkownika jako uprawnionego do wysyłania poczty. To najgroźniejszy, bo najłatwiejszy do wykorzystania typ serwera *open relay*.

Istnieją inne, nieco trudniejsze do wykorzystania przez spamerów *open relay*. Przykładem może być jeden z kilku źle skonfigurowanych serwerów pocztowych polskiego portalu *O2* – *kogut.o2.pl*. Jak widać na Listingu 2 – wystarczyło domyśleć się nazwy użytkownika, by podszyć się pod niego i wysłać list. W przypadku niektórych serwerów wystarczy podać nazwę lokalnej domeny – nie musi nawet istnieć użytkownik pod którego się podszywamy.

Podobną sytuację widać na Listingu 3 – znów mamy tu do czynienia z serwerem pocztowym jednego z większych polskich portali – tym razem *Onet* (co ciekawe, *Onet* deklaruje się jako aktywnie walczący ze spamem...). Jest to tzw. *multistage open relay* (*multihop open relay*). Oznacza to, że list jest przyjmowany przez jedno IP, a wysyłany przez inne.

Listing 4. *Nagłówki Received poczty otrzymanej z serwera multistage open relay.*

```
Received: from smtp8.poczta.onet.pl (213.180.130.48)
  by mail.hakin9.org with SMTP; 23 Feb 2004 18:48:11 -0000
Received: from  mail.hakin9.org ([127.0.0.1]:10248 "helo hakin9.org")
  by ps8.test.onet.pl with SMTP id <S1348420AbUBWSrW>;
  Mon, 23 Feb 2004 19:47:22 +0100
```

Jak uniknąć zostania *open relayem*?

Protokół SMTP umożliwia:

- przyjęcie poczty od użytkownika (MUA) i przesłanie jej do innego serwera (MTA),
- przyjęcie poczty od innego serwera (MTA) i przesłanie jej do użytkownika lokalnego (MUA),
- przyjęcie poczty od jednego serwera (MTA) i przesłanie jej do drugiego serwera (MTA).

Nie ma żadnej różnicy między sposobem przesyłania listu przez MUA i przez MTA. Najważniejsze jest to, czy adres IP nadawcy jest zaufany (np. w sieci lokalnej) i czy adresat jest w domenie lokalnej, czy zewnętrznej.

Przesyłka poczty poza nasz serwer nazywana jest relayowaniem (*relaying*). Aby spamerzy nie mogli wykorzystać naszego serwera do prowadzenia swojej działalności nie możemy zezwolić na relayowanie bez autoryzacji. Dlatego też przy konfiguracji serwera SMTP należy poczynić następujące założenia:

- Jeśli list jest przeznaczony do jednej z domen obsługiwanych przez nasz serwer – musi zostać przyjęty bez autoryzacji.
- Jeśli list jest wysyłany przez użytkownika lokalnego (z MUA na serwerze), w sieci lokalnej lub z autoryzowanego, stałego IP, a adresatem jest użytkownik zewnętrzny, list może zostać przyjęty bez autoryzacji (sugerowane jest jednak wymaganie autoryzacji).
- Jeśli list jest wysyłany przez użytkownika zewnętrznego (np. z dynamicznego IP), a adresatem jest użytkownik zewnętrzny, list nie może zostać przyjęty bez autoryzacji.

Aby się o tym dowiedzieć musimy przeanalizować nagłówki `Received` (patrz Ramka) otrzymanego listu. Jak widać na Listingu 4 list został przyjęty przez *ps8.test.onet.pl* (213.180.130.54), a wysłany do odbiorcy przez *smtp8.poczta.onet.pl* (213.180.130.48). To utrudnia wykrycie, że serwer jest skonfigurowany jako *open relay*, ale wcale nie przeszkadza w wykorzystaniu go do rozsyłania spamu.

Innym typem *open relay* są serwery z błędnie skonfigurowaną autoryzacją nadawcy (SMTP-AUTH). Umożliwiają wysłanie poczty po podaniu dowolnego loginu i hasła. Najczęściej zdarza się to początkującym administratorom *qmaila*, którzy nie przeczytali dokumentacji łatki SMTP-AUTH i w nieprawidłowy sposób wywołują *qmail-smtpd.*

Program *qmail-smtpd* z zaaplikowaną łatką wymaga trzech argumentów: FQDN, programu sprawdzającego hasło (kompatybilnego z *checkpassword*) i dodatkowego parametru dla programu sprawdzającego hasło. Przykład: `qmail-smtpd hakin9.org /bin/checkpassword /bin/true`. Najczęstszym błędem jest podanie `/bin/true` jako drugiego parametru – wtedy sprawdzenie hasła jest zawsze udane (niezależnie od podanego loginu i hasła). Spamer może również spróbować ataku słownikowego – dobrze więc, by hasła użytkowników stosowane do autoryzacji SMTP nie były zbyt banalne.

Nagłówki Received

Nagłówki `Received` są obowiązkowym elementem każdego listu. Każdy serwer pocztowy przez który przejdzie list dodaje własny nagłówek `Received` nad pozostałymi, nadanymi przez poprzednie serwery. Tak więc czytając nagłówki od dołu do góry poznamy drogę listu od nadawcy do odbiorcy. Spamer może dodać poniżej istniejących własne nagłówki, starając się zatrzeć swoją tożsamość i zwalić winę na kogoś innego. Prawdziwe są z pewnością nagłówki wstawiane przez serwer odbiorcy (najwyższe). Pozostałe mogą być fałszywe.

Jedynie dzięki nagłówkom `Received` możemy zidentyfikować IP faktycznego nadawcy listu, a także często rozpoznać, czy list został nadany za pośrednictwem *open relay*, czy *open proxy.* Analiza nagłówków nie jest jednak łatwa, ponieważ nie istnieje ustalony standard ich formułowania i każdy serwer pocztowy podaje zawarte w nich informacje w nieco innej kolejności.

Serwery *open proxy*

Innym typem źle skonfigurowanych serwerów, które mogą być wykorzystane przez spamerów są *open proxy*, czyli serwery proxy, do których mogą łączyć się nieautoryzowani użytkownicy. Serwery *open proxy* mogą działać w oparciu o różne oprogramowanie i różne protokoły. Najczęściej jest to protokół HTTP-CONNECT, ale zdarzają się też *open proxy* umożliwiające połączenie przez protokoły HTTP-POST, SOCKS4, SOCKS5 i inne.

Open proxy może być wykorzystane przez spamera w identyczny sposób, jak *open relay* – do przesłania nieautoryzowanej poczty. Wiele z nich umożliwia dodatkowo ukrycie swojego adresu IP. Takie proxy to smakowity kąsek dla spamera.

Wykorzystanie *open proxy*

Na Listingu 6 widzimy przykład wykorzystania open proxy umożliwiającego połączenie przez HTTP-CONNECT na porcie 80. Większa część połączenia to komunikacja z *open relay* (te same komendy, co na Listingu 2). Zanim jednak połączymy się z serwerem SMTP, nawiązujemy kontakt z *open proxy* i dopiero za jego pomocą łączymy się z MTA. Podczas połączenia deklarujemy, że komunikacja będzie odbywała się za pomocą protokołu HTTP/1.0, jednak wcale nie musimy go używać.

Sytuacją najbardziej komfortową dla spamera jest znalezienie serwera *open proxy*, który ma jednocześnie zainstalowany lokalny serwer pocztowy. W większości przypadków MTA przyjmuje bez autoryzacji połączenia od lokalnego proxy traktując

Listing 5. *Serwer open relay z błędną konfiguracją SMTP-AUTH.*

```
$ telnet mail.example.com 25
< 220 mail.example.com ESMTP
> ehlo hakin9.org
< 250-mail.example.com
< 250-PIPELINING
< 250-8BITMIME
< 250-SIZE 10485760
< 250 AUTH LOGIN PLAIN CRAM-MD5
> auth login
< 334 VXNlcm5hbWU6
> cokolwiek
< 334 UGFzc3dvcmQ
> cokolwiek
< 235 ok, go ahead (#2.0.0)
> mail from:<hakin9@hakin9.org>
< 250 ok
> rcpt to:<nobody@nowhere.com>
< 250 ok
> data
< 354 go ahead
> Subject: test
>
> To jest test
> .
< 250 ok 1077563277 qp 13947
> quit
< 221 mail.example.com
```

je na równi z użytkownikami lokalnymi. W takiej sytuacji spamer nie musi znać ani jednego serwera *open relay* i może komfortowo, anonimowo prowadzić swoją działalność na czyjś koszt i na czyjąś odpowiedzialność, znacznie utrudniając wykrycie

Zombie

Najnowszą metodą wykorzystywaną przez spamerów do przenoszenia kosztów i odpowiedzialności na inne osoby są tzw. *zombie*. Technika ta łączy wirusa (worma) z koniem trojańskim. Celem jest utworzenie *open proxy* na komputerze zainfekowanym przez wirus. W ten sposób powstaje ogromna sieć anonimowych *open proxy* z których korzystają spamerzy z całego świata.

Najbardziej znanym przykładem zombie są wirusy z serii *Sobig*. Oto jak działa wariant *Sobig.E*:

- Człon pierwszy po zainfekowaniu komputera użytkownika (po uruchomieniu załącznika) rozsyła się do wszystkich adresów znalezionych w plikach .txt i .html na twardym dysku.
- Między godziną 19 a 23 czasu UTC łączy się z jednym z 22 zawartych w kodzie wirusa adresów IP na porcie 8998 UDP aby uzyskać adres URL, z którego może pobrać drugi człon.
- Po pobraniu drugiego członu (konia trojańskiego), jest on instalowany i uruchamiany; adres IP zainfekowanego komputera jest przesyłany do autora zombie, a następnie pobierany jest trzeci człon.
- Trzeci człon to zmodyfikowany program *Wingate*, który po automatycznej instalacji uruchamia na komputerze użytkownika *open proxy*.

Więcej informacji na temat działania wirusów z serii *Sobig* można znaleźć pod adresem *http://www.lurhq.com/sobig.html*.

Jedyną skuteczną metodą zabezpieczenia przed zombie jest stosowanie programów antywirusowych i systemów IDS (*Intrusion Detection System* – np. *Snort*), które pomogą wykryć *open proxy* w naszej sieci.

Przezorny zawsze ubezpieczony

Jak widać, wykorzystanie źle zabezpieczonych serwerów nie jest trudne. Skutki dla administratora dziurawego serwera mogą być zaś bardzo poważne, a spamer najprawdopodobniej nie poniesie żadnej odpowiedzialności. Uruchamiając więc serwer proxy upewnijmy się, że prawo do korzystania z niego mają tylko użytkownicy sieci lokalnej, a najlepiej wymusić autoryzację dla wszystkich nadawców. Będzie się to wiązało z mniejszym komfortem użytkowników, ale o celowości takiego działania nie musimy chyba nikogo przekonywać.■

Listing 6. *Serwer open proxy wykorzystany do przesłania anonimowo poczty przez open relay.*

```
$ telnet 204.170.42.31 80
> CONNECT kogut.o2.pl:25 HTTP/1.0
>
< HTTP/1.0 200↵
  Connection established
< 220 o2.pl ESMTP Wita
> helo hakin9.org
< 250 kogut.o2.pl
> mail from:<ania@o2.pl>
< 250 Ok
> rcpt to:<hakin9@hakin9.org>
< 250 Ok
> data
< 354 End data with↵
  <CR><LF>.<CR><LF>
> Subject: test
>
> To jest test
> .
< 250 Ok: queued as 5F4D41A3507
> quit
< 221 Bye
```

Skąd spamerzy biorą adresy *open relay* i *open proxy*?

Samodzielne znalezienie źle zabezpieczonych serwerów może być kłopotliwe. Jednak wystarczy otrzymać spam wysłany przez *open relay* lub *open proxy*, by móc je wykorzystać. Do sprawdzania, czy pod adresem IP jest *open relay* można wykorzystać skrypt *rlytest* (*http://www.unicom.com/sw/rlytest/*), zaś do wykrycia *open proxy* – *pxytest* (*http://www.unicom.com/sw/pxytest/*). Oba umieściliśmy na naszym CD.

Spamerzy, którym zależy na usprawnieniu działalności korzystają przede wszystkim z płatnych baz adresów *open relay* i *open proxy*. Nietrudno je znaleźć – wystarczy wpisać w dowolnej wyszukiwarce słowa *open proxy* lub *open relay* i kliknąć w kilka pierwszych linków (np.: *http://www.openproxies.com/* – 20 USD za miesiąc, *http://www.openrelaycheck.com/* – 199 USD za pół roku).

Inną metodą uzyskania adresów jest pobranie strefy zawierającej adresy *open relay* lub *open proxy* od jednego z serwerów DNSBL (patrz artykuł Ochrona przed spamem na serwerze). Listę takich serwerów można znaleźć np. pod adresem *http://www.declude.com/junkmail/support/ip4r.htm*. Do pobrania strefy można wykorzystać aplikację *host*: `host -l <nazwa strefy> <adres DNSBL>`. Uwaga: wiele serwerów DNSBL uniemożliwia pobieranie całych stref.