Uniwersytet w Białymstoku

Wydział Matematyki i Informatyki

Instytut Matematyki

Łukasz Kruszewski

Filtrowanie niechcianych wiadomości na serwerze pocztowym

Praca dyplomowa napisana
pod kierunkiem
dr. Mariusza Żynela

Białystok 2016

Składam serdeczne podziękowania
dr. Mariuszowi Żynelowi
za poświęcony czas oraz
cenne rady związane z
niniejszą pracą, a także
za wsparcie i wyrozumiałość.

Łukasz Kruszewski

Spis treści

Wstęp

Wszyscy korzystamy z poczty elektronicznej. Wraz z upływem lat stała się rzeczą tak powszechną, że korzystanie z niej jest instynktowne. Jednak nie każdy ma świadomość zagrożeń idących drogą poczty elektronicznej. Nachalne reklamy, wirusy, to tylko niektóre z nich. By usprawnić pracę na naszym serwerze pocztowym, niezbędny jest system który przefiltruję naszą korespondencję.

Celem mojej pracy, jest instalacja i konfiguracja oprogramowania do filtrowania niechcianej poczty, które będzie gotowe do zastosowania np. na prywatnym serwerze pocztowym. Aby w pełni zrozumieć realizowane przeze mnie zagadnienie, warto by było zacząć od podstaw, takich jak przybliżenie dokładnego działania poczty elektronicznej. W pracy znajdziemy opis zagrożeń, jakie mogą kryć się w wiadomości na naszej skrzynce. Następnie poznamy techniki filtrowania niechcianej poczty, a na końcu mamy wdrożenie systemu filtrowania i dostarczania. Część teoretyczną mojej pracy stanowią pierwsze trzy rozdziały. Czwarty i zarazem ostatni rozdział mojej pracy to opis części praktycznej.

W pierwszym z nich przybliżę jak działa poczta elektroniczna, pojawią się tu takie pojęcia jak SMTP, IMAP czy POP jako elementarz, zwroty te będą przewijać się przez całą pracę i ułatwią zrozumienie całego algorytmu. W drugim rozdziale przyjrzymy się temu co należy filtrować na naszej poczcie. Wirusy, Spam, Ransomware czy Phishing to niebezpieczeństwa których chcemy się wystrzegać i dowiemy się jak to robić poprzez dokładny opis technik filtrowania. Trzecim rozdziałem powoli zahaczamy o część praktyczną naszej pracy. Znajdziemy w niej opis użytych technologii, a diagram podróży wiadomości od nadawcy do odbiorcy pomoże nam lepiej zrozumieć działanie całego wdrażanego mechanizmu. Sendmail, RBL, Milter, LMTP i SIEVE to rzeczy bez których nie byłoby możliwe stworzenie efektywnego systemu filtrowania naszej korespondencji, a więc niezbędne jest wyjaśnienie jak działa każda z tych usług. Ostatni rozdział, to opis części praktycznej, przedstawię w niej konfigurację Sendmail'a i innych procesów zależnych do prawidłowego działania naszej maszynerii. Zaprezentuję wyniki działania wtyczek smf-clamav, smf-spamd i smf-zombie jako dowód efektywności systemu, do filtrowania niechcianej poczty.

Rozdział 1

Poczta elektroniczna

Poczta elektroniczna e-mail - czyli usługa internetowa, służąca do przesyłania wiadomości tekstowych pomiędzy komputerami posiadającymi dostęp do sieci.

Poczta zawdzięcza swoje powstanie, systemowi który odegrał znaczącą rolę w rozwoju komunikacji elektronicznej. Compatible Time-Sharing System, bo o nim mowa to system powstały w 1961r. w Massachusetts Institute of Technology, mający na celu stworzenie zdalnego dostępu użytkowników z terminali podłączanych za pośrednictwem m.in. sieci telefonicznej. Główną usługą udostępnianą przez ten system była możliwość zapisywania i odczytywania plików przechowywanych na dysku serwera. Użytkownicy, szybko zaczęli wykorzystywać jego funkcjonalność, do udostępniania na serwerze plików tekstowych zawierających krótkie wiadomości, dzięki czemu usprawniali pracę nad wspólnie tworzonym projektem. Obserwujący ten trend, administratorzy CTSS postanowili stworzyć narzędzie, które ułatwiałoby komunikację między użytkownikami, a także umożliwiłoby proste wysyłanie wiadomości od administratorów do szerszej grupy użytkowników.

E-mail został wymyślony w roku 1965. Autorami pomysłu byli: Louis Pouzin, Glenda Schroeder i Pat Crisman. Wówczas jednak usługa ta służyła jedynie do przesyłania wiadomości pomiędzy użytkownikami tego samego komputera a adres poczty elektronicznej jeszcze nie istniał. Usługę polegającą na przesyłaniu wiadomości tekstowych pomiędzy komputerami wymyślił w roku 1971 Ray Tomlinson, wybrał również znak @ do rozdzielania nazwy użytkownika od nazwy komputera, a później nazwy domeny internetowej. Na początku do wysyłania e-maili służył protokół komunikacyjny CPYNET. Później wykorzystywano FTP, UUCP i wiele innych protokołów, a w 1982 roku Jon Postel opracował do tego celu protokół SMTP, używany do dzisiaj. [18][22]

1.1 Działanie poczty elektronicznej

Bez wątpienia wysyłanie wiadomości e-mail jest jedną z popularniejszych czynności wykonywanych w internecie, jednak czy na pewno wiemy jak wygląda droga przesyłki od nadawcy do odbiorcy i co odpowiada za dostarczenie takiej wiadomości?

Cały proces możemy podzielić na wysyłanie i odbieranie. Za wysyłanie odpowiada protokół SMTP (Simple Mail Transfer Protocol), natomiast za ich odbieranie odpowiadają protokoły IMAP lub POP.

Czego potrzebujemy do odbioru poczty?

1. Mail User Agent (MUA) - interfejs użytkownika, z którego możemy odbierać jak i wysyłać wiadomości. Najpopularniejszymi są Thunderbird, MS Outlook, Zimbra Desktop. Warto jednak nadmienić, że interfejs Gmail'a czy wp.pl to również MUA.

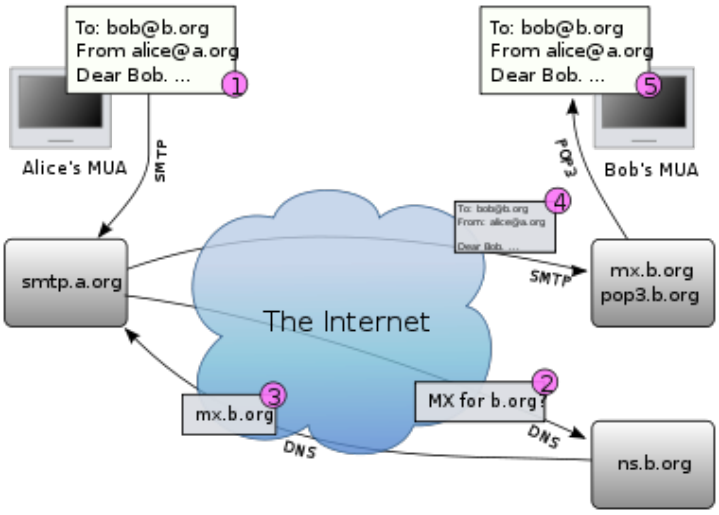
2. Mail Submission Agent (MSA) - oprogramowanie na serwerze pocztowym, którego zadaniem jest przekazanie wiadomości od MUA do MTA. Zazwyczaj przekazanie wiadomości poprzedzone jest autentykacją i autoryzacją nadawcy.

3. Mail Transfer Agent (MTA) - jest odpowiedzialne za przekazanie wiadomości na serwer pocztowy odbiorcy i następnie przekazanie jej MDA (Mail Delivery Agent). Najpopularniejszymi MTA są Sendmail i Postfix.

4. Mail Delivery Agent (MDA) - odpowiada za bezpośrednie rozdzielanie poczty i dostarczenia jej do skrzynki użytkownika.

5. DNS - protokół umożliwiający zamianę adresów użytkowych na adresy tworzącę sieć komputerową, czyli popularne adresy IP. Dzięki DNS popularna domena wp.pl jest tłumaczona na odpowiadający jej adres IP, czyli 212.77.98.9. [12]

6. Rekord MX - porcja informacji w DNS która wskazuje serwer odpowiedzialny za przyjmowanie poczty dla określonej domeny. [6]



Rysunek 1.1: Diagram przedstawiający podróż wiadomości od nadawcy do odbiorcy.

Na podstawie powyższego diagramu przyjrzyjmy się jak przebiega przesłanie wiadomości

1. Użytkownik, w tym wypadku Alice, korzystając ze swojego programu pocztowego, czyli MUA, wysyła e-mail łącząc się z MSA zgodnie z protokołem SMTP.

2. Serwer MTA nadawcy analizuje część domenową adresu odbiorcy by zlokalizować serwer MTA odbiorcy na jaki dostarczyć wiadomość. W tym celu odpytuje DNS pobierając rekord MX przypisany domenie odbiorcy.

3. W odpowiedzi na pytanie MTA nadawcy serwer DNS zwraca rekord MX domeny odbiorcy.

4. Serwer MTA nadawcy, używając protokołu SMTP, przekazuje wiadomość serwerowi MTA odbiorcy.

5. Odbiorca, w tym wypadku Bob, odbiera wiadomość, na swoim MUA, za pomocą protokołu POP lub IMAP.

Warto zwrócić uwagę, że w tym schemacie uczestniczy co najmniej siedem różnych programów. Powodem tego jest fakt, że po drodze od MTA nadawcy do MTA odbiorcy może być kilka MTA pośredniczących w przekazaniu wiadomości. Czasem rolę MSA i MTA nadawcy pełni rolę jeden program i proces. Tak jest w przypadku Sendmail'a. [15]

1.2 Protokół SMTP

SMTP - protokół odpowiadający za wysyłanie wiadomości na poczcie elektronicznej. Zdefiniowany został w dokumencie RFC 821, a następnie w 2008r. w RFC 5321 zaktualizowany. Domyślnie używa portu 25. [2]

Najważniejsze komendy tego protokołu to:

helo - polecenie rozpoczynające sesję z serwerem pocztowym,

mail from - deklaracja nadawcy,

rcpt to - deklaracja odbiorcy,

data - rozpoczyna wpisanie treści wiadomości, aby zakończyć wpisywanie treści i ewentualnych załączników wpisujemy kropkę w osobnej linii,

quit - polecenie kończące sesję z serwerem pocztowym.

Zobaczmy przykładową sesję SMTP. Po wywołaniu polecenia

```
telnet 10.18.0.123 25
```

dialog pomiędzy serwerem (S) a klientem (C) może wyglądać następująco:

```
S: 220 omega.uwb.edu.pl ESMTP Sendmail 8.15.2/8.15.2;
C: helo localhost
S: 250 omega.uwb.edu.pl Hello geo-01 [10.18.0.122],
   pleased to meet you
C: mail from: <lukasz@solaris-x86.net>
S: 250 2.1.0 <lukasz@solaris-x86.net>... Sender ok
C: rcpt to: <lukasz@omega.uwb.edu.pl>
S: 250 2.1.5 <lukasz@omega.uwb.edu.pl>... Recipient ok
C: data
S: 354 Enter mail, end with "." on a line by itself
C: Subject: test
C:
C: test
C:
C: .
C:
S: 250 2.0.0 u7U8JrOh000570 Message accepted for delivery
C: quit
```

Bardzo ważnym składnikiem wiadomości, zwykle ukrywanym przed użytkownikiem z uwagi na jego techniczny charakter, jest zestaw tak zwanych nagłówków. Nagłówki zawierają wiele cennych informacji o danej wiadomości. Najważniejsze z nich to:

Subject - wpisanie treści tematu,

From - informacja z jakiego adresu przyszła do nas wiadomość,

To - wyświetla odbiorcę wiadomości,

Received - pokazuję drogę, jaką przebyła wiadomość, zanim dotarła do odbiorcy.

Warto w tym miejscu zwrócić uwagę na fakt, że zarówno adres nadawcy, jak i odbiorcy podawane są dwa razy. Analogicznie jak w zwykłej poczcie mamy adresy na kopercie określone przez **mail from** i **rcpt to** oraz ich odpowiedniki wewnątrz listu **From** i **To**. Końcowy użytkownik nie widzi niestety koperty, a nagłówki **From**, **To** pełnią jedynie charakter informacyjny i są wyświetlane w programach pocztowych jako nadawca i odbiorca. Wielu hakerów w prosty sposób wykorzystuje to by podszywać się pod osoby trzecie.

1.3 Protokoły IMAP i POP

Protokół POP

POP to protokół umożliwiający odbiór poczty elektronicznej z serwera do lokalnego komputera poprzez połączenie TCP/IP. Połączenie POP odbywa się na porcie 110, a w wersji szyfrowanej SSL, na porcie 995. Kiedy chcemy skorzystać z poczty nawiązujemy połączenie TCP z serwerem, a następnie wysyłane zostaje powitanie. Taką kolej rzeczy nazywamy stanem autoryzacji. Wszelkiego rodzaju zapytania do serwera nazywamy stanem transakcji, a ostatni stan to update czyli aktualizacja danych serwera POP. [11]

Przykład powitania z serwerem w wersji POP3:

```
S: +OK omega Cyrus POP3 v2.4.16 server ready
  <5034054857336343972.1473606041@omega>
```

Użytkownik by móc korzystać z poczty musi się określić względem serwera. Komenda `USER` pozwala na wpisanie nazwy użytkownika, jeżeli nazwa użytkownika jest poprawna, komendą `PASS` wpisujemy hasło. Przykład takiego logowania:

```
S: +OK omega Cyrus POP3 v2.4.16 server ready
    <5034054857336343972.1473606041@omega>
C: USER lukasz@omega.uwb.edu.pl
S: +OK Name is a valid mailbox
C: PASS spinacz
S: +OK Mailbox locked and ready SESSIONID=<omega-466-1473606041-1>
```

Komenda `APOP` umożliwia nam możliwość szyfrowanego logowania. `APOP` ma dwa argumenty: nazwa użytkownika i zaszyfrowane hasło w formacie szesnastkowym w kodzie ASCII. Gdy weryfikacja przebiegnie poprawnie sesja jest w stanie transakcji. Przykład logowania APOP:

```
S: +OK omega Cyrus POP3 v2.4.16 server ready
    <5034054857336343972.1473606041@omega>
C: APOP lukasz@omega.uwb.edu.pl c6787a8d88e819d88b9c8a99f561c2
```

Powyższy ciąg znaków, to zakodowane hasło. Będąc już zalogowany, użytkownik może używać następujących komend:

1. `STAT` - Zwraca ilość i sumaryczny rozmiar wiadomości w bajtach na serwerze.

2. `LIST` - Zwraca listę wiadomości podając numer i rozmiar wiadomości w bajtach.

3. `RETR` - Zwraca wiadomość o numerze podanym przez argument.

4. `TOP` - Zwraca część wiadomości o numerze podanym w pierwszym argumencie. W drugim argumencie podaje się ilość pierwszych wierszy wiadomości jaka ma być zwrócona, 0 oznacza same nagłówki.

5. `DELE` - Serwer oznacza wiadomość o przekazanym w argumencie numerze jako skasowaną, kasuje ją jednak dopiero, gdy sesja przejdzie w stan "UPDATE".

6. `NOOP` - Serwer po otrzymaniu tej komendy wysyła pozytywną odpowiedź "+OK".

7. `RSET` - Zmienia status wiadomości skasowanych na nieskasowane.

Po zakończeniu stanu transakcji przechodzimy w stan aktualizacji. To tutaj finalnie akceptujemy wszystkie polecenia wprowadzone w stanie transakcji. Po zakończeniu tego stanu sesja jest skończona. Przykładowa sesja POP3:

```
S: +OK omega Cyrus POP3 v2.4.16 server ready
    <5034054857336343972.1473606041@omega>
C: USER lukasz@omega.uwb.edu.pl
S: +OK Name is a valid mailbox
C: PASS spinacz
S: +OK Mailbox locked and ready SESSIONID=<omega-466-1473606041-1>
C: STAT
S: +OK 13 4159
C: LIST
S: +OK scan listing follows
S: 1 1536
S: 2 1535
S: 3 1088
S: .
C: TOP 1 0
S: +OK Message follows
S: Received: from omega.uwb.edu.pl ([unix socket])
S:         by omega (Cyrus v2.4.16) with LMTPA;
S:         Mon, 05 Sep 2016 12:43:07 +0200
```

```
...
S:
S: .
C: QUIT
S: DONE
```

Protokół IMAP

IMAP to internetowy protokół do odbierania poczty elektronicznej, następca POP. Korzystanie z tego protokołu jest idealnym rozwiązaniem dla użytkowników chcących odbierać pocztę korzystając z wielu komputerów i mieć dostęp do całej zawartości skrzynki. IMAP jest wykorzystywany w najpopularniejszych serwisach z pocztą elektroniczną jak gmail.com, wp.pl itp. Komunikacja z usługą IMAP odbywa się na porcie 143, a w wersji szyfrowanej SSL, na porcie 993. [14]

Przykład sesji IMAP przy użyciu programu telnet:

```
S: * OK [CAPABILITY IMAP4rev1 LITERAL+ ID ENABLE AUTH=PLAIN AUTH=LOGIN
    SASL-IR] omega Cyrus IMAP v2.4.16 server ready
S: a1 LOGIN lukasz@omega.uwb.edu.pl spinacz
S: a1 OK [CAPABILITY IMAP4rev1 LITERAL+ ID ENABLE ACL RIGHTS=kxte QUOTA
    MAILBOX-REFERRALS NAMESPACE UIDPLUS NO_ATOMIC_RENAME UNSELECT CHILDREN
    MULTIAPPEND BINARY CATENATE CONDSTORE ESEARCH SORT SORT=MODSEQ
    SORT=DISPLAY THREAD=ORDEREDSUBJECT THREAD=REFERENCES ANNOTATEMORE
    LIST-EXTENDED WITHIN QRESYNC SCAN XLIST URLAUTH URLAUTH=BINARY
    LOGINDISABLED AUTH=PLAIN AUTH=LOGIN COMPRESS=DEFLATE IDLE]
    User logged in SESSIONID=<omega-274-1473606142-1>
C: a1 LIST "" "*"
S: * LIST (\HasChildren) "/" INBOX
S: * LIST (\HasNoChildren) "/" INBOX/SPAM
S: * LIST (\HasNoChildren) "/" INBOX/Sent
S: * LIST (\HasChildren) "/" INBOX/Trash
S: * LIST (\HasNoChildren) "/" INBOX/Trash/TEST
S: a1 OK Completed (0.000 secs 6 calls)
C: a1 EXAMINE INBOX
S: * 13 EXISTS
S: * 0 RECENT
S: * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)

S: * OK [PERMANENTFLAGS ()] Ok
S: * OK [UIDVALIDITY 1473065766] Ok
S: * OK [UIDNEXT 14] Ok
S: * OK [HIGHESTMODSEQ 28] Ok
S: * OK [URLMECH INTERNAL] Ok
S: a1 OK [READ-ONLY] Completed
C: a1 FETCH 1 ALL
S: * 1 FETCH (FLAGS (\Seen) INTERNALDATE " 5-Sep-2016 12:43:08 +0200"
    RFC822.SIZE 1536 ENVELOPE ("Mon, 5 Sep 2016 11:43:06 +0200 (CEST)"
    "test" (("Mariusz Zynel" NIL "mariusz" "geo-01.uwb.edu.pl"))
    (("Mariusz Zynel" NIL "mariusz" "geo-01.uwb.edu.pl")) (("Mariusz Zynel"
    NIL "mariusz" "geo-01.uwb.edu.pl")) ((NIL NIL "lukasz"
```

```
    "omega.uwb.edu.pl")) NIL NIL NIL
    "<201609050943.u859h6nh001700@geo-01.uwb.edu.pl>"))
S: a1 OK Completed (0.000 sec)
a1 LOGOUT
S: * BYE LOGOUT received
S: a1 OK Completed
read:errno=0
```

Porównanie IMAP i POP

Wybór właściwego dla nas protokołu nie jest jednoznaczny. Każdy z nich ma swoje wady i zalety, a to z jakiego powinniśmy skorzystać, zależy od charakterystyki naszej pracy z pocztą elektroniczną.

Protokół POP działa "offline". Łączymy się z serwerem raz, by zgrać całą zawartość skrzynki do katalogu na naszym komputerze, a cała jej zawartość jest domyślnie usuwana z serwera. Czytanie wiadomości jak i pisanie odpowiedzi może być wykonywane bez dostępu do sieci, a połączenie jest nam potrzebne wyłącznie do samego wysłania korespondencji lub pobrania zawartości skrzynki jeszcze raz. Ten tryb pracy jest dla użytkowników posiadających ograniczone połączenie do sieci. Brak stałego połącznia obniża zużycie transferu. Problemem tego protokołu jest obsługiwanie skrzynki na wielu komputerach. Domyślnie poczta po zapisaniu na dysku komputera jest usuwana z serwera, wówczas nie możemy mieć na każdym z komputerów całej odebranej dotychczas korespondencji. Istnieje opcja pozostawienia na serwerze pobranej poczty jednak to prowadzi do przepełnienia pojemności skrzynki, a w protokole POP, nie ma możliwości korzystania z innych katalogów poczty. Następny problem POP'a to brak możliwości usunięcia listu bez pobrania go na komputer. W przypadku wiadomości potężnych rozmiarów i słabego łącza, nie pozostaję nam nic innego jak wielogodzinna jak nie wielodniowa udręka.

Protokół IMAP działa w inny sposób i obecnie jest dużo częściej spotykany. Poczta jest zamieszczona na serwerze skrzynki pocztowej i posiada dodatkowe katalogi. Po połączeniu z klientem pocztowym transmitowane są nagłówki. Pobranie wiadomości następuje po jej otwarciu, dzięki czemu jest możliwość usuwania korespondencji bez pobierania jej na nasz komputer. W przypadku kiedy odbieramy naszą pocztę na wielu urządzeniach, w różnych miejscach, doskonale sprawdza się protokół IMAP.

Dlaczego POP?

1. Wiadomośći składowane lokalnie, zawsze dostępne nawet bez połączenia z siecią.
2. Połączenie z internetem potrzebne jedynie do wysyłania i odbierania wiadomości.
3. Oszczędność miejsca na serwerze.
4. Opcja pozostawienia kopii wiadomości na serwerze.

5. Możliwość podłączenia wielu kont e-mail do jednej skrzynki odbiorczej.

Dlaczego IMAP?

1. Wiadomości dostępne w wielu miejscach na różnych urządzeniach.
2. Szybszy przegląd poczty dzięki ściągnięciu samych nagłówków.
3. Oszczędność miejsca na dysku twardym.
4. Możliwość składowania poczty lokalnie.

Reasumując, jeśli z jakichś przyczyn chcielibyśmy odbierać wiadomości na jednym komputerze np. firmowym lub też nie mamy stałego połączenia do sieci warto wybrać protkół POP, jednak jeśli chcemy odbierać naszą korespondencję w wielu miejscach na różnych urządzeniach, powinniśmy korzystać z poczty z protkołem IMAP. [21]

Rozdział 2

Niechciane wiadomości

2.1 Spam

Spam – niechciane lub niepotrzebne wiadomości elektroniczne wysyłane najczęściej masowo, mające charakter reklamowy, lub maile zawierające szkodliwe oprogramowanie mające na celu kradzież naszych danych osobowych, lub wyłudzenie dostępu do naszych komputerowych danych. Nadmierna ilość spamu powoduje zapychanie skrzynek pocztowych. Spam najczęściej jest rozpowszechniany przez pocztę elektroniczną, jednak niechciane wiadomości występują również na serwisach społecznościowych, komunikatorach internetowych czy w postaci SMS'ów.

Słowo "SPAM" wywodzi się od nazwy mielonki wieprzowej sprzedawanej od lat 30-tych w Stanach Zjednoczonych. Nie wiadomo, dlaczego twórcy poczty wybrali akurat nomenklaturę kulinarną. Można doszukiwać się analogii patrząc na skład konserwy. Są to odpady z mięsa wieprzowego, mięso gorszego sortu, mało wartościowe zupełnie jak te, zapychające naszą skrzynkę wiadomości.

Spam rozwijał i kształtował się wraz z powstawaniem internetu. Pierwsza masowa wysyłka drogą elektroniczną to 1978r. Niejaki Einar Stefferud, korzystając z sieci Arpanet, postanowił zaprosić na swoje przyjęcie urodzinowe, niespełna 1000 osób wysyłając przy tym do każdego z nich wiadomość zawierającą zaproszenie na owe przyjęcie. Nie spotkało się to z aprobatą i pierwszy "spamer" został ukarany, twarde dyski na jego serwerze zostały zablokowane.

Najgłośniejszą akcją komercyjną, dotyczącą rozesłania wiadomości reklamowych sięga roku 1994r. Laurence Canter i Martha Siegel prowadzili firmę adwokacką, by ją wypromować, rozesłali korzystając z sieci Usernet tysiące wiadomości zawierających ofertę kancelarii. Prawnicy otrzymali duży odzew, w większości negatywny, co poskutkowało zablokowaniem przez dostawcę internetowego, dostępu do poczty elektronicznej. O sprawie było głośno w mediach, a to zdarzenie umocniło w świadomości ludzi termin "spam". [2]

2.2 Wirusy

Wirusy wysyłane drogą mailową, nie różnią się od tych które możemy napotkać w sieci, jednak samo opakowanie złośliwego oprogramowania może zmylić użytkowników poczty elektronicznej. Wirusy wysyłane drogą mailową, zazwyczaj umieszczone są w załączniku lub jako kod wiadomości. Otwarcie takiego załącznika może być bardzo niebezpieczne, wirusy mogą spowodować utratę danych na naszym komputerze, a wiadomość automatycznie rozesłana do wszystkich w naszej książce adresowej. Jak bronić się przed wirusami?

- Nie pobierać załączników od nieznanych odbiorców.
- Zainstalowanie na naszym serwerze pocztowym oprogramowania antywirusowego np. ClamAV.
- Oprogramowanie antywirusowe na naszym komputerze lokalnym.

Najpopularniejszymi wirusami rozprzestrzenianymi drogą elektroniczną to "Melissa virus macro virus" i "ILOVEYOU virus". Wszystkie rozsyłane w różnych kombinacjach, zazwyczaj zmieniany jest adresat i temat wiadomości. [16]

2.3 Ransomware

Ransomware - rodzaj oprogramowania używanego przez hakerów mające na celu ograniczenie naszego dostępu do komputera poprzez zablokowanie funkcji systemowych. W zamian za usunięcie ograniczeń, hakerzy żądają korzyści majątkowych. Samo oprogramowanie najczęściej wysyłane jest w załącznikach wiadomości. Złośliwe oprogramowanie może również uzyskać dostęp do komputera ofiary poprzez sieć. Niezbędnym do usunięcia Ransomware jest antywirus i wbudowane w nim narzędzie przeznaczone do jego usunięcia. Oprócz dobrego antywirusa warto zadbać o to by nasza przeglądarka i używane przez nas paski narzędzi były aktualne, a zapora ogniowa zaktualizowana. [3]

2.4 Phishing

Phishing - to typ oszustwa internetowego, mający na celu wyłudzenie danych użytkownika, w większości przypadków chodzi o kradzież danych kont bankowych, haseł i wszelkich poufnych informacji. Adresat wiadomości phishingowych podszywa się pod duże instytucje zazwyczaj finansowe. Treść wiadomości zmusza do aktualizacji danych, w przeciwnym razie grozi utratą danych, czy awarią systemu. Głośny przypadek z przed dwóch lat dotyczył podszywania się pod bank PKO, gdzie internauci dostawali maile dotyczące zablokowania ich aplikacji bankowej, by odblokować dostęp należało wpisać podane hasło oraz wysłać jednorazowy kod SMS. Warto pamiętać, że żaden bank nigdy nie żąda

podania hasła w mailach, więc dokładna analiza adresata jak i załączników jest konieczna i na ogół wystarczy by ustrzeć się przed hakerami. [23]

2.5 Techniki filtrowania spamu

2.5.1 Analiza słownikowa

Ta metoda opiera się na tworzeniu listy słów które najczęściej występują w masowych wiadomościach. Na podstawie tej listy sprawdza się częstotliwość występowania ich w korespondencji. Metoda jest łatwa do zaimplementowania a jej skuteczność zależy od doboru niepożądanych zwrotów. Zbyt surowe podejście może prowadzić do tego, że większość korespondencji będzie blokowana. Zdecydowaną wadą tej metody jest dosyć proste obejście przez spamerów, stosując słowa z niewielką zmianą znakową (np. spam na sp@m) w tym wypadku nasz słownik musi być często aktualizowany, co wiążę się z dużym nakładem pracy, a niekiedy może stać się bardziej czasochłonne niż ręczne usuwanie niechcianych wiadomości. [8]

2.5.2 Czarne listy (RBL)

Listy RBL (Real-time Black Lists) to zbiór adresów IP rozsyłających spam, gdzie większość z nich to adresy serwerów open relay (tj. niezabezpieczone serwery pocztowe przed nieautoryzowanym wykorzystaniem do wysyłania wiadomości, najczęściej spamu). Serwery open relay przyczyniają się do zaśmiecania kont pocztowych niechcianymi wiadomościami, gdyż umożliwiają ich przekazywanie bez dokonania autoryzacji nadawcy. Podobnie wykorzystywane są serwery open proxy, które wskutek błędnej konfiguracji są dostępne dla wszystkich. Spamerzy wykorzystują serwery open relay i open proxy by zamaskować swój adres IP, przez co pozostają anonimowi. Istnieje duże ryzyko, że owe serwery znajdą się na czarnej liście przez przypadek, jednak wszystkie niesłusznie zablokowane serwery zostają odblokowane dzięki szybkiej interwencji administratorów serwerów pocztowych.

Każdy komputer podłączony do sieci może stać się zainfekowanym serwerem rozsyłającym spam. Grupy komputerów zainfekowanych działających pod kontrolą osoby trzeciej to tzw. botnety. Eksperci twierdzą, że 70 procent spamu pochodzi właśnie z wykorzystaniem botnetów.

Blokowanie adresów dynamicznych nie jest w interesie małych firm czy ludzi którzy chcą posiadać własny serwer pocztowy dostosowany do ich potrzeb. Adresy hosta mogą również być blokowane, gdy w przypadku prowadzenia listy dyskusyjnej nie jest konieczne potwierdzenie subskrypcji. Poprawnie działająca lista umożliwia wypisanie się z dyskusji w dowolnym czasie, a sam proces rejestracji musi zakończyć się potwierdzeniem np. otwarciem wskazanego linku. W przypadku wielu nieuczciwych firm, rejestracja odbywa się bez wiedzy

zainteresowanego i nie wymaga ona potwierdzenia. Adresy takie są blokowane, jednak niesie to ryzyko odrzucenia pożądanych wiadomości. [8]

RBL działa w oparciu o system DNS. Wpis do RBL polega na dopisaniu adresu IP jako hosta w sieci danej organizacji realizującej RBL. Na przykład adres 95.51.4.34 jest umieszczony na liście Barracuda. Możemy się o tym przekonać korzystając z polecenia `host`:

```
host 34.4.51.95.b.barracudacentral.org
34.4.51.95.b.barracudacentral.org has address 127.0.0.2
```

Nie ma go natomiast na liście SORBS:

```
host 34.4.51.95.dnsbl.sorbs.net
Host 34.4.51.95.dnsbl.sorbs.net not found: 3(NXDOMAIN)
```

2.5.3 Szare listy (greylisting)

Metoda opracowana przez Harrego Evansa mogącą pochwalić się bardzo wysoką efektywnością kosztem opóźnienia w dostarczeniu wiadomości do adresata. Technika szarych list przyjmuję następujące założenia:

a) spamer do rozsyłania wiadomości korzysta z programu który nie posiada pełnej funkcjonalności serwera pocztowego

b) spam nie jest wysyłany dwukrotnie do tego samego adresata

Wiadomości rozpoznajemy dzięki trzem informacjom w nich zawartym: numer IP komputera skąd wiadomość została zaadresowana, adres nadawcy i adres odbiorcy. Te informacje zwane Trójkami (z ang. Triplets) są przechowywane w bazie danych szarych list. Wiadomość przychodząca na serwer jest badana na podstawie zapisanych w bazie Trójek. Ta która zawiera dotychczas nie napotkaną Trójkę jest odrzucana i tymczasowo zapisywana w bazie danych. Następna wiadomość o tej samej Trójce jest przyjmowana. Zgodnie z przyjętym założeniem oprogramowanie wykorzystywane przez rozsyłających niechciane wiadomości nie jest w pełni funkcjonalne jak normalny serwer pocztowy. SMTP z którego korzystają spamerzy nie jest w stanie rozpoznać tymczasowego od stałego odrzucenia wiadomości. Rozbudowane serwery pocztowe mają wbudowany system kolejkowania wiadomości, które są przetwarzane w ustalonych odstępach czasu, jeżeli w tym czasie wystąpi kolejna próba wysłania wiadomości zostanie ona dostarczona na serwer obsługiwany przez greylist. Ta metoda posiada małe ryzyko odrzucenia pożądanej przez nas wiadomości, jest w pełni zautomatyzowana i nie wymaga ingerencji użytkownika, a spam nie musi być przetwarzany przez inne dodatkowe filtry klasyfikujące. [8] [7]

2.5.4 Pytanie - odpowiedź

Pytanie - odpowiedź (challenge - response) to system działający analogicznie do szarej listy. Metodę cechuje wysoka skuteczność, jednak jej wadą jest przyczynianie się do zapychania skrzynek pocztowych. Działanie metody polega na wysłaniu wiadomości zwrotnej z żądaniem potwierdzenia tożsamości adresata zanim jego wiadomość zostanie dostarczona. Metoda zakłada, że nadawca nie odbiera wiadomości z adresu, z którego rozsyłany jest spam. Jeżeli nadawca przejdzie proces weryfikacji oryginalna wiadomość zostaję dostarczona. Metoda wydaję się być skuteczna jednak dosyć trywialna, a potwierdzanie swojej tożsamości przy każdym wysłanym mailu jest niewygodne dla użytkownika. [8]

2.5.5 Filtr Bayesowski

Metoda w oparciu o analizę statystyczną, a dokładniej naiwny klasyfikator bayesowski. Metoda cechująca się prostotą, ale i efektywnością, a w dodatku jest jedną z prostszych metod do zaimplementowania na naszym serwerze pocztowym. Filtr przygotowany w oparciu o powyższą metodę dzieli przesyłki na dwa zbiory: wiadomości pożądane i spam. Filtr czyta wystąpienia poszczególnych wyrazów zawartych w przesyłce i wyznacza prawdopodobieństwo w oparciu o wynik wystąpień. Przetworzenie nowej wiadomości polega na zliczeniu całkowitego prawdopodobieństwa dla wszystkich wyrazów zawartych w wiadomości, w oparciu o wynik, filtr decyduję co zakwalifikować jako spam, a co przepuścić przez serwer pocztowy. [5]

2.5.6 Filtr SPF

Ta metoda, utrudnia spamerom podszywanie się pod adresy użytkowników, korzystając z innych domen. Metoda SPF opiera się o dodatkowe informacje o domenie wpisane do serwera DNS w postaci rekordów TXT. Owe rekordy określają listę adresów IP, z których nadawca może potencjalnie nadać wysyłkę. Jeżeli otrzymana wiadomość i adres z którego została dostarczona nie znajduje się na liście adresów danego rekordu, filtr oznacza e-mail jako podejrzany i blokuje go. [7]

2.5.7 Mechanizm DK/DKIM

Skrót od ang. Domain Keys/Domain Keys Identified Mail, metoda kryptograficzna, podobnie jak SPF, stara się określić skąd pochodzi dana wiadomość i czy została nadana bezpośrednio u nadawcy z tą różnicą, że ingeruje również w treść e-maila. Podobnie jak w SPF, do informacji o domenie nadawcy w DNS dołączany jest specjalny klucz w postaci rekordu TXT. W nagłówku każdej wiadomości jest umieszczany specjalny znak DKIM, który odpowiada

zapisanemu kluczowi przy domenie nadawcy. Jeżeli weryfikacja znaku, powiedzie się, tzn. jest on zapisany w rekordzie, wiadomość zostanie dostarczona. [7]

Rozdział 3

System filtrowania i dostarczania poczty

3.1 Sprzęt

Celem pracy jest przygotowanie serwera pocztowego, tak aby do użytkowników docierało możliwie najmniej niechcianych wiadomości. Zadanie to realizujemy na komputerze z systemem Sun Solaris 10 Update 8. Dokładne parametry nie są tak bardzo ważne, aczkolwiek oprogramowanie ClamAV i SpamAssassin lubi mieć sporo pamięci RAM i wydajny procesor, najlepiej wielordzeniowy. Sam program Sendmail nie jest specjalnie wymagający.

Jeśli chodzi o skrzynki pocztowe użytkowników warto na nie przewidzieć więcej miejsca, bo niektórzy lubią gromadzić swoje wiadomości, często z dużymi załącznikami. Zależy to jednak od ilości kont i rodzaju działalności użytkowników. Wydaje się rozsądnym przyjąć, że jedna skrzynka to w sumie około 5GB danych. Warto też zadbać o redundancję zasobów dyskowych, aby ustrzec się przed awariami sprzętu i wykonywać regularne kopie zapasowe, także po to by ewentualnie móc odzyskać niechcący usunięte przez użytkowników stare wiadomości.

Do realizacji zadania, testowania i uruchamiania oprogramowania, wykorzystujemy także analogiczny system zainstalowany na VirtualBox.

3.2 Schemat działania

Przygotowanie i potem konserwacja systemu pocztowego jest bardzo trudna oraz wymaga sporego nakładu pracy. Wszystko dlatego, że zasada działania poczty e-mail nie zmieniała się od jej powstania, czyli w zasadzie przed erą Internetu, gdy korzystała z niej garstka naukowców i praktycznie nie istniał spam, wirusy, phishing, ransomware i tym podobne. Teraz, administrator poczty jest nękany przez użytkowników bo jedni dostają za dużo spamu, a innym ginie oczekiwana korespondencja. To wymaga ciągłego tuningu systemu

pocztowego. Z drugiej strony zdarzają się użytkownicy niespecjalnie dbający o siłę hasła i narażają swoje konto na przejęcie przez hakerów i nadużycie naszego serwera pocztowego do rozsyłania spamu. Inaczej mówiąc, generalnie administracja poczty to trudne i niezbyt wdzięczne zajęcie.

Podstawowe założenia jakie przyjęliśmy opracowując nasz system pocztowy są następujące:

1. Nie przyjmujemy poczty z serwerów, które znajdują się na listach RBL. Przyjmujemy wszystkie pozostałe wiadomości.

2. Filtrujemy wirusy i nie przepuszczamy zainfekowanych wiadomości do użytkowników, oznaczamy je jako Positive w nagłówku X-Antivirus i trafiają one do kwarantanny. Wykorzystujemy tutaj oprogramowanie ClamAV.

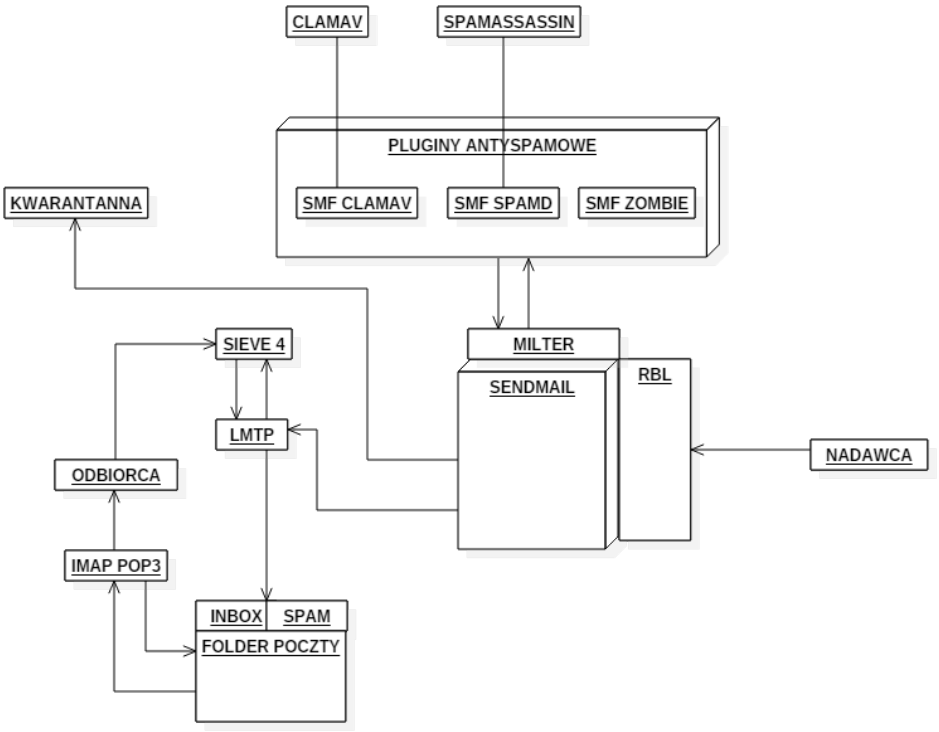
3. Filtrujemy wiadomości pod kątem poprawności składni oraz sprawdzamy, czy są prawidłowo zaadresowane (undisclossed-recipient jest zabronione). Jeśli nie to oznaczamy je jako Positive w nagłówku X-Antizombie.

4. Filtrujemy wiadomości pod kątem zawartych treści. Wykorzystujemy do tego celu oprogramowanie SpamAssassin. Wiadomości, które uzyskały 5 lub więcej punktów oznaczamy jako Positive w nagłówku X-Antispam.

5. Wiadomości oznaczone jako Positive (poza zainfekowanymi, które lądują w kwarantannie) umieszczamy w katalogu SPAM u danego użytkownika, do którego adresowana jest wiadomość. Może on wtedy w bezproblemowy sposób sprawdzić, czy oczekiwana wiadomość nie została przypadkowo zakwalifikowana jako spam (tzw. false-positive).

6. Każdy użytkownik ma możliwość tworzenia własnych, prostych filtrów pocztowych na serwerze opartych o oprogramowanie Cyrus Sieve. W przeciwieństwie do filtrów MUA (np. w Mozilla Thunderbird) takie filtry działają zawsze, gdy nadejdzie wiadomość, niezależnie od stosowanych MUA.

3.3 Schemat naszego systemu filtrowania i dostarczania poczty



Rysunek 3.1: Schemat naszego systemu filtrowania i dostarczania poczty

3.3.1 Sendmail

Jeden z najpopularniejszych serwerów pocztowych (MTA) udostępniany na darmowej licencji. Umożliwia przesyłanie i odbieranie poczty, głównie korzystając z protokołu SMTP. Napisany w 1982r. przez Erica Allmana. Popularność Sendmaila spada, ze względu na trudność w jego konfiguracji oraz problemy z bezpieczeństwem w dostarczaniu wiadomości. Sendmail nie jest oczywiście jedynym dostępnym serwerem pocztowym, inne tj. Qmail, Postfix, Oracle Communications Messaging (poprzednio: Sun ONE Messaging Server), Sendmail Switch również mają swoich fanów jednak nie przyjęły się tak dobrze wśród administratorów poczty. Sendmail to najprostsza droga na transport wiadomości od nadawcy do odbiorcy. Używa on protokołu DNS by zamienić adres hosta na adres domeny. Jak łatwo zauważyć, funkcjonalność ta jest bardzo pomocna, gdyż nie bylibyśmy w stanie zapamiętać adresów internetowych, z których korzystamy na co dzień gdyby ich adres identyfikowany był jako ciąg liczb.

Sendmail jest złożony z kilku części, włączając w to programy, pliki czy skrypty, które rozszerzają jego funkcjonalność. Sercem serwera pocztowego jest jego plik konfiguracyjny, który najprościej mówiąc zawiera cały plan działania zanim wiadomość dotrze do odbiorcy. Frustrująca dla użytkownika jest składnia kodu. Sendmail został zaprojektowany by jego działanie było jak najszybsze i cel niewątpliwie został osiągnięty, jednak dla użytkownika kod jest nieczytelny i wygląda bardziej na pracę kryptografa niż programisty. Stąd ingerencja w działanie Sendmaila i jego konfiguracja są czynnościami skomplikowanymi.

Funkcjonalność, której warto się przyjrzeć, to kolejka. Nie wszystkie wiadomości mogą dotrzeć od razu. Jeżeli coś spowoduje, że wiadomość nie dotarła, Sendmail przetrzymuje wiadomości w kolejce, by spróbować przesłać je później. Wiadomość może być umieszczona w kolejce:

- Gdy serwer pocztowy, do którego chcemy przesłać wiadomość jest nieosiągalny lub wyłączony.

- Gdy wiadomość ma wielu odbiorców. Większość wiadomości może dotrzeć od razu jednak te, w których wystąpią pewne komplikacje są umieszczane w kolejce.

- Gdy mail jest zbyt duży, np. ma zamieszczone duże załączniki, a szybkość łącza w danej chwili spadła, serwer może umieścić wiadomość w kolejce i wznowić wysyłkę, gdy prędkość się ustabilizuje.

Kolejną ważną funkcjonalnością, są aliasy. Sendmail wykorzystuje je by wiadomość wysyłana na jeden adres mogła być przekierowana na inny. Są to swego rodzaju przydomki nadane adresowi e-mail. Pocztę można przekazać na parę różnych sposobów:

- root: sysadm - przekazanie poczty konta root do skrzynki użytkownika sysadm.

- http-errors: mark, mariusz, lukasz - poczta adresowana do http-errors jest przekazywana do trzech skrzynek użytkowników poczty mark, mariusz, lukasz.

- lukaszmail: /dev/null - przekazanie poczty do pliku, ścieżka wyznacza lokalizację pliku.

Jak widzimy rola Sendmaila jest duża w prawidłowym przepływie wiadomości poczty elektronicznej. Łączy się z on z siecią by pobrać adresowane do nas listy, a odebrane wiadomości, możemy w wygodny dla nas sposób przekierowywać. Droga jaką przebywa e-mail od nadawcy do odbiorcy jest skomplikowana, ale dobrze skonfigurowany serwer pocztowy pomaga nam by sprawnie i przede wszystkim bezpiecznie zarządzać naszą pocztą. [1]

3.3.2 RBL

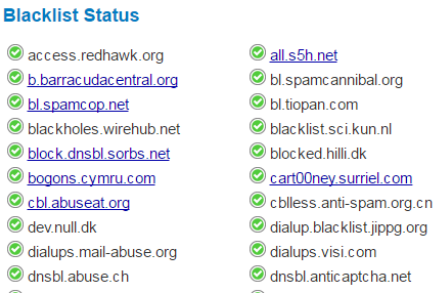
Gdy wiadomość przychodzi na naszą pocztę, Sendmail w pierwszej kolejności, tuż po nawiązaniu połączenia z MTA nadawcy sprawdza czy nie ma go na liście. Wiadomości z adresów wpisanych na listę są blokowane. Serwery działają niezależnie na podstawie analizy, serwer sam dodaje adresy które uznał za podejrzane, jednak użytkownik również ma możliwość zgłoszenia podejrzanego nadawcy i wpisanie go na RBL.

By zobrazować, cały proces skorzystam ze strony

`http://whatismyipaddress.com/`

na której znajduje się kilkadziesiąt serwerów RBL, po wpisaniu naszego adresu:

wyświetla nam się listę serwerów i ich status. W przypadku gdy adres nie znajduje się na RBL mamy znak koloru zielonego, natomiast w przypadku gdy jest na którymś z serwerów znak koloru czerwonego.
Serwerów RBL są setki i od nas zależy jak wiele z nich umieścimy na naszej poczcie, warto mieć na uwadze, że im więcej ich dodamy, tym dłużej będzie trwał cały proces porównywania i analizy adresów. Im bardziej surowy serwer, tym więcej wiadomości będzie blokowanych. Analogicznie w przypadku zbyt dużej wyrozumiałości, możemy natknąć się na niechcianą pocztę.



3.3.3 Milter

Samo słowo to połączenie angielskich słów mail i filter. Jest to dodatek do serwera pocztowego stworzony przez twórców programu Sendmail, umożliwiający dodanie filtrów antyspamowych i antywirusowych. Milter i Sendmail nie są tymi samymi procesami, milter działa jako zupełnie inny proces jednak porozumiewają się ze sobą za pomocą dedykowanego protokołu nazywanego z ang. "milter protocol", a samo połączenie, nosi nazwę sesji. Z głównej strony producenta możemy pobrać około 100 filtrów uzupełniających działanie Sendmaila. [1]

W naszej implementacji serwera Sendmail korzystamy z trzech filtrów milter:

1. **SMF-CLAMAV** - wtyczka do Sendmail'a, działająca w tle, służąca do skanowania wiadomości pocztowych wykorzystująca oprogramowanie ClamAV. Wtyczka ta przekazuje załączniki wiadomości do działającego w tle programu ClamAV, który z koleji skanuje je w poszukiwaniu kodu wirusa. Pozytywna odpowiedź programu ClamAV, czyli wykrycie wirusa powoduje umieszczenie wiadomości w kwarantannie. Tam wiadomość przebywa przez 7 dni, po czym jest usuwana. Odpowiednia informacja o wyniku skanowania dołączana jest do wiadomości jako nagłówek: X-Antivirus. Wszystkie informacje i nagłówki działań wtyczki są zapisywane w syslog'u.

2. **SMF-SPAMD** - wtyczka do Sendmail'a, umożliwiająca działanie SpamAssasina. Przekazuje wiadomość do działającego w tle SpamAssassin jako proces spamd. Wiadomości mają swój własny wynik z ustalonego przedziału liczbowego, jeżeli wiadomość otrzyma wysoki wynik, wtyczka zwraca wynik testu jako "pozytywny", wszystkie informacje i nagłówki są zapisywane w pliku syslog.

3. **SMF-ZOMBIE** - wtyczka do Sendmail'a służąca do filtrowania wiadomości w poszukiwaniu spamu. SMF-ZOMBIE skupia się na nadawcy wiadomości. Blokuje przesyłki od nieujawnionych adresatów, na podstawie rozszerzenia pliku odrzuca podejrzane załączniki by zapobiec rozprzestrzenianiu się wirusów. Wszystkie działania są zapisywane w pliku syslog.

3.3.4 LMTP

Rozszerzenie protokołu ESMTP (ESMTP rozszerzał protokół SMTP). Jest to protokół do zarządzania kolejką dostarczania poczty. Metaforycznie, LMTP można porównać do listonosza który dostarcza naszą pocztę. To MDA odpowiada bezpośrednio, gdzie umieści naszą wiadomość, czy będzie to katalog główny, czy folder spam. Zwykle MDA, realizujące protokół LMTP, uruchamia się na tej samej maszynie co MTA, ale nie jest to reguła. LMTP jest bardzo podobny do SMTP. Składnia i semantyka jest identyczna, jednak występuje kilka różnic. [19]

Komendy HELO I EHLO zostały zastąpione komendą LHLO. W komendzie DATA została dodana dodatkowa restrykcja, i jedna zmiana po końcowej kropce kończącej treść wiadomości. Zmiana z kropką, polega na tym, że serwer zwraca odpowiedź dla każdej poprzedniej poprawnie zadeklarowanej komendy RCPT, w kolejności w jakiej były deklarowane. Nawet jeżeli poprawne deklaracje RCPT dawały te samą ścieżkę dostępu, to i tak generowana jest odpowiedź

dla każdej oddzielnie. Jeżeli komenda RCPT jest poprawnie zadeklarowana, serwer bierze odpowiedzialność za dostarczenie wiadomości,

Przykład:

```
S: 220 foo.edu LMTP server ready
C: LHLO uwb.edu.pl
S: 250-uwb.edu.pl
S: 250-PIPELINING
S: 250 SIZE
C: MAIL FROM:<arkadiusz@bar.com>
S: 250 OK
C: RCPT TO:<omega@uwb.edu.pl>
S: 250 OK
C: RCPT TO:<mariusz@uwb.edu.pl>
S: 550 No such user here
C: RCPT TO:<lukasz@uwb.edu.pl>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Przykładowa treść maila...
C: ...itp. itp. itp.
C: .
S: 250 OK
S: 452 <lukasz@uwb.edu.pl> is temporarily over quota
C: QUIT
S: 221 uwb.edu.pl closing connection
```

3.3.5 SIEVE

To język interpretujący działanie skryptów do filtracji wiadomości e-mail, opisany w dokumencie RFC 3028. Został zaprojektowany by działać kompatybilnie z najpopularniejszymi serwerami pocztowymi np. Sendmail. Skrypty napisane w SIEVE wykonywane sa przez agenta LMTP podczas finalnego dostarczenia wiadomości, gdy jest ona przekazywana do skrzynki odbiorczej użytkownika. Istotną zaletą takiego rozwiązania jest niezależność od klienta pocztowego, czyli MUA. Na przykład w Thunderbird, czy MS Outlook, możemy ustawiać analogiczne filtry jak w SIEVE, ale wykonane zostaną one dopiero przy odbieraniu poczty. Po drugie, gdy odbieramy pocztę na różnych urządzeniach, korzystając z różnych programów, musielibyśmy wszędzie mieć taki sam zestaw filtrów. Tak więc dużym plusem SIEVE jest działanie po stronie serwera.

Bardzo ważną cechą SIEVE jest prostota kodu. Język jest przejrzysty, przez co każdy użytkownik może go modernizować własne filtry zgodnie ze swoimi potrzebami. Można to zrobić na dwa sposoby: logując się na serwer

oraz edytować i kompilować bezpośrednio plik zawierający skrypt SIEVE, albo wykorzystać wygodną wtyczkę do programu Thunderbird.

Jako przykład skryptu SIEVE obejrzyjmy skrypt, który wykorzystujemy w systemie, który wdrażamy. Jest to skrypt globalny, to znaczy, będzie wykonywany dla wszystkich użytkowników serwera. Znajduje się on w pliku:

```
/var/cfw/mda/sieve/global/spam.script
```

natomiast plik:

```
/var/cfw/mda/sieve/global/spam.bc
```

to jego postać skompilowana programem `sievec`. Treść skryptu jest następująca:

```
require ["fileinto"];

if header :contains "X-Antispam" "Positive" {
     fileinto "INBOX/SPAM";
     stop;
}

if header :contains "X-Antizombie" "Positive" {
     fileinto "INBOX/SPAM";
     stop;
}
```

Jak widać, składnia jest bardzo prosta i nawet nie znając sztuki programowania można takie filtry tworzyć samodzielnie. Pierwsza instrukcja warunkowa `if` sprawdza, czy nagłówek X-Antispam zawiera tekst Positive. Jeśli tak to wiadomość trafia do podkatalogu o nazwie SPAM i kończone jest wykonanie skryptu. Druga, analogiczna instrukcja dotyczy nagłówka X-Antizombie. [20]

Rozdział 4

Wdrożenie systemu filtrowania i dostarczania poczty

4.1 Pobranie odpowiednich pakietów

Na wstępie zaczniemy od pobrania odpowiednich pakietów ze strony

`http://math.uwb.edu.pl/ftp/solaris/x86/5.10/64/`

to jest

- Clamav - program antywirusowy,
- MDA - Mail Delivery Agent (Cyrus IMAP + skrypty administracyjne),
- Sendmail - Mail Transfer Agent,
- Perl - pakiet, w którym znajduję się SpamAssassin,
- Open SSL - biblioteka pozwalająca na szyfrowane połączenia SSL,
- MySQL - do stworzenia bazy danych, w której będą przechowywane konta pocztowe.

Ściągnięte pakiety należy wypakować poleceniem `gunzip`. Przykład:

```
gunzip clamav-0.99.1-Solaris10-x86.pkg.gz
```

Do zainstalowania pakietów wykorzystujemy polecenie `pkgadd`.

Przy pomocy opcji `-d` wskazujemy plik, z którego należy zainstalować odpowiedni pakiet. Przykład:

```
pkgadd -d clamav-0.99.1-Solaris10-x86.pkg
```

Jeżeli w systemie występują kolidujące pakiety, z tymi które chcemy zainstalować, należy je najpierw usunąć poleceniem `pkgrm`. Przykład:

```
pkgrm CFWsendmail
```

Po wypakowaniu i instalacji ściągniętych pakietów możemy przejść do konfiguracji zainstalowanego oprogramowania.

4.2 Konfiguracja oprogramowania

Konfiguracja Clamav'a

Zainstalowane programy należy uruchomić jako usługi. Poleceniem `svcs -p nazwa_usługi` podglądamy jej obecny stan z wyświetleniem wszystkich procesów uruchomionych w ramach tej usługi.

```
omega# svcs -p clamav
STATE          STIME    FMRI
disabled       11:34:53 svc:/application/security/clamav:default
omega#
```

Rysunek 4.1: Po instalacji Clamav'a jego status to disabled (wyłączony).

W tej chwili uruchomimy usługę clamav następującym poleceniem:

```
svcadm enable clamav
```

```
omega# svcs -p clamav
STATE          STIME    FMRI
online         11:53:14 svc:/application/security/clamav:default
               11:53:11      936 clamd
omega#
```

Rysunek 4.2: Status usługi Clamav został zmieniony na online, teraz jest aktywna.

Aby regularnie aktualizować sygnatury wirusów programu clamav dodajemy zadanie cron dla użytkownika clamav. W tym celu w katalogu

`/var/spool/cron/crontabs`

tworzymy plik o nazwie clamav, tak jak nazywa się użytkownik który będzie tę aktualizację uruchamiał. W pliku tym umieszczamy linijkę postaci:

```
30 4 * * * /opt/cfw/bin/freshclam --quiet
```

co oznacza, że codziennie o 4:30 będzie uruchamiany program `freshclam`, który pobiera aktualizację sygnatur. Dla sprawdzenia warto przełączyć się na użytkownika clamav poprzez `su - clamav` i uruchomić

`/opt/cfw/bin/freshclam`

```
omega# su - clamav
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
$ /opt/cfw/bin/freshclam
ClamAV update process started at Wed Aug 31 12:06:25 2016
WARNING: Your ClamAV installation is OUTDATED!
WARNING: Local version: 0.99.1 Recommended version: 0.99.2
DON'T PANIC! Read http://www.clamav.net/documents/upgrading-clamav
Empty script main-56.cdiff, need to download entire database
Downloading main.cvd [ 42%]
```

Rysunek 4.3: Aktualizacja sygnatur ClamAV.

Konfiguracja MySQL

Konfigurację MySQL zaczynamy od utworzenia katalogu i nadania mu odpowiednich uprawnień:

```
mkdir /data/mysql
chown mysql:mysql /data/mysql
```

Następnie instalujemy systemową bazę danych poprzez polecenie:

```
mysql_install_db
```

i uruchamiamy usługę poleceniem `svcadm enable mysql`. Do zarządzania bazą danych zakładamy hasło użytkownika root:

```
/opt/cfw/bin/mysqladmin -u root password 'spinacz'
```

Nadaliśmy hasło 'spinacz' teraz przygotujemy bazę danych do współpracy z programami IMAP, POP i Sendmail. W tym celu logujemy się jako root (konto, które ma pełną kontrolę nad systemem) do naszego MySQL'a

```
mysql -p -u root
```

Tworzymy bazę danych o nazwie dbISP

```
mysql > create database dbISP;
```

oraz tabelę o nazwie tMail, poprzez wykonanie gotowego skryptu SQL

```
mysql -p -u root < tMail.sql
```

W tabeli tMail będą przechowywane informacje o wszystkich kontach pocztowych. A wśród nich dwa o specjalnym przeznaczeniu o nazwie cyrus i mail. Są to konta administracyjne dla programów Cyrus IMAP i POP. Teraz ustawiamy w tabeli tMail domenę na nazwę hosta, na którym uruchomione są programy IMAP i POP (w naszym wypadku omega)

```
mysql > update tMail set domain = 'omega';
```

Na koniec tworzymy użytkownika w bazie MySQL, który będzie miał dostęp do tabeli tMail, nadając mu wszystkie uprawnienia z nią związane:

```
mysql > grant all on dbISP.tMail to mail identified by 'ma1l';
```

Użytkownik nazywa się mail i nadajemy mu hasło 'ma1l'. Login i hasło należy zapamiętać by móc konfigurować programy IMAP, POP i Sendmail.

```
mysql> select * from tMail
    -> ;
+------------+---------------------+---------------------+----------+-----------+---------------+--------+------------+-------------+
| cID        | cTS                 | cCT                 | username | domain    | password      | active | start_date | expire_date |
+------------+---------------------+---------------------+----------+-----------+---------------+--------+------------+-------------+
| 0000000001 | 2016-08-31 12:37:17 | 2016-08-31 12:37:17 | mail     | localhost | abgvB8R/yFO52 | Y      | 1970-01-01 | 2100-01-01  |
| 0000000002 | 2016-08-31 12:37:17 | 2016-08-31 12:37:17 | cyrus    | localhost | abgvB8R/yFO52 | Y      | 1970-01-01 | 2100-01-01  |
+------------+---------------------+---------------------+----------+-----------+---------------+--------+------------+-------------+
2 rows in set (0.00 sec)

mysql> update tMail set domain = 'omega';
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from tMail
    -> ;
+------------+---------------------+---------------------+----------+--------+---------------+--------+------------+-------------+
| cID        | cTS                 | cCT                 | username | domain | password      | active | start_date | expire_date |
+------------+---------------------+---------------------+----------+--------+---------------+--------+------------+-------------+
| 0000000001 | 2016-08-31 12:39:51 | 2016-08-31 12:37:17 | mail     | omega  | abgvB8R/yFO52 | Y      | 1970-01-01 | 2100-01-01  |
| 0000000002 | 2016-08-31 12:39:51 | 2016-08-31 12:37:17 | cyrus    | omega  | abgvB8R/yFO52 | Y      | 1970-01-01 | 2100-01-01  |
+------------+---------------------+---------------------+----------+--------+---------------+--------+------------+-------------+
2 rows in set (0.00 sec)

mysql> grant all on dbISP.tMail to mail identified by 'ma1l';
Query OK, 0 rows affected (0.00 sec)
```

Rysunek 4.4: Sprawdzamy czy domena rzeczywiście została zmieniona z 'localhost' na 'omega'.

Konfiguracja MDA

Następnym krokiem jest konfiguracja pakietu MDA, czyli programów Cyrus, IMAP i POP. Zaczynamy od utworzenia katalogu /data/mail w którym gromadzone będą wiadomości użytkowników z uwzględnieniem domeny, nazwy konta oraz folderów w skrzynce pocztowej. Nadajemy temu katalogowi uprawnienia aby mogły z nich te programy korzystać. Będą one uruchomione jako użytkownik 'mail'.

```
chown mail:mail /data/mail
```

Aby uruchomić IMAP i POP w wersji szyfrowanej SSL niezbędny jest certyfikat. Wykorzystamy ten wystawiony na domenę: uwb.edu.pl. Do katalogu `/opt/cfw/etc/mda` kopiujemy trzy pliki: `cacert.pem`, `server.crt` oraz `server.key`.

Właścicielem tych trzech plików musi być użytkownik `mail`, ten sam który jest używany do uruchomienia programów w usłudze MDA, i mają być one tylko do odczytu przez właściciela:

```
chown mail:mail server.* cacert.pem
chmod 0400 server.* cacert.pem
```

Teraz uruchamiamy usługę MDA:

```
svcadm enable mda
```

i możemy założyć konto użytkownika:

```
mdauseradd nazwa_użytkownika@domena
mdapasswd nazwa_użytkownika@domena
```

Konfiguracja Sendmail'a

Zaczynamy od modyfikacji opisu usługi Sendmail, czyli tzw. manifestu SMF, aby dostosować usługi od których zależy. Standardowo usługa sendmail zależy od usług smf-grey, smf-spamd, smf-zombie, smf-clamav oraz kilku innych. W naszej konfiguracji nie korzystamy z szarej listy, bo wielu użytkowników nie toleruje opóźnień w dostarczaniu poczty, dlatego usuwamy smf-grey z listy zależności sendmail'a. W tym celu należy odpowiednio zmodyfikować plik: `/var/svc/manifest/cfw/network/smtp-sedmail.xml`

Następnie zaimportować zmieniony plik XML wywołując polecenie:

```
svccfg import smtp-sendmail.xml
```

Poza modyfikacją usługi SMF musimy zmodyfikować plik konfiguracyjny Sendmail'a czyli plik: `/opt/cfw/etc/mail/sendmail.cf` i wyrzucamy z niego wpisy dotyczące smf-grey.

Następną zmianą w konfiguracji Sendmail'a jest modyfikacja pliku:

`/opt/cfw/etc/mail/mailertable`

Dopisujemy linijkę:

```
omega.uwb.edu.pl        cyrus:/var/cfw/mda/socket/lmtp
```

która mówi, że jeśli przyjdzie wiadomość do użytkownika w domenie: omega.uwb.edu.pl to zostanie ona przekierowana do LMTP.

Generujemy wersję binarną pliku konfiguracyjnego:

```
makemap hash mailertable < mailertable
```

Sendmail jako MSA do wysłania poczty wymaga autoryzacji użytkownika. Autoryzacja odbywa się za pośrednictwem biblioteki Cyrus SASL. Jej konfiguracja znajduję się w pliku: `/opt/cfw/lib/sasl2/Sendmail.conf.` Aby Sendmail mógł działać w wersji szyfrowanej STARTTLS w komunikacji z innymi MTA wymagane są odpowiednie certyfikaty. Instalujemy je w katalogu `/opt/cfw/etc/mail/certs/.` W naszym przypadku korzystamy z tych samych certyfikatów co dla MDA. Kopiujemy 3 pliki `ca.crt, server.key i server.crt.` Generujemy link symboliczny do pliku certyfikatu:

```
ln -s server.crt `openssl x509 -noout -hash -in server.crt`.0
```

Warto się upewnić czy root jest właścicielem tych plików i czy są one tylko do odczytu dla roota.

```
omega# ls -l
total 18
-r--------   1 root     root        4780 Sep  5 12:26 ca.crt
-r--------   1 root     root        1586 Sep  5 12:26 server.crt
-r--------   1 root     root        1675 Sep  5 12:26 server.key
omega# openssl x509 -noout -hash -in server.crt
f39cf96e
omega# ln -s server.crt
omega#
omega# ln -s server.crt
ln: cannot create ./server.crt: File exists
omega# ln -s server.crt f39cf96e.0
omega# ls -l
total 20
-r--------   1 root     root        4780 Sep  5 12:26 ca.crt
lrwxrwxrwx   1 root     root          10 Sep  5 12:30 f39cf96e.0 -> server.crt
-r--------   1 root     root        1586 Sep  5 12:26 server.crt
-r--------   1 root     root        1675 Sep  5 12:26 server.key
omega#
omega#
omega#
```

Rysunek 4.5: Właścicielem musi być root.

By uruchomić Sendmail'a wraz z procesami zależnymi, wpisujemy polecenie:

```
svcadm enable -r sendmail
```

4.3 Testy i uruchamianie

4.3.1 Zwykłe wiadomości

Po uruchomieniu naszego systemu wypadałoby go przetestować. Na początek sprawdzimy czy poczta dociera tam gdzie powinna. W tym celu z innego komputera w tej samej sieci wysyłamy maila testowego w następujący sposób. Wołamy standardowe polecenie:

```
mailx adres_odbiorcy
```

w naszym przypadku, odbiorcą będzie `lukasz@omega.uwb.edu.pl`.

```
mailx lukasz@omega.uwb.edu.pl
Subject: test

test

.
EOT
```

Wysłana wiadomość dotarła, co możemy zweryfikować w logu systemu na komputerze omega:

```
Sep  5 12:43:06 omega sendmail[1271]: [ID 801593 mail.info] u85Ah5G1001271:
   from=<mariusz@geo-01.uwb.edu.pl>, size=562, class=0, nrcpts=1, msgid=
   <201609050943.u859h6nh001700@geo-01.uwb.edu.pl>, proto=ESMTPS, daemon=
   MTA, relay=geo-01 [10.18.0.122]
Sep  5 12:43:06 omega sendmail[1271]: [ID 801593 mail.info] u85Ah5G1001271:
   Milter add: header: X-Antizombie: Negative: checked in 0.346sec at
   omega.uwb.edu.pl ([10.18.0.123])
Sep  5 12:43:06 omega sendmail[1271]: [ID 801593 mail.info] u85Ah5G1001271:
   Milter add: header: X-Antivirus: Negative: scanned in 0.039sec at
   omega.uwb.edu.pl ([10.18.0.123])
Sep  5 12:43:07 omega smf-spamd[1115]: [ID 810766 mail.info] u85Ah5G1001271:
   HAM (2.6/5.0) 1.498sec
Sep  5 12:43:07 omega sendmail[1271]: [ID 801593 mail.info] u85Ah5G1001271:
   Milter add: header: X-Spam-Checker-Version: SpamAssassin 3.3.2
   (2011-06-06) on\tomega.solaris-x86.net
Sep  5 12:43:07 omega sendmail[1271]: [ID 801593 mail.info] u85Ah5G1001271:
   Milter add: header: X-Spam-Status: No, score=2.6 bayes=0.5 required=5.0
   tests=ALL_TRUSTED=-1,\tDKIM_ADSP_NXDOMAIN=0.8,DNS_FROM_AHBL_RHSBL=2.438,
   NO_DNS_FOR_FROM=0.379\tautolearn=no version=3.3.2
Sep  5 12:43:07 omega sendmail[1271]: [ID 801593 mail.info] u85Ah5G1001271:
   Milter add: header: X-Antispam: Negative: score 2.6/5.0, scanned
   in 1.498sec at omega.uwb.edu.pl ([10.18.0.123])
Sep  5 12:43:08 omega sendmail[1273]: [ID 801593 mail.info] u85Ah5G1001271:
   to=<lukasz@omega.uwb.edu.pl>, delay=00:00:02, xdelay=00:00:01,
   mailer=cyrus, pri=120562, relay=localhost, dsn=2.0.0, stat=Sent
```

Standardowo w logu systemowym, Sendmail wpisuje wszelkie informacje o przetwarzanej poczcie i ewentualnych problemach. W przypadku naszego maila testowego, który otrzymał identyfikator: u85Ah5G1001271, kolejno:

- wiadomość została odebrana przez MTA,
- wiadomość jest przetwarzana przez filtr SMF-ZOMBIE,
- wiadomość jest przetwarzana przez filtr SMF-CLAMAV, czyli jest sprawdzana pod kątem obecności wirusów,
- wiadomość jest sprawdzana przez SpamAssassin, filtr smf-spamd umieszcza w wiadomości nagłówek podsumowujący tego sprawdzenia,
- na koniec informacja o tym, że wiadomość została dostarczona do skrzynki odbiorcy.

Treść wysłanej przez nas kontrolnej wiadomości, łącznie ze wszystkimi nagłówkami wygląda następująco:

```
Return-Path: <mariusz@geo-01.uwb.edu.pl>
Received: from omega.uwb.edu.pl ([unix socket])
 by omega (Cyrus v2.4.16) with LMTPA;
```

```
 Mon, 05 Sep 2016 12:43:07 +0200
X-Sieve: CMU Sieve 2.4
Received: from geo-01.uwb.edu.pl (geo-01 [10.18.0.122])
by omega.uwb.edu.pl (8.15.2/8.15.2) with ESMTPS id u85Ah5G1001271
(version=TLSv1 cipher=DHE-RSA-AES256-SHA bits=256 verify=NO)
for <lukasz@omega.uwb.edu.pl>; Mon, 5 Sep 2016 12:43:06 +0200 (CEST)
Received: from geo-01.uwb.edu.pl (localhost [127.0.0.1])
by geo-01.uwb.edu.pl (8.14.3/8.14.3) with ESMTP id u859h6WM001701
for <lukasz@omega.uwb.edu.pl>; Mon, 5 Sep 2016 11:43:06 +0200 (CEST)
Received: (from mariusz@localhost)
by geo-01.uwb.edu.pl (8.14.3/8.14.3/Submit) id u859h6nh001700
for lukasz@omega.uwb.edu.pl; Mon, 5 Sep 2016 11:43:06 +0200 (CEST)
Date: Mon, 5 Sep 2016 11:43:06 +0200 (CEST)
From: Mariusz Zynel <mariusz@geo-01.uwb.edu.pl>
Message-Id: <201609050943.u859h6nh001700@geo-01.uwb.edu.pl>
To: lukasz@omega.uwb.edu.pl
Subject: test
X-Antizombie: Negative: checked in 0.346sec at omega.uwb.edu.pl
   ([10.18.0.123])
X-Antivirus: Negative: scanned in 0.039sec at omega.uwb.edu.pl
   ([10.18.0.123])
X-Spam-Checker-Version: SpamAssassin 3.3.2 (2011-06-06) on
   omega.solaris-x86.net
X-Spam-Status: No, score=2.6 bayes=0.5 required=5.0 tests=ALL_TRUSTED=-1,
   DKIM_ADSP_NXDOMAIN=0.8,DNS_FROM_AHBL_RHSBL=2.438,NO_DNS_FOR_FROM=0.379
   autolearn=no version=3.3.2
X-Antispam: Negative: score 2.6/5.0, scanned in 1.498sec at omega.uwb.edu.pl
   ([10.18.0.123])
```

4.3.2 Nieprawidłowo sformatowana wiadomość - działanie usługi SMF-ZOMBIE

Jeżeli antispam lub antizombie zwrócą wynik "Positive" wiadomość trafi do spamu. Analogicznie jeżeli X-spam status zwróci wynik "Yes" wiadomość również trafi do katalogu SPAM.

Przykład wiadomości dającej pozytywny wynik usługi smf-zombie

```
Return-Path: <mariusz@math.uwb.edu.pl>
Received: from omega.uwb.edu.pl ([unix socket])
 by omega (Cyrus v2.4.16) with LMTPA;
 Tue, 06 Sep 2016 11:09:41 +0200
X-Sieve: CMU Sieve 2.4
Received: from localhost (geo-01 [10.18.0.122])
by omega.uwb.edu.pl (8.15.2/8.15.2) with ESMTP id u86994lh000582
for <lukasz@omega.uwb.edu.pl>; Tue, 6 Sep 2016 11:09:12
+0200 (CEST)
Date: Tue, 6 Sep 2016 11:09:06 +0200 (CEST)
Message-Id: <201609060909.u86994lh000582@omega.uwb.edu.pl>
Subject: test 5
\begin{textbf}
From: Gall Annonim <gdzies@niewiadomo.gdzie>
```

```
To: <undisclosed-recipient@omega.uwb.edu.pl>
\end{textbf}
X-Antizombie: Positive: Message for undisclosed recipients,
checked in 34.888sec at omega.uwb.edu.pl ([10.18.0.123])
X-Antivirus: Negative: scanned in 0.003sec at omega.uwb.edu.pl ([10.18.0.123])
X-Spam-Checker-Version: SpamAssassin 3.3.2 (2011-06-06) on omega.uwb.edu.pl
X-Spam-Status: No, score=4.9 bayes=0.5 required=5.0 tests=ALL_TRUSTED=-1,
DKIM_ADSP_NXDOMAIN=0.8,DNS_FROM_AHBL_RHSBL=2.438,MISSING_DATE=1.396,
TO_MALFORMED=1.247 autolearn=no version=3.3.2
X-Antispam: Negative: score 4.9/5.0, scanned in 0.378sec at
omega.uwb.edu.pl ([10.18.0.123])
```

Wiadomość od użytkownika `Gall Annonim` do

`undisclosed-recipient@omega.uwb.edu.pl`.

Usługa smf-zombie zauważyła niezidentyfikowanego odbiorcę i skierowała wiadomość do katalogu SPAM.

4.3.3 Wirusy - działanie usługi SMF-CLAMAV

Próba wysłania zainfekowanego pliku w załączniku, kończy się niepowodzeniem, tzn. wiadomość nie trafia do odbiorcy lecz do kwarantanny. Informację o tym możemy podejrzeć w logu systemowym:

```
Sep  6 12:11:39 omega sendmail[808]: [ID 801593 mail.info] u86ABbS6000808:
  from=<mariusz@math.uwb.edu.pl>, size=1266, class=0, nrcpts=1,
  msgid=<67b6ef44-1c5e-312b-99c8-a4a8c6ed335d@math.uwb.edu.pl>,
  bodytype=8BITMIME, proto=ESMTPS, daemon=MTA, relay=geo-01 [10.18.0.122]
Sep  6 12:11:39 omega sendmail[808]: [ID 801593 mail.info] u86ABbS6000808:
  Milter add: header: X-Antizombie: Negative: checked in 0.341sec
  at omega.uwb.edu.pl ([10.18.0.123])
Sep  6 12:11:39 omega smf-clamav[451]: [ID 358538 mail.notice]
  u86ABbS6000808: Sanesecurity.Foxhole.Zip_com.UNOFFICIAL, 0.006sec,
  geo-01, <mariusz@math.uwb.edu.pl> -> <lukasz@omega.uwb.edu.pl>
Sep  6 12:11:39 omega sendmail[808]: [ID 801593 mail.info] u86ABbS6000808:
  Milter add: header: X-Antivirus: Positive:
  virus Sanesecurity.Foxhole.Zip_com.UNOFFICIAL detected, scanned in
  0.006sec  at omega.uwb.edu.pl ([10.18.0.123])
Sep  6 12:11:39 omega sendmail[808]: [ID 801593 mail.info] u86ABbS6000808:
  milter=smf-clamav, quarantine=Virus detected
Sep  6 12:11:39 omega smf-spamd[433]: [ID 810766 mail.info] u86ABbS6000808:
  HAM (1.4/5.0) 0.424sec
Sep  6 12:11:39 omega sendmail[808]: [ID 801593 mail.info] u86ABbS6000808:
  Milter add: header: X-Spam-Checker-Version: SpamAssassin 3.3.2
  (2011-06-06) on omega.uwb.edu.pl
Sep  6 12:11:39 omega sendmail[808]: [ID 801593 mail.info] u86ABbS6000808:
  Milter add: header: X-Spam-Status: No, score=1.4 bayes=0.5 required=5.0
  tests=ALL_TRUSTED=-1,\tDNS_FROM_AHBL_RHSBL=2.438 autolearn=no
  version=3.3.2
Sep  6 12:11:39 omega sendmail[808]: [ID 801593 mail.info] u86ABbS6000808:
  Milter add: header: X-Antispam: Negative: score 1.4/5.0, scanned in
```

```
  0.424sec at omega.uwb.edu.pl ([10.18.0.123])
Sep  6 12:11:39 omega sendmail[808]: [ID 801593 mail.info] u86ABbS6000808:
  to=<lukasz@omega.uwb.edu.pl>, delay=00:00:00, mailer=cyrus, pri=31266,
  quarantine=Virus detected, stat=quarantined
```

Ostatnia linijka informuję nas o tym, że wiadomość znajduję się w kwarantannie. Możemy się o tym przekonać inaczej. W tym celu przechodzimy do katalogu: `/var/spool/mqueue` znajdują się tam dwa pliki o nazwach:

```
/var/spool/mqueue
omega#
omega# ls
dfu86ABbS6000808  hfu86ABbS6000808
omega# ls -l
total 6
-rw-------   1 root     smmsp        694 Sep  6 12:11 dfu86ABbS6000808
-rw-------   1 root     smmsp       1655 Sep  6 12:11 hfu86ABbS6000808
omega#
```

W pliku hfuu86ABbS6000808 są nagłówki wiadomości z których możemy odczytać, że rzeczywiście program antywirusowy znalazł tam w załączniku wirusa.

Wiadomość z kwarantanny możemy przenieść do skrzynki poleceniem:

```
unquarantine u86ABbS6000808
```

4.3.4 Spam - działanie usługi SMF-SPAMD

By zbadać działanie wtyczki pod kątem spamu, wysyłamy testową wiadomość zawierającą słowa które znajdują się na liście naszej usługi jako często występujące w niechcianej poczcie, działa tu technika filtrowania metodą analizy słownikowej. X-Spam-Status mówi nam o tym czy dana wiadomość powinna zostać oznaczona jako niechciana. W tym celu ma skalę punktową na podstawie której ocenia korespondencję. Progiem jest 5 punktów. Wszystkie wiadomości powyżej progu, trafiają do katalogu Spam.

Przykład Spam'u:

```
Return-Path: <mariusz@geo-01.uwb.edu.pl>
Received: from omega.uwb.edu.pl ([unix socket])
 by omega (Cyrus v2.4.16) with LMTPA;
 Tue, 06 Sep 2016 12:24:14 +0200
X-Sieve: CMU Sieve 2.4
Received: from geo-01.uwb.edu.pl (geo-01 [10.18.0.122])
by omega.uwb.edu.pl (8.15.2/8.15.2) with ESMTPS id u86AOCu0000824
(version=TLSv1 cipher=DHE-RSA-AES256-SHA bits=256 verify=NO)
for <lukasz@omega.uwb.edu.pl>; Tue, 6 Sep 2016 12:24:14 +0200 (CEST)
Received: from geo-01.uwb.edu.pl (localhost [127.0.0.1])
by geo-01.uwb.edu.pl (8.14.3/8.14.3) with ESMTP id u869OD6I001608
for <lukasz@omega.uwb.edu.pl>; Tue, 6 Sep 2016 11:24:13 +0200 (CEST)
Received: (from mariusz@localhost)
```

```
by geo-01.uwb.edu.pl (8.14.3/8.14.3/Submit) id u869ODwC001607
for lukasz@omega.uwb.edu.pl; Tue, 6 Sep 2016 11:24:13 +0200 (CEST)
Date: Tue, 6 Sep 2016 11:24:13 +0200 (CEST)
From: Mariusz Zynel <mariusz@geo-01.uwb.edu.pl>
Message-Id: <201609060924.u869ODwC001607@geo-01.uwb.edu.pl>
To: lukasz@omega.uwb.edu.pl
Subject: test spam viagra earn lots of money
X-Antizombie: Negative: checked in 0.340sec at omega.uwb.edu.pl
  ([10.18.0.123])
X-Antivirus: Negative: scanned in 0.007sec at omega.uwb.edu.pl
  ([10.18.0.123])
X-Spam-Checker-Version: SpamAssassin 3.3.2 (2011-06-06) on omega.uwb.edu.pl
X-Spam-Status: Yes, score=7.8 bayes=0.5 required=5.0 tests=ALL_TRUSTED=-1,
  DKIM_ADSP_NXDOMAIN=0.8,DNS_FROM_AHBL_RHSBL=2.438,DRUGS_ERECTILE=2.221,
  NO_DNS_FOR_FROM=0.379,SERGIO_SUBJECT_VIAGRA01=2.983 autolearn=no
  version=3.3.2
X-Antispam: Positive: score 7.8/5.0, scanned in 0.417sec at omega.uwb.edu.pl
  ([10.18.0.123])


viagra viagra viagra earn money fast and viagra
```

Jak widzimy, X-Spam Status dał wynik 7.8, czyli o 2.8 punktu powyżej progu. E-mail trafia do katalogu SPAM.

Bibliografia

[1] Costales B., Jansen G., Assmann C. *Sendmail*, O'Reilly Media, 1993.

[2] https://alterweb.pl/-spam-definicja-rodzaje-historia-powstania\-oraz-sposoby-ochrony

[3] https://avast.com/pl-pl/c-ransomware

[4] https://drzewo-wiedzy.pl/?page=artykul&id=88

[5] https://e-marketing.pl/artyk/artyk74.php

[6] https://en.wikipedia.org/wiki/MX_record

[7] https://itl.pl/index.php/user_templates,user_tpl,docs4h.html

[8] https://kis.pwszchelm.pl/publikacje/VII/Zarychta.pdf

[9] http://makeuseof.com/tag/pop-vs-imap/

[10] https://milter-manager.sourceforge.net/reference/introduction.html

[11] https://nss.et.put.poznan.pl/study/projekty/sieci_komputerowe/protokoly_pocztowe/html/pop3.html

[12] https://pl.wikipedia.org/wiki/Domain_Name_System

[13] https://pomoc.home.pl/baza-wiedzy/czym-sa-i-do-czego-sluza-listy-rbl/

[14] https://pomoc.home.pl/baza-wiedzy/dlaczego-powinienem-korzystac-z-protokolu-pocztowego-imap/

[15] https://runbox.com/email-school/how-email-works/

[16] https://searchmidmarketsecurity.techtarget.com/definition/email-virus

[17] https://securelist.pl/threats/internal/7058,co_to_jest_phishing.html

[18] https://staff.elka.pw.edu.pl/~mgolisze/papers/Transport_poczty_elektronicznej_art.pdf

[19] https://tools.ietf.org/html/rfc2033

[20] https://tools.ietf.org/html/rfc5228.html

[21] https://uci.umk.pl/index.php/Rónice_pomidzy_protokoaami_POP3_i_IMAP

[22] https://udyomedia.pl/def-E-mail.html

[23] https://wyborcza.biz/biznes/1,147883,16407710,Zalew_wirusow_w_polskim_internecie__Hakerzy_podszywaja.html