

## SMTP, Email Tracing, and Spam Filtering Modified from Jake Robert's "SMTP and Tracing E-mail"

### Objective:

The primary objective of this lab is to gain a deeper understanding of Email systems, how they work, how they are abused, and how they can be protected. In order to accomplish this, the lab will involve several sub-objectives. These secondary objectives include: directly using SMTP to send an email, forging an email's headers, experimenting with a Spam reporting utility (SpamCop.net), and the installation of a Spam filtering system for an Email server (using SpamAssassin/Sendmail/Procmail).

### References:

Jake Robert's IT Security lab "SMTP and Tracing Email"

<http://www.stopspam.org/email/headers.html>

<http://www.spamcop.net>

<http://www.spamassassin.org>

<http://evillair.netdojo.com/howto/spamassassin.html>

### Equipment/Programs Used:

PC with RedHat 9.0  
PC with Windows XP and telnet client  
SpamAssassin  
Sendmail  
Procmail

### Procedures:

This section will describe the process of learning about the SMTP protocol, learning how it is used and abused, and the process of installing and configuring a Spam filter for a Linux Email server. This section is further broken down into the following subsections:

Use SMTP to send an email \*\*

Forge an email's headers \*\*

Trace an email's origin \*\*

---

\*\* The procedures for these items were almost entirely taken from Jake Robert's original document, with a few minor adjustments and corrections.

## Report Spam to SpamCop Install and configure a Spam filter **Task 1**

### Use SMTP to Send an Email

For this task an SMTP server is needed. Most ISPs provide an SMTP server, however, it can be difficult to use the SMTP server due to many of the anti-spam measures in place.

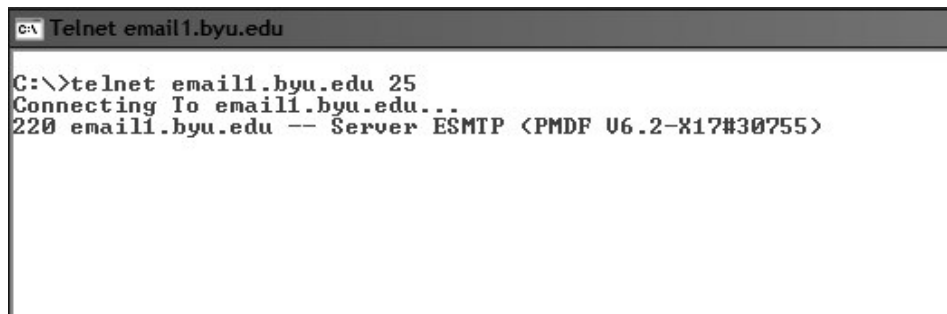
BYU's email servers will allow anyone to send email from an on-campus IP address so we'll use it for the demonstration.

If an email server is not available there are some free SMTP servers that can be installed on your local machine.

### Connect to the SMTP Server

Connect to the email server using a telnet client. SMTP servers use TCP port 25. In windows this can be done from the RUN window. Execute the command:

```
telnet email1.byu.edu 25
```

A screenshot of a Windows command prompt window titled "C:\ Telnet email1.byu.edu". The window shows the following text:

```
C:\>telnet email1.byu.edu 25
Connecting To email1.byu.edu...
220 email1.byu.edu -- Server ESMTP (PMDF U6.2-X17#30755)
```

### The Basic SMTP Conversation

Once connected to the SMTP server you are presented with a greeting:

```
220 email1.byu.edu -- Server ESMTP (PMDF V6.2-X17#30755)
```

So, being a polite user, say hello and introduce yourself. The SMTP command for hello is HELO or EHLO:

```
HELO shrek.byu.edu
```

The server responds letting me know if its okay to send mail:

```
250 email1.byu.edu OK, [10.0.2.100].
```

Now you need to identify what email address you are sending from:

```
MAIL FROM: jake_roberts@byu.edu
```

The server responds:

```
250 2.5.0 Address Ok.
```

Now you tell the server the email address you wish to send to:

```
RCPT TO: jwr24@email.byu.edu
```

And the server responds:

```
250 2.1.5 jwr24@email.byu.edu OK.
```

That is everything the SMTP server needs to send the email. Now we can send the message data. This should include everything that the email client on the other end will want to know including the actual message. To indicate to the server that you are sending the message send the command:

```
DATA
```

The server responds:

```
354 Enter mail, end with a single ".".
```

Now send all your data. When you're done place a single period (.) on a line by itself:

```
From: Jacob Roberts <jake_roberts@byu.edu>
To: Jake Roberts <jwr24@email.byu.edu>
Subject: This is an email sent by hand.
This is the message body. Congratulations you've sent an
email.
.
```

The server responds:

```
250 2.5.0 Ok.
```

The email has been sent. Now that we're done let's say good-bye:

```
QUIT
```

The server responds:

```
221 2.3.0 Bye received. Goodbye.
```

The DATA section is the most interesting for our purposes. As far as the SMTP server is concerned the DATA section is all message data and does not direct how the message is sent. The From, To, and Subject lines have no meaning for the server. This information is for the recipient's email client and shows up in the To, From, and Subject lines when the email is received.

This is the basic process an email client uses to send SMTP email messages.

### **Forge and Email's Header**

Now let's think like a spammer. We want to hide our identity if we can. People seem to get angry at all the friendly messages we are sending out. The SMTP conversation above can be changed to hide our identity:

```
220 email1.byu.edu -- Server ESMTP (PMDf V6.2-X17#30755)
HELO nowhere.com
```

```
250 email1.byu.edu OK, [10.0.2.100].
MAIL FROM: fakeaddress@nowhere.com
250 2.5.0 Address Ok.
RCPT TO: jwr24@email.byu.edu
250 2.1.5 jwr24@email.byu.edu OK.
DATA
354 Enter mail, end with a single ".".
From: Some False Name <not_A_real_ADDRESS@fake.org>
To: Not Sent to <you@yourdomain.com>
Subject: This is a forged Email
Hehe. Now you can't tell who I am.
.
250 2.5.0 Ok.
QUIT
221 2.3.0 Bye received. Goodbye.
```

The commands we sent to the server are highlighted. We faked our HELO information and our MAIL FROM information. We have to put the real recipient address in the RCPT TO command so the email gets to where it needs to go. But in the DATA section we can put a false From address and a false To address.

Remember, the server doesn't care what is in the DATA section. It knows the real recipient address because we gave it in the RCPT TO command. Email clients use the To and From information in the DATA section. So instead of showing the real recipient (jwr24@email.byu.edu) the email client shows that the email was sent to: *Not Sent to <you@yourdomain.com>* just as we specified in the DATA section.

## Trace an Email's Origin

If a spammer (or virus) can forge their identifying information what can be done to stop them? Fortunately we are able to trace the path the email takes across the Internet to our email box. Most email clients have the ability to show the email's headers.

## Email Headers

The email headers include a set of information about the email message. The From, To, and Subject lines are part of the email headers. Each SMTP server that the email travels through records information about the transaction and appends it to the top of the header as a Received line. Using the Received lines we can trace the email to the IP address of the sender by following the chain of servers that sent and received the email.

Here is a set of real email headers from a spam I received:

```
Received: from lyra.hurrah.com (lyra.hurrah.com [208.151.247.37])
by bfish.hurrah.com (8.11.3/8.11.3) with ESMTP id hA657jQ24363
for <ancientamerica.org.postmaster@mail.hurrah.com>; Wed, 5 Nov
2003 21:07:45 -0800
```

**Received:** from mail.voxmail.com.br ([200.190.61.200])  
by lyra.hurrah.com (8.9.3/8.9.3) with ESMTP id VAA11194  
for <"bowler, matthew,1147 south main orem, ut 84058-6847  
us,,,,,8012242218,,matt"@mail.ancientamerica.org>; Wed, 5 Nov 2003  
21:07:02 -0800 (PST)

**From:** jpaterr@yahoo.com  
**Message-Id:** <200311060507.VAA11194@lyra.hurrah.com>  
**Received:** (qmail 26557 invoked by uid 7794); 4 Nov 2003 15:29:29  
-0000

**Received:** from jpaterr@yahoo.com by mail.voxmail.com.br by uid  
7791 with qmail-scanner-1.16  
(clamscan: 0.60. Clear:. Processed in 8.279132 secs); 04 Nov 2003  
15:29:29 -0000

**Received:** from pe244065.user.veloxzone.com.br (HELO  
smtp0391.mail.yahoo.com) (webmaster@200.164.244.65)  
by mail.voxmail.com.br with SMTP; 4 Nov 2003 15:29:17 -0000

**Date:** Wed, 5 Nov 2003 17:29:12 GMT  
**X-Priority:** 3  
**To:** bowler@lyra.hurrah.com, matthew@lyra.hurrah.com,  
1147.south.main.orem@lyra.hurrah.com, ut.84058-  
6847.us@lyra.hurrah.com,  
8012242218@lyra.hurrah.com, matt@mail.ancientamerica.org  
**Subject:** WOW VIDEO E-MAIL....A GOLDEN OPPORTUNITY\$\$\$\$  
**Mime-Version:** 1.0  
**Content-Type:** text/plain; charset=us-ascii  
**Content-Transfer-Encoding:** 7bit

I've indented and highlighted each header item to help the headers more readable. The email headers are open to any application to add anything it wishes and they often do. These usually are in the form *X-[Name]: [information]*, but not always. Most of this information helps the email client know how to handle or format the message once it has been received.

Each SMTP server places a Received line at the top of the headers as it processes the email. So to trace the message we have to work from the bottom up. Be careful though because a clever spammer can insert their own falsified Received lines to try and throw you off their track.

The first received line says mail.voxmail.com.br received this message from pe244065.user.veloxzone.com.br and it notes that in the HELO message the sender claimed to be smtp0391.mail.yahoo.com.

The next Received line looks like some internal server scanning and doesn't help further the trace. It confuse us a little because it breaks up the by/from chain. I think we can safely ignore this Received line.

The third Received line says that lyra.hurrah.com received the message from mail.voxmail.com.br and it records its IP address (SMTP servers often record the IP address so it can be verified against the HELO message). This Received line seems valid. The chain is intact: From veloxzone.com to voxmail.com to hurral.com.

The fourth line shows that lyra.hurrah.com delivered the message to bfish.hurrah.com which is my email server. This looks legitimate because the chain is still intact. Besides, I trust my own email servers.

Usually the Received lines will reveal the IP address of my spammer, but in this case all we have is a DNS name. So a simple ping will hopefully retrieve the IP address:

```
ping pe244065.user.veloxzone.com.br
Pinging pe244065.user.veloxzone.com.br [200.164.244.65] with 32 bytes of data:
Reply from 200.164.244.65: bytes=32 time=549ms TTL=45
Ping statistics for 200.164.244.65:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 549ms, Maximum = 549ms, Average = 549ms
```

So our spammer is at 200.164.244.65.

Now that we have the spammers IP address we have to resist the urge revenge and be responsible Netizens. Most ISPs have policies against using accounts for spamming and we should contact them to see if they will shut down this spammer account.

IP addresses are registered to the ISPs who must make this information public. Most registries have websites with whois tools that allow us to find the ISP and their contact information. Judging from the DNS name this system is in Brazil (br) so we need to find a registry in South America. Unfortunately I don't know where that registry is so let's try the American registry. This is located at [www.arin.net](http://www.arin.net). ARIN stands for America Registry for Internet Numbers.

The ARIN website has a whois search on their front page:



By entering our spammers IP address we can get information about the correct registry to consult. Among the information it returns is this note:

```
Comment: This IP address range is under LACNIC responsibility for further
Comment: allocations to users in LACNIC region.
Comment: Please see http://www.lacnic.net/ for further details, or check the
Comment: WHOIS server located at whois.lacnic.net
```

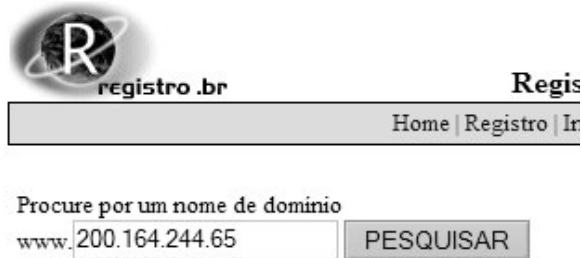
Now let's consult the LACNIC (Latin American and Caribbean IP address Regional) registry. They also have a whois tool on the front page:



Among the results is this note:

remarks: These addresses have been further assigned to Brazilian users.  
remarks: Contact information can be found at the WHOIS server located  
remarks: at whois.registro.br and at http://whois.nic.br

It looks like we have one more registry to go. They also have a whois tool on the front page. I hope you can read Portuguese:



This registry finally returns the contact information we are looking for:

inetnum: 200.164/16  
asn: AS7738  
ID abusos: CGR13  
entidade: Telemar Norte Leste S.A.  
documento: 002.558.134/0001-58  
responsável: Marcello Lugon  
endereço: Rua Humberto de Campos, 425, 7º andar  
endereço: 22430-190 - Rio de Janeiro - RJ  
telefone: (021) 31311343 []

ID: CGR13  
nome: Centro de Gerencia de Rede TELEMAR  
e-mail: **abuse@TELEMAR.NET.BR**  
endereço: Praia de Botafogo, 166, 7 andar  
endereço: 22250-040 - Rio de Janeiro - RJ  
telefone: (21) 080028234 []  
criado: 05/06/2000  
alterado: 13/08/2003

ID: MAL516  
nome: Marcello Lugon  
e-mail: mlugon@TELEMAR.COM.BR  
endereço: Rua Humberto de Campos, 425, 7º andar  
endereço: 22430-190 - Rio de Janeiro - RJ  
telefone: (021) 3131-1343 [-]  
criado: 09/10/2000  
alterado: 12/09/2002

remarks: Security issues should also be addressed to  
remarks: nbsso@nic.br, <http://www.nbsso.nic.br/>  
remarks: Mail abuse issues should also be addressed to  
remarks: **mail-abuse@nic.br**

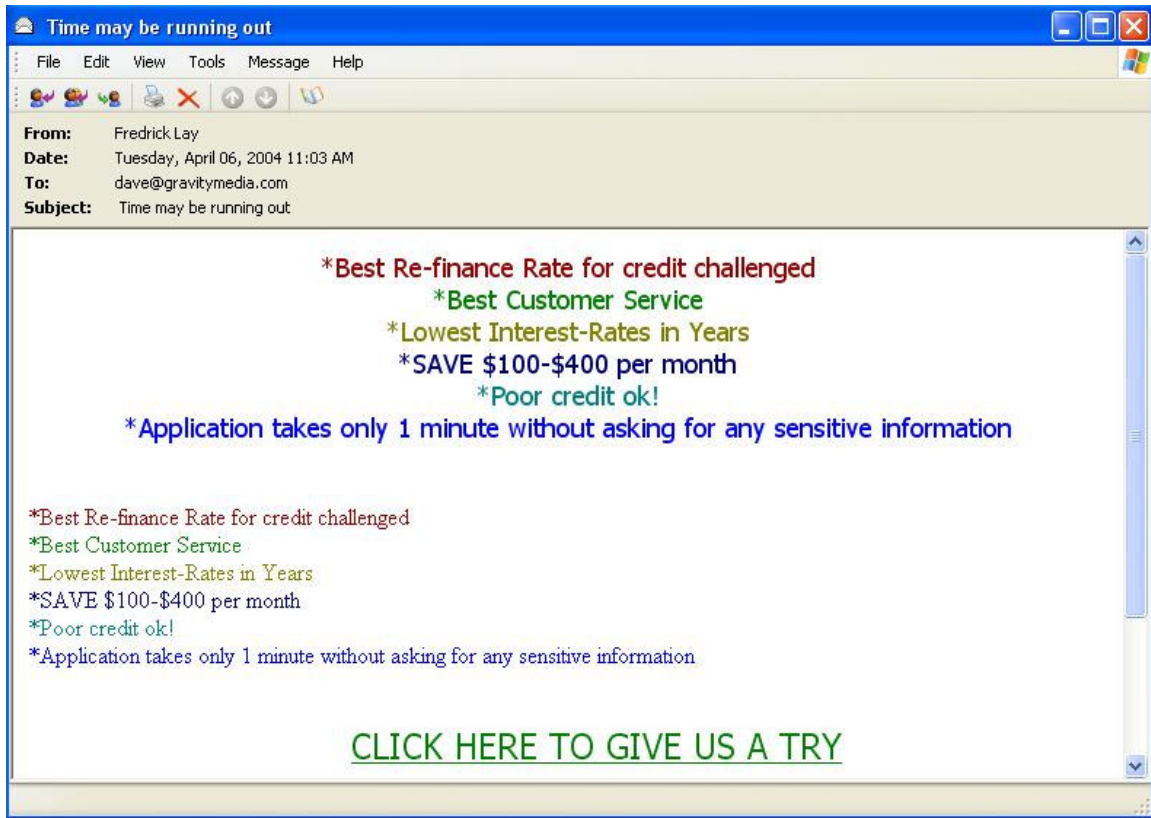
From this we find two contact email addresses (highlighted) that we can report this spammer to. Now we should report as much detail to them as we have. We need to forward the original spam message to them. It is very important to copy in the email headers because the email client will not include them. This allows the ISP to verify that an IP they are registered to really sent the email. It may also be helpful if we explain how we found the contact information we are using. Of course, our complaint could just be deleted since we don't speak Portuguese, but at least we did our part.

### **Report Spam to SpamCop**

SpamCop.net is a free service provided to help reduce the amount of Spam that is being generated on the internet. SpamCop uses an automated system to receive sample Spam emails from the public to analyze the email and determine the source of the Spam. Once the email is analyzed and is confirmed (by the user) to be Spam, SpamCop sends an automatic report to the ISP for the origin of the mail.

In order to use SpamCop's services, the user must first register (for free) to use the service. The registration allows for anonymous posting of Spam. Due to the rapid change in web structure, no screenshots of the registration process for SpamCop are provided. Once a user has registered to use SpamCop, they are assigned an email address that will be used in reporting all future spam. The email address that was assigned for this example was [submit.WmtIQe8grd3Sn0tu@spam.spamcop.net](mailto:submit.WmtIQe8grd3Sn0tu@spam.spamcop.net).

Once the email address is received, the Spam reporting process is very simple. Once Spam is received by the user, it simply needs to be forwarded to the email address provided by SpamCop. It is important that the user forward the email as an attachment in order to preserve the original email headers. If this is not done, then the email will not be properly analyzed and the Spam cannot be reported. Out of convenience, SpamCop also allows for more than one Spam attachment to be included at the same time. A copy of the sample Spam that was sent to SpamCop is shown below.



To report this Spam, it was simply forwarded as an attachment to the SpamCop address provided after registering for the service. Shortly after being submitted to SpamCop, an email from SpamCop was received with a link to continue the process of reporting the Spam. A partial result of the Spam analysis is provided below.

**Parsing header:**

Received: from c-67-168-191-41.client.comcast.net (c-67-168-191-41.client.comcast.net [67.168.191.41]) by tristan.gravitymedia.com (8.12.8/8.12.8) with SMTP id i374C6Xr027025 for <x>; Tue, 6 Apr 2004 22:12:07 -0600

67.168.191.41 found  
host 67.168.191.41 = c-67-168-191-41.client.comcast.net (cached)  
host c-67-168-191-41.client.comcast.net (checking ip) = 67.168.191.41  
Possible spammer: 67.168.191.41  
67.168.191.41 is not an MX for c-67-168-191-41.client.comcast.net  
host c-67-168-191-41.client.comcast.net (checking ip) = 67.168.191.41  
Received line accepted

Received: from pinch-s21.cod.aol.com ([224.252.16.216]) by as3-r06.hotmail.com with Microsoft SMTPSVC(5.0.2195.6824); Tue, 06 Apr 2004 12:12:06 -0500

224.252.16.216 found  
host 224.252.16.216 (getting name) no name  
67.168.191.41 not listed in dnsbl.njabl.org  
67.168.191.41 listed in cbl.abuseat.org ( 127.0.0.2 )  
Open proxies untrusted as relays

**Tracking message source: 67.168.191.41:**

[Routing details for 67.168.191.41](#)

[\[refresh/show\]](#) Cached whois for 67.168.191.41 : abuse@comcast.net

Using abuse net on abuse@comcast.net  
abuse net comcast.net = abuse@comcast.net  
Using best contacts abuse@comcast.net

**Yum, this spam is fresh!**

67.168.191.41 not listed in dnsbl.njabl.org  
67.168.191.41 not listed in dnsbl.njabl.org  
67.168.191.41 listed in cbl.abuseat.org ( 127.0.0.2 )  
67.168.191.41 is an open proxy  
67.168.191.41 not listed in plus.bondedsender.org  
67.168.191.41 not listed in query.bondedsender.org  
67.168.191.41 not listed in iadb.isipp.com

**Resolving link obfuscation**

Once the email has been analyzed and the user verifies that it is indeed unwanted Spam, then the user simply submits the analysis and SpamCop sends an automatic report to the ISP of the traced origin to inform them that Spam is coming from the designated source IP.

This process makes reporting Spam very simple and is quite effective in helping to reduce the amount of Spam that is generated. Due to the widespread use of SpamCop's services, and particularly their networked Blacklists, ISPs are often very responsive to SpamCop's reports. Although the user will not hear back about the result, they can rest assured knowing that a detailed and complete report of the Spam was sent in their behalf. This eliminates the need for the administrator of a mail system to analyze and report mail abuse on an individual basis.

## **Install and Configure a Spam Filter**

This section is further broken down into the following sections: RedHat 9 Pre-Install Requirements, Installing and Configuring SpamAssassin, Configuring Procmail, and Testing the Spam Filter.

### **RedHat 9 Pre-Install Requirements**

Although installing a Spam filter using SpamAssassin is fairly straightforward, there are a few basic requirements that must be met. The installation of Linux used must contain Sendmail and Procmail for this installation example to work properly. The default installation of RedHat 9 comes with both of these already installed. Also, in order to provide more functionality to the mail server and to better simulate a corporate mail server, other services such as POP3 and IMAP were also included in the installation. These options may be selected during the install process of RedHat 9. If the machine being used is not being installed with a fresh copy of Linux, then any POP3 server will work adequately for the examples used in this lab.

### **Installing and Configuring SpamAssassin**

First, relocate to the `/usr/local/src` directory to download the following packages (using `'wget'`):

<http://eu.spamassassin.org/released/RPMs/perl-Mail-SpamAssassin-2.63-1.i386.rpm>

<http://eu.spamassassin.org/released/RPMs/spamassassin-2.63-1.i386.rpm>

<http://eu.spamassassin.org/released/RPMs/spamassassin-tools-2.63-1.i386.rpm>

Once the RPMs have been successfully downloaded, install them by issuing the following commands in the order given:

```
rpm -ivh perl-Mail-SpamAssassin-2.63-1.i386.rpm
```

```
rpm -ivh spamassassin-2.63-1.i386.rpm
```

```
rpm -ivh spamassassin-tools-2.63-1.i386.rpm
```

If there were no problems encountered, then SpamAssassin is now installed on the machine. However, it is not yet analyzing mail as it comes into the server. To do this, procmail needs to be configured to send incoming mail through spamassassin for analysis.

## Configuring Procmail

On a default RedHat 9 installation, sendmail is already configured to send all mail through procmail, however, there is no procmailrc file set up. In order to accomplish this task, the procmail file needs to be created in the `/etc` directory. Create the proper file by issuing the command `'touch /etc/procmailrc.'` Then edit this file to look like the following:

```
DROPPRIVS=yes
LOGFILE=/var/log/procmail

:0fw
* < 256000
|/usr/bin/spamassassin -P

:0e
{
EXITCODE=$?
}
```

This will now send all mail through Spamassassin before delivering it to the intended recipient. By default, if an email receives 5 points from SpamAssassin, then it is tagged as spam, and the subject line will begin with `*****SPAM*****` instead of the regular subject line (the original subject is in-tact, the spam tag is merely appended to the beginning).

In order to add some filtering capability, it is first necessary to understand what the procmailrc file is doing. The first line, `"DROPPRIVS=yes"`, merely tells procmail to drop root privileges. This adds some measure of security but will then prevent procmail from being able to create directories or files (which may be useful as will be shown). The second line, `"LOGFILE..."`, is for logging purposes and can be left out if no logging is desired. The next three lines (beginning with `":0..."` and ending with `"|/usr..."`) tell procmail that if the email is less than 256K in size, send it on to spamassassin. This is done because spam is rarely ever larger than this size and it helps to reduce the overhead caused by spamassassin on larger emails. The remaining section tells procmail to then exit and continue with normal processing. As can be seen, there is no filtering being done—only analysis.

To begin filtering, the administrator must first decide how to handle the filtering. In this example, all email that is marked to have received more than 8 spamassassin points will be sent to a dynamically built file based on the current date. Everything else will be delivered as normal to the users of the system. This allows the email to be saved by the system administrator in case anybody complains of having missed an email, but helps to reduce the amount of spam going through to user's inboxes. To accomplish this filtering, the procmailrc file should look like the following:

```
DROPPRIVS=no
LOGFILE=/var/log/procmail
DATE=`date +%Y-%m-%d`

:0fw
```



**Results:**

The SMTP protocol was learned about and used to send an email directly through a telnet client without an email client. The header information of an email was successfully forged and the source of a real Spam email was able to be traced. Several Spam were also submitted to SpamCop.net, an automated spam reporting service. In addition to this, a very successful spam filtering system was deployed on a live business network and proved to be very valuable in filtering out unwanted emails. Nearly 25,000 emails were filtered in slightly longer than a month, including almost 1,200 viruses. Less than one quarter of one percent of legitimate email was falsely marked as spam.

**Conclusions:**

Email is critical to any business. Therefore, as an IT professional it also becomes vital to understand how email, particularly SMTP, functions on a network. This lab helped the student to gain a better understanding of how email can be abused on a network. The SMTP protocol truly is “simple,” as the name suggests. It was surprisingly easy to forge the headers of an email, without being detected by the mail server. Unfortunately, this ease of being able to forge an email has facilitated the widespread abuse of email. The ability to forge the source of an email has made spamming a very easy and widely-used technique for marketing.

The lab also provided a means to better understand some of the tools that are available to help assist in preventing the abuse, or at least in preventing the annoyance of an abused email system. The company that was used to test the deployment of the spam filtering system was very grateful for the services that spam filtering provided. Through a relatively simple process, the company was able to save time, money, and frustration by not having to deal with the thousands of unwanted emails that normally plagued the users' inboxes.