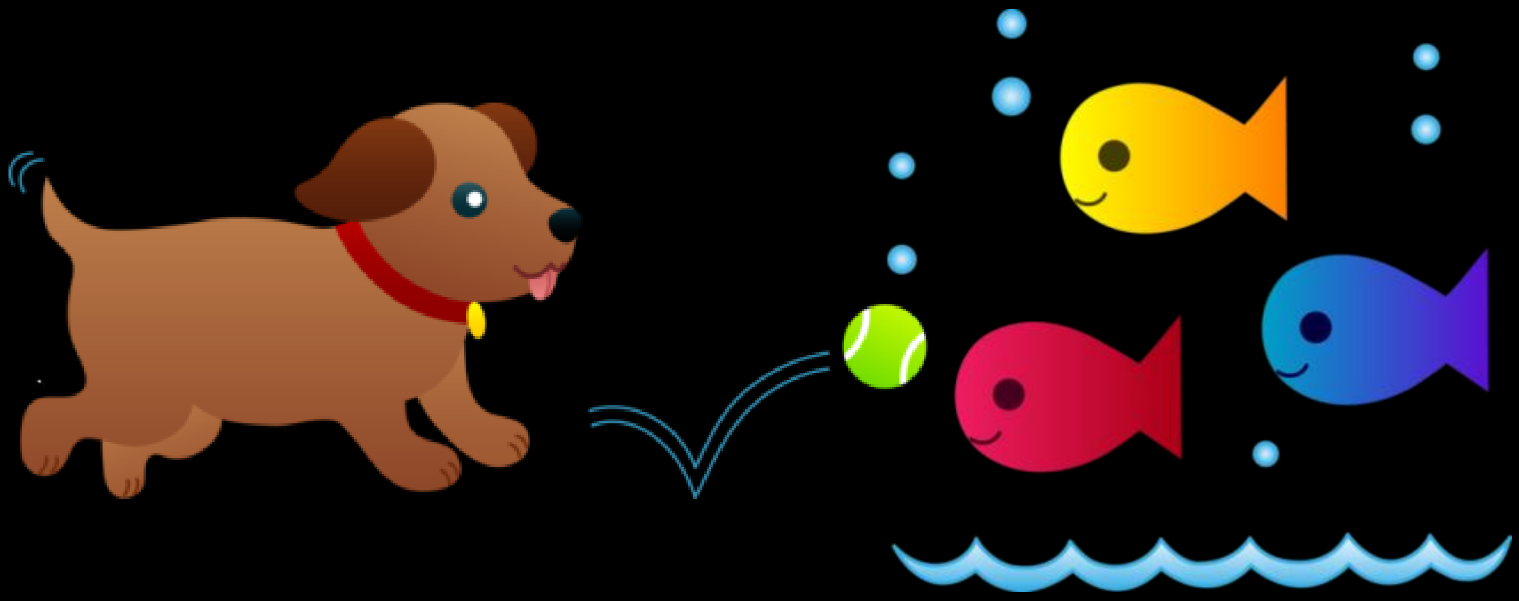
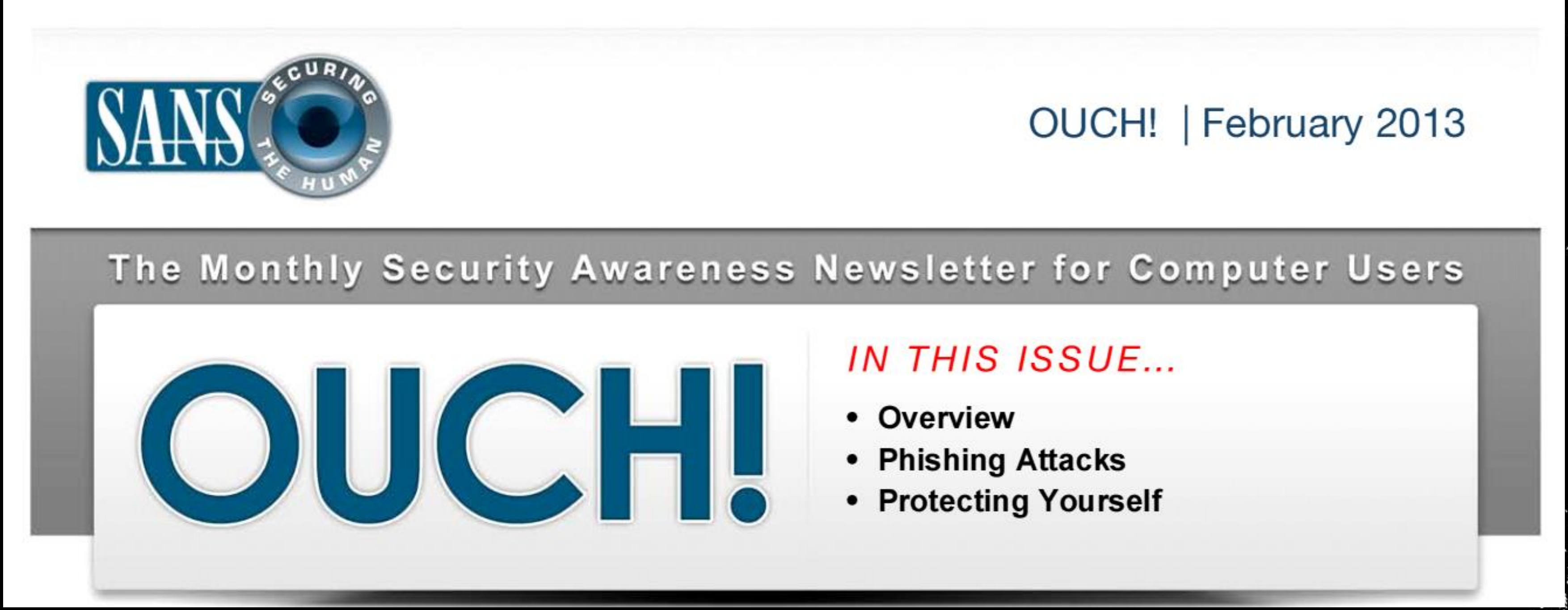
Fetching & Phishing



Phishing IS Not Going Away



SANS SECURING THE HUMAN

OUCH! | February 2013

The Monthly Security Awareness Newsletter for Computer Users

OUCH!

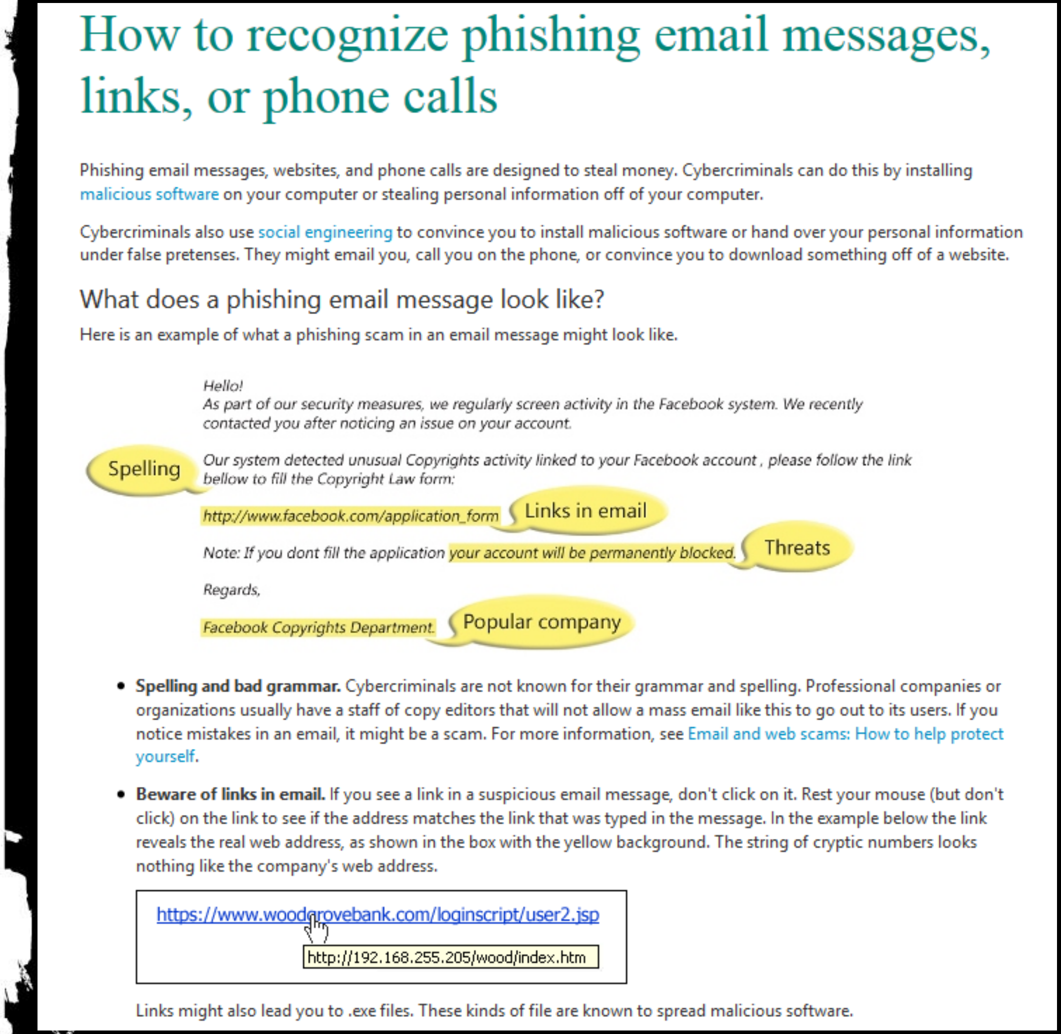
IN THIS ISSUE...

- Overview
- Phishing Attacks
- Protecting Yourself

http://www.securingthehuman.org/newsletters/ouch/issues/OUCH-201302_en.pdf

"OUCH! FEBRUARY 2013" ON PHISHING...STILL?
GENTLEMEN, YOU HAD MY CURIOSITY.
BUT NOW YOU HAVE MY ATTENTION.
memegenerator.net

Power to the User?



How to recognize phishing email messages, links, or phone calls

Phishing email messages, websites, and phone calls are designed to steal money. Cybercriminals can do this by installing malicious software on your computer or stealing personal information off of your computer.

Cybercriminals also use social engineering to convince you to install malicious software or hand over your personal information under false pretenses. They might email you, call you on the phone, or convince you to download something off of a website.

What does a phishing email message look like?

Here is an example of what a phishing scam in an email message might look like.

Hello!
As part of our security measures, we regularly screen activity in the Facebook system. We recently contacted you after noticing an issue on your account.

Our system detected unusual Copyrights activity linked to your Facebook account , please follow the link bellow to fill the Copyright Law form:

http://www.facebook.com/application_form

Note: If you dont fill the application your account will be permanently blocked.

Regards,

Facebook Copyrights Department.

- **Spelling and bad grammar.** Cybercriminals are not known for their grammar and spelling. Professional companies or organizations usually have a staff of copy editors that will not allow a mass email like this to go out to its users. If you notice mistakes in an email, it might be a scam. For more information, see Email and web scams: How to help protect yourself.
- **Beware of links in email.** If you see a link in a suspicious email message, don't click on it. Rest your mouse (but don't click) on the link to see if the address matches the link that was typed in the message. In the example below the link reveals the real web address, as shown in the box with the yellow background. The string of cryptic numbers looks nothing like the company's web address.

https://www.woodgrovebank.com/loginscript/user2.jsp
http://192.168.255.205/wood/index.htm

Links might also lead you to .exe files. These kinds of file are known to spread malicious software.

The SANS OUCH! report links to the Microsoft site to help users learn how to detect phishing emails, etc.

Can the majority of users handle this?

NOPE!

Especially with phishing campaigns that are crafted from legitimate emails.

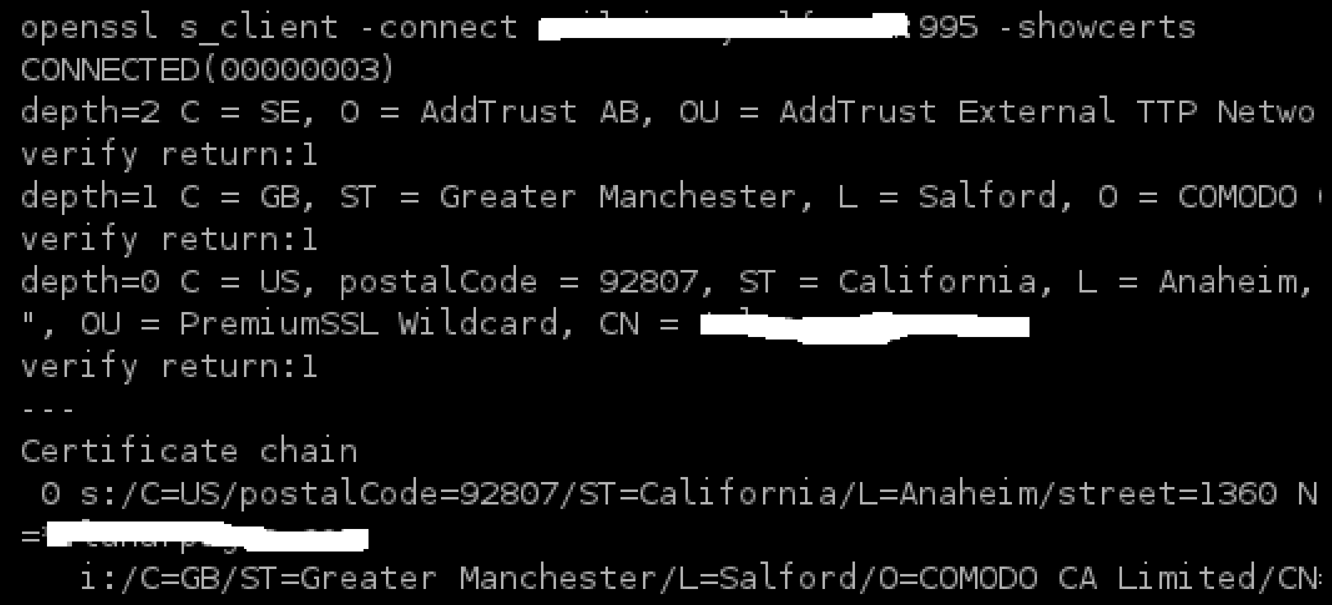
Forgot About Spam Folder

- **Wanted to download my spam folder for offline processing**
- **NO GUI please, command line**
- **Decided to try fetchmail**
- **Was about to use:** *fetchmail -u <username> -a -p POP3 --bsmtp /<path>/<text_filename> <mail_server>*
- **That method is insecure and will give you:** *Warning: the connection is insecure, continuing anyways. (IT WILL CONTINUE WITHOUT USER ACCEPTING THE RISK!)*
- **Forgetting my spam folder was like a First World Problem ☺**

LISTENS TO HIP-HOP
FORGOT ABOUT DRE
memegenerator.net

Grab CertifiCATS

- **Fetchmail does support ssl > YEAH.txt**
- **So I had to get the certificate from my mail server using:** *openssl s_client -connect <mail server>:995 –showcerts*



```
openssl s_client -connect [redacted]995 -showcerts
CONNECTED(00000003)
depth=2 C = SE, O = AddTrust AB, OU = AddTrust External TTP Netwo
verify return:1
depth=1 C = GB, ST = Greater Manchester, L = Salford, O = COMODO
verify return:1
depth=0 C = US, postalCode = 92807, ST = California, L = Anaheim,
", OU = PremiumSSL Wildcard, CN = [redacted]
verify return:1
---
Certificate chain
 0 s:/C=US/postalCode=92807/ST=California/L=Anaheim/street=1360 N
=[redacted]
   i:/C=GB/ST=Greater Manchester/L=Salford/O=COMODO CA Limited/CN
```

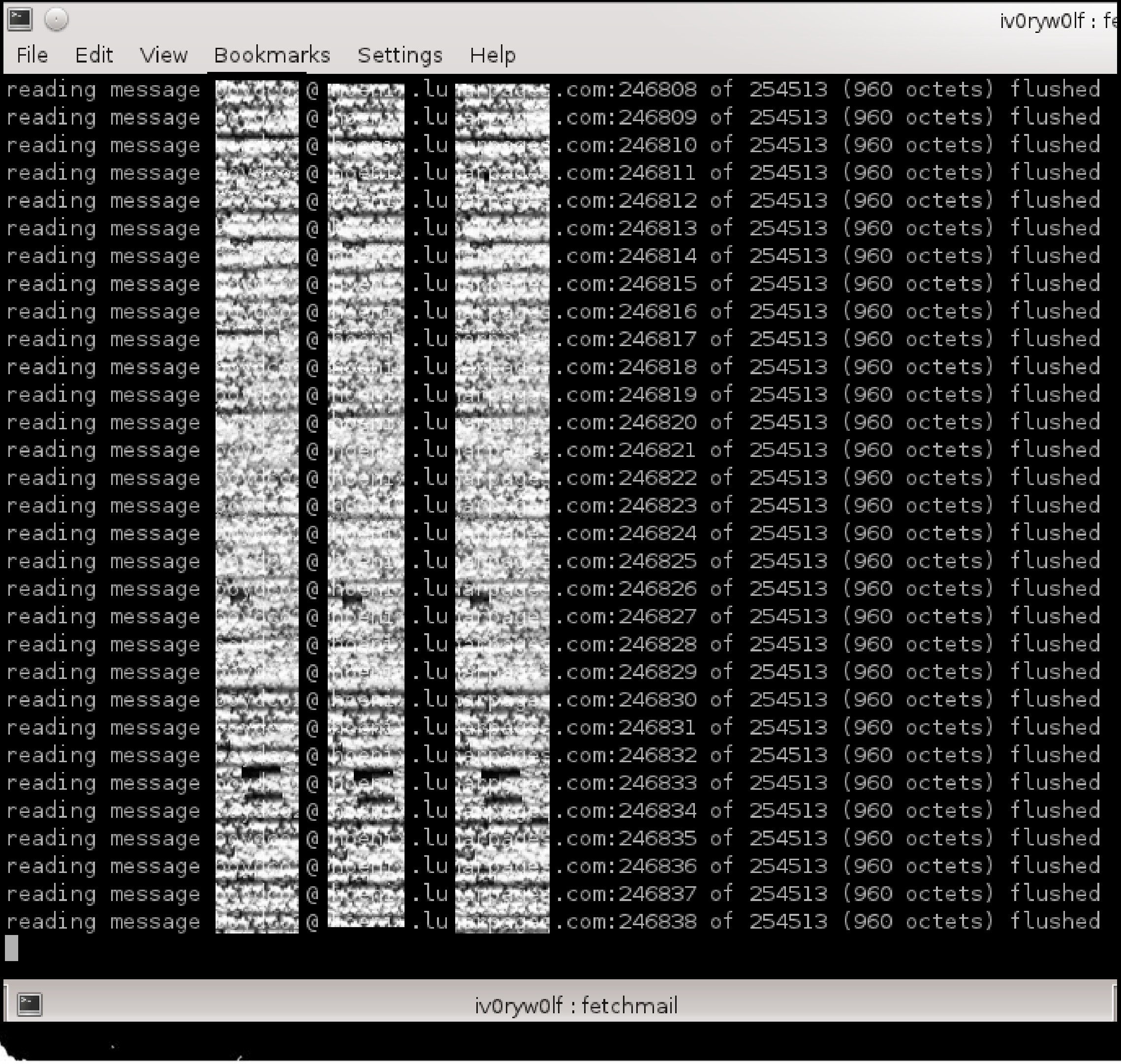
CertifiCAT!!!

-----BEGIN CERTIFICATE-----
MIIF0zCCBLugAwIBAgIRANkCbs6awHrRokj
gYkxCzAJBgNVBAYTAkdCMRswGQYDVQQIExJ
BgNVBAcTB1NhbGZvcmQxGjAYBgNVBAoTEUN
VQQDEyZDT01PRE8gSGlnaC1Bc3N1cmFuY2U
MjAzMDYwMDAwMDBaFw0xNDAzMTAyMzU5NTl
A1UEERMFOTI4MDcxEzARBgNVBAgTCkNhbGl
aW0xGzAZBgNVBAkTEjEzNjAgTiBIYW5jb2N
dCBJbmMuMRwwGgYDVQQLExNMdW5hcnBhZ2V
...
dHA6Ly9vY3NwLmNvbW9kb2NhLmNvbTArBgN
Y29tgg5sdW5hcnBhZ2VzLmNvbTANBgkqhki
UCEhYhYevf2L91ewPFODszlLlNxzvb4CMbY
MIupUZgpV33Gf/LUO8tGAdzQUBnah3/Pcfv
bgb6sDovpJuyU1LLekuS6mNrMr3TQdiUvBG
0Z9yc12CjohGsamz9I4YEKa9lielatVuQPj
15kZiCuj0Zrg+H/KCoR4S/o9Q6+gR5QIcSS
3g46rqrQng==
-----END CERTIFICATE-----

CertifiCAT Usage

- Copy the first base64 string, including -----BEGIN CERTIFICATE----- & -----END CERTIFICATE-----, to <filename>.pem
- Then you can run the fetchmail command with ssl
- fetchmail -u <username> -a -p POP3 --sslcertck --keep --sslcert=/<path>/<filename>.pem --bsmtp /<path>/<filename>.txt <mail server>
- The previous error message should be nonexistent

Fetchmail Output Example



```
reading message [redacted]@[redacted].lu[redacted].com:246808 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246809 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246810 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246811 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246812 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246813 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246814 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246815 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246816 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246817 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246818 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246819 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246820 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246821 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246822 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246823 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246824 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246825 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246826 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246827 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246828 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246829 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246830 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246831 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246832 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246833 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246834 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246835 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246836 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246837 of 254513 (960 octets) flushed
reading message [redacted]@[redacted].lu[redacted].com:246838 of 254513 (960 octets) flushed
```

Fetchmail Options

-p <proto> | --proto <proto> | --protocol <proto>

Specify the protocol to use when communicating with the remote mailserver. If no protocol is specified, the default is AUTO. proto may be one of the following:

AUTO Tries IMAP, POP3, and POP2 (skipping any of these for which support has not been compiled in).

POP2 Post Office Protocol 2 (legacy, to be removed from future release)

POP3 Post Office Protocol 3

APOP Use POP3 with old-fashioned MD5-challenge authentication. Considered not resistant to man-in-the-middle attacks.

RPOP Use POP3 with RPOP authentication.

KPOP Use POP3 with Kerberos V4 authentication on port 1109.

SDPS Use POP3 with Demon Internet’s SDPS extensions.

IMAP IMAP2bis, IMAP4, or IMAP4rev1 (fetchmail automatically detects their capabilities).

ETRN Use the ESMTP ETRN option.

ODMR Use the On-Demand Mail Relay ESMTP profile.

All these alternatives work in basically the same way (communicating with standard server daemons to fetch mail already delivered to a mailbox on the server) except ETRN and ODMR. The ETRN mode allows you to ask a compliant ESMTP server (such as BSD sendmail at release 8.8.0 or higher) to immediately open a sender-SMTP connection to your client machine and begin forwarding any items addressed to your client machine in the server’s queue of undelivered mail. The ODMR mode requires an ODMR-capable server and works similarly to ETRN, except that it does not require the client machine to have a static DNS.

Fetchmail Options

-a | --all | (since v6.3.3) --fetchall

Retrieve both old (seen) and new messages from the mailserver. The default is to fetch only messages the server has not marked seen. Under POP3, this option also forces the use of RETR rather than TOP. Note that POP2 retrieval behaves as though --all is always on (see RETRIEVAL FAILURE MODES below) and this option does not work with ETRN or ODMR. While the -a and --all command-line and fetchall rcfile options have been supported for a long time, the --fetchall command-line option was added in v6.3.3.

-k | --keep

Keep retrieved messages on the remote mailserver. Normally, messages are deleted from the folder on the mailserver after they have been retrieved. Specifying the keep option causes retrieved messages to remain in your folder on the mailserver. This option does not work with ETRN or ODMR. If used with POP3, it is recommended to also specify the --uidl option or uidl keyword.

-K | --nokeep

Delete retrieved messages from the remote mailserver. This option forces retrieved mail to be deleted. It may be useful if you have specified a default of keep in your .fetchmailrc. This option is forced on with ETRN and ODMR.

Fetchmail Options

--sslcert <name>

For certificate-based client authentication. Some SSL encrypted servers require client side keys and certificates for authentication. In most cases, this is optional. This specifies the location of the public key certificate to be presented to the server at the time the SSL session is established. It is not required (but may be provided) if the server does not require it. It may be the same file as the private key (combined key and certificate file) but this is not recommended. Also see --sslkey below.

--sslcertck

Causes fetchmail to strictly check the server certificate against a set of local trusted certificates (see the sslcertfile and sslcertpath options). If the server certificate cannot be obtained or is not signed by one of the trusted ones (directly or indirectly), the SSL connection will fail, regardless of the sslfingerprint option.

For more check the man pages > http://fetchmail.berlios.de/fetchmail-man.html

What’s Next?

- At this point I have about 2 GB of spam and legitimate emails to compare
- Parse through the weeds of emails for statistics
- Maybe create a phishing engine to input legit and spit out spam
- Questions?

Forgotten Fetchmail Option

-p MMMBOP

You’re Welcome!