

# CAN YOU TRUST YOUR EMAIL?

Kyle Cassidy (cassidy@saturn.rowan.edu)  
Office of Academic Computing, Rowan College of New Jersey

A. Michael Berman (berman@elvis.rowan.edu)  
Department of Computer Science, Rowan College of New Jersey

*To be presented at the Eastern Small College Computing Conference,  
New Rochelle, NY, October 1995.*

## ABSTRACT

Everyday, millions of electronic mail messages (email) pass through the Internet. Most academics depend upon email to do their jobs. However, email is not trustworthy. Specifically, it is almost never possible to verify, using the email alone, that a received message has come from the apparent sender. The problem can come from at least three sources: the design of the Internet mail protocol; specific attacks designed and distributed by hackers; and generally lax security standards, particularly at academic institutions. The issue of trust in communications is not entirely new -- after all, forgery has been recognized as a crime almost since the invention of writing. However, the lack of general understanding of this new medium, combined with the lack of non-digital information associated with paper mail (in particular, the signature) have created an environment in which forged messages are easy to send and hard to recognize. We discuss two specific actions academics ought to take that can help the situation: user education and improved system security. Finally we briefly describe a technical approach -- the digital signature -- that promises to greatly reduce the problem in the future.

## THE PROBLEM

Electronic Mail is used every day, across the Internet, by millions of users. Most users accept the validity and security of email. They seldom question whether a piece of email did indeed come from the person whose name appears at the top and with their consent. However, making use of one of several security flaws, it is possible for any user on the Internet to send mail to any other user on the Internet in the guise of anyone. It is just as simple for me to send you mail with the return address `thoreau@walden.pond.com` as it is for me to write it on the outside of an envelope.

## Case Study

At Rowan College we have been concerned about this gaping hole for some time, knowing that "Security Through Obscurity" would eventually lead to everyone being in on the secret. The walls were finally breached one afternoon in February of 1995 when a student forwarded to our system administrator an email that appeared to be a threat from another student. The sender of the mail was confronted and denied sending the message at all. Checking the user logs, the system administrator found out that the alleged "sender" was not logged on at the time the mail was sent. However, an examination of system accounting information revealed the identity of the actual sender.

## WHY ELECTRONIC MAIL IS NOT TRUSTWORTHY

### The Simple Mail Transfer Protocol (SMTP): Email's Greatest Flaw

SMTP, the Simple Mail Transfer Protocol, underlies the entirety of electronic mail [1]. SMTP transfers mail from one computer to another computer, regardless of the type of computer on either end. In order to do this simply and efficiently, all transactions use simple ASCII, and no password is required from a remote computer when allowing mail to come in. Using passwords would require that every computer on the Internet know the password of every other computer on the Internet, but know them in a way that users would not be able to find them out. The creators and implementors of SMTP opted instead for "Security Through Obscurity" -- don't tell anyone about it, and hope they don't figure it out. As a result, anyone with access to the program TELNET and the RFC for SMTP (freely available on the Internet) can "spooof" email. (RFC stands for "Request For Comments"; this is the name given to the documents that define Internet standards.) What we get in efficiency and simplicity, we lose in security.

SMTP consists of a small number of commands sent back and forth by two email gateways to establish where a message belongs and get it there as quickly as possible. Let us say that mail is being sent from `president@whitehouse.gov` to `cassidy@saturn.rowan.edu`. In order for the mail to get from the President's gateway to my mailbox a number of things occur:

- 1) The mail gateway at the White House contacts the mail gateway at Rowan College. This is almost always done by telnetting to port 25 at the remote machine, port 25 being the default for mail.
- 2) `whitehouse.gov` introduces itself and asks if "cassidy" is a valid user.
- 3) After being assured that "`cassidy@saturn.rowan.edu`" does indeed receive mail there, `whitehouse.gov` tells `saturn.rowan.edu` who the message is from. The text of the message then follows.

The entire transaction looks like this:

```
$ telnet saturn.rowan.edu 25
Trying 150.250.1.8 ...
Connected to saturn.rowan.edu.
Escape character is '^]'.
220 saturn.rowan.edu MX V4.1 VAX SMTP server ready at Sat, 17 Jun 1995
23:07:12T

MAIL FROM: <president@whitehouse.gov>
250 MAIL command accepted.
RCPT TO: <CASSIDY@SATURN.ROWAN.EDU>
250 Recipient okay (at least in form)
DATA
354 Start mail input; end with <crLf>.<crLf>
Hello Kyle, I was wondering if you could come over on Friday and give me
a hand with the big summit. You know how highly I value your opinion.

Thanks,
Bill
.

250 Message received and queued.
quit
```

The problem lies in that SMTP doesn't care *who* telnets to port 25 and talks to it. And to make matters worse, SMTP has no way of knowing, nor does it care, if the sender's address is valid. So the problem comes to fruition: anyone on the Internet can telnet to almost any mail

gateway anywhere and send whatever data they care to with any return address they care to. SMTP even has a simple "help" feature which will assist you in spoofing email if you forget the proper syntax. Therefore not only is it an absurdly simple task for me to send *you* email with the return address "president@whitehouse.gov", it would be just as trivial for me to send the President mail with *your* return address.

```
214-Commands:
214- HELO MAIL RCPT DATA RSET
214- NOOP QUIT HELP VRFY EXPN
214-.
214-For more info use "HELP <topic>".
214-
214 End of HELP info
```

SMTP has help available for all of its commands -- a great advantage for absent-minded spoofer.

## Attacks on Operating System Weaknesses

Some email spoofing preys on holes particular to various operating systems. Subsequent investigation of the mail spoofing incident on our campus lead system administrators to discover a program called `switch.com` in the perpetrators' directory. *Switch* is a DCL script which exploits a security hole in the way VMS handles electronic mail, allowing one user to send mail as another. This program appears to have been widely distributed via the Internet. The user needs no

knowledge about email or the Internet to use this program, which simply prompts for a return address to use on the message.

## Weak Standards for System and Network Security

Much electronic grief comes when users leave themselves logged in and go away from their terminals. Recently, the Secret Service investigated a message sent from our campus that threatened the President. Our conclusion was that the message was sent from a computer in a public laboratory when a user left his machine logged on after he went home. In this case, there was no way to determine who actually sent the message.

## Why is Email Different from Paper Mail?

Suppose you receive a (paper) letter from someone who purports to be President Clinton. There are certain things which might lead you to accept the validity of this mail. It might have the return address of the White House for example, and a Washington D.C. postmark. The signature might also match one associated with President Clinton. If someone uses a program such as `switch.com` to send you email as President Clinton there are ways to check things which are the equivalent of postmarks and signatures. If you care to investigate and know how to, there are ways to find out where the mail really came from, or at least get a good idea. Let us say however that President Clinton was sitting in the Oval Office, writing electronic mail, got up to answer the door, and someone sat down at his terminal and sent you mail from his account. For all the electronic investigation you can do, this email is from President Clinton -- literally, his electronic identity has been stolen.

# DEALING WITH THE PROBLEM

## User Education

As with most issues of security, a well informed user is a better protected user. Users should be aware that the return address doesn't necessarily tell who the mail came from. They need to understand that if they receive a strange, threatening, or offensive message, they should not automatically assume that it was sent by the person whose email address appears on the message. In addition, they should also know better than to leave their terminals logged in or to share passwords.

## Legal and Policy Approaches

While sending death threats to the President is illegal, the rapid expansion of computer technology has left something of a legal void, and many issues of computer security have yet to be addressed by lawmakers. It is therefore necessary for colleges to have well-designed campus policies concerning computer usage. Students, staff, and faculty should be made aware what practices

("spoofing" mail, stealing passwords, sharing accounts, etc.) are not permissible. Policies should be reasonable and not overly restrictive, while at the same time discouraging blatantly improper use of the network.

## Using Cryptography to Create Digital Signatures

There are software solutions to some of this. The technique known as Public-Key Cryptography can create what's called a "digital signature". The program PGP (Pretty Good Privacy) provides one method of Public-Key Cryptography [2, 3]. PGP is free and easily available via the Internet. PGP is a "Public Key Encryption" algorithm where every user has two "keys", one public and one private. The public key can be freely distributed and is used for encrypting a message *to you*. Your private, or "secret" key, is kept by you and you alone. Only your private key can decrypt a message sent to you, encrypted with your public key. The public key can not be used to decrypt files, nor can it be used to somehow regenerate the private key.

You can also use PGP to sign a document electronically. Using your secret key, you add a digital signature to a text file (it looks like numbers and letters, not a real signature) and mail it to someone. They can then use your public key to verify that the file came from you. Nothing aside from your secret key can create the signature. In one way, a digital signature is even better than one in ink -- a digital signature is based on a mathematical function of the text of the entire message, so that if someone changes even one letter of the message, the receiver will be able to detect the change.

### An Example:

Bob's Widget Factory in Modesto, California, sells widgets via the Internet. One day the factory receives an electronic Purchase Order for 1,000 widgets, at \$175 per widget, from Rowan College in New Jersey. Without a digital signature, or some other form of confirmation, the factory is unwilling to risk shipping 1,000 widgets. After all, suppose a student from the college really sent the order, or a hacker changed the order from 10 widgets to 1,000? However, if Bob's Widget Factory and Rowan College both use PGP, Rowan College can supply a digital signature along with the order. Unless someone has compromised the security of the purchasing office, the message cannot be bogus. Bob's Widget Factory can ship \$175,000 worth of widgets with confidence.

### Using PGP

We ran the MacPGP 2.3 on a PowerBook 145b and found it quick and easy to use. Both encrypting/decrypting and signing/verifying are simple tasks with this version of PGP. Implementations of PGP have been written for almost every popular computer platform (VAX/VMS, Unix, Atari, MS-DOS, etc.)

Note that using PGP on a time-shared computer system compromises security, as your secret key is not only in a place where a cracker might come across it, but it is probably on God knows how many backup tapes made by your system administrator every night. PGP depends upon your key, as well as a secret "passphrase". If someone has a copy of your secret key and

then while walking past your desk one afternoon, finds the yellow Post-it note stuck to your monitor that says "pgp passphrase: *linda is a hot tamale*" you are out of luck -- that person can now forge messages in your name, as well as read encrypted messages intended for you. PGP has provisions for invalidating a compromised key, but it involves telling everybody in the world that your key has been stolen and having everyone believe you -- after all, a smart cracker would think to call all your friends and say "Hi, this is, uh, like Kyle, heh heh, right, and I, uh, forgot my PGP passphrase heh heh heh, so my new public key is...."

## CONCLUSION

As the Internet community grows, security becomes more of an issue. It is irrational to expect that system administrators only need ask their users to behave responsibly and it will happen. As has always been the case with the Internet, with the good comes the bad -- the rapid growth means that there are many more malicious users than a few years ago, and even more who are naive. Steps must be taken now to assure that two years from now we are not suddenly faced with the grim reality that the infrastructure will not support us.

### References

[1] Postel, Jonathan B. "RFC 821: Simple Mail Transfer Protocol," on-line document, August 1982.

[2] Stallings, William. *Protect Your Privacy*. Prentice-Hall, Clifton N.J., 1994.

[3] Zimmerman, Phillip, "Pretty Good Privacy User's Guide" (two volumes), on-line document, March 1993