1-1-2007

Collaborative intelligent email ranking system

Nathaniel H Whittacre
University of Nevada, Las Vegas

COLLABORATIVE INTELLIGENT

EMAIL RANKING SYSTEM

by

Nathaniel H. Whittacre

Bachelor of Science
University of Nevada, Las Vegas
2002

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
December 2007

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval

The Graduate College
University of Nevada, Las Vegas

NOVEMBER 16TH, 2007

The Thesis prepared by

NATHANIEL H. WHITTACRE

Entitled

COLLABORATIVE INTELLIGENT EMAIL RANKING SYSTEM

is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative

1017-53

ABSTRACT

Collaborative Intelligent Email Ranking System

by

Nathaniel H. Whittacre

Dr. Yoohwan Kim, Examination Committee Chair
Assistant Professor of Computer Science
University of Nevada, Las Vegas

Email has become one of the most powerful communication tools today. It is has widely proliferated in both business and personal use. It allows for fast communication between multiple parties that can be easily understood by even the most novice user, and allows for advanced transfer of data for power users. Even with that, it is one of the most abused systems on the Internet. Email systems have allowed for widespread distribution of the worst viruses on the Internet, causing billions of dollars in damage. Most of the technologies that have been deployed to prevent these types of attacks have been thwarted.

A new email protocol is required to implement accreditation, authentication and reputation to overcome these issues. This new system is a combination of currently accepted systems, along with additions to make them more effective as a whole. This new system is called Collaborative Intelligent Email Ranking System (CIERS).

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ACKNOWLEDGEMENTS

I am eternally grateful for my wife, Ruth, who has supported me through this entire process. She is my inspiration for succeeding in all that I do. I would also like the thank my children, Natali, Noah and Kathryn, for understanding the time that I have dedicated to accomplishing this. I also acknowledge the guidance that my parents gave me to accomplish great things. Finally, I would like to thank all the professors that have given their great knowledge to me through my many years at the University of Nevada, Las Vegas.

CHAPTER 1

INTRODUCTION

Internet mail has become one of the most useful communication tools in recent time. It is used heavily in both in business transactions and in personal communications. Moreover, it has become an easy way of sending and receiving data over the Internet in a very simple system that most people can easily use. Unfortunately, many people have found ways to abuse electronic mail; thus giving intruders ways to send unsolicited mail, including advertisements, pornography, and harmful viruses to millions of people every day. For years, smart people have tried to stop these malicious attacks into users personal mailboxes, but each new protection method has led to even more advanced techniques to get past them. Currently, there are a few methods that protect the user most of the time, while still allowing good functionality. This paper proposes a new idea to both secure electronic mail transmissions and block unsolicited mail.

There are three main protocols used for electronic mail (e-mail) transmissions: simple mail transport protocol (SMTP), post office protocol version 3 (POP3), and Internet message access protocol version 4 (IMAP4). The combination of these protocols allows e-mail to flow easily from client to server to server and then be retrieved by another user. SMTP is the protocol primarily used on the Internet for sending e-mail from either client to server or server to server. POP3 is the

prominent protocol for simple retrieval of e-mail from a message storage system. IMAP4 is a complicated system of server side e-mail storage and maintenance, along with client retrieval.

These protocols are very antiquated, especially SMTP, which was originally defined in 1982. One of the strengths of SMTP is also its weakness; any machine connected on the Internet can easily act as a mail server. Because of this, anyone, from individuals to small and large businesses can easily install a server on the Internet and start sending and receiving e-mail. Even more interesting, the SMTP protocol is used to transfer mail from the email client the the SMTP server acting as the mail transfer agent (MTA). This means that every desktop is essentially acting as a mail server, with the ability to send mail to any system connected to the Internet. Hackers have exploited this to infect workstations and then send mail directly to mail exchange (MX) servers rather than going through a MTA for that network.

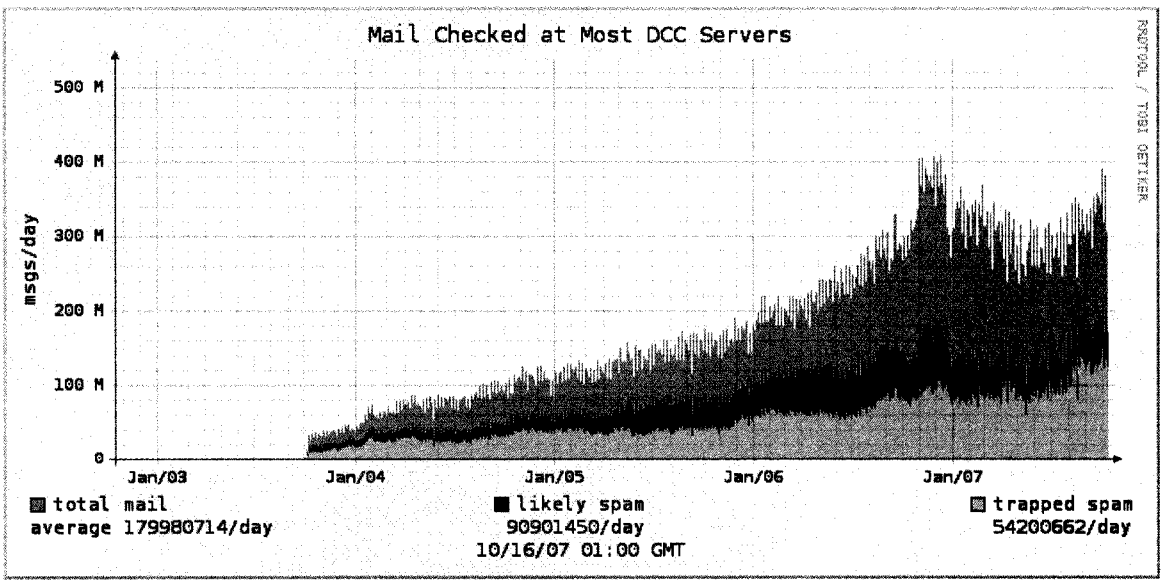

Illustration 1: SPAM messages checked by DCC Servers [DCC07]

Because of this, unsolicited mail (SPAM) has proliferated on the Internet. According to some sites, SPAM messages may be as much as 90% of all e-mail traffic. It is difficult to judge how much actual SPAM is going through the Internet because the amount of SPAM to each host can vary greatly; however, it is enough to say that a large percentage of messages on the Internet are SPAM. Illustration 1 shows a information from a group a servers that collaborate together to identify and fight SPAM, called Distributed Checksum Clearinghouse (DCC). This organization tracks and identifies SPAM and distributes this information among its members. According to the messages tracked in this graph, just over 50% of all email is likely SPAM. The bandwidth and processing power required to deal with this is tremendous.

1.1 Collaborative Intelligent Email Ranking System

In order to prevent the ability to manipulate the SMTP system, this thesis presents a new system to provide authentication, accreditation and reputation among mail servers. The Collaborative Intelligent Email Ranking System (CIERS) implements existing protocols, including SMTP and secure socket layer (SSL) transport to provide normal delivery and storage of email, but adds on a central authentication and reputation management authority to log and coordinate e-mail transfer among members of CIERS. CIERS operates on the simple principle that a human interpreter will be able to do a much better job at identifying unwanted e-mail than any computer process, and that a large effort among users will quickly identify mail servers that are sending out large amounts of SPAM. Moreover, additional mail servers cannot be added to the CIERS

network without first properly registering to the network and identifying the administrators of the servers. This registration process includes an authentication processes that makes it difficult for random servers to be added quickly and taken down quickly.

Although the process of authentication, accreditation and reputation will be described in more detail later in this document, Illustration 2 gives an overview of the process. The basic design centers around a CIERS authority server authenticating and logging email transmitted between two email servers. All the authentication information between the servers and the CIERS authority is done in XML, thus allowing the system to be easily expanded upon as need arises to respond to future threats to the email system. The mail servers communicate using SSL encryption, relying on the public key infrastructure that is widely used to authenticate e-mail servers. The CIERS authority acts as the certificate authority for the public keys; therefore, the mail servers can be accredited by a third party. Finally, after the mail is received by the end user, that user has the option to report the e-mail as an unsolicited e-mail. This communication is also done in XML. The CIERS authority tracks this communication and computes its Trust Value, which it uses to rank e-mail domains on their SPAM sending levels. All of these techniques combined result in a secure e-mail system that prevents large amounts of unsolicited mail to travel through the mail system. It also completely prevents unauthorized and unauthenticated hosts from being able to send email to other hosts.

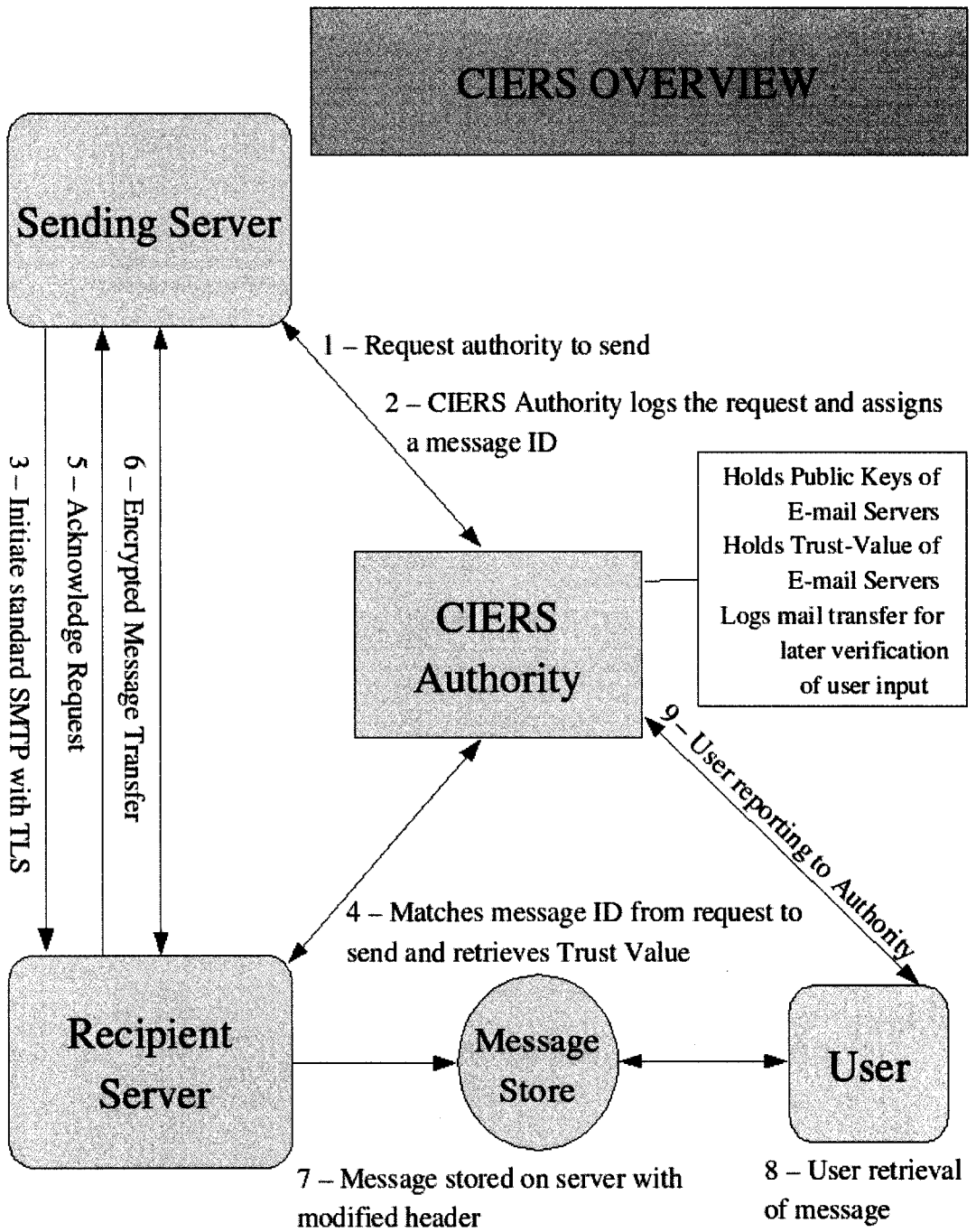

Illustration 2: CIERS Overview

1.2 Thesis Overview

Chapter 2 of this thesis begins with an overview of current mail protocols and their uses. The protocols discussed are Simple Mail Transfer Protocol (SMTP), Post Office Protocol Version 3 (POP3), and Internet Message Access Protocol (IMAP4). It gives a detail discussion of how hosts communicate with example exchanges between servers and clients. As these protocols are discussed, weaknesses in them are identified. Examples of attacks on these protocols, especially SMTP are identified and examples of attacks are given.

Chapter 3 discusses the current state of SPAM protection. Several protection algorithms are discussed, including naïve-Bayesian filtering and some collaborative e-mail reporting lists. Included in this discussion is an explanation of where these techniques are failing and how spammers have circumvented them.

A detailed discussion of TLS encryption protocols is given in chapter 4. This chapter will give reasoning behind these encryption protocols, along with an algorithm analysis of the protocols. It also addresses attacks on the data transfer using this encryption technology. It will give some examples of encryption handshaking, including an actual TLS session.

Chapter 5 details the CIERS protocol. It gives the syntax for the authentication, accreditation and reputation transmission between mail servers and the CIERS authority. Additionally, it discusses the communication between the e-mail client and the CIERS authority; moreover, the method for computing the trust value. Finally, this chapter details example transmissions between hosts using the CIERS protocol.

Chapter 6 concludes the thesis with discussion on way to deploy CIERS across the Internet, including additional functionality that may be required to facilitate performance benchmarks required for global deployment. A discussion of possible weaknesses in the protocol will be discussed, along with way to avoid those weaknesses.

Finally, Appendix 1 includes example code written to demonstrate the CIERS system. This code is written in PHP and C++ and is used to demonstrate the effectiveness of the CIERS system.

CHAPTER 2

MAIL TRANSFER PROTOCOLS

Mail protocols have been around since nearly the beginning of the Internet. Simple Mail Transport Protocol (SMTP) dates back to 1982. It is still widely used today, and has been modified to increase security and capabilities. SMTP is used to send e-mail from one host to another. It is implemented in both mail clients and mail servers. Moreover, it is the standard for transmitting e-mail on the Internet.

SMTP does not offer a way to retrieve mail from a server store, so several protocols have been created for that purpose. Post Office Protocol Version 3 (POP3) is the most basic protocol for e-mail retrieval from a mail server store. It is widely used, mostly for client computers to retrieve e-mail from their Internet Service Provider's (ISP) mail servers. Internet Mail Access Protocol Version 4 (IMAP4) is also widely used and offers many more capabilities than POP3, including server stored folders. These two protocols offer the bulk of the open protocols for mail retrieval on the Internet. Other proprietary protocols, like Microsoft's Messaging Application Programming Interface protocol (MAPI protocol), are widely used in corporate environments and facilitate more extensive interfaces than basic mail retrieval. Because this protocol is proprietary, it is beyond the scope of this paper.

2.1 SMTP Protocol

The SMTP protocol is a very robust system of sending e-mail between client and server and server and server. It is also a very simple system, which has allowed it to expand from a simple text based system, to a system capable of handling large amounts of binary data. SMTP was originally defined in RFC 821 in 1982. To allow for additional sending methods, it was update by RFC 2821 in 2001. One of the beauties of SMTP is that it is completely transport independent. Although it is mostly used across TCP/IP networks, it can also be used in other transport environments.[KLENSIN01] The basic model for SMTP follows in Illustration 3:

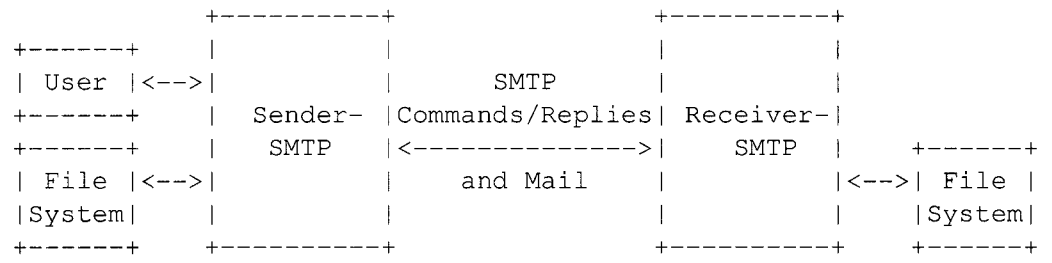

Illustration 3: SMTP Transport from RFC 821

In each SMTP transaction, there is a sender SMTP, which can also be the originating client and a receiver SMTP server. The sender SMTP machine initiates communication with the receiver SMTP server. They perform some protocol negotiation and the message is transmitted. The receiver SMTP server does not necessarily need to be the final destination of the e-mail. This a

great benefit to the SMTP protocol, because it allows for relaying across the Internet. The sender SMTP server sends the message to the receiver SMTP server, and the receiver SMTP server can either store it for local retrieval or pass it on in a chain to the destination server. SMTP relaying is a great benefit to the protocol, but also a huge problem. Because of the ability to relay, and the fact that most SMTP servers will allow connection from most other SMTP servers automatically, any person can setup a SMTP server on the Internet and begin to send messages. For this specific reason, this document will propose a central authentication system to register and track SMTP servers on the Internet.

The following is an example SMTP session:

S: 220 mail.whittrio.com; ESMTP Tue, 25 Jan 2005 22:36:36 -0800

C: EHLO whittrio.com

S: 250-mail.whittrio.com Hello traffic.whittrio.com [67.104.119.206], pleased to meet you

S: 250-ENHANCEDSTATUSCODES

S: 250-PIPELINING

S: 250-8BITMIME

S: 250-SIZE

S: 250-DSN

S: 250-AUTH DIGEST-MD5 CRAM-MD5

S: 250-DELIVERBY

S: 250 HELP

C: MAIL FROM:<nathan@whittrio.com>

S: 250 2.1.0 <nathan@whittrio.com>... Sender ok

C: RCPT TO:<nathan@stimulustech.com>

S: 250 2.1.5 <nathan@stimulustech.com>... Recipient ok

C: DATA

S: 354 Enter mail, end with "." on a line by itself

C: This is a test

C: Have a nice day

C:

C: .

S: 250 2.0.0 j0Q6aakE018722 Message accepted for delivery

The previous transaction would produce the following message:

Return-Path: <nathan@whittrio.com>

Received: from whittrio.com (traffic.whittrio.com [67.104.119.206])

by mail.whittrio.com (8.12.11/linuxconf) with ESMTP id j0Q6aakE018722

for <nathan@stimulustech.com>; Tue, 25 Jan 2005 22:37:00 -0800

Date: Tue, 25 Jan 2005 22:36:36 -0800

From: nathan@whittrio.com

Message-Id: <200501260637.j0Q6aakE018722@mail.whittrio.com>

This is a test

Have a nice day

There are a few items of interest in this simple transaction. First, the receiver SMTP mail server initially communicates the capabilities of the server. Some of the additional capabilities allows for encrypted transmissions and authentications, which have proven important in current implementations of SMTP. One major insecurity of the SMTP protocol is the spoofing of e-mail addresses. The receiver SMTP server generally accepts the senders e-mail address without any way of authenticating that e-mail address. It is not built into the protocol to request authenticity of that e-mail address unless that sender resides locally as a user of the receiver SMTP server, which is unlikely.

Take the following real e-mail as an example:

Return-Path: <ywozupvyfs@FreeMail.nl>

Received: from c-24-14-164-93.client.comcast.net (c-24-14-164-93.client.comcast.net [24.14.164.93])

by mail.whittrio.com (8.12.11/linuxconf) with SMTP id j0R4eqv6008931;

Wed, 26 Jan 2005 20:40:52 -0800

Received: from 124.144.110.228 by web783.mail.yahoo.com; Wed, 26 Jan 2005 20:37:12 -0800

Message-ID: <meridionalSFBRKRKAPBEFDIGPXKMVHWJ@coolmail.to>

From: "Lacy Stahl" <gdhdpbwawh@coolmail.to>

To: nwhit@whittrio.com, whit@whittrio.com

Subject: Hey,

Date: Wed, 26 Jan 2005 20:37:12 -0800

MIME-Version: 1.0

Content-Type: multipart/alternative;

boundary="--869530864330165980 29"

----8695308643301659802 9

Content-Type: text/plain;

Content-Transfer-Encoding: 7Bit

No time, no problem. Find a woman for a day. Real people, real life, in your area.

Married woman looking for part time friend. We have close to 1 million online!

< Http://trw.cvbg.info >

First and foremost, the receiver SMTP server, mail.whittrio.com, has no way to authenticate if c-24-14-164-93.client.comcast.net is a valid mail server. There are several checks, to be discussed later, to see if it is on a list of bad servers, but for the most part, the receiver SMTP server will accept the mail as long as the recipient is a local user on the server. Moreover, the sending e-mail user may not be a valid e-mail address, even though the domain could be valid. There are no

safeguards built into the protocol to allow for user authentication. Even notice that the e-mail address specified in the return path is quite different than the sender's e-mail address. It is even harder for the user to distinguish if this is a valid e-mail before opening it, because the sender's name could be spoofed into being a name that the person may be familiar with, or generic enough that the user will always open it.

2.2 POP3 Protocol

The SMTP protocol does not allow for a way for the user to place a request to the message store to retrieve the e-mail. The POP3 protocol is one system to allow for this. Post Office Protocol Version 3 is defined by RFC 1939 in 1996. POP3 is a very simple protocol, allowing a users to log into a mail server, request mail in the main message store, retrieve the mail and delete it from the server.[MYERS96] The following is an example of a POP3 transaction:

S: +OK POP3 mail.whittrio.com v2003.83rh server ready

C: USER nathan

S: +OK User name accepted, password please

C: PASS mypassword

S: +OK Mailbox open, 429 messages

C: STAT

S: +OK 429 4928409

C: RETR 1

S: +OK 6584 octets

S: Return-Path: <Tom.Wilson@Borland.com>

S: Received: from sbmail02.starbase.com (mail.premia.com [209.162.219.9])

S: by mail.whittrio.com (8.11.6/linuxconf) with ESMTP id h2HMeE504481

S: for <nathan@whittrio.com>; Mon, 17 Mar 2003 14:40:14 -0800

S: Received: by sbmail02.starbase.com with Internet Mail Service (5.5.2653.19)

S: id <GPRR8X98>; Mon, 17 Mar 2003 13:50:32 -0800

S: Message-ID: <1D0763828FC2D511A6670090279C1C8C774305@sbmail02.starbase.com>

S: From: Tom Wilson <Tom.Wilson@Borland.com>

S: To: "'nathan@whittrio.com'" <nathan@whittrio.com>

S: Subject: CodeWright support for Borland C++ Builder

S: Date: Mon, 17 Mar 2003 13:50:31 -0800

S: ...Message Text...

C: DELE 1

S: +OK Message deleted

C: QUIT

S: +OK Sayonara

POP3 also allows for encrypted logins, but not for encrypted transmissions of the message. POP3 is generally the most widely used protocol among Internet Service Providers (ISPs) for mail retrieval from their servers. There are several disadvantages of POP3. First, by default the

password and retrieval information is sent in plain text. This allows for easy interception of the data by a man-in-the-middle. Second, there is no server side storage of the mail. Only incoming mail is stored on the server until retrieval by the client. Finally, there is no feedback mechanism for reporting the quality of the message. In other words, whether the message is considered SPAM or not.

2.3 IMAP4 Protocol

IMAP4 overcomes several of the problems of POP3, mostly server side storage and login encryption. IMAP4 was defined by RFC 1730 in 1994 and was updated in 1996 by RFC 2060. IMAP4 defines methods for server side message store, mail and data management systems, and much better mail retrieval techniques than POP3. Along with all these techniques comes a complex instruction set, much more complex than POP3 [CRISPIN96]. The following example is similar to the basic POP3 retrieval instruction shown above:

S: * OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS STARTTLS AUTH=LOGIN] mail.whittrio.com IMAP4rev1 2003.338rh at Wed, 26 Jan 2005 22:28:57 -0800 (PST)

C: a001 LOGIN nathan mypassword

S: a001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS BINARY UNSELECTS

S: CAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND] User nathan authenticated

C: a412 select inbox

S: * 428 EXISTS

S: * NO Mailbox vulnerable - directory /var/spool/mail must have 1777 protection

S: * 0 RECENT

S: * OK [UIDVALIDITY 1047941245] UID validity status

S: * OK [UIDNEXT 1194] Predicted next UID

S: * FLAGS ($MDNSent Junk NonJunk $Forwarded \Answered \Flagged \Deleted \Draft \Seen)

S: * OK [PERMANENTFLAGS ($MDNSent Junk NonJunk $Forwarded * \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags

S: * OK [UNSEEN 33] first unseen message in /var/spool/mail/nathan

S: a412 OK [READ-WRITE] SELECT completed

C: a003 fetch 12 full

S: * 12 FETCH (FLAGS (\Seen NonJunk) INTERNALDATE "16-Oct-2003 08:14:59 -0800"RFC8

S: 22.SIZE 1864 ENVELOPE ("Thu, 16 Oct 2003 08:00:48 -0700" "CodeWright Seminar Reminder" (("Borland Beaverton" NIL "Borland.Beaverton" "Codewright.com")) (("Borland Beaverton" NIL "Borland.Beaverton" "Codewright.com")) (("Borland Beaverton" NIL "Borland.Beaverton" "Codewright.com")) (("'nathan@whittrio.com'" NIL "nathan" "whittrio.com")) NIL NIL NIL

"<1D0763828FC2D511A6670090279C1C8C01F0E15A@SBMAIL02>") BODY ("TEXT"

"PLAIN" ("CHARSET" "iso-8859-1") NIL NIL "8BIT" 1002 39))

S: a003 OK FETCH completed

C: a004 fetch 12 body[header]

S: * 12 FETCH (BODY[HEADER] {862}

S: Return-Path: <Borland.Beaverton@Codewright.com>

S: Received: from sbmail02.starbase.com (mail.premia.com [209.162.219.9])

S: by mail.whittrio.com (8.12.9/linuxconf) with ESMTP id h9GFEwLc023266

S: for <nathan@whittrio.com>; Thu, 16 Oct 2003 08:14:58 -0700

S: Received: by SBMAIL02 with Internet Mail Service (5.5.2653.19)

S: id <48MQRCHC>; Thu, 16 Oct 2003 08:00:49 -0700

S: Message-ID: <1D0763828FC2D511A6670090279C1C8C01F0E15A@SBMAIL02>

S: From: Borland Beaverton <Borland.Beaverton@Codewright.com>

S: To: "'nathan@whittrio.com'" <nathan@whittrio.com>

S: Subject: CodeWright Seminar Reminder

S: Date: Thu, 16 Oct 2003 08:00:48 -0700

S: MIME-Version: 1.0

S: X-Mailer: Internet Mail Service (5.5.2653.19)

S: Content-Type: text/plain;

S: charset="iso-8859-1"

S: Content-Transfer-Encoding: 8bit

S: X-MIME-Autoconverted: from quoted-printable to 8bit by mail.whittrio.com id h9GFEwLc023266)

S: a004 OK FETCH completed

C: a006 logout

S: * BYE mail.whittrio.com IMAP4rev1 server terminating connection

S: a006 OK LOGOUT completed

As one can easily see, the interaction is much more detailed between the client and the server. IMAP4 offers greater flexibility of use, but is much more difficult to implement. It also lacks feedback tools that would allow a user to give feedback on the quality of the message.

CHAPTER 3

CURRENT STATE OF SPAM PROTECTION

All the widely used protocols mentioned above were developed before the Internet became a large commercial environment. Especially with SMTP, there is trust and automatic acceptance built directly into the protocol. This does have its advantages, allowing the protocol to be used for many years without many major changes. Along with these advantages comes many disadvantages, including:

- SMTP, by itself, does not define standard ways to transmit the mail encrypted – this encryption must be done by the user agent before the e-mail is sent, or using additional protocols during the mail transfer
- There is no central authentication of hosts – anyone can setup a mail server on the Internet without any registration process, like the ones required for domains and SSL certificates
- All mail servers on the Internet are automatically accepted by all the other mail servers on the Internet
- A person can spoof the mail path
- A person can spoof the sender and receivers address in the header of the mail

Because of these general issues with the standard mail protocols, people can take advantage of

the system and send unsolicited mail, usually untraceable. This allows for the world of SPAM.

SPAM is defined by mail-abuse.com as the following:

An electronic message is "spam" IF: (1) the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients; AND (2) the recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent; AND (3) the transmission and reception of the message appears to the recipient to give a disproportionate benefit to the sender.

Under this definition, mail can be considered SPAM even if it does not have a commercial or offensive intent, it just has to be unsolicited. Spam has been on the increase dramatically since 2002. In 2003, SPAM started to account for over 50% of all e-mail and continues to increase as a percentage of total e-mail. Because of the growth of SPAM, there have been many proposals to eliminate it. Some of these proposals are very simple and some are complex algorithms for SPAM detection.

3.1 SPAM Blacklists

One of the original ideas for SPAM blocking was to create a list of servers or e-mail users that sent SPAM, and the user would reject all mail from that source. This method is very simplistic and only works on a limited basis for know sites that generate a lot of SPAM. Usually a server administrator subscribes to a globally maintained blacklist that many sites contribute to. Unfortunately, this can be easily circumvented by spammers by simply changing their IP address

or server name. This does protect well, though against open relays or mail server that have been infected by a virus or trojan. A real mail-server is blacklisted until the server administrator corrects the problem with the server and can prove the problem is fixed. This method is used widely on the Internet, but does not solve any of the underlying problems with Internet mail. There are many companies willing to offer maintained blacklists for free or for a price, including:

- spamcop.net [SC05]
- mail-abuse.com
- ordb.org
- razor.sourceforge.net
- rhyolite.com/anti-SPAM/dcc/
- pyzor.sourceforge.net
- And many others.

A more restrictive alternative method to blacklists is whitelisting. This method only allows mail from a set of servers allowed to send to a certain server or user. The advantage to whitelists is that one will only get the mail that one wants. It is a 100% solution. Even with this advantage, whitelisting does not allow for the complete philosophy of the Internet, namely being able to communicate with many different people easily. On the other hand, whitelists can be used in combination with other systems to produce a more accurate SPAM detection system.

3.2 Filtering

The two methods mentioned do not accurately nor absolutely detect and remove SPAM. With this in mind, several advanced algorithms have been developed for SPAM detection and removal. These algorithms are generally similar and use weighted averages of different systems, including blacklists and whitelists. Moreover, they implement advanced filtering techniques to detect unsolicited mail from standard mail. One of the more common filtering methods is call naïve-Bayesian filtering. Bayesian is an algorithm for classifying probabilities of occurrence of certain text inside a whole document. Based on these probabilities, the filter can learn good mail from bad mail. There are other more advanced filtering methods built into the more modern scanners that when used in conjunction with other methods, prove to be highly effective, with little false-positives. Some of the current products available include:

- SpamAssassin [SA05]
- Symantec Anti-virus Gateway
- Trend Micro ScanMail
- Mcafee SpamKiller
- Systems built into mail clients like Outlook 2003 and Thunderbird

3.3 Authentication

Even with these advanced algorithms, they are only 90% accurate. Moreover, the SPAM still takes up bandwidth and processing power to be detected and dealt with appropriately. Many

people want to eliminate SPAM at the root of the problem: take away the spammers ability to send e-mail to servers that will automatically accept it and disallow them from spoofing e-mail addresses. The IETF has had several working groups participating in this problem recently. One such working group, MARID, was set to the task of authenticating e-mail addresses using the domain name service (DNS). The group accepted a combined effort by Meng Weng Wong of pobox.com and Microsoft, called Sender ID.[LYON04] The basic concept of Sender ID is for the domain administrator to publish a list of allowed e-mail users from a domain in the DNS records of that domain. When email is received by a server, the sender's address is checked by the recipient's server against the DNS records of the sender's domain. If that sender is a valid sender and all the records match, the mail is allowed to go through. [HARDIE05]

In principle, the Sender ID system is a good way of forcing authentication of messages. No longer could spammers spoof the sender's e-mail address in mail. Opponents of the Sender ID claim that spammers can easily, and have already, registered domains, setup necessary DNS records and started spamming using the Sender ID technology. Because other mail servers that implement Sender ID automatically allow authenticated mail to enter, this allows spammers to circumvent other SPAM detection systems. Another large problem with Sender ID is that it was developed by Microsoft, who holds certain software patents on the underlying technology driving Sender ID. For this reason, IEFT has temporarily rejected Sender ID as a viable solution for SPAM detection because of the potential conflicts with licensing that Microsoft might incur. Moreover, the open source community, whose software drives the majority of the Internet servers,

refuses to implement Sender ID because of the potential licensing problems that Microsoft will not resolve.[ROBERTS 04]

Another method for SPAM detection is proposed by Yahoo!, who has also submitted a Internet-draft to the IETF. The principle is very similarly to Sender ID, except that each e-mail is digitally signed using a private key by the sending mail server. The signature is attached to the e-mail header and sent using standard SMTP protocols. When the recipient server receives the message, it obtains the public key from the sender's DNS tables and uses it to verify the signature in the message header. If the signature does not authenticate, the message can be discarded or flagged. If the signature is verified, the message is then allowed to pass through. The same issue exists with Yahoo! DomainKeys as with Sender ID: the spammer can setup a temporary viable domain, setup the keys, SPAM and the receiving server with authenticate the messages.[YAHOO05]

During a summit from November 9-10, 2004 held by the Federal Trade Commission, several proposals were submitted and discussed to solve the SPAM problem.[FTC05] Both Sender ID and Yahoo! DomainKeys were discussed as ways to validate the users authenticity. The part that is missing in those proposals is certifying the users reputation. A few proposals were discussed including one from Truste called Bonded Sender.[TRUSTE05] The concept of Bonded Sender is a follows. An organization subscribes to this server for a fee, depending on the number of e-mails sent through the system. The organization must go through an extensive certification program and follow guidelines on security, server installation and setup, and transparency. Truste continues to

monitor the domain for compliance with its specifications and debits the server's account if there are problems with the domain. Users are able to subscribe to the list for free and also post comment back on the sending servers.[TRUSTE04] Their design is to be used on top of other authentication schemes such as the ones mentioned above and other advanced algorithms.

CHAPTER 4

ENCRYPTION

One of the core aspects to CIERS is the requirement to encrypt the all communication between the mail exchange servers. The encryption protocol that is implemented in CIERS is Transport Layer Security (TLS), formerly known as Secure Socket Layer (SSL), a standard that is used on the Internet to provide a high level of security and privacy between hosts. This protocol is widely used, especially to implement the HTTPS protocol that is used for banking, security transactions, and many other transactions that require 100% security.

4.1 TLS Overview

TLS is implemented at the application layer of the network stack, but is used in conjunction with other application layer protocols. In other words, after a TLS connection is negotiated, another application layer protocol rides over the encrypted tunnel created between the two hosts. TLS requires a stateful TCP/IP connection between two hosts.[PETERSON00] One of the interesting aspects to TLS is that it does not specify exactly which cipher must be used in the encryption of the data; alternately, it allows the hosts to negotiate that cipher in the handshaking phase of the connection. Some of the cipher algorithms that can be used are:

- For key exchange: RSA, Diffie-Hellman, DSA, SRP, PSK
- For symmetric ciphers: RC4, 3DES, AES
- For cryptographic hash functions: HMAC-MD5 or HMAC-SHA

There are three basic phases to the TLS protocol:

1. Peer negotiation and handshaking to decide the encryption algorithm
2. Key exchange and key authentication
3. Symmetric cipher encryption and message authentication

As shown in Illustration 4, TLS uses a public/private key combination to authenticate each host. The illustration demonstrates an optional security level of security, which is mutual authentication, where both hosts must present a valid public key, thus verifying that both hosts can be trusted. The CIERS protocol recommends mutual authentication, but it is not required, because additional authentication steps are taken in the CIERS negotiation to authenticate that the sender (client as described in TLS) is valid.

After the cipher is negotiated, the hosts must validate the public certificate with a certificate authority (CA). TLS generally implements X.509 certificates. The detail of these certificates are discussed in more detail in section 4.3. X.509 certificates rely on a certificate hierarchy for authentication of a valid certificate. The root certificate of the hierarchy is the public certificate of a CA. Most systems have a list of valid trusted CA root certificates in the machines database, so the machine can safely validate the presented certificate for validity without having to contact an additional host, and creating a possible man-in-the-middle attack.[DA99]

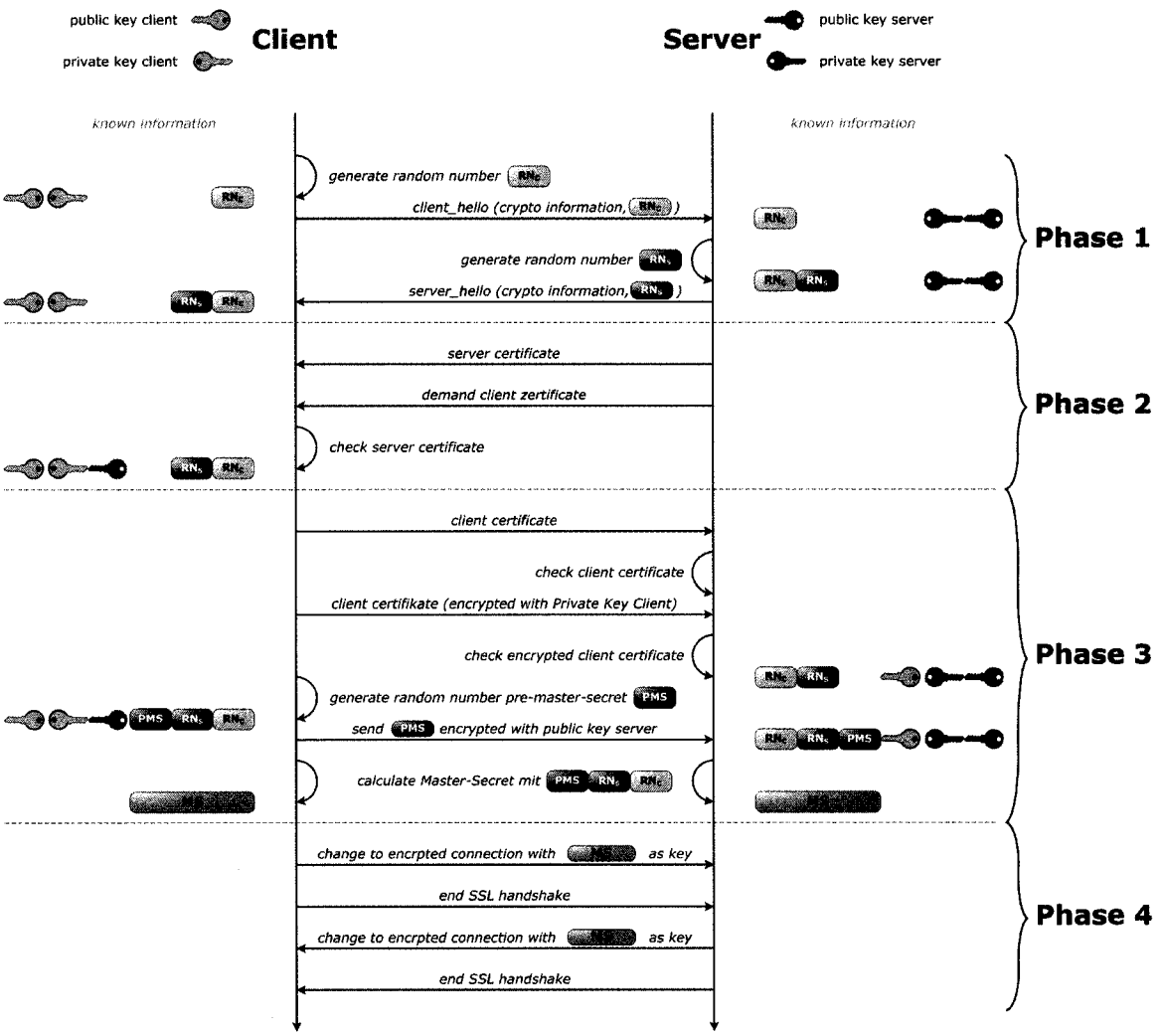

Illustration 4: TLS Handshake Phase - [FRIEDRICH07]

Finally, after the certificate is verified, the client encrypts a random number with the server's public key, which only the server is able to decrypt with its private key. Because of this, only the client and server have knowledge of the session keys, and they remain hidden from any third party that may be tracking the session. From this random number, the client and server generate appropriate key material for the agreed upon symmetric cipher, and the encrypted data stream can continue. All forthcoming transmission between the two hosts are securely encapsulated in the secure tunnel.

4.2 TLS Handshaking Detail

To start a TLS negotiation, the client host first makes a TCP stateful connection to the server computer. All communication between the two hosts is transported using this TCP connection. After the TCP connection is established, the client sends the Client Hello handshake message. The most important parts of the message are shown in illustration 5. The message starts with the content length, followed by the handshake message type, which is this example is Client Hello (0x01). Following the message type, the client identifies the TLS version, in this case, is TLS v1.0 (0x0301). Additionally, the client sends some additional message length information, followed by the cipher specs that the client supports. Each cipher spec is a 4-byte code that defines the key exchange algorithm and symmetric cipher algorithm. Some examples from the illustration are:

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x000039)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x000035)

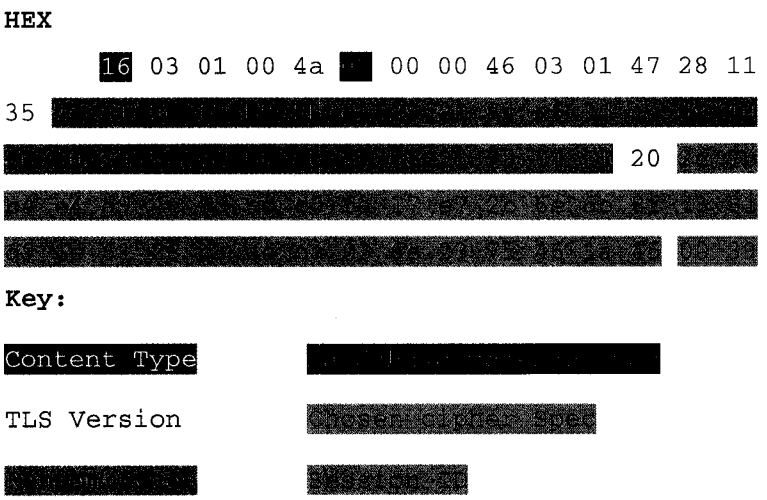

Illustration 5: Example ServerHello Message

- TLS_RSA_WITH_RC4_128_MD5 (0x000004)
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x000016)
- TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x000064)

Once the server receives the Client Hello message, it choose the cipher algorithm that offers the highest level of security from the list of algorithms supplied by the client. If the client's algorithm list does not offer a security level high enough, the server can break the connection. The server responds back to the client with a Server Hello message. This message identifies to the client which cipher spec was chosen. In this example, TLS_DHE_RSA_WITH_AES_256_CBC_SHA was the chosen spec. The server also includes a set of random bytes. These random bytes, along with a PreMasterSecret from the client, are used to create the Master Secret, which can be computed by both the client and server. All other

cryptographic information is derived from this Master Secret. Additionally, the server assigns a Session ID, which is an identifier for the entire TLS session. This Session ID can also be used to resume a previously negotiated TLS session without having to go through the handshake process again. The server also includes its X.509 certificate, along with any chain X.509 certificates required for validation of its certificate. Finally, it includes a Server Hello Done message, to notify the sender that all the messages are complete.[DA99]

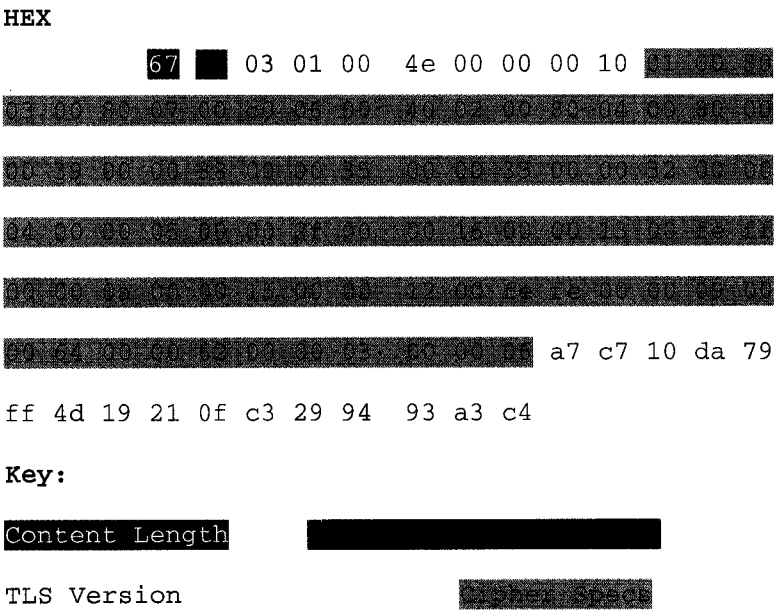

Illustration 6: Example ClientHello Message

The client then responds with a Client Key Exchange message. At this point, the client generates a random key, called the Pre Master Secret, that it encrypts with the server's public key.

It sends this random number to the server and the client and server generate the master secret. This secret is what is used to encrypt all data in the tunnel. Additionally, the client sends a Change Cipher Spec message to the server, letting the server know that it is now sending only encrypted data. Finally, the client sends a Finished message, which includes a hash and Message Authentication Code (MAC) of the previous handshake conversation. The server computes the same hash and MAC and compares it against the message sent from the client. If they match, the handshake is considered successful and all future communication between the client and server will be encrypted. Otherwise, the connection should be terminated and renegotiated.[DA99]

4.3 X.509 Certificate

The X.509 standard was initially issued by the International Telecommunication Union (ITU) as a standard for public key infrastructure in 1988. It has been updated several times since, and is currently in version 3 as defined by RFC 3280. X.509 is also commonly referred to as PKIX or Public Key Infrastructure. The X.509 standard relies on a set of trusted root certificates, which are usually preinstalled in the application that will be validating the certificates. These root certificates are usually self-signed certificates from trusted Certificate Authorities (CA). When another server wants to have a X.509 certificate for their server, the administrator would make a certificate request to a CA to generate a certificate for that user. The CA signs the requested certificate creating a hash of the certificate and encrypting it using the CA's private key. To validate an assigned certificate, the client uses the the CA's public key to decrypt the hash and compare it

against the hash that it computed for the certificate. If the decrypted hash and the computed hash match, the certificate is valid. The signature algorithm is generally RSA or Diffie-Hellman and the hash function is usually MD5 or SHA-1. [HPFS02]

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 347679 (0x54e1f)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=US, O=Equifax Secure Inc., CN=Equifax Secure Global eBusiness CA-1
        Validity
            Not Before: Jan 22 17:42:07 2007 GMT
            Not After : Jan 23 17:42:07 2008 GMT
                        Subject:    C=US,    O=4salebyu.com,    OU=GT18246483,    OU=See
www.geotrust.com/resources/cps   (c)07,   OU=Domain   Control   Validated   -   QuickSSL(R),
CN=4salebyu.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:cd:cb:04:02:3f:95:6f:01:d3:b9:a3:97:b3:a6:
                    51:c6:11:35:e1:9b:1d:bb:94:ec:63:25:b5:18:1c:
                    72:cc:86:c0:39:ca:f3:50:e8:28:9d:cb:4b:38:da:
                    30:b9:0c:49:1e:95:78:ae:5d:99:23:c3:66:02:8b:
                    cc:d6:6c:13:31:1e:16:e7:b4:87:52:3b:9c:a4:29:
                    60:e4:cd:0b:ae:31:02:29:56:50:be:84:06:47:b3:
                    82:27:ac:76:d2:0a:03:43:36:f8:3f:1d:d6:7e:f0:
                    df:18:31:d9:d7:1a:a5:ea:2b:28:fa:07:ac:50:f7:
                    4f:d8:13:69:89:30:e7:eb:43
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
            Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
            X509v3 Subject Key Identifier:
            37:89:22:37:C4:59:AD:9C:AE:1C:34:AD:1B:59:E9:1C:C5:92:3B:C7
            X509v3 CRL Distribution Points:
            URI:http://crl.geotrust.com/crls/globalca1.crl

            X509v3 Authority Key Identifier:
            keyid:BE:A8:A0:74:72:50:6B:44:B7:C9:23:D8:FB:A8:FF:B3:57:6B:68:6C

            X509v3 Extended Key Usage:
            TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Basic Constraints: critical
            CA:FALSE
    Signature Algorithm: md5WithRSAEncryption
        b6:32:43:8b:b4:bb:73:30:a6:be:05:ff:43:c2:c7:07:42:b0:
        74:2d:8d:b3:a7:98:58:d4:43:e1:86:17:1c:5c:78:e5:1a:26:
        a9:94:1a:63:70:fa:e4:d8:35:61:9e:e6:d6:5e:24:d6:c7:0d:
        4e:67:b2:91:44:15:94:11:22:22:e2:59:d3:44:76:14:db:75:
        5b:2d:a8:0f:d7:49:00:49:e3:f5:ea:19:9c:2b:e9:28:92:9c:
        31:f4:bf:aa:d4:45:4d:ba:dc:da:ee:81:08:49:cc:d3:6a:c7:
        f5:5d:f5:8b:4f:5c:bf:e0:84:78:93:99:3b:b5:61:24:ef:fc:
        c5:f4
```

Illustration 7: Example X.509 v3 Certificate

4.4 RSA Encryption

Since RSA encryption is heavily used in TLS, it is important to provide an overview of the algorithm. The RSA algorithm involves a public key and a private key. The public key is designed to be viewed by anyone and is used to encrypt messages. The private key is held secret by the owner and is used to decrypt the messages previously encrypted by the public key. The following steps are taken to generate the public and private keys for the RSA algorithm [RSA78]:

1. Choose two distinct large random prime numbers p and q.
2. Compute $n = p \cdot q$ (Note that n is used as the modulus for both the public and private keys)
3. Compute the Euler totient: $\Phi(n) = (p - 1)(q - 1)$
4. Choose an integer e such that $1 < e < \Phi(n)$, and *greatest common divisor*$(e, \Phi(n)) = 1$.

   e is the public key exponent. A popular choice for the public key exponent is 65537, as seen the X.509 key in Illustration 7.
5. Compute d to satisfy the congruence relation $d \cdot e \equiv 1 \pmod{\Phi(n)}$, ie $d \cdot e = 1 + k\Phi(n)$ for some integer k.

   d is the private key exponent.

The public key is the modulus n and the public exponent e. The private key is the modulus n and the private exponent d.

To encrypt a message m, the sender first turns m into a number where $m < n$. Then the sender computes the cipher text using the following formula:

$$c = m^e \pmod n$$

The sender then transmits c to the recipient, who then performs the following calculation to recover m:

$$m = c^d \pmod n$$

This works because of the following:

$$c^d \equiv (m^e)^d \equiv m^{ed} \pmod n$$

$ed \equiv 1 \pmod{(p-1)(q-1)}$ and hence $ed \equiv 1 \pmod{p-1}$ and $ed \equiv 1 \pmod{q-1}$

which can be rewritten as

$ed = k(p-1) + 1$ and $ed = h(q-1) + 1$ for proper values of k and h. If m is not a multiple of p then m and p are coprime because p is prime. Therefore

$$m^{(p-1)} \equiv 1 \pmod p$$

so, by using the first expression for ed

$$m^{ed} = m^{k(p-1)+1} = (m^{p-1})^k m = 1^k m = m \pmod p$$

If instead m is a multiple of p, then

$$m^{ed} \equiv 0^{ed} = 0 \equiv m \pmod p$$

Using the second expression for ed, it is concluded that

$$m^{ed} = m \pmod q$$

Since p and q are distinct prime numbers, applying the Chinese remainder theorem yields

$$m^{ed} \equiv m \pmod{p \cdot q}$$

Therefore

$c^d \equiv m \pmod{n}$.

The security of the RSA algorithm is based on the mathematical premise that the problem of factoring large numbers is hard. There is no efficient polynomial time algorithm to for factoring large integers, the a brute force attack is very difficult on RSA. It is currently believed that a 2048-bit key is sufficiently large enough to not be broken in the near future and larger keys are not breakable for the foreseeable future.[MOV96]

4.5 TLS Overhead

Two of the main issues that comes with using TLS is additional computation overhead on the servers and additional bandwidth consumption in the protocol exchange. First and foremost, encryption techniques, especially RSA, are CPU intensive. These extra CPU cycles required to perform the encryption can be expensive to implement on the mail and web servers that are deployed to handle the CIERS protocol.[APS99] Also, there is additional bandwidth required to perform the handshaking for TLS.

The amount of additional bandwidth required for the TLS handshake varies depending on the key and cipher algorithms used. As shown in Illustration 8, the amount of data required for this communication is between 719 bytes and 4260 bytes. This additional bandwidth is minimal compared to the overall transmission of the e-mail.[FKMT06]

In a recent study, the performance of web servers using TLS encryption compared to unsecured transmission was analyzed. In this study, several different CPU configurations were

Mechanism	Data Amount (bytes)		
	512-bit key	1024-bit key	2048-bit key
Plain PSK	586		
DH_anon	719	911	1295
DHE_PSK	791	983	1367
RSA	1242	1439	1861
DHE_DSS	1558	1950	2821
RSA mutual auth	1994	2388	3298
DHE_DSS mutual auth	2417	3009	4260

Illustration 9: TLS handshake bandwidth overhead

compared to study the effects on TLS on normal e-commerce web traffic. The e-commerce web traffic is similar to the average e-mail sizes, approximately 7KB, so this study is relevant to the type of data handled in the CIERS protocol. The study concluded that TLS imposes a factor of 3.4 to 9 overhead over insecure traffic, as shown in Illustration 9.[CDW06] This is a large amount of overhead, and would need to be carefully considered in the implementation of the TLS protocol on mail servers. Additional resources would have to be dedicated for even small mail servers.

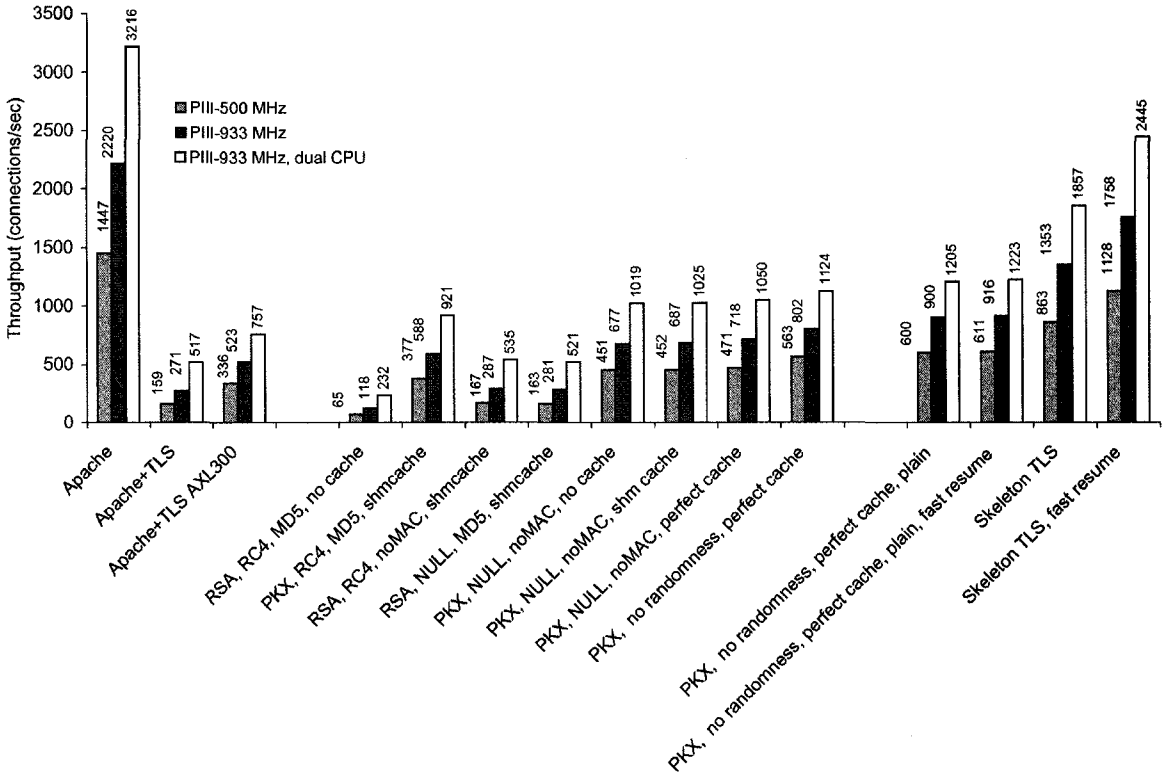

Illustration 10: TLS Perfomance Analysis [CDW06]

CHAPTER 5

COLLABORATIVE INTELLIGENT EMAIL RANKING SYSTEM

To solve some of the issues with DNS based systems and Bonded Sender, I propose a similar method that combines ideas from many of the different methods available. The first objective would be to setup a central authentication system that allows e-mail servers and individual e-mail users to sign up for an account. When the user signs up for an account, a certificate is generated for that user. The certificate consists of a public and private key. The public key is kept and published by the central authentication server, whereas the private key is kept secret by the user. The user is also assigned a Trust Value. To begin with, the Trust Value is high. This Trust Value is used to indicate to other servers how much they should trust the e-mail coming from that server. The value is based on a weighted average of user feedback, other blacklists and whitelists and the amount of unreported e-mail sent through the system.

The sending and receiving process in this system is as follows:

1. The sending server requests the authorization to send from the authentication server. If the server is listed in the service, the corresponding authorization is provided. If not, the whole process is skipped and standard SMTP transport is used to mail service
2. The authentication server logs the request from the sending server.

3. The sending server uses standard TLS encryption to communicate with the recipient server.
4. The recipient server authenticates the sending server's request to send through the CIERS authority.
5. The authority matches the recipients request to the sender's logged request. This match is then stored for later authentication against any complaints against the sender.
6. The recipient server tags the e-mail header with the sender server's identification, message identification and trust value.
7. The recipient server stores the message using standard server store techniques.
8. The user then request the e-mail using standard protocols such as POP3 and SMTP.
9. When the user views the message, he can decide if the message is SPAM or a valid message. If the user considers the message as SPAM, the client program will have easy functionality to send the message to the authentication server as SPAM. The authentication server matches the sending and receiving requests to its database to insure that the message was really sent through the server. It it was, the administrator of the sending server is notified and requested to take action. The Trust Value of the sending server is degraded until the problem is resolved.

This system allows for both authenticity and verification of e-mail. Servers registered that send SPAM will be immediately downgraded and other servers will quickly stop receiving mail from the other servers. Because the authentication server logs all the traffic, the servers can be reported to the proper governmental agencies for appropriate action, if necessary.

5.1 Authentication Authority Registration Process

The most essential aspect of CIERS is to have a central authority. When an user signs up for the CIERS service, that user must prove its identity as a real organization. The user must also provide authenticated contact information for dispute resolution between members. The CIERS authority's responsibility is to guarantee this information and maintain a large database off all the participating organizations. Once a user is registered into the database, it can add domains that are allowed to authenticate through the CIERS system. Each domain receives an individual certificate and is tracked separately. Additionally, each MX server is required to have its own valid and signed X.509 certificate for TLS encryption. This certificate can be signed by the CIERS authority or another CA, but it must always be a valid certificate.

CIERS requires the following information for user registration. This information is essential to authenticate the validity of the user. It is store in a confidential area of the site, but other users can view this information for dispute resolution. All this information is collected through a web portal sign-up process.

- Organization Information – Name and organization type
- Primary Contact Name
- Address
- Country
- Email Address
- Phone Number

- Username
- Password

The email address and phone number will be verified for authenticity by an automated system. The web server will send out an automatic e-mail to the user and must be responded to, in order to activate the account. Once the email is authenticated, the user will be contacted by an automated phone system message that must be able to contact the user directly by the phone number given. The user would then be required to enter in a code to validate the account. Once these two authentication methods are completed, the account is active. The user will then be able to add domains to the account.

A user can add as many domains to its account as desired. Each domain, though, must be allowed to be added to the account by the primary domain contact listed in the registrar's database. If the domain contact does not permit the user to add the domain, then that domain is rejected as a user's domain. The domain addition protocol is as follows:

1. A user submits a domain to be added.
2. The CIERS authority sends an e-mail to the primary domain contact requesting permission to allow that domain to be added to the user's account.
3. The primary domain contact can reject or accept the request. Either way, the user is notified of the domain contact's decision via e-mail.
4. If the request is accepted, the domain is assigned a 20-character randomly-generated, unique alphanumeric server identification string.

5. The domain is also assigned a private 20-character randomly-generated, unique alphanumeric server passphrase.
6. A SSL certificate is also generated for each mail server in the domain, if there is not a valid certificate existing for that server.
7. An initial trust value of 20 is assigned to the e-mail domain. All domains are considered fully trusted initially.

Once a domain is configured on the CIERS authority system it can begin sending and authenticating e-mail using the CIERS system.

5.2 Mail Server Communication

Several standard mail communication protocols are used in conjunction with a new defined XML request and response system to access the server identification, email identification and Trust Value of the server. We propose adding an extension onto the existing SMTP protocol definition name CIERS. This new extension will be discussed in detail proceeding the initial communication between SMTP mail servers.

An CIERS authenticated SMTP session proceeds as follows:

The sending server must first request the authority to send an e-mail from the CIERS authority. To do so, it initiates an HTTPS connection to a CIERS server. Using the SSL protocol for communication with the CIERS authority prevents a man-in-the-middle attack on the request. Also, using a standard communication protocol allows for easy implementation across the global

Internet. The sending server then sends an XML request to the CIERS authority server. The XML request format is as follows:

```
<CIERS request_type="auth_to_send">
        <sender_server_id><!-- Assigned Sender's Server ID -->
                </sender_server_id>
        <sender_passphrase><!-- Assigned Sender's Passphrase -->
                </sender_passphrase>
        <recipient_domain><!-- The recipient's e-mail domain -->
                </recipient_domain>
</CIERS>
```

If the recipient domain is valid, the CIERS Authentication server will respond with either the recipient server's ID, message ID and valid mail-exchange servers for the domain. On the other hand, if the recipient domain is invalid, the CIERS server will respond with a domain invalid message. The XML format for this communication follows:

```
<CIERS response_type="auth_to_send">
        <message_id><!-- Randomized Message Identification Unique for this
                Recipient Domain --></message_id>
        <valid_mx_servers><!-- May contain one or more MX servers that are
                        registered with the CIERS authority -->
                <mx_server><!-- Server Name or IP Address -->
                        </mx_server>
                <recipient_server_id><!--Valid Recipient Server ID -->
                </recipient_server_id>
        </valid_mx_servers>
</CIERS>
```

If the server does not have the requested domain on record it would respond:

```
<CIERS response_type="auth_to_send">
        <domain_invalid/>
</CIERS>
```

The sending SMTP server connects to the remote SMTP server using the standard SMTP port. After the sending SMTP server receives the header communication from the mail server, it initiates the EHLO string to the remote server.

Immediately after receiving an acknowledgment from the remote SMTP server, the sending server sends the STARTTLS command. The process for this is defined in RFC 2487.[HOFFMAN99] The SMTP servers negotiate an encrypted communication channel using the standard TLS protocols. If an appropriate TLS communication cannot be established, the sending server can choose to send the e-mail without using the CIERS authentication system, or reject the message back to the sender because of the communication failure. The CIERS protocol is not allowed to proceed unless the encrypted channel can be established using the authenticated public key of the recipient server.

Once a TLS session is initiated, the sending SMTP server sends the CIERS string to the remote server. The recipient SMTP server can respond with the following acknowledgments:

220 Ready to start CIERS

501 Syntax error

454 CIERS not available due to a temporary reason

If the response is 220, the sending server can proceed with the request to initiate CIERS send. If the response is 501, the sending server can either continue with the communication without CIERS authentication or reject the message back to the sender. If the response is 454, the sending server can either continue with the communication without CIERS authentication or queue the

message for a later attempt.

Once a 220 message is received, the sending server will initiate the CIERS send message. This message contains the sender server's identification and the message identification that is received from the CIERS authority. These two IDs become the key that the recipient server uses to authenticate that this is a valid sending server and that the server has not been spoofed by another user. Because the sending server must identify itself with the CIERS authority with the private passphrase and the message ID is given by the CIERS authority, the recipient server can guarantee that this is a valid e-mail from that domain. This request to send is sent in the following XML format:

```
<CIERS request_type="initiate_send">
        <sender_server_id><!-- Sender's Public ID --></sender_server_id>
        <message_id><!-- Unique assigned message ID --></message_id>
</CIERS>
```

Once the recipient server receives this request, it initiates a HTTPS communication with the CIERS authority to validate the request from the sending server. This communication is encrypted as to prevent a man-in-the-middle attack. The request is XML formatted as follows:

```
<CIERS request_type="trust_value">
        <recipient_server_id><!-- Recipient public server identification -->
                </recipient_server_id>
        <recipient_passphrase><!-- Recipient private passphrase -->
                </recipient_passphrase>
        <sender_server_id><!-- Sender's Public ID --></sender_server_id>
        <message_id><!-- Assigned message ID --></message_id>
</CIERS>
```

The CIERS authority would then lookup the message identification to match it against the previous request of the sending server. If the identification matches and the recipient server ID and passphrase are correct and matching the sending server's request, the CIERS authority will respond with a valid transmission message and the trust value of the sending domain. If the identification information does not match, the CIERS authority responds with an invalid transmission message, at which point, the recipient server can reject the message or still accept delivery of the message without CIERS authentication. The format for a validated XML response from the CIER authority is as follows:

```
<CIERS response_type="trust_value">
        <valid_transmission/>
        <trust_value><!-- Trust value between 0 – 20 --></trust_value>
</CIERS>
```

An invalid message would be:

```
<CIERS response_type="trust_value">
        <invalid_transmission/>
</CIERS>
```

Upon receiving a valid transmission message from the CIERS authority, the recipient server would then send to the sending server the following message:

```
<CIERS response_type="initiate_send">
        <request_accepted/>
</CIERS>
```

Alternately, if the recipient server received an invalid transmission message from the CIERS authority, the recipient server would then send this message:

<CIERS response_type="initiate_send">
<request_denied/>
</CIERS>

The recipient server would then either begin receiving the message or end the SMTP session in the usual manner. Additionally depending on the trust value of the sending server, the recipient server could reject the message for too low of a score. This option should be configurable in the SMTP server.

5.3 Additional Header Information

Upon receiving a valid CIERS authenticated message from the sending server, the recipient server would add header information to the message so that the client can identify the message. The additional information in the header is as follows:

X-CIERS-ServerID: <Sending Server ID>

X-CIERS-MessageID: <Message ID>

X-CIERS-TrustValue: <Trust Value>

This information is pertinent to the email client in identifying the message as SPAM and also in the response system to the CIERS authority. This system depends on the collaborative effort of all e-mail recipients to rate validity of the e-mail domains.

5.4 Client Response Mechanism

The central tenant to making this system work is having a collaborative response from many parties to identify which servers are sending SPAM. We can now guarantee that we can identify that the mail came from a certain domain, because the servers can no longer be spoofed; therefore, we can guarantee that the response mechanism will accurately depict whether spam was really coming from a certain domain, or just a rogue machine on the Internet.

To do this, an add-in must be added to the e-mail client software. The add-in will analyze the header information added in by the recipient SMTP server. Additionally, the add-in will have settings to allow restriction of e-mail delivery to the user based on the trust value. Finally, if the user views the e-mail and identifies it as SPAM, it will allow the user to send a response back to the CIERS authority to flag the e-mail as SPAM. The response is sent using HTTPS and XML in the following format:

```
<CIERS request_type="client_response">
        <server_id><!--The sending server ID from the e-mail header-->
                </server_id>
        <message_id><!-- The message ID from the e-mail header -->
        </message_id>
        <spam/>
</CIERS>
```

The server can respond can respond by accepting the response, which would require the CIERS authority to match the server ID and message ID to its database, or to reject the response. The accepting message is a XML reply as follows:

```
<CIERS response_type="client_response">
        <response_accepted/>
</CIERS>
```

The server can also respond by rejecting the request:

```
<CIERS response_type="client_response">
        <response_denied/>
</CIERS>
```

5.5 Processing of Client Responses

Once a response is received from a client, the CIERS authority allows a period of time for the CIERS user to respond to an allegation of SPAM abuse. The user that is registered is contacted via email to notify that user that a SPAM message has been flagged. The user can then login into its account and view all the messages that have been flagged as SPAM. For each email that has been flagged, the user can view the time and date that the email was sent, from which server it was sent from, to which domain it was sent, and when the e-mail was flagged as SPAM. The CIERS authority does not have the actual e-mail that was sent, so the user must match the e-mail to the log of its mail servers.

The user is given a chance to reject the claim of the SPAM abuse. There can be two ways the user can accomplish this:

1. Contact the administrator of the recipient domain and have that user mark in their user portal that the message is not SPAM.

2. Contact the actual user that received the e-mail, and have that user send a response using an CIERS e-mail response system to identify that the message was not actually SPAM abuse.

Because either of these two methods is tedious, it encourages the domain administrators to install necessary protection on their domains to prevent SPAM messages from being sent through their servers. Additionally, even if the message that was originally marked as SPAM is ultimately proved not to be SPAM, the system still takes into account these messages in the trust value score.

5.6 Scoring the Trust Value

The trust value is a simple average calculation between 0 and 20. This value is recalculated upon any e-mail processing by the CIERS authority. The formula for trust value calculation is as follows:

*Trust Value = 20 - ((total number of SPAM e-mails / total number of e-mails) * 20)*
*- ((total number of refuted SPAM e-mails / total number of e-mails) * 5)*

The trust value is recomputed for each email or client response that is sent through the system, so there is an immediate change to the trust value on any activity.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

As originally discussed, one of the most important aspects to the probability of CIERS working to decrease overall SPAM on the Internet is global acceptance of the protocol. If the majority of the servers on the Internet implement this technology, it would, theoretically, eliminate unsolicited email. There are several considerations, though, that would need to be addressed to allow for global implementation:

- Mail server programs would have to be written to include the TLS protocol (if they are not already capable) and the CIERS protocol.
- TLS encryption requires additional computation overhead on mail servers, by a factor of up to 9.
- Each mail server would need a valid X.509 certificate.
- The central CIERS authority would have to have tremendous bandwidth and storage capacity to handle all the email that is processed on the Internet each day.

The first consideration is moderately difficult because this would require vendor agreement on the part of the mail server program providers. There would most likely be an Internet draft process that would require a working group to be setup and study the protocol. After the working group

accepts the protocol, it must go through a review process and public comment period.[BRADNER96] If it is finally accepted, there would be an implementation period that would require vendor compliance with the protocol. Some shortcuts to waiting for new versions, is to develop a wrapper software that performs the CIERS protocol and then transmits the completed message to the existing mail server.

The second and third considerations would require a financial burden to be placed on the organizations that host and process e-mail. Fortunately, hardware is getting much less expensive and processing power is increasing exponentially, therefore CPU power required to perform TLS is not a large concern, unless the system hosts many mail accounts that transmit a large amount of mail. In this case, the cost could be tremendous. Additionally, X.509 certificates are not cheap. These certificates range from $20 to $1499 / year. [VERISIGN07][GODADDY07] If a company is running many mail servers, this would be quite expensive, and would require additional cost per year. These financial burdens would need to be weighed against the financial burdens placed on organizations because of the cost of dealing with SPAM.

6.1 CIERS Global Processing

The final factor that needs to be considered in implementing the CIERS protocol is the required facilities to fully implement the CIERS authority. With billions of e-mails going through the Internet daily, the bandwidth and storage requirements to handle that type of traffic would be tremendous. These implementation considerations are out of the scope of this document, but

several proposals could be considered:

1. CIERS could be handled by a government sponsored organization. That organization would need to be given the appropriate resources to cope with and scale to the demand put on it. Additionally, the government could mandate compliance within a certain period to the new standards.
2. A distributed CIERS system could be configured, with existing security organizations providing the backbone of the the system.

If the second option were to be implemented, additional protocols would need to be developed. These protocols could be an extension of the previously developed XML formats. This would allow for much easier scalability and deployment. For example, organization A and organization B could each offer certificates along with CIERS authentication. Organization A would then setup a trust relationship with organization B to share information about the different hosts that they authenticate for. If a request to send mail comes from a server that is subscribed to organization B to send to a server subscribed to organization A, organization B could retrieve the relevant information to process the CIERS authentication from organization A and would store the transaction in its database. In this method, the bandwidth and storage requirements would be spread through many different organizations, in a method similar to DNS registrars. These trust relationships could be controlled by a central authority that could certify organizations as compliant with the CIERS system.

6.2 Conclusion

It is obvious that there is no silver bullet to resolving the problems with unsolicited mail on the Internet. To solve this problem, it would take a tremendous effort by the entire Internet community. Even with that in mind, the CIERS protocol, in conjunction with additional protocols outlined in this document, this problem can be solved. The CIERS protocol can solve the issues of accreditation, authentication and reputation of mail servers. Additional techniques, such as Sender ID, could be used to authenticate individual user accounts on those mail servers. Combined with existing blacklists and filtering systems, the threat of unsolicited mail could be significantly reduced.

REFERENCES

[APS99] George Apostolopoulos, Vinod G. J. Peris, Debanjan Saha: Transport Layer Security: How Much Does It Really Cost? INFOCOM 1999: 717-725.

[BRADNER96] Bradner, S. The Internet Standards Process. RFC 2026. ftp://ftp.rfc-editor.org/in-notes/bcp/bcp9.txt. Oct 1996. Accessed 10/29/07.

[CRISPIN96] Crispin, M. RFC 2060 http://www.ietf.org/rfc/rfc2060.txt. Dec 1996. Accessed 1/25/2005.

[DCC07] DCC Graphs http://www.dcc-servers.net/dcc/graphs/. Accessed 10/16/2007.

[DA99] Dierks, T. and C. Allen. RFC 2246 The TLS Protocol version 1.0. http://www.ietf.org/rfc/rfc2246.txt. Jan 1999. Accessed 10/29/07.

[FTC05] Federal Trade Commission. Email Authentication Summit. http://www.ftc.gov/bcp/workshops/e-authentication/index.htm. Accessed 4/2/2005.

[FRIEDRICH07] Friedrich, Christian. Schematic representation of the SSL handshake protocol with two way authentication with certificates. http://en.wikipedia.org/wiki/Image:Ssl_handshake_with_two_way_authentication_with_certificates.png. Accessed 10/29/2007.

[FKMT06] Xiaoming Fu, Fang-Chun Kuo, Fabian Meyer, Hannes Tschofenig, Comparison Studies between Pre-Shared Key and Public Key Exchange Mechanisms for Transport Layer Security (TLS). Technical Report No. IFI-TB-2006-01, Institute for Informatics, University of Göttingen, Göttingen, Germany, ISSN 1611-1044, January 2006.

[GODADDY07] GoDaddy. http://www.godaddy.com. Accessed 10/29/07.

[HARDIE05] Hardie, Ted. IETF MARID Working Group to Close. Http://xml.coverpages.org/MARID-SenderID-Close.html. Accessed 4/2/2005.

[HOFFMAN99] Hoffman, Paul. RFC 2487 http://www.ietf.org/rfc/rfc2487.txt. Jan 1999. Accessed 10/4/2007

[HPFS02] R. Housley, W. Polk, W. Ford, and D. Solo. RFC 3280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List Profile. http://www.ietf.org/rfc/rfc3280.txt. April 2002. Accessed 10/29/07.

[KLENSIN01] Klensin, J., Editor. RFC 2821 http://www.ietf.org/rfc/rfc2821.txt. Apr 2001. Accessed 1/25/2005.

[LYON04] Lyon, J. and M. Wong. MTA Authentication Records in DNS. http://tools.ietf.org/html/draft-ietf-marid-core-01. May 2004. Accessed 4/2/2005.

[MOV96] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press. 1996. Pages 283-291

[MYERS96] Myers, J. and M. Rose. RFC 1939 http://www.ietf.org/rfc/rfc1939.txt. May 1996. Accessed 1/25/2005.

[PETERSON00] Peterson, Larry L. and Bruce S. Davie. "Computer Networks: A Systems Approach". Morgan Kaufman, Pages 371-397, 2000.

[RSA78] R. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Vol. 21 (2), pp.120–126. 1978.

[ROBERTS04] Roberts, Paul. IETF deals Microsoft's e-mail proposal a setback. Http://www.infoworld.com/article/04/09/14/HNietfmsblow_1.html. September 14, 2004. Accessed 4/2/2005.

[SA05] SpamAssassin. Http://www.spamassassin.org. Accessed 2/10/2005.

[SC05] SpamCop. Http://www.spamcop.net. Access 2/10/2005 and 10/16/2007.

[TRUSTE04] Truste. IETA: Summary of Progress, View of the Future. http://www.ftc.gov/bcp/workshops/e-authentication/presentations/11-10-04_maier_TRUSTe.pdf. November 10, 2004. Accessed 4/2/2005.

[TRUSTE05] Truste. Bonded Sender Program. Http://www.truste.com. Accessed 4/2/2005.

[VERISIGN07] Verisign. http://www.verisign.com. Accessed 10/29/07.

[YAHOO05] Yahoo!. Yahoo! DomainKeys. http://antispam.yahoo.com. Accessed 2/10/2005.

APPENDIX

SOURCE CODE

VITA

Graduate College
University of Nevada, Las Vegas

Nathaniel H. Whittacre

Home Address:
969 Ostrich Fern Court
Las Vegas, NV 89183

Degrees:
Bachelor of Science, Computer Science, 2002
University of Nevada, Las Vegas

Thesis Title: Collaborative Intelligent Email Ranking System

Thesis Examination Committee:
Chairman, Dr. Yoohwan Kim, Ph. D.
Committee Member, Dr. Thomas Nartker, Ph. D.
Committee Member, Dr. Laxmi Gewali, Ph. D.
Graduate Faculty Representative, Dr. Mei Yang, Ph. D.