SECURITY

Email Security with Cisco IronPort

The definitive guide to deploying and maintaining secure email architectures with Cisco IronPort ESA

ciscopress.com

Chris Porter

Email Security with Cisco IronPort

Chris Porter

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

Email Security with Cisco IronPort

Chris Porter

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing April 2012

Library of Congress Cataloging-in-Publication data is on file.

ISBN-13: 978-1-58714-292-5

ISBN-10: 1-58714-292-9

Warning and Disclaimer

This book is designed to provide information about network security. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The author, Cisco Press, and Cisco Systems, Inc., shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact: U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact: International Sales international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Brett Bartow

Managing Editor: Sandra Schroeder

Project Editor: Mandie Frank

Editorial Assistant: Vanessa Evans

Cover Designer: Sandra Schroeder

Composition: Mark Shirar

Business Operation Manager, Cisco Press: Anand Sundaram

Manager Global Certification: Erik Ullanderson

Development Editor: Eleanor C. Bru

Copy Editor: Sheri Cain

Technical Editors: Ben Hartwell, Tim Draegen

Proofreader: Sarah Kearns

Indexer: Tim Wright

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

About the Author

Chris Porter was one of the first field systems engineers hired by IronPort Systems in 2003, around the time of the launch of the ESA C-series product. He has served as systems engineer, SE manager, and now technical solutions architect at Cisco, who acquired IronPort in June 2007.

Chris has been involved in planning, deploying, and configuring Email Security Appliances (ESA) at hundreds of organizations, with a chief role in both pre-sales engagements and post-sales support. His experience has made him a trusted voice in ESA product design decisions.

Chris holds a bachelor's and master's degree in Computer Science from Stevens Institute of Technology in Hoboken, NJ, and a CCNA certification. Chris is currently a technical solutions architect at Cisco, specializing in content security and the IronPort email and web-security products and services.

About the Technical Reviewers

Tim Draegen has a long history in the email space. As Agari's director of technical strategy, Tim is responsible for technical innovate and advocacy. Tim works closely with numerous financial institutions, ISPs, and enterprises to design, develop, and adopt the technologies that enable secure, authentication-based email channels. Tim chairs the OTA Email Authentication Committee and is tasked with making the value of email authentication clear to everyone—not just email gurus or security professionals. Prior to Agari, Tim spent a dozen years as a software engineer and security researcher, including a stint at Cisco/IronPort, where in late 2004, he recognized the importance of email authentication and implemented DomainKeys, setting the stage for enterprise-grade protection of email domains and PayPal's successful 2007–2008 DMARC prototype.

Benjamin Hartwell started with IronPort Systems in 2002 and remained associated with IronPort/Cisco until the fall of 2011. Ben tested, managed, or was a technical advisor for the ESA from version 2.0 to version 8.0. Ben has been involved in quality assurance for 13 years, testing both consumer and enterprise products. He is currently continuing work in the security space, focusing on web security.

Acknowledgments

I had the pleasure of working with a fantastic team of editors on this, my first technical book. The team at Pearson has been extraordinary, and has my sincerest thanks. Ellie Bru guided me through the technical-review process and provided great feedback on content and layout. Thank you to Mandie Frank, who managed this book through its iterations in final editing. Special thanks go to Brett Bartow, without whose encouragement and guidance, this book would not have even been started.

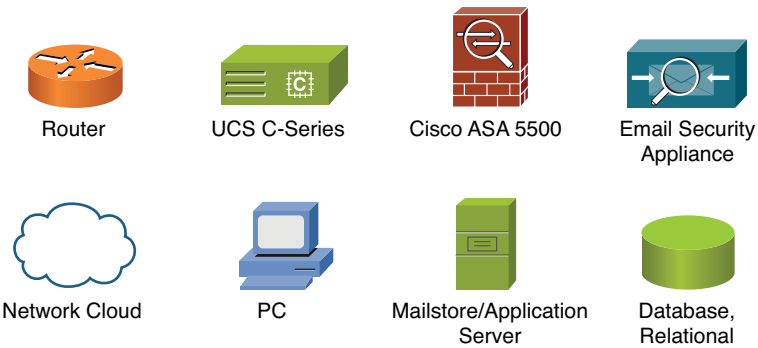
I owe a great debt to my technical editors, Ben Hartwell and Tim Draegen. Their long experience and technical expertise shaped this guide into the final product you see here. They questioned everything, asked for clarification on my statements, and suggested better examples. This book is all the better because of their work. Thank you again!

For more than eight years, I've been a part of an extraordinary team, first at IronPort Systems and now in Security at Cisco. I've never worked harder or had more fun at a job. It has been an honor serving with such an intelligent, motivated, and downright crazy group of individuals. My thanks to everyone involved with designing, building, selling, and supporting the ESA product.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the AsyncOS Configuration Guide. The Configuration Guide describes these conventions as follows:

- Boldface indicates commands and keywords that are entered literally, as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a status command).
- Italics indicate new terms, emphasized words, and command-line variables; for command-line variables, the italicized text is a placeholder for the actual name or value.

Introduction

The Cisco IronPort Email Security Appliances (ESA) have been deployed in thousands of networks to accept, filter, and deliver email messages. The ESA is easy to deploy and its security-filtering settings are effective right out of the box. However, many organizations are looking for more from their messaging environment and have barely tapped the potential of the ESA product line.

Email on the Internet is powered by the Simple Mail Transfer Protocol (SMTP). The simplicity of the protocol is its strength and weakness. Lack of authentication and the notion that one should accept all messages, gracefully, means that it is the most abused protocol on the Internet today. Spam is the most obvious form of abuse, but other dangers lurk, like bounce storms that can force servers offline in a denial of service (DoS) attack. Many organizations struggle with having email messages rejected or dropped by overzealous filtering.

The requirements laid on SMTP have changed over the years: Most businesses consider email to be their most important communication medium, ranking above telephone. Spam volumes have increased and changed form, and legitimate message traffic is growing in size and complexity as HTML has come to be the dominant format and the use of attachments has become widespread. New standards, such as SPF and DKIM, promise to bring proper authentication to email messages.

This book introduces the challenges facing messaging environments today and offers effective solutions on the ESA. It provides a series of recipes for solving particular messaging problems, delves into obscure features, makes recommendations on improving security, and shines light on oft-ignored issues like bounce blowback. Architecture recommendations are provided for deploying multiple ESAs in a variety of organizations, with the goal of improving reliability and automatically handling failure scenarios.

Goals and Methods

The goal of this book is to contribute to a more thorough understanding of Internet email, its inherent problems, and the solutions that the ESA product provides. I intend that for every area of functionality on the ESA, I provide not only how to configure, but whether and why you should. ESA is fundamentally a security product, but also a proper foundation for a reliable email gateway environment, and I hope to prove that it's far more than just a spam filter.

My method is to tackle the biggest problems that I've seen in more than 8 years of installing and configuring the ESA, and preface it with the relevant introductory material to make it accessible. Everything you'll read here is drawn from my personal experience with the product in hundreds of organizations, from small businesses to multinational corporations, higher education, service providers, and government.

Who Should Read This Book?

This book was written with a broad audience in mind. Email administrators that are using or considering ESA will learn everything from basic setup and configuration to advanced filtering rules, deployment architectures, and integration with external systems. Network architects will learn some of the root causes of email reliability problems and how to effectively configure an environment of multiple ESAs and related services. Security teams will gain deeper understanding of email as one of the primary vectors of malware and fraud, and learn about the less-visible aspects of email security. Anyone involved in designing a new gateway email environment based on ESA, or redesigning an existing one, will find this book valuable. Messaging directors will gain insight into some of challenges presented by internal email users as well as external senders, and be able to make better decisions about filtering policy, disaster recovery architecture, and email authentication standards.

How This Book Is Organized

Although you could read this book cover-to-cover, it is designed to be flexible and allow you to easily move between chapters and sections of chapters to cover only the material you need. If you do intend to read them all, the order in which they are presented is an excellent sequence.

Chapters 1 through 15 cover the following topics:

- **Chapter 1, "Introduction to Email Security"**: This chapter provides an introduction to the Cisco ESA and the AsyncOS software, their history and features, and an overview of the problems it is designed to solve. This chapter also covers the basics of SMTP and its dependency on DNS. The history and nature of SMTP abuses are explored.
- **Chapter 2, "ESA Product Basics"**: This chapter gives an overview of all the ESA features and the basic connection and setup process. It reviews differences between the various hardware models. The chapter walks you through the System Setup Wizard step by step, and then reviews the configuration you'll have after completing it. After showing a simple deployment example, more complex network installations are discussed.
- **Chapter 3, "ESA Email Pipeline"**: Messages accepted and delivered by the ESA pass through the filtering pipeline step by step, and this chapter covers each stage in detail. Important concepts, such as listeners and the work queue, are introduced. The configuration options for each filtering stage are discussed and best practices are presented. After reading this chapter, you should understand what happens at each stage of the pipeline, from initial connection to final delivery, and the order in which they occur.

- **Chapter 4, "ESA Web User Interface"**: This chapter covers the main administrative interface of the ESA: the web user interface. It describes how to connect to the WUI and provides an overview of all the administrative web pages. Where appropriate, it provides the equivalent CLI commands. The chapter covers other important administrative concepts, such as the commit process, user privilege levels, and centralized management through the WUI.
- **Chapter 5, "Command-Line Interface"**: This chapter discusses the command-line administrative interface. Knowing how to use the CLI is critical for power users and essential for automation. The chapter covers connecting to the CLI using common SSH clients and using public-key authentication. All available commands are described, including some hidden and undocumented commands. You learn how to investigate and troubleshoot email delivery issues with examples of CLI command usage.
- **Chapter 6, "Additional Management Services"**: This chapter covers other monitoring and information services available on the ESA. You'll learn how to enable SNMP and use the ESA MIB to query SNMP attributes and react to traps. The chapter covers the use of the ESA filesystem and how to transfer data to and from the ESA. Lastly, it covers ESA logging including descriptions of all of the log types and an in-depth analysis of the primary mail log format.
- **Chapter 7, "Directories and Policies"**: This chapter covers LDAP directories and the ESA features that make use of your directory information. You'll learn how security and performance are improved by using LDAP features and the options for tying ESA policy to directory membership attributes. It then covers basic and advanced LDAP queries, AD specifics, and troubleshooting LDAP. It ends with a series of recipes for common directory integration problems.
- **Chapter 8, "Security Filtering"**: This chapter covers anti-spam, antivirus, and Outbreak Filters, the chief security engines of the ESA. It describes the process of enabling scanning and rule updates. You'll learn the options you have for handling junk and malicious message verdicts and how to tune the settings for your organization. This chapter also covers using content filters for interaction with engine verdicts and writing filters for common security needs.
- **Chapter 9, "Automating Tasks"**: This chapter provides an understanding of how to automate administrative tasks from remote systems using the CLI and WUI. It includes descriptions of automatic configuration backups using expect and Perl, retrieving XML status data, and exporting reporting information for processing in external applications.
- **Chapter 10, "Configuration Files"**: This chapter covers the XML file format used by the ESA for saving and restoring configuration files. The structure and components of the file are discussed, and examples of configuration file manipulation are provided.

- **Chapter 11, "Message and Content Filters"**: This chapter focuses on the two custom rules engines available on the ESA: message and content filters. Every filter has one or more conditions and actions, and all of them are described here. You'll learn the difference between the two engines and when to use each. Many examples are provided, and several difficult examples are illustrated. Performance problems and other limitations are explored.
- **Chapter 12, "Advanced Networking"**: This chapter provides depth on all the possible network models you can use to deploy an individual appliance. It covers single and multihomed deployments, the virtual gateways feature, and advanced routing and DNS configurations. Other network topics, like TLS for secure connections and DSR for load balancing, are discussed.
- **Chapter 13, "Multiple Device Deployments"**: Almost every ESA deployment will involve multiple appliances, and this chapter reviews the challenges and solutions whether you have two or 20. Load balancing for reliability and performance is discussed.
- **Chapter 14, "Recommended Configuration"**: This chapter covers best practices for ESA deployment and configuration. Security options for incoming mail are reviewed. You'll learn about handling outgoing mail properly, and what it means to be a good Internet email sender. Different types of organizations and their challenges, from small and large businesses to universities and service providers, are discussed.
- **Chapter 15, "Advanced Topics"**: This chapter covers two major topics that have emerged in the last few years, and the ESA approach to the challenges they create. Two email authentication standards, SPF and DKIM, have become widespread, and we'll examine how these work and the arguments for using them. Regulatory and Data Loss Prevention (DLP) policies have put new pressures on email administrators and policy makers, and you'll learn how the ESA provides capabilities to meet these requirements.

Chapter 1

Introduction to Email Security

In this chapter, you will learn the following:

- Brief overview of the Cisco IronPort Email Security products
- A history of the AsyncOS software releases that run the Email Security Appliances
- Basic topics of Internet email, focusing on the Simple Mail Transfer Protocol
- Definition of email security and overview of the threat landscape

Overview of Cisco IronPort Email Security Appliance (ESA)

The Cisco IronPort Email Security Appliance (ESA) family was launched in 2003. IronPort Systems, then an independent security company, created ESA as a self-contained hardware and software product to provide high-performance email security. The goal was to provide a single product that accepted, filtered, and delivered Internet email messages, and to do so reliably and quickly. Initially, the ESA provided basic message features along with filtering for spam and virus messages, but subsequent releases have dramatically expanded the range of capabilities.

ESA was certainly not the first system that performed virus or spam filtering for email messages; Bulk Unsolicited Commercial Internet email (UCE, the technical term for *spam* email) has been around almost as long as Internet email itself. It was one of the first, however, to combine these features into a single product, and to do so on a purpose-built platform with extremely high performance. Prior to the introduction of messaging appliances, typical email architectures included multiple layers of filtering, either as separate products running in series or as filtering products that *plugged in* to messaging servers. These multilayer designs used either proprietary application programming interfaces (API) for moving message contents or used Simple Mail Transfer Protocol (SMTP)

to deliver messages from one system to another. The all-in-one appliance form factor removes the layering complexity, lowering the cost of providing effective reliable email security. Messaging servers are freed from the role of email filtering, allowing resources to be used to serve end users.

Because of its primary goal of security filtering, the ESA does not act as a message store and does not provide end-user access protocols, like Post Office Protocol (POP) or Internet Message Access Protocol (IMAP). It does not provide a webmail client interface. To compose and read messages, your environment must have other servers for messages and end-user interfaces.

IronPort continued to deliver new security and filtering services, enhancements to the base platform features, and increased message throughput with new hardware models and software releases. Cisco Systems acquired IronPort in June 2007. The IronPort product line continues its strong history as part of the Cisco Security Technologies business unit. The ESAs are also referred to as the *C-series* line of Cisco appliances, due to most of the model numbers starting with the letter C, such as the C370 and C670. The exception is the X-series, like the X1070. Although the *X-series* models are ESAs and offer the same functionality, their high-end hardware and resulting high throughput is suitable for carriers, service providers, and extremely large enterprises. The X- moniker distinguishes the *carrier class* products. The original IronPort models—the A50 and A60—had only email acceptance and delivery capabilities, without security filtering, and have reached end-of-life status.

Note All the ESA models have the same software features and differ only in the hardware platform, with some exceptions. The chief difference between the models is CPU count and speed, and RAID disk count and RAID mode. There's no difference in the availability of software features, and this guide's configuration guidelines apply to all models. The smaller 1U hardware units (like the C150 and C160) have differences, like two physical network interface cards (NIC) instead of three, which are important in some configurations. We note these differences where they are relevant.

In this guide, we refer to all the hardware models as Email Security Appliances (ESA). The product has also been called a Message Transfer Agent (MTA) and Messaging Gateway Appliance (MGA), and sometimes incorrectly referred to as the *IronPort Spam Filter*. Because it offers far more than spam filtering, we stick with the acronym ESA.

The term MTA refers to the servers tasked with accepting and delivering messages and, in Internet SMTP architecture, is distinguished from mail delivery agents (MDA), which store messages and provide user access, and mail user agents (MUA) that allow users to retrieve, display, and compose messages. Some MTA products provide MDA capabilities directly on the same system, but the ESA does not.

AsyncOS

AsyncOS is the name given to the collected software running on the Cisco IronPort appliances. It includes the base operating system (OS), device drivers, memory management, process scheduling, and all the application and scanning software. The OS fundamentals are built on FreeBSD, with significant portions specifically altered for messaging tasks. Low-level components are written in the C programming language, while most of the application software and all the management interfaces are written in Python and use a coroutine-based model called *sbrapnel*. This high-performance threading library was specifically built for the processing needs of email, allowing the ESA to handle thousands of simultaneous connections.

AsyncOS also refers to the messaging software, all the security filtering, the web-based user interface (WUI), and the command-line interface (CLI). AsyncOS versions are referred to by a *Major.Minor.Point-Build* number format, such as 7.1.0-310. Each AsyncOS software build is complete and self-contained. Upgrades from one version to another involve an entire build image, instead of individual upgrades to components. The only exception is the security engines, whose software versions are automatically independently upgraded by the system. Security engine updates are dynamic in order to provide real-time protection against the latest virus and spam variants.

Security Management Appliances (SMA)

Throughout this book, we often refer to a security management appliance (SMA) (or the *M-series* appliances). These separate Cisco IronPort appliances complement the ESA and provide centralized features, such as email reporting, message tracking, and the end-user IronPort Spam Quarantine (ISQ). It is typical to deploy one or two SMAs in conjunction with two or more ESAs to provide these centralized services for the environment. In larger deployments, of four or more ESAs, the SMA is indispensable. Despite the name, as of this writing, the SMA provides no actual configuration management for ESA devices; centralized management is done directly on the ESAs with a dedicated clustering feature.

The SMA is indispensable in ESA deployments, because it provides a single centralized interface for email reports and message tracking. The SMA consolidates the data spanning many ESAs and provides a single interface analysis or investigation. The reporting and tracking features that the SMA provides are also part of the standard ESA feature set, although limited to a single appliance when run on the ESA. Most of the reporting and tracking features we discuss are available as described in either ESA or SMA. The SMA provides much higher capacity for storage and much higher import rates for log data. The higher capacity of the SMA allows for a larger ISQ, which provides storage and user access to messages deemed to be a spam threat. Quarantine is one of the possible actions for filtered spam messages and, like the other features, is available on both the ESA and the SMA, but the SMA provides a single centralized interface and more capacity.

Another benefit of an SMA is that it offloads the processing work of tasks like message tracking and quarantine access, which are unpredictable and can place high load on an ESA when used. For this reason, in most environments, it's preferable for centralized reporting, tracking, and quarantine to be run on the SMA. Like ESA, there are different models of SMA, differing only in performance and capacity.

The SMA is built on the same code base as the ESA, and so its user interfaces, administration features, configuration options, and monitoring and reporting are similar. However, because the SMA is not intended to accept, filter, and deliver email, those portions of the configuration aren't available. Where the ESA and SMA are similar, those parts of this book are applicable to both families of appliances.

History of AsyncOS Versions

AsyncOS was first publicly available in general release with software version 2.0 for the A60 model appliance in November 2002. Versions 2.0 and 2.5 focused on high-performance message delivery and features designed to allow businesses that rely on email communications to quickly deliver email. It was quickly adopted by retail, banking, insurance, and service-provider companies that needed to manage and deliver large email campaigns.

In June 2003, version 3.0 was the first that focused on high-performance MTA features for both incoming (receiving) and outgoing (sending) SMTP mail. Table 1-1 lists the major AsyncOS releases.

Table 1-1 *History of Major AsyncOS Releases*

Version	Highlights
AsyncOS 3.0, June 2003	Public listeners and inbound mail support.
AsyncOS 3.5, December 2003	Brightmail Anti-Spam, LDAP directory integration features, and Sophos Anti-Virus.
AsyncOS 4.0, October 2004	Virus outbreak filters, dramatic WUI overhaul, including policies and content filters, policy quarantine, SMTP AUTH, HTTPS streaming upgrade.
AsyncOS 4.5, September 2005	IronPort Anti-Spam, centralized management, domain keys signing.
AsyncOS 4.7, July 2006	Bounce verification, LDAP failover features, network diagnostics, ISQ.
AsyncOS 5.0, March 2007	Enhancements to set up defaults, spam quarantines, spam filtering, LDAP settings, and notifications.
AsyncOS 5.5, October 2007	Introduction of ESA envelope encryption for email, LDAP enhancements, DKIM verification, AsyncOS reversion.

Version	Highlights
AsyncOS 6.0, March 2008	New reporting and tracking engine on ESA, introduction of the SMA features for centralized tracking. System Setup Wizard and LDAP Wizard for AD were added to simplify initial configuration.
AsyncOS 6.3, July 2008	New hardware platform support, introduction of Image Analysis engine.
AsyncOS 6.5, December 2008	External authentication and new user roles. Enhancements to destination controls, system setup, and encryption. New upgrade methods for accelerated AsyncOS upgrades.
AsyncOS 7.0, November 2009	RSA DLP engine, IPAS "marketing" verdict, intelligent multiscan, prioritized SMTP routes.
AsyncOS 7.1, April 2010	"No-auth" envelope encryption, weighted SMTP routes, TLS improvements.
AsyncOS 7.5, December 2011	Introduced SMTP "call-ahead" for recipient validation. Virus Outbreak Filters enhanced and renamed Outbreak Filters (OF). Role-based access control, password complexity, and expiration rules for user accounts. Combined text and HTML disclaimers.

Software Features

The ESA provides features for all stages of accepting, filtering, and delivering email messages with SMTP. In a typical deployment, an ESA is deployed as the first mail server for email coming from the Internet, and the last mail server on the path out to the Internet. All models of appliance provide the following features:

- **Connection controls and rate limiting:** In the past, email has the unstated assumption that anyone who wants to send mail to your organization will be allowed. Not so with the ESA, which can strictly control the connections and access of junk senders.
- **Email acceptance and delivery:** Only accept the right messages, and quickly get them to the right destination. Ensure that email delivery is reliable, highly available, and extremely high performance.
- **Security filtering:** Spam, viruses, fraud, phishing, and other kinds of unwanted messages threaten the reliability and security of your entire computing environment. ESA shuts down this infection vector and delivers clean, legitimate email.
- **Data loss prevention and encryption:** Whether through oversight or intent, users can potentially send confidential or personally identifiable information via email, in violation of company policy or state or federal regulations. ESA has filtering features to detect and stop these messages or, when the transmission is allowed but must be secure, provides means to encrypt the content in transit.

- **Custom filtering:** For other filtering challenges, ESA provides a flexible filtering engine to identify content based on the sender, recipient, subject, or body of email messages, and even within attachments in some 400 different formats.

Email Security Landscape

What does the term *email security* mean? When we think of the problems associated with Internet email, the immediate thought is usually the most visible problem: spam. Formally defined as bulk, unsolicited commercial email (UCE), spam is a clear reminder of the problems that plague email systems, aggravating administrators and users and interfering with legitimate business use of email. Although it is the most visible email problem, it's not the only threat to a company's email infrastructure.

As the single most important means of business communication, email availability is vital to any company's daily operations. It's also one of the chief means by which employees interact with the outside world. We're generally concerned with three primary tasks for email security: stopping the bad mail from coming in while allowing legitimate mail to flow unimpeded; controlling outgoing mail that might contain sensitive information, or that might be sent to unauthorized third-parties; and dealing with any unplanned situation that might cause email receipt and delivery to be stopped.

Note A Cisco study conducted in 2006 showed Internet email to be approximately 75% spam; a similar analysis in May 2010 showed 85%. For the first time, however, the second half of 2010 showed a sustained decline of spam volume. Estimated average daily volume of spam was approximately 100 billion messages per day in December 2010 compared with over 300 billion messages (average) per day from May to July 2010. It remains to be seen if this is just a brief respite to be followed by another increase or a major inflection point in the business of spam.

Email Spam

Despite being simple for humans to identify, spam is difficult to define precisely and thus difficult for software to identify. Most users classify any unwanted message as spam, with a definition of unwanted that varies daily. There's a difference between messages sent in bulk by legitimate senders, such as retailers, banks, media outlets, social networking, auction sites, and the unsolicited, usually fraudulent advertising in spam messages. We can break down spam into various categories, including phishing (fraudulent messages that seek to fool recipients into revealing personal information or login credentials), advertising, advance-fee fraud, *419* scams (money-courier schemes that promise to reward recipients with a portion of the money being transferred), but it's all a problem, and it all needs to be filtered.

For this book, we use the term *spam* to refer to unsolicited email messages related to criminal activity, be it advertising fraudulent products or services or enticing users to

provide information or participate in a fraud. Bulk messages sent by legitimate senders, even if unwanted, are more properly termed *marketing* or *broadcast* messages. Of course, some grey area exists here, and the conduct of legitimate senders sometimes crosses the line. Your organization may have some automated or bulk messaging, as it is an extremely effective and inexpensive means of contact with customers, partners, and vendors. (Chapter 14, "Recommended Configurations," addresses the topic of being a good bulk sender.)

In the U.S., several states and the federal government passed legislation defining and prohibiting unsolicited commercial email. In general, these laws prohibit sending email of a commercial nature without some pre-existing business relationship with the recipient and requiring messages to contain contact information and opt-out instructions. Unfortunately, as is often the case with legislation over technology, the results have been mixed. Many spam originators operate from countries with little or no technology laws, and other originators toe the line, claiming to have the pre-existing relationship with the recipient that, in practice, is nearly impossible to verify.

Viruses and Malware

Messages that contain binary attachments, whatever the source, can potentially include malicious executable software. A *virus message* is any email message that contains one or more malicious executables. We can make distinctions between different types of viral messages, like those that use social-engineering versus *Trojan horse* software that purports to have a legitimate purpose, but for us, it's all a threat that needs to be eliminated.

The world of viruses and malware has changed dramatically over the years. Viruses and worms were initially spread as a means of proving a point or gaining notoriety for their authors. Today, the motivation is almost purely criminal: to steal user credentials and personal information by means of keystroke logging and system monitoring, to establish a software foothold inside of a corporate network, and to spread the infection to other users. Email is often only one vector for infection, and the software that succeeds in bypassing security filters today is extremely sophisticated, capable of phoning home for instruction, using multiple protocols and vectors, installing new components to morph over time, and hiding from virus scanners and even the OSs themselves.

We are, of course, concerned about the email infection vector, and there, the situation has been changing. Because of the widespread deployment of email security systems, the use of broad attachment-filtering rules, improvements to mail clients and OSs, and the effectiveness of network security solutions, email has recently become less effective as a vector for distributing malware. Email-borne viruses have become more targeted, seeking out specific individuals or organizations, or exploiting social network ties and email address books to mimic communication among friends and associates.

In place of the infected email-attachment vector, more attackers have been using URLs to malicious software and messages that entice the user to click the link. The enticement takes on many different forms: Common ones are the promise of money, revealing gossip, threats of account closure, or claims of having some embarrassing information. The

end goal is the same: A user clicks the link that leads to malicious software. The software-delivery methods also vary; some sites claim that the user needs updated plugins or toolbars installed, while others rely on unpatched browser software to execute a silent *drive-by* download. Whatever the message and the means, the messages represent a significant threat and a security target for the ESA.

Protecting Intellectual Property and Preventing Data Loss

Email security also means examining outgoing mail sent by local users to recipients on the Internet. This communication—to partners, contractors, suppliers, media, or the public at large—represents a public face to your organization. That public-facing nature requires the same kind of brand protection and communication policy that your organization mandates for any public communication. It also represents a serious risk, because it allows internal users with access to sensitive or confidential information a direct communication path to the Internet.

In a typical deployment, ESA is situated as the first email *hop* on the way in, and the last hop on the way out. When architected this way, the ESA is an ideal chokepoint for examining both inbound and outbound email messages and applying actions like encryption.

Protecting intellectual property in an organization is a big topic, and email is only a part of it. But, the same steps taken to identify, classify, and secure data can be applied to ESA email policies, and the rules about what can and cannot be sent to external recipients can be controlled there.

The latest emerging pressure on email environments is the introduction of legislation from state and federal governments over the transport and disclosure of certain kinds of electronic information, and email is certainly covered under these regulations. In the financial industry, these requirements have existed for years, but in other verticals, the pressure is new. The legislation is often not specific enough to dictate exact policies on electronic communication, but the ESA provides a variety of tools to allow your organization to implement the controls it deems necessary to comply.

Regulatory compliance typically focuses on a few classes of information generally encompassed under the term Personally Identifiable Information (PII): payment card numbers, bank routing numbers, and other financial account information, government ID numbers, personal names, addresses, telephone numbers, and healthcare records. The ESA's Data Loss Prevention (DLP) features provide rules for identifying these classes of data, or defining your own classes, and taking action on the messages as appropriate. A common policy is to encrypt content that contains sensitive information, when that message would otherwise be sent to an external recipient in the clear. Encrypting email content satisfies the requirement that prohibits sending personal information in the clear.

Other Email Security Threats

Aside from the obvious threats that spam and viruses pose, and the challenge of filtering outgoing mail, your organization may face a number of other email-specific problems that affect email availability:

- **Denial of service (DoS):** Almost any protocol can be compromised by DoS. In email, this can be intentional with floods of email traffic, or accidental, as with misdirected bounces or notifications. ESA provides several features to protect against these kinds of messages.
- **Fraud and impersonation:** Because of the lack of authentication in SMTP, sender addresses can be *spoofed* or impersonated. This can be an issue for both inbound and outbound traffic: Your users can be fooled into trusting message content that appears to be from reputable sources, and your own local user accounts and domains can be impersonated, potentially affecting your reputation online and potentially leading your customers or partners into fraud. This is a multifaceted problem, and there is no silver-bullet solution. However, there are some new industry efforts and technology features on the ESA that can help mitigate or eliminate this problem.
- **Online activism:** Email is an inexpensive and widely available service, and the letter-writing campaign of yesteryear lives on in email campaigns created by activists. Because of the quick spread of information and the ease of sending email addresses, anyone with a point to make can quickly do so via email and encourage or enable others to do so, too. Because it's usually easy to guess email accounts for executives or other prominent individuals in an organization, it's easy to send them a lot of email. Regardless of the message, high volumes of email can cause problems for the organization and its targeted individuals. The ESA provides filtering capabilities, allowing you to tailor a solution that fits your organization.
- **Blacklisting:** One early solution to the problem of spam, and the computers that were sending it, was to create a public listing of IP addresses that were seen sending spam. These services are known as spam black-hole lists, or blacklists. Because they typically provide their listing using Domain Name Service (DNS), they are often called DNSBLs. These have been effective as a spam-fighting mechanism, and their effectiveness has forced spammers into new techniques to avoid them. Unfortunately, some blacklists are better than others, resulting in some lists that are easy to get on and difficult to get off. When your organization is added to a public blacklist, it can affect delivery of all email to any destination that uses that blacklist, and it can be time-consuming to arrive at a root cause and get off the list. We discuss strategy and ESA features that can help you stay off of these lists in Chapter 14.

Simple Mail Transfer Protocol (SMTP)

Internet email is driven by SMTP, which is one of the most venerable Internet standards. SMTP was first formally defined by Jon Postel in RFC 821, published in August 1982. It

was not the first Internet messaging protocol. SMTP evolved from experience with earlier protocols, some based on FTP, for delivering electronic messages on the ARPANET. For some time after the ARPANET transitioned into the modern Internet, SMTP was a complement to Unix-to-Unix Copy (UUCP) mail, which has since virtually disappeared. The legacy UUCP "bang path" addressing can still be found in SMTP, unfortunately only used for exploiting vulnerable systems.

The most recent specifications for SMTP and Extended SMTP (ESMTP) can be found in RFC 5321, published on October 2008. (In this book, we use the acronym SMTP in most cases, even though we are almost always referring to ESMTP. Where the distinction is important, we note it.)

Note Virtually every software, hardware, or service-based Internet email product supports ESMTP, although the ESMTP functions that are supported vary. Because it's rare to find a pure SMTP-only client or server, we make the assumption that we're always working with ESMTP-capable agents.

The RFCs for SMTP reserves destination TCP port 25 for its use. Although there are some cases for running SMTP services on ports other than 25, that's the industry standard, and the ESA defaults to port 25 in all cases. If not otherwise specified, you can assume port 25 when we're talking about SMTP. As with most application layer protocols, the TCP source port for clients is a random high number.

RFC 821 states in the first line of its introduction that the goal of SMTP is "to transfer mail reliably and efficiently." SMTP uses TCP connections for transport, although it is technically independent of transport protocol and requires only a reliable, ordered data stream. SMTP has built-in mechanisms to ensure reliable delivery. The *store-and-forward* approach that most SMTP software uses means that a message is either delivered or it's not, and the final disposition of any message should never be unknown. Transient (non-permanent) errors are a standard part of the protocol, and it is typical for clients to hold messages and attempt redelivery should a temporary error occur during an SMTP transaction.

SMTP is a plain-text protocol (technically, it was originally defined to support a 7-bit character set) and is intended to be easily human readable. In fact, messages can be transmitted manually using Telnet to an appropriate SMTP server.

I refer to an SMTP session between client and server as an *SMTP conversation*. After a TCP connection is made, roles are defined for the client, which sends commands and data, and the server, which parses the commands and responds. SMTP is fairly interactive with a back-and-forth of commands and responses between client and server. Email software that understands SMTP can act as either client, server, or both; MTAs, like ESA, routinely serve both needs. We look at the need for and use of MTAs later in this chapter.

Example 1-1 is a simple SMTP conversation example. The line numbers are not part of the session; they're here so we can refer to the commands. The client's commands are in plain text while the server's response appears in italics.

Example 1-1 *Simple SMTP Conversation Example*

```
<client connects to server esa02.cisco.com>
1   220 esa02.cisco.com ESMTP
2   HELO external-sender.com
3   250 esa02.cisco.com
4   MAIL FROM: <sender@external-sender.com>
5   250 sender <sender@external-sender.com> ok
6   RCPT TO: <chriport@cisco.com>
7   250 recipient <chriport@cisco.com> ok
8   DATA
9   354 go ahead
10  Subject: Example Message
11
12  This is the text of an example message.
13  .
14  250 ok: Message 31274 accepted
15  QUIT
16  221 esa02.cisco.com
```

Line 1 is called a banner, or greeting, that the server sends to the client. The 220 is a three-digit response code from the server, and this one indicates success, or at least, no error to this point. SMTP uses three-digit codes that RFC 5321 refers to as xyz, where x can be 2, 3, 4, or 5. Any code beginning with 2 is a *success* code, often referred to as 2yz or 200-class. The RFCs define the response codes that should be used in SMTP conversations, but in practice, the RFC definitions aren't always precisely followed. We can usually count on the first digit being accurate: 2yz for success, 3yz for success (but waiting for more data), 4yz indicating a transient (non-fatal) error, and 5yz is any permanent, fatal error. Context for the response comes from the point in the conversation where the error occurred. The second and third digits in a response code can be revealing, but because of variable interpretation by different products, you shouldn't rely on them to be perfectly accurate. Some systems respond with an error code of 550 for all mail-delivery errors and don't distinguish between causes. Table 1-2 shows some common SMTP response codes. The ESA references refer to the default configuration; many of the response codes on ESA can be customized.

Table 1-2 *Common SMTP Response Codes*

Response Code	Typical Use
220	Used as the success code for the initial SMTP server greeting.
250	Success, the command was accepted. The context of this response will determine exactly what was successful.
354	Response to an SMTP client's DATA command. It can be interpreted as *OK so far, waiting for message data*. It's the only 3yz code you'll see from an ESA.
421	Temporary failure or temporary rejection at the connection level. Usually means that the server is too busy to handle any more connections, or is deliberately rate-limiting the sender. ESA uses this to reject new connections when it reaches its global limit or has reached a policy limit for this connecting IP. Could also indicate that the service is temporarily not available (for example, if shutting down or if license key has expired).
452	Temporary failure or rejection at the message or recipient level. ESA uses this response code to reject individual recipients on a message when the sender has exceeded the limits for recipient count, or exceeded the limit for invalid recipients.
550	Permanent failure at the recipient level. This usually means that the mailbox does not exist. ESA uses this response code when a recipient address is invalid.
554	Permanent failure at the service or connection level. ESA uses this to reject SMTP connections outright, in place of a 220 banner, for low-reputation or explicitly listed hosts.

Line 2 of Example 1-1 is the first command issued by the client, HELO, and line 3 is the server response. In SMTP, a line is a string of characters terminated by a carriage return (ASCII hex character 0x0D) immediately followed by a linefeed (character 0x0A). This line termination sequence is usually indicated as <CRLF>. The "hello" is literally an introduction, and it allows the client to identify itself to the server. In practice, the use of HELO is deprecated in favor of EHLO, but the distinction isn't important in this simple example. The string that follows HELO is arbitrary, but the RFCs encourage the use of the sender's Internet domain. This is where we first encounter one of the serious limitations of SMTP: the lack of authentication. SMTP dates from a time when servers could trust clients and vice versa. There is no reliable mechanism by which we can verify that any of the information provided by the client is accurate. The HELO string can easily contain almost any value up to the maximum string length supported by the server, typically 1024 bytes. We can't trust the value contained here to always be valid or invalid and, so, it's unwise to make any filtering decisions on the basis of it. HELO/EHLO strings also come up when we discuss Transport Layer Security (TLS) for secure email delivery and Sender Policy Framework (SPF) for sender authentication.

Line 4 is the MAIL command, which tells the server that the client is beginning a new message, and it specifies the sender address. Email address literals should always be surrounded by angle brackets (< and >) and, on Internet clients and servers, should always include a fully qualified domain name (FQDN) after the @ sign. A blank sender address, specified as

```
MAIL FROM: <>
```

is valid, but is reserved for use by system messages, usually for Delivery Status Notification (DSN) messages or *bounces*. Line 5 is the response from the server. If this had been a temporary or permanent error (4yz or 5yz response code), the client would not be able to deliver this message. At that point, the client may disconnect with QUIT, use RSET to reset the connection, or simply start a new message with a new MAIL FROM command.

Line 6 begins the listing of recipients of this email message using the command RCPT TO. This example has only one recipient. Recipients specified in the SMTP RCPT command are usually referred to as envelope recipients, and they may or may not match the value in the To field you would see in your email client. This is addressed later when we discuss the difference between envelope and message body. Line 7 is the response from the server; this recipient is accepted. It's perfectly reasonable on a multirecipient message that some addresses may not be accepted. It's also reasonable for an SMTP server not to accept more than one recipient per message or to limit the number of recipients per message.

Line 8 is the DATA command, and line 9 is the 354 response. 3yz response codes translate as "OK so far, go ahead with transmit." It's not a guarantee that the server will accept the message, but up to this point, the transmission hasn't hit any reason to be rejected. After the 354 response, the sender transmits the message body line by line, which includes both the SMTP headers and what any user would identify as the "body" of the message. It would also include any attachments, which are just encoded portions of the body following the MIME specifications. (We discuss MIME in more detail later in this chapter.) This message has no attachments; it's just plain ASCII text. The message is terminated by the sequence <CRLF>.<CRLF>. That is the carriage return/linefeed combination, surrounding a period character. In hex, this combination is 0x0D0A2E0D0A. This is the only proper message-termination sequence, and sending SMTP clients must be sure never to transmit this sequence for anything but a message termination. Lines of a message body that contain nothing but a period character as typed by the sender need to be careful not to transmit it in the raw.

The blank line (line 11) indicates the break between message headers and message body. The message header is all the standard and custom headers attached to a message. This example has only one: the standard Subject header that will be displayed to the recipient in the MUA. As you can see, it's not required that the From or To headers be included, and in fact, even the Subject isn't strictly required.

Line 14 immediately follows the data termination sequence and includes the response code and text from the server. The 250 here indicates that the message was accepted.

The response provided by the server may or may not have information about the message. Often, the response is a simple variation on "message accepted, thank you," but many will systems will often return a local message identifier that can be used to trace the message as it passes from server to server. These identifiers are usually only locally significant. The ESA response in this example tells us that the message was accepted and was given the internal message ID (MID) of 31274. An administrator for this ESA could use that MID to trace it through that appliance.

SMTP Commands

Aside from the required minimal SMTP commands HELO (or EHLO), MAIL, RCPT, DATA, and QUIT, numerous other commands are defined by the RFCs. Unfortunately, many of these commands have proven to be useful to criminals in exploiting systems and stealing addresses. Because of this, the use of such commands as EXPN (Expand) or VRFY (Verify) has fallen out of favor, and ESA doesn't even implement them, although it may respond to them. The SMTP commands that the ESA responds to is listed in Table 1-3.

Table 1-3 *SMTP Commands the ESA Honors*

Command Format	Purpose
EHLO <domain>	Client greeting. Clients are required to provide an identifying domain or hostname, but this requirement is often ignored.
MAIL	Indicates the start of a new SMTP message, and the envelope sender address. Because the full command is MAIL FROM, we often refer to this address that way.
RCPT	Indicates a new envelope recipient of the message. The response from the server indicates whether that recipient is valid; a message needs at least one valid recipient, but the client can proceed even with one or more rejected recipients.
DATA	Client is ready to send message data, including both headers and body of the message. Message transmission must conclude with the SMTP termination string <CRLF>.<CRLF>.
RSET	Reset. Resets the current conversation back to a default state. All sender, recipient, and message data is discarded. Some products issue this command between messages when delivering more than one message per connection.
VRFY <email address>	Verify. Asks the receiving MTA if the email address of the recipient is valid on the destination system. Because this is easily exploited by dictionary attack to provide an address harvester with all legitimate email addresses on a system, it is not used on Internet-facing SMTP servers. The ESA does not implement this command and will respond with "252 ok" in all cases, regardless of the provided email address.

Command Format	Purpose
EXPN <email address>	Expand. For a known recipient email address, asks the receiving MTA whether this is a mailing list with multiple recipients, and if so, to return the expanded list of all members. The potential for abuse is staggering. The ESA does not implement this command and returns "500 command not recognized" in response to it.
QUIT	Terminates the SMTP conversation and closes the TCP connection.
NOOP	No operation. Self-explanatory, really. It takes no arguments and provides no function. ESA always returns "250 ok" in response. Its only purpose is as a keepalive to avoid having server timeout inactive sessions.
HELP	Help on commands. Because of security, AsyncOS strictly limits its response. There's little call for using the HELP command other than for troubleshooting, and even then, it's of marginal use.
AUTH	Provide credentials for SMTP AUTH, which, if authenticated, allows a sender to enjoy relay privileges through the ESA. Typical only in service provider and some education environments. Credentials are passed in the clear, which is a security concern unless combined with TLS.
STARTTLS	Instructs the server that the client wants to use TLS to transmit message data and to prepare for the TLS protocol negotiation. This command is only available if using ESMTP and if the ESA policy allows for the use of TLS.

ESMTP Service Extensions

When a client connects to a server and uses the EHLO greeting, in addition to identifying itself, it indicates to the server that it is ESMTP capable. Servers like the ESA then respond with a list of ESMTP *extensions* that it supports. Extensions are a way for the capabilities of SMTP clients and servers to be modified and improved without having to alter the protocol itself. Standard extensions must be registered with IANA. Vendor-specific extensions may be created, but these extensions must use names that start with X.

For example, an ESA responds to a client greeting with three standard extensions, like this:

```
EHLO cisco.com
250-mail.chrisporter.com
250-8BITMIME
250-SIZE 20971520
250 STARTTLS
```

The three extensions listed respectively tell the client that the system supports 8-bit characters in MIME messages, will accept messages up to 20 MB, and offers secure connection support using TLS. Another standard extension that ESA supports is AUTH for providing remote authentication of mail clients.

SMTP Message Headers and Body

All lines transmitted after the data command, up to the final termination sequence, can be considered the "body" of the SMTP message. The data is composed of two parts. A blank line separates the message headers and the message body proper, as we saw in line 11 of the simple SMTP example. The distinction between headers and body are important for MUAs, as the headers, with a couple of exceptions, are typically not displayed for the end user. Headers provide information about the message that's important to receiving and transmitting systems, including several that are required by RFC. RFC 5322 defines a number of standard headers, the most common of which are listed in Table 1-4.

Table 1-4 *Common RFC-Defined Headers*

Header	Purpose
From	The body sender address, distinguished from the envelope sender address. This header is for end user's benefit and is not typically examined by the ESA. Usually, the envelope sender and the body From header are the same email address, but can differ for mailing list or other automated software, for example. If this header is not present in a message received, the receiving MTA is required to create it, based on the envelope sender address. The value of the From field usually takes the form From: Firstname Lastname <user@domain.tld>, but many variants are possible.
To	The body recipient address. This "visible header" is typically displayed by the MUA and is not necessarily the same as the envelope recipients. The ESA can look at this header but, aside from some operations like masquerading, it's not normally used by the ESA other than to pass it on unchanged.
CC	Carbon copy. Indicate recipients that are not the primary recipient but are still visible in the headers of the message. Note that there is no equivalent "BCC" header; recipients that appear in the envelope but not in the To or CC are considered Blind Carbon Copied.
Subject	Obviously, the subject of the message as supplied by the sender, whether a human user or an automated system. The addition of RE: or Re: for replies and Fwd: or FW: for message forwards is purely a human and software custom, not a requirement for the Subject field.
Reply-To	A header that specifies where replies to this message should be sent. Although it can be a different address than that in the From header, it's usually the same for person-to-person messages. Messages and group email lists often make use of the Reply-To header so that replies go back to the list software and not the original author.

Header	Purpose
Received	Indicates when, how, and from where a message was received. RFCs require that all SMTP servers add a Received header to every message that it accepted, while preserving earlier Received headers. There are typically multiple copies of the Received header, and collectively, they tell the hop-by-hop history of this message. Received headers are used for message-loop determination and are extremely important for troubleshooting.
Date	A date and timestamp on a message, usually added by the sending mail client. If not present, an MTA, like the ESA, will add it.
Content-Type	Indicates the MIME type of data in the body of the message. This header can also provide additional parameter values, such as "charset" (Character Set) for text types and "boundary" for multipart types.
MIME-Version	Indicates the version of the MIME standard the message adheres to. As of this writing, only version 1.0 is defined.
Content-Transfer-Encoding	Originally, SMTP was only capable of transmitting data in 7-bit US ASCII with limited line lengths. Even today, support for 8-bit email cannot be relied on. 8-bit binary or character data must be encoded with schemes like Base64 or quoted-printable, and this header indicates which encoding was used. It can also specify that the data is unencoded, because it's already 7-bit clean or that it's unencoded 8-bit binary data.
Message-ID	Contains a single unique message ID, usually added by the originating software. If not present, an MTA like the ESA will add it. Note that although the Message-ID is recorded by the ESA, internally it uses a separate message ID called the MID to uniquely identify each message. The MID is locally significant and is not guaranteed to be unique across ESAs.

Custom headers not defined in RFCs can be added to any SMTP message. The name of these headers must start with *X-* and follow the RFC-defined Name: Value format and adhere to SMTP rules regarding header length and line wrapping. These are typically vendor-specific messages, although products like ESA allow organizations to add and remove headers to messages.

Message headers are extremely valuable for troubleshooting delivery and filtering issues. The standard headers provide information about the source and the intended destination and should be adequate to trace a message through multiple systems to determine an error point. They also provide enough information to the recipient so that their email client, or MUA, can compose replies and forwards.

Envelope Sender and Recipients

SMTP messages really have two sender addresses and two (or more) recipient addresses. Addresses specified during the SMTP conversation prior to DATA are the *envelope*

addresses. The sender address specified during the SMTP MAIL FROM command is called the *envelope sender* address. The recipients listed in one or more RCPT TO commands are considered the *envelope recipient* addresses. These addresses are used in routing messages to their destination or sending back notifications to the sender—not the visible To: and From: fields that you see in a mail client.

This is an important distinction, because when an ESA configuration setting, table, or filter refers to sender or recipient, it is almost always referring to the envelope. In fact, the ESA rarely ever examines or alters the visible To, From, CC, or Reply-To headers unless specifically set with a filter to do so.

Transmitting Binary Data

Normally, SMTP doesn't distinguish between plain-text or other human-readable parts of the message and binary attachments. SMTP also does not natively define messages with multiple parts, such as text in different formats. Binary data and multipart messages are handled through the Multipurpose Internet Mail Extensions (MIME) standard. MIME is also used in other protocols, like HTTP, to send binary data over a plain-text protocol. The vast majority of email messages sent will be in MIME format, even very simple messages. MIME is an extensive specification covered by numerous RFCs. It's not required to be familiar with all of it, but it is important to understand how binary data is transmitted and how the ESA deals with "bodies" and "attachments" of messages.

MIME begins with a MIME-Version header that must be present in order to be considered a MIME format message. MIME-Version, of course, specifies the version number of the MIME standard that the creator of the message used. The other important header is the Content-Type header, which specifies the overall global *MIME type* of the entire message, in a format of *type/subtype* that the MIME RFCs specify. If the message is composed of just one part, this header indicates its type. A *MIME part* refers to logically grouped data: the entire message text or the entire attached file. If the message is composed of multiple parts, the global type in Content-Type will be multipart/alternative, or multipart/mixed, as this one is. The Content-Type header can also include other information about the data in the message in the form of *parameter=value* pairs. The parameters will vary depending on the type. For text types, the character set is defined in the charset parameter. For multipart types, the MIME boundary is defined in the boundary parameter. This is the ASCII text string that acts as a delimiter between parts of the message. A boundary is present even if there is only a single part to a message. Typically, the MUA generates the boundary strings, and so the format varies considerably. It must be a string that is not found anywhere else in the message other than the boundaries.

An example MIME-formatted SMTP message is shown in Example 1-2. This is the 7-bit ASCII text representation of a message, with the Base64-encoded binary data shortened for convenience—this 37 K attachment has almost 700 lines.

Example 1-2 *Example MIME-Formatted SMTP Message*

```
Received: from unknown (HELO mailstore.chrisporter.com) ([10.60.10.20])
  by mail.chrisporter.com with ESMTP; 15 Jan 2011 16:22:30 -0500
Received: by mailstore.chrisporter.com (Postfix, from userid 1001)
      id 9D52697D7F; Sat, 15 Jan 2011 16:22:28 -0500 (EST)
Received: from localhost (localhost [127.0.0.1])
      by mailstore.chrisporter.com (Postfix) with ESMTP id C98BB97D7A
      for <chriport@cisco.com>; Sat, 15 Jan 2011 16:22:27 -0500 (EST)
Date: Sat, 15 Jan 2011 16:22:27 -0500 (EST)
From: Chris Porter <cporter@chrisporter.com>
To: chriport@cisco.com
Subject: Document you requested
Message-ID: <alpine.DEB.1.10.1101151621410.24513@mailstore.chrisporter.com>
User-Agent: Alpine 1.10 (DEB 962 2008-03-14)
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="1007293450-747258728-1295126547=:24513"

  This message is in MIME format.  The first part should be readable text,
  while the remaining parts are likely unreadable without MIME-aware tools.

--1007293450-747258728-1295126547=:24513
Content-Type: TEXT/PLAIN; format=flowed; charset=US-ASCII

Chris,
      Here's that document that you requested.

Thanks
Chris

--1007293450-747258728-1295126547=:24513
Content-Type: APPLICATION/ZIP; name=credt.xlsx
Content-Transfer-Encoding: BASE64
Content-ID: <alpine.DEB.1.10.1101151622270.24513@mailstore.chrisporter.com>
Content-Description:
Content-Disposition: attachment; filename=credt.xlsx

UEsDBBQABgAIAAAAIQDdsQorbwEAAMQEAAATAM0BW0NvbnRlbnRfVHlwZXNd
LnhtbCCiyQEooAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
.
.
.
q5itppkBAAAoAwAAEAAAAAAAAAAAAAAAAAAB9jwAAZG9jUHJvcHMvYXBwLnht
bFBLBQYAAAAACwALAMUCAABMkgAAAA=

--1007293450-747258728-1295126547=:24513--
```

By using multipart/mixed, we tell the receiving MUA that there is more than one type in the message. The type of each part is defined in its MIME header, listed just after the boundary string. Each part includes its own Content-Type and, possibly, other headers. It is possible that the MIME type of a part can also be multipart/mixed, with its own *subparts*, and in fact, the nesting can continue, creating a tree structure. This example has only two levels: the root with two nodes.

The receiving MUA is free to interpret the MIME parts as it sees appropriate, but most will treat the first text/plain or text/html part (or both, if combined into a multipart/alternative) as the message body and all remaining parts as attachments. Usually, the MUA displays to the user a filename and download options for binary attachments, but it's possible to take other actions. Some email clients, upon receiving encrypted message parts, automatically decrypt parts that are encrypted with the recipient's public key (for example, assuming that the private key is available).

Note Multipart/alternative is a special multipart type that is extremely common. The root part has two or more subparts, but instead of being independent parts, the subparts all display the same information in different representations. The recipient's MUA is free to pick the representation that best fits its display. The most common multipart/alternative messages have the same message content in two parts: one in text/plain and the other in text/html. The email client picks the best version for display to the end user and ignores the other. Two different email clients might display different parts; for example, a client on a mobile device may choose the plain-text part while a desktop OS client displays the HTML.

This message has a Microsoft Excel worksheet file attached, with filename credt.xlsx. Despite that, the MIME type of this part is APPLICATION/ZIP, because the Microsoft Office formats use a compressed file type that the MUA identified as ZIP.

MIME Types

The MIME RFCs (RFCs 2045 through 2049) define not only the headers and organizational requirements of MIME messages, but also the global MIME type categories and standard subcategories, along with a procedure for defining and registering new subcategories. The five standard global categories of MIME types are listed in Table 1-5.

Table 1-5 *Top-Level Global MIME Categories*

Category	Description
`text/`	Plain-text data stored in 7-bit ASCII. text/plain refers to text data with no formatting whatsoever. Other text types include text/html and text/richtext. The most common types are text/plain and text/html.
`image/`	Any data stored in an image format that can be displayed on a monitor. Common types include image/jpeg and image/gif. Some MUAs display these images natively if they understand the image type. Text mail clients can safely ignore these types.

Category	Description
`video/`	Video data, covering multimedia formats that usually include both moving image and audio soundtrack data. The most common type is video/mpeg, but a type exists for almost every common format. Some MUAs can natively display multimedia, but most will only offer a way to save the data to a file on disk.
`audio/`	Audio-only data like music or voice recordings. Common formats are audio/mpeg for MP3 and audio/wav. Usually, the mail client launches an external player to handle these types.
`application/`	Any other kind of binary data that doesn't fit into the other categories. These types are typically associated with a software application, and so they are numerous. There are hundreds of different application/ types in use today. The special type application/octet-stream is used for unknown formats of binary data.

The last important MIME topic is the way that binary data is represented in the plain text SMTP protocol. In our example, the binary attachment has a header called Content-Transfer-Encoding with a value of Base64. This specifies that the binary data are represented by a set of 64 ASCII characters. *Base64* encoding is defined in RFC 2045, as is another binary-to-text encoding called *quoted-printable*. The ESA handles all these encodings transparently, and all filtering is performed on unencoded content; you don't need to convert search strings into Base64 format. ESA can even examine text content stored within binary attachments and can open compressed formats to access the files contained therein.

Note Because the Base64 algorithm encodes every three binary bytes as four text characters, they expand the data storage and transfer requirements by a third. This means that if you send a 1 MB (1024 KB) attachment in email, its size in SMTP transmission is actually about 1365 KB, plus more for MIME headers, padding, and line breaks. If you want your environment to accept 5 MB attachments, you have to set the ESA limits to 6.7 MB (6827 KB) or higher to allow for the expansion that Base64 causes.

Character Sets

SMTP was originally defined only for the 7-bit US-ASCII character set, convenient only for representing the English language. Other languages that use a Latin character set, but with diacritics like accents or tilde, can be represented, but suffer some information loss. Languages that use non-Latin characters, like those written in Greek or Cyrillic alphabets, or those that use thousands of characters in their written form, like Chinese or Korean, have to use some other format to be transmitted over SMTP. For this reason, MIME supports the transmission of data in arbitrary character sets, such as Big 5, or the ISO-8859- series, or more importantly, Unicode encodings, like UTF-8 and UTF-16. The ESA can accept and deliver messages in any character set.

The character set used by text MIME types is specified in the Content-Type header with the charset parameter. For example, 7-bit clean-text message body might use

```
Content-Type: text/plain; charset=US-ASCII
```

Where an HTML format message in Unicode might use

```
Content-Type: text/html; charset=utf-8
```

If the content is encoded with a scheme like Base64, the Content-Type character set refers to the character set of the original data, and that character set should be used for the decoded data. The MIME-encoding schemes all use US-ASCII for their encoded representation, regardless of the original source character set.

Domain Name Service (DNS) and DNS MX Records in IPv4 and IPv6

SMTP is intimately connected with the Internet Domain Name System (DNS). DNS Mail Exchanger (MX) records replaced earlier MD and MF records in RFC 973 in January 1986, some time after SMTP had been in general use. Early SMTP implementations relied on HOSTS.TXT instead of DNS.

To deliver Internet mail, any SMTP client must first determine the domain of each of the recipients of the message it is attempting to deliver. This is normally the portion of the email address after the @ sign. RFC 5321 states, unambiguously, the process for SMTP clients to follow when determining the destination host. The client must first perform a DNS MX lookup on the domain. If no MX record is found, the client must lookup an A record for the domain, and if present, treat it as it would an MX record.

An MX record is a domain-level resource record. An MX record for a domain supplies three key pieces of information: first, the Time to Live (TTL) that's common for all DNS resource records; second, one or more hostnames of servers capable of receiving email via SMTP for that domain; third, the numerical priority of each host, alternately described as MX preference, weight, or cost, of each host.

Once the list of hosts in the MX record is found, the client selects the lowest-numbered host (the lowest cost) and performs an A lookup to determine the correct destination IP. The client then makes a connection attempt on port 25 to that destination IP. Some clients, like ESA, look up all the A records for all MXs at once for efficient lookups. If there is more than one equal-cost MX, all of them are looked up and the client is free to select any at random. When a destination IP is unavailable, the client must attempt all of the equal cost hosts, before moving to the next-lowest-cost host or hosts. The process continues until the first IP that accepts connections on destination port 25 is found.

In IPv6, MX records haven't changed at all, other than the fact that the hosts they list may have both A and AAAA records, or just AAAA records. As of this writing, the number of domains publishing MX and hosts with AAAA records is fairly low, and the number of sites accepting SMTP over IPv6 is likewise low, but growing.

Message Transfer Agents (MTA)

We've used the acronym MTA repeatedly, so before we go much further, we now describe MTAs and the functions they provide. Because the ESA is, at its heart, a purpose-built MTA, it's important to know what they are and are not.

Here are the chief responsibilities of an MTA for Internet email at an organization:

- **Accepting Internet mail for local domains:** The MTA accepts connections from Internet senders for email being sent to local recipients. MTAs should not accept email for recipients in domains other than those of the local organization. Accepting mail for non-local domains means the MTA is an *open relay*, meaning that it allows any client to send email to any domain. Because this allows malicious clients to use your servers to deliver their email, your organization becomes a spam and virus source by proxy. If your servers are accidentally configured as open relays, their IP addresses will be added to the blacklists that track sources of junk email.
- **Verifying local recipient addresses:** MTAs should verify that a recipient email address exists before accepting it. This isn't required, but is strongly recommended, both for the health of your own environment and for being a better Internet citizen.
- **Queuing mail and retrying messages:** Although not strictly required, most MTAs are store-and-forward systems that operate asynchronously. That is, the transmission from client to MTA, and MTA to destination, are performed independently, at separate times and over different connections. In between acceptance and delivery, MTAs are expected to store their messages, and in the event of an unavailable destination or an error in transmissions, retry periodically until the message is delivered.
- **Directing recipients to an appropriate destination:** Local recipient mailboxes may reside across a number of different local or remote servers. MTAs typically provide features to identify recipients, look up the appropriate destination in a table or in a directory, and deliver the mail there.
- **Delivering mail from local users to the Internet:** Individual email clients and mailstores like Microsoft Exchange or Lotus Notes send all nonlocal traffic to an MTA for final delivery to an Internet destination. The MTA deals with all the vagaries of Internet email: connection reliability, DNS records, down hosts, and message queuing and retries.
- **Filtering messages:** Many MTA software packages provided basic filtering of messages, or plugin approaches to integrating external filtering, but the introduction of the ESA made junk filtering a primary task of gateway MTAs.

MTAs are one portion of email infrastructure that also typically includes a mail store or database (like Microsoft Exchange or IMAP servers) and MUAs, like Microsoft Outlook, Lotus Notes, Mozilla Thunderbird, or web-based mail clients. MTAs do not duplicate the functions of these products, and so do not provide retrieval, composition, or archiving of messages. MTAs do not typically store messages—only keeping them in memory or on disk until delivered.

It is possible to have a single server running MTA, MUA, and mailstore, although the combination of mailstore and MTA is the most common. In fact, this dual role is likely the most common experience with MTAs that administrators have.

From the basic of MTA functionality, various opensource and commercial products have expanded the scope of capabilities, transforming the traffic cop directing a few email messages an hour into a gateway security device responsible for tens or hundreds of thousands per hour.

Abuse of SMTP

Abusing the SMTP protocol is almost as old as the protocol itself. The latest RFC, 5321, deprecates certain SMTP features, or allows servers to ignore them, because of their use in exploits. SMTP was built in an era of trust and the hosts connected to the early Internet were known organizations and were trustworthy. Because SMTP requires no authentication, it's easy to forge information, and there are a number of threats related to specific SMTP tricks that the ESA can protect against.

Relaying Mail and Open Relays

Once upon a time, an SMTP server owner could trust that clients that connected would only attempt to deliver messages to recipients that were local to the server's organization. This makes perfect sense, because a client should only be aware of a given SMTP server through MX record lookup for the target organization. As a result, many SMTP servers were configured to relay mail for all destinations so that if recipients were somehow mistakenly sent to a particular server it could forward to the right destination.

This nice feature was, of course, one of the first exploits that spammers found, allowing them to deliver messages to unsuspecting legitimate servers that dutifully forwarded their junk on to a third party. Such hosts are known as open relays if they relay for all destinations. Today, all SMTP server solutions prevent relaying except by strict permission. ESA goes one step further, identifying repeated relay attempts as a sign of junk mail sources, and rate limiting or dropping these connections.

Relaying still exists today on the Internet, because many SMTP servers are configured to allow relaying by authorized clients that can successfully authenticate by using the AUTH command. This is convenient for remote users, but by having a single factor of authentication, is dangerous when publicly available. Malicious senders exert a lot of effort to steal email account credentials from users, either with social-engineering email messages, fake *password reset* websites, or keystroke loggers. Stolen credentials can be used to get a relaying "free pass" for a spam sender. Another common relay vector is web-based email open to the public Internet, protected only by username and password. Once compromised, local user accounts are used to send more junk email, using your servers and ruining their good Internet reputation.

Bounces, Bounce Storms, and Misdirected Bounces

Another standard behavior of messaging systems that assumed trustworthy senders was the Delivery Status Notification, or a *bounce* message. It's also often called a Non-Delivery Record (NDR). In technical terms, a bounce is just another SMTP message, composed by a system at some point in the delivery path back to the original envelope sender. The message usually contains information about the original message, and any information it has about why the delivery failed. Formats vary; some systems send back a copy of the entire original message and others create generic bounces with no information about the original, not even the recipient address that failed.

What happens if the sender address is forged? Well, any receiving system that accepts the message, only to have it bounce later, will send it back to the forged address. The bounce message may contain content from the original, including any malicious URLs, making this a potential threat vector for your organization. Even if the bounces don't contain malicious content, they can be confusing and aggravating to users. Additionally, although a few simple bounce messages may not be much of a threat, en masse, they represent a significant problem. If thousands of servers across the Internet each send thousands of bounces per hour to your organization, the effect is a distributed denial of service (DDoS) that can render all of your MTAs too overloaded to handle legitimate mail.

Some of these storms are intentional. If an attacker composes thousands of messages with your email address as sender and sends it to known-invalid recipients at SMTP servers that exhibit this accept first, bounce later behavior, you will get thousands of bounce messages. Figure 1-1 illustrates this problem.

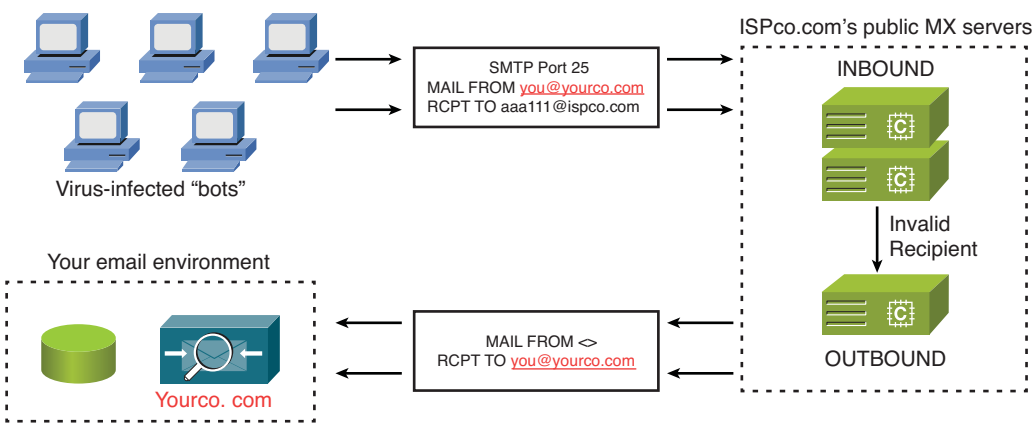


Figure 1-1 *Misdirected Bounce or Bounce Storm Problem*

The ESA provides a dedicated Bounce Verification (BV) feature to protect against this kind of attack. BV marks up messages sent by your organization, so that if they bounce, it can be recognized as having originally coming from your environment. Bounce messages lacking the markup can be discarded quickly and safely.

The other side of the problem is that you don't want to be a source of these bounces, either. The ideal solution to avoid being a source of storms is to never accept recipients that will ultimately not be delivered. This can be done by checking a table or (preferably) a directory of valid recipient addresses at SMTP connection time and providing a response code back to the sender indicating success or failure. ESA provides this through the static Recipient Access Table, via LDAP Accept queries, or a combination of both.

Directory Harvest Attacks

The directory checking and SMTP conversation-time rejection or acceptance of recipients leads to yet another problem. A system that reliably reports a recipient as valid or not can be repeatedly checked for all possible recipients. The search space of alphanumeric characters is small enough that it can be brute-force attempted in hours or minutes. From this, the attacker now has a thorough list of all the valid addresses at a given domain, and these addresses can be sold or used in spam campaigns. Example 1-3 demonstrates an SMTP harvest attack conversation.

Example 1-3 *Directory Harvest Attack*

```
220 esa02.cisco.com ESMTP
HELO external-sender.com
250 esa02.cisco.com
MAIL FROM: <sender@external-sender.com>
250 sender <sender@external-sender.com> ok
RCPT TO: chris@cisco.com
550 #5.1.0 Address rejected.
RCPT TO: cporter@cisco.com
550 #5.1.0 Address rejected.
RCPT TO: chrisporter@cisco.com
550 #5.1.0 Address rejected.
RCPT TO: chriport@cisco.com
250 recipient <chriport@cisco.com> ok
RCPT TO: chrisp@cisco.com
550 #5.1.0 Address rejected.
```

The ESA provides thorough Directory Harvest Attack Prevention (DHAP) through Mail Flow Policies. Each policy dictates the maximum number of invalid recipients per hour that are considered acceptable. Any sender that exceeds the maximum number is disconnected and cannot reconnect for a full hour, making brute-force harvests impossible. By setting the value appropriately, depending on the policy and the size of the environment, we can allow legitimate senders to be quickly and accurately notified of undeliverable mail while identifying and stopping harvesters. Example 1-4 shows an example of DHAP in use.

Example 1-4 *Directory Harvest Attack Prevention*

```
RCPT TO: <aaron@cisco.com>
550 #5.1.0 Address rejected.
RCPT TO: <abby@cisco.com>
550 #5.1.0 Address rejected.
RCPT TO: <adam@cisco.com>
550 #5.1.0 Address rejected.
RCPT TO: <alan@cisco.com>
250 recipient <alan@cisco.com> ok
RCPT TO: <alfred@cisco.com>
550 #5.1.0 Address rejected.
550 Too many invalid recipients
Connection closed by foreign host.
```

Summary

As we've seen, email security is a multifaceted problem that encompasses more than just spam and virus filtering. Cisco's ESA is designed to solve a wide variety of email delivery, reliability, and security issues, with a feature set that has evolved continually since the product's launch in 2003.

At the heart of email security and the ESA feature set is SMTP, the protocol underpinning all Internet email. SMTP is a scalable, sturdy protocol that has served its purpose well for more than 30 years, but has limitations that are important to understand. We discussed the critical parts of SMTP, DNS, and Internet email message formats.

This page intentionally left blank

Chapter 2

ESA Product Basics

In this chapter, you will learn the following:

- Models of appliances and their hardware features
- Running the basic system setup wizard, and the resulting configuration
- Basic networking options and recommended configurations
- Configuring and testing network access for product updates
- Configuring the security filtering features

Hardware Overview

The Cisco IronPort ESAs are 1U and 2U hardware appliances in a 19-inch rack-mount form factor. Because they're based on standard Intel platform computer server hardware, they fit in most any standard data center 19-inch racks. The appliances are supplied with sliding rails for rack mounting. Table 2-1 lists the various Email Security Appliance (ESA) models that are shipping and supported as of this writing. Earlier models included the C30, C300, C350, C60, C600, C650, X1000, and X1050 models, which are no longer available for sale.

Table 2-1 *ESA Models and Characteristics*

Platform	Form Factor	CPU and RAM	Mail Queue	RAID	Ethernet
C160	1U	One dual-core 2.4 Ghz, 4 GB RAM	10 GB	Software RAID 1 (Mirror)	Two 10/100/1000 copper
C170	1U	One dual-core 2.8 Ghz, 4 GB RAM	10 GB	Software RAID 1 (Mirror)	Two 10/100/1000 copper
C360	2U	One quad-core 2.3 Ghz, 4 GB RAM	35 GB	RAID 1 (Mirror)	Three 10/100/1000 copper
C370	2U	One quad-core 2 Ghz, 4 GB RAM	35 GB	RAID 1 (Mirror)	Three 10/100/1000 copper
C660	2U	Two quad-core 2.3 Ghz, 4 GB RAM	70 GB	RAID 1+0 (Mirrored and striped)	Three 10/100/1000 copper
C670	2U	Two quad-core 2 Ghz, 4 GB RAM	70 GB	RAID 1+0 (Mirrored and striped)	Three 10/100/1000 copper
X1060	2U	Two quad-core 2.8 Ghz, 4 GB RAM	70 GB	RAID 1+0 (Mirrored and striped)	Three 10/100/1000 copper
X1070	2U	Two quad-core 2.66 Ghz, 4 GB RAM	70 GB	RAID 1+0 (Mirrored and striped)	Three 10/100/1000 copper

2U Enterprise Models

As of this writing, the current 2U enterprise models are the C370, C670, and X1070 ESAs. All have redundant, hot-swappable power supplies, one or two quad core CPUs, 4 GB RAM, and hardware RAID, either with two drives mirrored (RAID 1) or four or six drives mirrored and striped (RAID 1+0, or RAID 10). All the disks are hot-swappable. All models have dual hot-swap replaceable AC power supplies. There are two 10/100/1000 copper network interface cards (NIC) labeled *Data 1* and *Data 2* that can be paired, and a third 10/100/1000 NIC labeled *Management*. Despite the labels, the Ethernet interfaces can be used for any supported services. Fiber-optic interfaces are available as an optional component. All models have a 9-pin serial port for console access and ship with a null modem cable.

1U Enterprise Models

The C170 is the smallest ESA and is a 1U 19-inch rack-mount appliance. It differs considerably from the larger appliances. It has a single dual-core CPU, 4 GB RAM, dual SATA drives in a software RAID 1 configuration, and two 10/100/1000 copper NICs, labeled *Data 1* and *Data 2*. It does not support NIC pairing, and it has a single AC power supply. It also provides a 9-pin serial port for console access.

Selecting a Model

The primary difference between the ESA models is performance, usually measured in terms of throughput: how many messages can be accepted, filtered, and delivered by the system in a given unit of time. Throughput is usually specified in messages per hour or messages per second, at a given average message size. Performance depends heavily on the configuration of the system, especially on the choice of security filters being run. In general, the units with more CPU cores, or faster clock CPUs, have higher throughput, as expected. In some configurations, however, the additional cores won't be used.

Another practical concern is the need for hardware redundancy. Because of hardware RAID, hot swap disk, and redundant hot-swap power supplies, the 2U models have higher Mean Time Between Failure (MTBF). Replacing drives or power supplies on the 1U models requires powering down the appliances and opening the chassis.

The 2U models also support NIC pairing, where the two physical jacks on the motherboard share the same MAC and IP address. One of the jacks is denoted as primary and is used for all network traffic until a failure occurs, and traffic is switched to the backup NIC. Cable, switch port, NIC failures, or any other problem with the physical connection triggers a failover to the backup port.

Basic Setup via the WUI System Setup Wizard

All ESA models ship in a *factory default* configuration with a single network interface preconfigured. None of the filtering engines, reporting, or even the basic Simple Mail Transfer Protocol (SMTP), is running. Logging and administrative subsystems are running at their default settings. No user accounts are available other than the superuser, *admin*. Before you do anything with the ESA, perform at least some basic setup. In this chapter, we cover the most common means of setup, using the web-based user interface (WUI) and its System Setup Wizard. Even if you're an expert, I recommend using the wizard—it's too easy to forget something doing it manually. Even after nearly 8 years installing ESAs, I start with the wizard.

Connecting to the ESA for the First Time

When you unbox the ESA, you find, along with the appliance itself, the usual accessories, like an Ethernet cable, AC power cords, and a 9-pin *null modem* cable for connecting to the ESA serial console. The Ethernet and AC power cables are just generic cables,