Global Information Assurance Certification Paper

A Robust Email Infrastructure using Sendmail 8.12

Alan Ptak
SANS Security Essentials GSEC
Practical Assignment (version 1.3)
6 February 2002

Summary

This document provides an overview of sendmail from a security perspective with a case study to show how sendmail and sound network security practices can be combined to create a robust scalable electronic mail infrastructure. Sendmail has matured and evolved into a robust security-aware message transport agent. Sendmail 8.9 and later versions include security features that can be adapted to a wide range of needs and situations. System security can be assured to a high degree by careful attention to file and directory ownership and modes. Numerous configuration options allow for extensive tuning of security, performance and behavior.

Sendmail 8.12 includes several new security options that help avoid local exploits by eliminating the need to have sendmail to run as a *set-user-ID root* program. Sendmail 8.12 provides a single sendmail binary that acts differently depending on its operating mode and supplied arguments. Sendmail runs as root to bind to port 25, to call the local delivery agent and to read .forward files. To write messages submitted via the command line to the queue directory, sendmail runs as a set-group-ID (SGID) program. A new group is created for the exclusive use of sendmail and the queue directory is group-writable for this group only. When used in conjunction with other security features and configuration guidelines, sendmail 8.12 can be a trusted service for enterprise electronic mail systems.

Security Concerns for Electronic Mail Systems

The key security issue is the concern that an attacker can use the mail system to gain unauthorized access, permissions or privileges by any means. Gaining root permissions is of course the worst case, however lesser degrees of access can be serious if, for instance, sensitive corporate or personal data is not sufficiently protected. Other security goals are to survive denial of service attacks by gracefully degrading service in the presence of increasing traffic to the server, minimizing unsolicited mail and especially messages containing viruses, and minimizing information leakage regarding the system, the network and the users.

The bad news is that security goals trade off poorly against functionality. The key to dealing with this difficult problem is to view security as risk management [Viega]. Intelligent choices can only be made for the security versus functionality tradeoff when the context within which these tradeoffs occur is well understood. Many factors will influence these decisions, including cost, schedule, reliability and external security compliance requirements.

Developers, system architects and administrators would do well to adhere to the principle of least privilege. This principle dictates that things (users, programs and processes, files and directories) should be given the minimum permissions that are needed to operate. This can be difficult to apply strictly in practice. Allowing users to read the aliases file to allow users to see what aliases are available for their use may be considered acceptable in may environments. On the other hand, system administrators that overly relax read permissions to system files and directories (e.g., /etc)

to avoid having to *su root* to read and modify configuration files unwittingly ease the task for intruders intent on attacking their system.

Hard and fast security recommendations that apply in all situations are difficult to make. The best advice may be to carefully understand and evaluate the tradeoffs involved and to make informed decisions about them. Involve the right people and processes to improve the quality of the tradeoff decisions. Include security requirements as early as possible to increase the likelihood of success and reduce the chance of unpleasant surprises later.

An Overview of Sendmail

Sendmail is a flexible and highly configurable internetwork mail routing facility that has greatly matured over time and benefited from its wide adoption and large user community. Sendmail is available as a freeware distribution from the Sendmail Consortium [SmlCons] and as a commercial product from Sendmail, Inc. [SmlInc1]. The standard reference for configuring sendmail is still *sendmail, Second Edition*, [Costales], although many significant additions and improvements have been added since it's writing in 1997. The Usenet news group comp.mail.sendmail is a popular source of information and advice on all aspects of sendmail.

Electronic mail is handled by two distinct and separate facilities: the Mail User Agent (MUA) and the Mail Transport Agent (MTA) [Costales]. The MUA is an application programs that people use to read, compose and send messages. Examples on UNIX are mh, elm, pine, zmail, mutt and dozens of others. In the Windows-dominated commercial world, Outlook, Lotus Notes and Eudora are common. These feature-rich applications are principally involved in managing, sending and receiving messages where the user is either the originator or the recipient of the message. In contract, users do not directly interact with the MTA, which transports messages and delivers them to recipients' mailboxes. Sendmail is one of the oldest MTA's and is widely used on a variety of platforms.

Sendmail is a collection of programs, files and utilities organized into three main parts: the *configuration file,* the *queue directory* and the *aliases file*. In sendmail 8.12, the traditional sendmail MTA has been further split into separate processes: the sendmail MTA and the sendmail delivery agent, each having its own configuration file.

The sendmail distribution includes generic scripts and utilities that are used to generate the necessary configuration files. The sendmail configuration is typically specified in the m4 scripts sendmail.mc and submit.mc, which are processed to create the configuration files sendmail.cf and submit.cf respectively. The sendmail program reads these configuration files when it is first invoked and whenever it receives the SIGHUP signal.

Queue directories hold messages until they can be delivered to recipients or destination mail exchangers. Mail is typically queued when the destination server is temporarily unavailable or busy. A message may be queued several times as it gets passed from one server to another on busy systems.

The aliases file is essentially a powerful name substitution facility. Aliases allow users to use several different email addresses to receive mail. An alias can be used so that a recruiter can receive mail addressed to jobs@mycompany.com without having to put her personal email address on the corporate web site. Aliases allow mailing lists to be created to distribute messages to a group of users (e.g., to several recruiters). Aliases can also pass messages to other programs for

additional processing (e.g., to page the system administrator when a message with a particular subject line is received) in addition to having it delivered to the user.

Sendmail Security

Attacks exist against sendmail configuration files, alias files, mail queues and against the sendmail programs themselves (e.g., [Zalesky]). Sendmail configuration files are obvious targets for attackers, since they contain parameters that critically affect the program's behavior. Alias files that contain arbitrary links to other programs can be easily exploited, by manipulating the alias file itself or the programs which may be invoked by aliases. Queue directories with unnecessary read and write permissions allow unauthorized access to user's mail, violating security policies and possibly privacy legislation and allowing traffic analysis attacks. In many cases, knowing that particular user accounts are highly active can be a significant source of information leakage even when the content of the messages is not revealed. Sendmail versions 8.9 and later offer vastly enhanced security, primarily through the use of the principle of least privilege by sendmail developers.

The queue directories, the /etc directory and the sendmail configuration files are the most sensitive system files and directories because of their impact on overall system integrity. The ownership and modes (read/write/execute permissions) of these files critically affect security. Sendmail allows 3 special users (DefaultUser, TrustedUser, and RunAsUser) to allow greater security tuning where necessary. Sendmail defaults to reasonable values when some or none of the users are defined on the system. The alias files should be routinely checked for default, unnecessary and bogus entries. Alias files should never contain programs that do not validate their input, such as uudecode and procmail, since they can be used to write arbitrary files. The aliases file should only be writable by root, and it should be in a protected directory.

The general rules for the most sensitive files and directories are outlined below and in the next section:

- /, /var and /var/spool should be owned by root with mode 755
- /etc and /etc/mail should be owned by root with mode 755
- /etc/mail/aliases should be owned by root with mode 644 or 600 (as discussed above)

An often-overlooked vulnerability is allowing users read access to system logs. Ideally, systems are configured to log events to a remote syslog server on a different network. This is easily accomplished, greatly simplifies monitoring and management of distributed systems and protects information that is essential in identifying intrusions and anomalous behavior.

Non-SUID root sendmail 8.12

Security conscious system architects and administrators have several new security-related options in Sendmail 8.12 that eliminate the need for sendmail to run as a SUID program. The goal is to eliminate the need to have the sendmail program to be set-user-ID root, to avoid local exploits. Sendmail needs to run as *root* for several reasons:

- Bind to port 25 to accept incoming SMTP connections
- Call the local mail delivery agent (LDA) as root to deliver local mail messages. By this arrangement, the LDA doesn’t need to be a SUID root program either.

- Read .forward files for users for special processing of user's mail.
- Write mail messages submitted at the command line to the queue directory.

A sendmail daemon started by root can perform the first 3 requirements. A separate set-group-ID mail submission program (MSP) submits messages to a group-writable queue directory to address the last requirement. Sendmail 8.12 addresses these requirements by providing a single sendmail binary that acts differently depending on its operating mode and supplied options. Separate configuration files are provided for each mode: sendmail.cf for the "normal" mode and submit.cf for the MSP mode.

Sendmail chooses the appropriate configuration file and operation mode based on the arguments passed to the program when it is invoked. Sendmail can be forced to use the sendmail.cf or submit.cf file with the –`Am` or –`Ac` arguments respectively. Otherwise, the submit.cf file is used only when sendmail is invoked in one of three modes:

- Deliver mail in the usual way (`-bm` flag)
- Use the SMTP protocol on standard input and output (`-bs` flag)
- Read messages for recipients (`-t` flag).

In all other modes, the sendmail.cf file is used. Sendmail is typically started in a script in /etc/init.d/rc2.d that invokes 2 instances of the sendmail daemon:

```
/usr/lib/sendmail –L sm-mta –bd –q30m
/usr/lib/sendmail –L sm-msp-queue –Ac –q15m
```

The sendmail binary is owned by root and the group *smmsp,* and the binary is set-group-ID (indicated by the 's' flag in the middle octet of the file permissions). The traditional mqueue directory remains essentially unchanged from previous releases, and is only accessible by root. A new group *smmsp* is created for the exclusive use of sendmail when in MSP mode. The MSP queue directory is owned by *smmsp.* This directory is group-writable for the user and group *smmsp* with no access privilege by any other user. The resulting directory permissions are summarized as follows:

```
drwxrwx---    smmsp  smmsp  ... /var/spool/clientmqueue
drwx------    root   wheel  ... /var/spool/mqueue
-r-xr-sr-x    root   smmsp  ... /usr/lib/sendmail
-r--r--r--    root   wheel  ... /etc/mail/sendmail.cf
-r--r--r--    root   wheel  ... /etc/mail/submit.cf
```

The *smmsp* and *wheel* groups do not exist by default on Solaris 7. These groups are added by including the following lines in the /etc/group file:

```
smmsp::25:
wheel::100:root
```

A case study: Widgets, Inc.

A case study is presented to illustrate several common elements that are present in many corporate electronic mail infrastructures. The discussion centers on security-related issues at the

network, server and sendmail configuration level for the fictitious Widgets, Inc. Several security-related issues for the network and SMTP server are discussed.

The operations of Widgets, Inc. are centered at their corporate headquarters with several branch offices and a number of mobile employees. The company uses electronic mail as its principal means of business and technical correspondence. Different Internet Service Providers (ISPs) provide Internet connections for each office location. Virtual private networks (VPN's) link branch offices with the company headquarters. Mobile employees use a commercial Internet service for the home offices and for dial-up access, and also access the headquarters over a VPN.

Figure 1 shows the system architecture. The electronic mail infrastructure is located in the company's headquarters. A SMTP gateway server in the DMZ at the company HQ receives all incoming mail. The SMTP gateway forwards mail to the internal mail server via SMTP, where it is written to the mail spool directory on a local disk for the user. Users retrieve their mail from the internal server via POP.

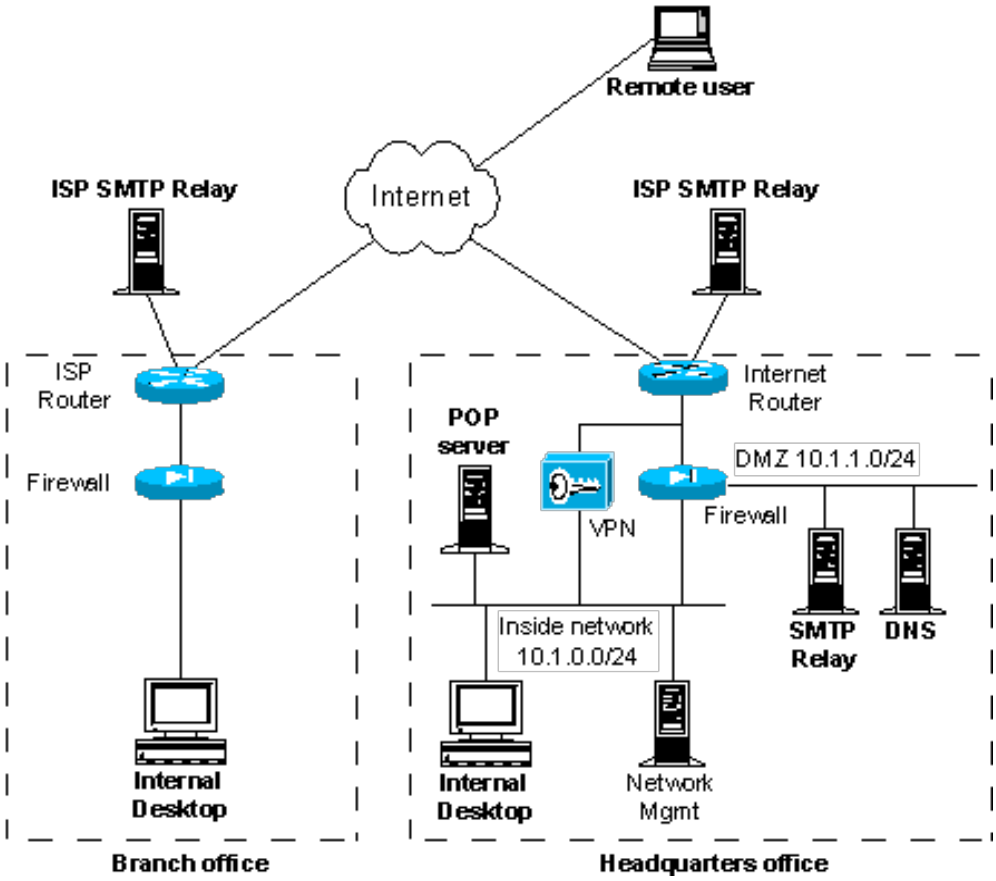

Figure 1: System architecture

Whereas the SMTP gateway is listed as the mail exchanger (MX) for all of the company's domains, the internal mail server is known and accessible only by the SMTP gateway. Using a defense-in-depth strategy, access lists on the Internet router and on the firewall are constructed to restrict outside network access to the internal mail server. The VPN concentrator is used only to terminate VPN connections from remote users without allowing any other traffic to bypass the firewall. Users at the HQ location have direct access to the Mail server on the local area network to retrieve their mail. Remote users and users at the company's branch offices have access to the Mail server (and other corporate Internet and file servers not described here) via the VPN.

Several other standard security practices provide additional layers of protection. The YASSP package (*Yet Another Solaris Security Package*) is installed to limit the security exposure [Chouanard]. TCP Wrappers [Wietse] on the SMTP and Mail servers restricts unauthorized remote connections. System events are logged using the syslog service running on the Network Management workstation on the Inside network. Tripwire supports configuration management and a rudimentary form of intrusion detection by periodically scanning the filesystem and reporting changes since the last scan [Tripwire].

Sendmail configuration for the SMTP Gateway

The SMTP gateway is the publicly accessible mail exchanger (MX) for all of the company's domains. SMTP gateways are severely restricted servers that can be very well secured when configured as bastion hosts. The SMTP gateway has no user accounts, does not allow user login or per-user forwarding, and does not perform any local delivery of mail. Sendmail is configured to simply accept SMTP connections from any source and to make SMTP connections to specific servers within the organization's protected network. Since there is no local delivery of mail, sendmail doesn't even need to run SGID.

Sendmail configuration for the POP/SMTP Internal Mail Server

The internal mail server is configured to allow Post Office Protocol (POP) and SMTP access while denying user logins. Common POP server packages, such as Qualcomm's qpopper authenticate against the system user accounts and passwords, however users are generally not given a login shell. Optionally, the system may be configured with no support for .forward files as an additional security measure, since users have no way to get at their home accounts. Given that users don't access sendmail directly, sendmail doesn't need to run as root after it binds to port 25 at start-up. The 3 special sendmail users previously mentioned can be used to provide this behavior.

Network security policy and firewall configuration

The headquarters firewall uses a Cisco Pix with three Ethernet interfaces:

```
nameif ethernet0 outside security0
nameif ethernet1 inside  security100
nameif ethernet2 dmz     security50
```

The security level for the interface is defined by the "`security`*`nnn`*" parameter in the firewall configuration file. Level 0 represents the lowest level of trust and 100 the highest. The DMZ is assigned level 50 so that network traffic between it and the inside and outside networks can be controlled by the firewall. Outbound access refers to any traffic from a higher security interface to a lower one. Two firewall policies are required for outbound traffic: a translation method and, optionally, an access control list that allows traffic to pass to the specific IP addresses, ports and protocols required.

Internal hosts and networks are assigned non-routable addresses as defined by RFC1918 for address allocation for private internets [Rekhter]. The firewall translates internal addresses to the

assigned addresses for each location. The firewall supports both static and dynamic address translation methods. The `static` command establishes a two-way one-to-one mapping from internal to external addresses across any two specific interfaces on the firewall. Servers require static addresses, whereas typical users that only initiate connects to servers can use dynamic addresses. Dynamic translations are configured using the `nat/global` commands. [CiscoNat] The `nat` command establishes a one-way translation for inside hosts to access outside networks that times-out after a period of inactivity. The time-out value is configurable and may vary from 10 minutes to 1 hour for TCP connections.

The Cisco Pix firewalls support access lists for inbound traffic only on a given interface, which focuses attention on egress traffic only. The default behavior allows all traffic to pass from high to low security zones. To support a "deny all" security policy, 3 access control lists are required for the headquarters firewall to override the default behavior:

The inbound access control list for the Outside interface (*acl_out*) determines what traffic may enter the HQ network from the Internet:

1. Allow SMTP (tcp, port 25) from the Outside to the SMTP server on the DMZ
2. Allow DNS (tcp, port 53) from the Outside to the DNS server on the DMZ
3. Allow DNS (udp, port 53) from the Outside to the DNS server on the DMZ
4. Deny anything else and log the access attempt

The inbound access control list for the DMZ interface (*acl_dmz*) determines what traffic may leave the DMZ for either the Internet or the Inside networks:

1. Allow DNS (TCP port 53) from the DNS and SMTP servers to any destination
2. Allow DNS (UDP port 53) from the DNS and SMTP servers to any destination
3. Allow SMTP (TCP port 25) from the SMTP servers to any destination
4. Allow SMTP (TCP port 25) from the SMTP servers to any destination
5. Allow SSH (TCP port 22) from the DNS and SMTP servers to the network management server on the Inside network
6. Allow syslog messages (UDP port 514) from the DNS and SMTP servers to the network management server on the Inside network
7. Deny anything else and log the access attempt

Finally, the inbound access control list for the Inside interface (*acl_inside*) determines what traffic may leave the Inside networks for either the DMZ or the Internet:

1. Allow DNS (TCP port 53) from any host on the Inside to the DNS server
2. Allow DNS (UDP port 53) from any host on the Inside to the DNS server
3. Allow SMTP (TCP port 25) from any host on the Inside to the ISP SMTP relay server.
4. Allow SSH (TCP port 22) from the Network Management workstation on the Inside network to the DNS and SMTP relay servers on the DMZ.
5. Deny anything else and log the access attempt

A partial listing of the firewall configuration is included in Appendix A.

Configuring the Server

The SMTP gateway is configured as a bastion host to mitigate its vulnerability to attacks. All the well-known rules apply: install only those packages that are required, apply all recommended vendor patches and updates, install Secure Shell for remote access and install and configure

YASSP, TCP Wrappers and Tripwire.

The SMTP gateway resides on the DMZ and is assigned the local address of 10.1.1.11/24, which is not routable on the Internet. Sendmail, however, needs to know the global IP address (6.1.1.4) and fully qualified name for the server (smtp1.widgets.com). The address translation occurs on the firewall and is completely transparent to the server. Sendmail normally learns at run-time by reading the interface configuration. A simple workaround for this problem is to configure the Ethernet interface (hme0) with the global name and assigned IP address and to add a logical sub-interface (hme0:1) using the local IP address and hostname:

```
# ifconfig hme0:1 plumb
# ifconfig hme0:1 10.1.1.11 netmask 0xffffff00 up
# route add default 10.1.1.1
# netstat -rn
Routing Table:
  Destination        Gateway           Flags  Interface
----------------- -------------------- -----  ---------
10.1.1.0          10.1.1.11             U     hme0:1
6.0.0.0           6.1.1.4               U     hme0
default           10.1.1.1              UG
127.0.0.1         127.0.0.1             UH    lo0
#
```

This completes the basic configuration of the major components for the system. Testing and tuning sendmail itself involves following the standard methods and techniques outlined in [Costales] and the man pages and other documentation supplied with the sendmail distribution.

Conclusions

Sendmail has matured and evolved into a robust security-aware message transport agent. Sendmail 8.9 and later versions include security features that can be adapted to a wide range of needs and situations. System security can be assured to a high degree by careful attention to file and directory ownership and modes. Numerous configuration options allow for extensive tuning of security, performance and behavior. Sendmail 8.12 eliminates the set-user-ID root problem, which further improves the security of sendmail installations by avoiding local exploits.

A systems approach to security with a defense in depth strategy has implications for the design and configuration of the network, the server and the sendmail program. Tradeoffs between security and functionality can best be made when adopting a risk management approach to security issues.

References

Cheswick Cheswick, W. and S. Bellovin. Firewalls & Internet Security. Boston: Addison-Wesley. 1994.

Chouanard Chouanard, Jean. "How to install Solaris and have a good host security" 19 Nov. 2000. URL: http://www.yassp.org/ (6 Feb. 2002)

CiscoPix Cisco Systems, Inc. " Cisco PIX Firewall and VPN Configuration Guide Version 6.1." (29 Jan. 2002) http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_61/config/ (5 Feb. 2002)

CiscoNat Cisco Systems. “Using nat, global, static, conduit, and access-list Commands and Port Redirection on PIX.” 9 Jan. 2002. URL: http://www.cisco.com/warp/public/707/28.html (4 Feb. 2002)

Costales Costales, Bryan with Eric Allman. Sendmail, Second Edition. Cambridge: O'Reilly & Associates, Inc. 1997.

Garfinkel Garfinkel, S. and G. Spafford. Practical UNIX and Internet Security, 2nd Edition. Cambridge: O'Reilly. 1996.

Rekhter Rekhter, Y. et al. “Address Allocation for Private Internets.” RFC-1918. February 1996. URL: ftp://ftp.isi.edu/in-notes/rfc1918.txt (4 Feb. 2002)

SmlCons Sendmail Consortium. “Sendmail 8.12.2.” URL: http://www.sendmail.org/8.12.2.html (4 Feb. 2002)

SmlInc1 Sendmail, Inc. “Sendmail.” URL: http://www.sendmail.com (4 Feb. 2002)

SmlInc2 "Securing Sendmail." URL: http://sendmail.net/000705securitygeneral.shtml (5 Feb. 2002)

Stevens Stevens, W. R. TCP/IP Illustrated, Volume 1 The Protocols. Boston: Addison-Wesley. 1994

Tripwire Tripwire, Inc. "Tripwire." URL: http://www.tripwire.com (6 Feb. 2002)

Wietse Wietse, Venema. " Wietse's tools and papers." URL: ftp://ftp.porcupine.org/pub/security/index.html (6 Feb. 2002)

Viega Viega, John and Gary McGraw. Building Secure Software. Boston: Addison-Wesley. 2002.

Zwicky Zwicky, Elizabeth D., Simon Cooper and D. Brent Chapman. Building Internet Firewalls, 2nd Edition. Cambridge: O'Reilly. 2000.

Zalewski Zalewski, Michael. "Multiple Local Sendmail Vulnerabilities" (10 March 2001) URL: http://www.securiteam.com/unixfocus/6F0010A2UI.html (6 Feb. 2002)

Appendix A

Partial PIX configuration file for the HQ firewall

```
!--- Name interfaces and assign security levels
nameif ethernet0 outside security0
```

```
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
!--- Assign IP addresses to interfaces
ip address outside 6.3.1.2 255.255.255.224
ip address inside 10.1.0.1 255.255.255.0
ip address dmz 10.1.1.1 255.255.255.0
!--- Assign routes and minimal logging
route outside 0.0.0.0 0.0.0.0 6.3.1.1 1
logging on
logging host inside 10.1.0.24
!--- Define access translation pools and PAT address
global (outside) 1 6.3.1.29
nat (inside) 1 10.0.0.0 255.0.0.0 0 0
nat (dmz) 1 10.1.1.11 255.255.255.255 0 0
nat (dmz) 1 10.1.1.12 255.255.255.255 0 0
!--- Define two-way translations
static (dmz,outside) 6.3.1.4 10.1.1.11 netmask 255.255.255.255 0 0
static (dmz,outside) 6.3.1.5 10.1.1.12 netmask 255.255.255.255 0 0
static (inside,dmz) 10.1.0.0 10.1.0.0 netmask 255.255.0.0 0 0
!--- Outbound access list for the Outside interface
access-list acl-out permit tcp any host 6.3.1.4 eq 22
access-list acl-out permit tcp any host 6.3.1.4 eq domain
access-list acl-out permit udp any host 6.3.1.4 eq domain
access-list acl-out permit tcp any host 6.3.1.5 eq 22
access-list acl-out permit tcp any host 6.3.1.5 eq domain
access-list acl-out permit udp any host 6.3.1.5 eq domain
access-list acl-out permit tcp any host 6.3.1.5 eq smtp
access-list acl-out deny ip any any log
!--- Outbound access list for the DMZ interface
access-list acl-dmz permit tcp host 10.1.1.11 any eq 22
access-list acl-dmz permit tcp host 10.1.1.11 any eq smtp
access-list acl-dmz permit tcp host 10.1.1.11 any eq domain
access-list acl-dmz permit udp host 10.1.1.11 any eq domain
access-list acl-dmz permit tcp host 10.1.1.12 any eq 22
access-list acl-dmz permit tcp host 10.1.1.12 any eq smtp
access-list acl-dmz permit tcp host 10.1.1.12 any eq domain
access-list acl-dmz permit udp host 10.1.1.12 any eq domain
access-list acl-dmz deny ip any any log
!--- Outbound access list for the Inside interface
access-list acl-ins permit tcp 10.1.0.0 255.255.255.0 any eq 22
access-list acl-ins permit tcp 10.1.0.0 255.255.255.0 any eq smtp
access-list acl-ins permit tcp 10.1.0.0 255.255.255.0 any eq domain
access-list acl-ins permit udp 10.1.0.0 255.255.255.0 any eq domain
access-list acl-ins deny ip any any log
!--- Bind access list to the interfaces
access-group acl-out in interface outside
access-group acl-dmz in interface dmz
access-group acl-ins in interface inside
```