Interested in learning more about security?

SANS Institute InfoSec Reading Room

A Secure Sendmail Based DMZ for the Corporate Email Environment

Adding a layer of simple mail transfer protocol (SMTP) infrastructure in the demilitarized zone (DMZ) between the Internet and your corporate email system is an effective way for corporate environments to enhance the security and functionality of their electronic email systems. We will explore the concept of using the sendmail SMTP application as a DMZ for the corporate email system (i.e. Exchange, GroupWise, Notes, etc). This concept has a number of benefits, including increased security, scalability, stability, and f...

A Secure Sendmail Based DMZ for the Corporate Email Environment

Jason D. McLellan

January 13, 2003

GSEC version 1.4b

ABSTRACT

Adding a layer of simple mail transfer protocol (SMTP) infrastructure in the demilitarized zone (DMZ) between the Internet and your corporate email system is an effective way for corporate environments to enhance the security and functionality of their electronic email systems.

We will explore the concept of using the sendmail SMTP application as a DMZ for the corporate email system (i.e. Exchange, GroupWise, Notes, etc). This concept has a number of benefits, including increased security, scalability, stability, and flexibility.

This approach will probably only be attractive to larger environments as smaller organizations will have a hard time cost justifying the extra server(s) required and supporting the added complexity. This approach should work equally as well with other SMTP systems such as Qmail, or Postfix that are appropriately secured.

THE EMAIL DMZ

The sendmail program certainly does not have the best reputation for being secure, and does not often get credited for increasing security. Much of its bad reputation can be attributed to its involvement in the Morris worm incident in 1988 [Kehoe]. There have been other more recent issues including the planting of a trojan in the source distribution [CERT CA-2002-28]. However, it is possible to securely deploy sendmail, and use it to add defense-in-depth to the corporate email environment.

The specific areas we will explorer are; the benefits of this approach, design considerations and example designs, sendmail compilation and configuration, and content security strategies for antivirus and unsolicited email control.

BENEFITS

Adding a secure SMTP buffer zone between your email system and the Internet has several advantages:

Security: The SMTP gateway of the internal mail system (the GWIA gateway in Novell's GroupWise as an example) is a direct path to the post office databases and end user data. Having an extra layer in front of that gateway will allow for better separation of processes and data. In addition, this layer gives the mail

administrator a shutoff valve (without losing mail). In an emergency (virus attack or alike) the mail administrator can simply turn off the internal gateway, disconnect, or otherwise disable it to allow the mail to simply queue on the Internet DMZ side. Without this layer you will be relying on the sending systems queuing policy for delivery once your back on line.

From a security standpoint the main issues this design will seek to addresses with the concept are:

- Add a layer of infrastructure (defense-in-depth) between our internal post offices (our data) and the Internet.
- Prevent our gateway from being used by unauthorized parties and bad-guys as a mail relay.
- Prevent viruses and lost data with scanning and queuing during an attack.
- Use mail blacklists and other sendmail features to reduce the quantity of unsolicited/unwanted/offensive email.

Scalability: Sendmail was built to handle the large volumes of email on the Internet; it runs on virtually any platform and can be scaled with many different technologies such as large UNIX servers, layer 4-7 switches, and software clustering.

In large corporate environments the SMTP gateways provided by the major email vendor's, can create challenges in scalability. This can be especially true if the gateway is a single device and is burdened both with sending and receiving processes. It is possible to use load balancing and clustering technologies with these gateways but you quickly run into nuances with access controls and other issues relating to relay prevention.

Stability: Sendmail is a very old and very mature application, being developed in the 1980's [Vixie, Avolio].

Flexibility: Due to its popularity many commercial mail add-on products are designed to work with, or are based directly on sendmail. This allows for more choices when selecting content security vendors. Even products that are not designed to use sendmail specifically can be integrated into this framework.

DESIGN CONSIDERATIONS

In the following section we are going to explore three design scenarios. These three scenarios are by no means presented as the best possible implementation of sendmail, but merely as a basis for exploring the concept of an email DMZ. All of these designs use some basic terminology in regards to function of the components. The three main components of this framework are the incoming layer, the content security layer, and the delivery layer.

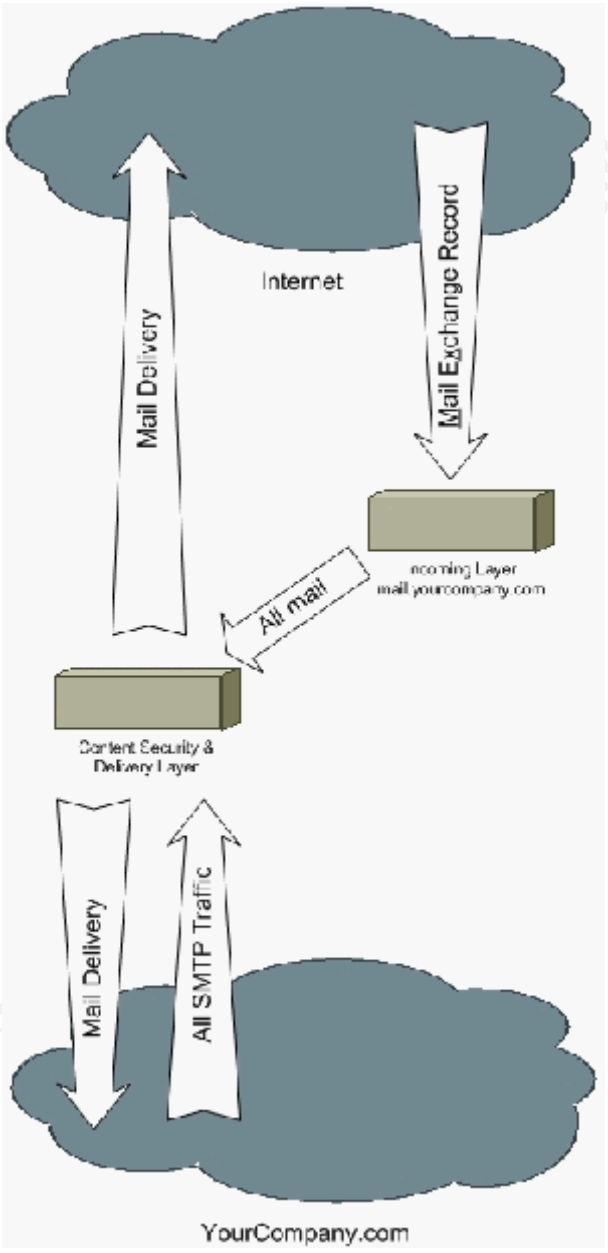
Incoming Layer: These mail systems service the domain name system (DNS) mail exchange (MX) record for your domain for incoming mail from the Internet. The sendmail on these systems should be configured to send all mail bound for your domain to the content security layer. At this layer all compliance checks should be in place, making sure simple things like senders domain exists, and the message conforms to RFC-821 [Postal] and RFC-1869 [Klensin, Freed, Rose, Stefferud, Crocker]. Also, at this level, you can use the "dnsbl" feature to check SPAM blacklists and your own local "access_db" for black and white listings. Built-in sendmail features for SPAM control should be implemented in the incoming layer and add-on products in the content security layer.

Content Security Layer: The content security servers configuration will vary depending on which vendor you choose. The content security server might have its own proprietary SMTP service, or might be configurable to use a local sendmail instance. In any case, the content security servers should be configured to send all mail to the delivery layer for final delivery. Internal systems should be configured to send all SMTP mail to the content security layer, thus insuring all SMTP traffic is scanned including SMTP-based mail originating from and bound to your internal company.

Delivery Layer: The delivery servers queue and deliver all content-screened mail to the final destination addresses for delivery, internally (your companies email gateway) or externally (Internet). The delivery layer can be combined with the content security layer depending on the volume requirements of the site, the vendor, and the amount of anti-spam/virus screening occurring.

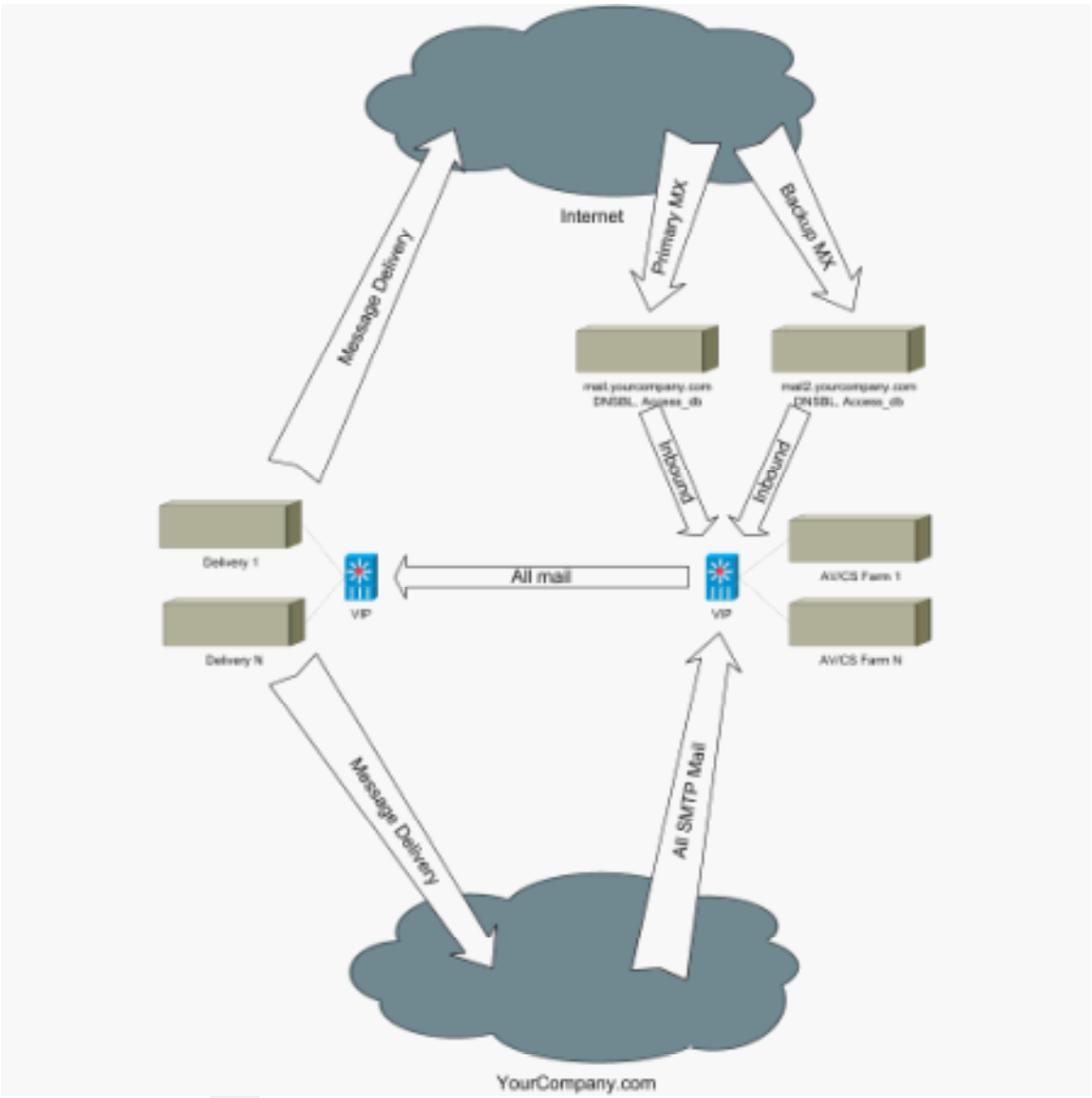
The following design diagrams use load balancing of the different layers to demonstrate one of the ways this concept can be scaled. Your specific systems requirements of uptime and volume will drive the need to use load balancing, single systems, or clustering. In the following diagrams the "VIP" icon designates layer 4-7 switching.

SCENERIO ONE: This scenario would support a small to medium sized shop with minimal uptime requirements. This concept could theoretically be done even on a single server depending on choice of content security vendor, and volume requirements. All of the configuration examples provided later will use this design to keep things as simple as possible.

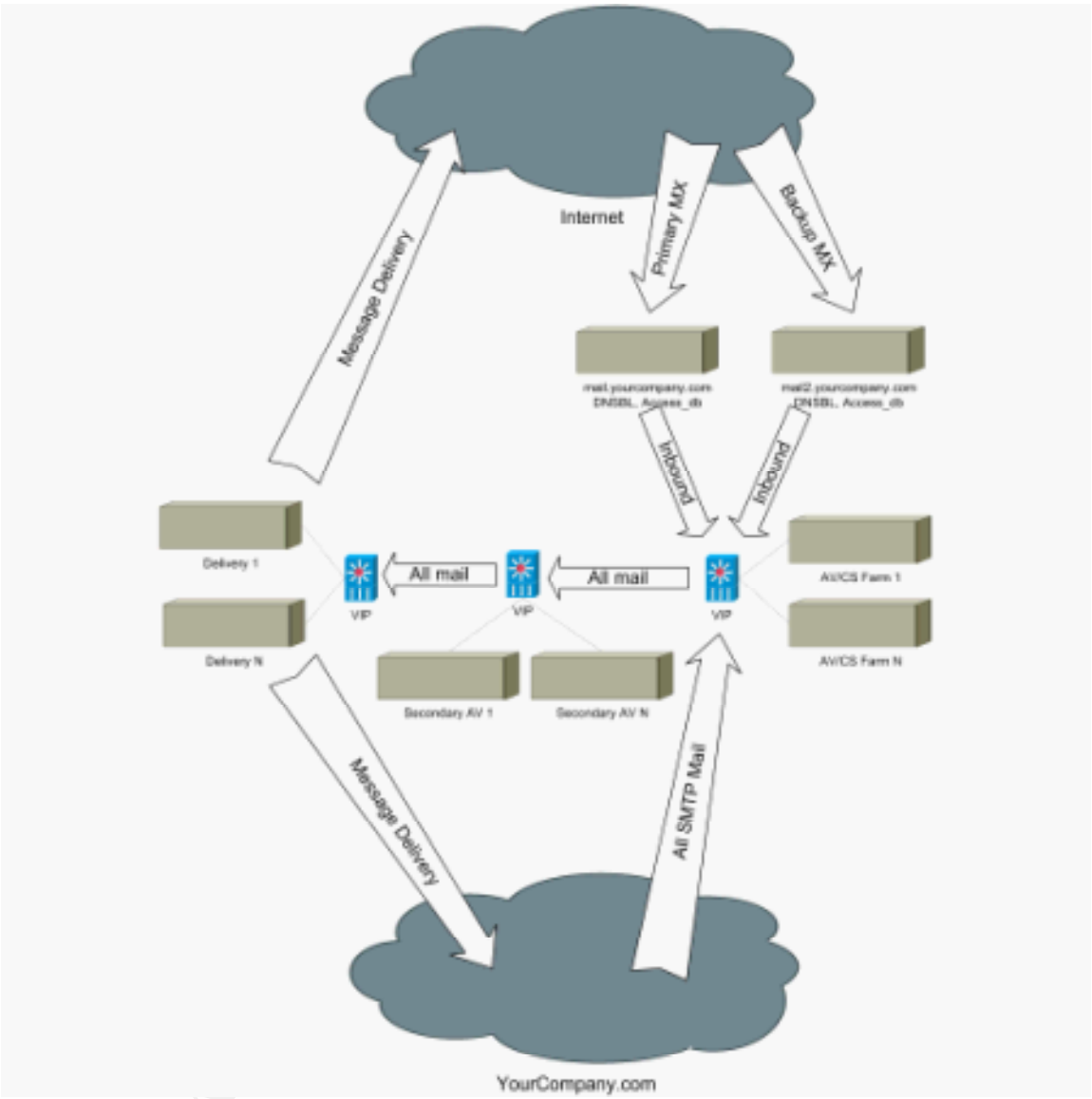


In this scenario we have one server handling the incoming layer and one server handling both content security and delivery layers. For this example, we are going to assume the content security layer is running a product that will use sendmail to deliver the filtered mail.

SCENERIO TWO: Three-tiered design separating the three layers on to separate servers.



SECENERIO THREE: In this case all layers are separated and we have expanded the content security layer into two separate pieces to demonstrate how multiple content security vendors could be used together.



BUILDING SENDMAIL

Now that we have some design ideas, we can look more closely at how to build sendmail. Again for simplicity sake, we will be focusing design scenario one, one server providing the incoming function, and one server providing delivery and content security functions. Also note, because UNIX examples are on Solaris (Sparc), structures and commands will likely vary on other platforms.

We will be building version 8.11.6 of the product. The italic convention *"#<some command>"* designates a UNIX root command prompt and command.

Building your own distribution: On your development system download, verify MD5 checksum or the PGP signature and unpack the source code.

Note: There has been a recent incident where an unknown attacker added a trojan to the sendmail source code at http://www.sendmail.org/ [CERT CA-2002-28].

To check the PGP signature you will need to have the PGP software installed on your development system (can be downloaded from http://www.pgpi.org):

#gunzip -c sendmail-8.11.6.tar.gz | gpg --verify sendmail-8.11.6tar.sig -

If your don't have PGP check the MD5 sum as follows:
Check http://www.cs.niu.edu/~rickert/sendmail/md5.html or another trusted site for the sum value. It will look something like this:

a57e7681d810d9d6400cbe6bbcf06aa0 sendmail.8.11.6.tar.gz

Now compare the value above with the output from the md5sum command (can be downloaded at http://www.sunfreeware.com):

#md5sum sendmail.8.12.6.tar.gz

After you have verified the source codes authenticity and integrity, uncompress and untar the distribution:

#zcat sendmail.8.11.6.tar.gz|tar xvf –

Make some missing manual page structures:

#mkdir /usr/share/man/cat8
#mkdir /usr/share/man/cat5
#mkdir /usr/share/man/cat1

Build componets:

#cd sendmail8.11.6
#cd sendmail
#sh Build
#sh Build Install
#cd ../mailstats
#sh Build

#sh Build Install
#cd ../smrsh
#sh Build
#sh Build Install
#cd ../makemap
#sh Build
#sh Build Install
#cd ../praliases
#sh Build
#sh Build Install
#cd ../vacation
#sh Build
#sh Build install

Note on Solaris and HPUX, do not compile mail.local use the existing local mailer /etc/lib/mail.local.

The macro configuration files, default macro and platform specific macro files are typically located in the /usr/lib/mail/cf structure. We need to refresh this structure from the distribution and make a backup copy.

*#tar –cvpf sendmail.backup /usr/lib/mail/**
#cd /usr/lib/mail/cf
#cp sendmail.mc sendmail.mc.upgrade
#cd /<distribution sendmail-8.11.6/cf>
*#cp –R * /usr/lib/mail*

Create your binary distribution with your preferred technique. Since sendmail is very simple my preference is to just create a tar file for distribution:

#tar –cvpf sendmail-8-11-6.tar /usr/lib/sendmail
#tar –uvpf sendmail-8-11-6.tar /usr/lib/smrsh
#tar –uvpf sendmail-8-11-6.tar /usr/sbin/mailstats
#tar –uvpf sendmail-8-11-6.tar /usr/sbin/makemap
#tar –uvpf sendmail-8-11-6.tar /usr/sbin/praliases
#tar –uvpf sendmail-8-11-6.tar /etc/mail/helpfile
#tar –uvpf sendmail-8-11-6.tar /etc/mail/statistics
#tar –uvpf sendmail-8-11-6.tar /usr/bin/newaliases
#tar –uvpf sendmail-8-11-6.tar /usr/bin/mailq
#tar –uvpf sendmail-8-11-6.tar /usr/bin/hoststat
#tar –uvpf sendmail-8-11-6.tar /usr/bin/purgestat
#tar –uvpf sendmail-8-11-6.tar /usr/bin/vacation
#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat8/sendmail.8
#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat8/mailstats.8
#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat8/smrsh.8
#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat8/makemap.8
#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat8/praliases.8
#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat5/aliases.5

#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat1/mailq.1
#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat1/newaliases.1
#tar –uvpf sendmail-8-11-6.tar /usr/share/man/cat1/vacation.1

You should also include other configuration files specific to your environment such as /usr/lib/mail/cf/sendmail.mc or /etc/mail/mailertable, i.e. configuration files that we will cover in great detail below.

Installing on the target system: Install the new binaries as appropriate to your packaging method. In the above case, simply stop sendmail and untar the files.

#pkill -x -u 0 sendmail
#cd /
#tar –xvpf sendmail-8-11-6.tar

Verify files permissions as follows:

#chown mail /var/spool/mqueue
#chmod 700 /var/spool/mqueue
#chmod 755 /var/spool
#chmod 644 /etc/mail/sendmail.cf
#chmod 644 /etc/mail/aliases
#chmod 644 /etc/mail/mailertable
#chmod 644 /etc/mail/local-host-names
#chmod 644 /etc/mail/virtualusertable
#chmod 644 /etc/mail/sendmail.cf

[http://www.sendmail.net/000705securitygeneral.shtml]

SECURING THE HOST OPERATING SYSTEM

One could write an entire book on securing Solaris and other UNIX operating environments (several have). Due to space constraints and scope creep we will not be covering it here.

A good starting place would be Lance Spitzners papers located at:
http://www.spitzner.net

There are two papers Lance has written that are particularly relevant.
http://www.spitzner.net/armoring.html - Armoring Solaris
http://www.spitzner.net/armoring2.html - Armoring Solaris II

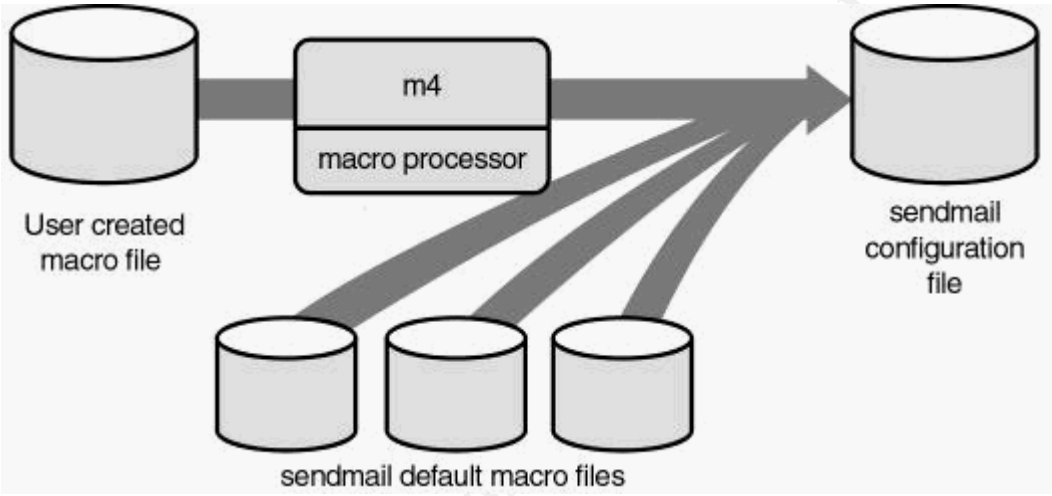
These papers focus around installing a Solaris based firewall but fit very nicely in what we are trying to do here.

Remember to disregard the portions pertaining to the removal of sendmail environment in Lance's papers.

CONFIGURING SENDMAIL

After we compile and install sendmail, we need to configure it to do what we need. Because of the number of options and its use of macro configuration files, databases, and texts files; configuring sendmail can seem overwhelming. Understanding the macro process and directives is key to properly configuring sendmail.

The main configuration file (sendmail.cf, a very large text file) is built with the M4 macro compiler with input from the sendmail.mc (user created text file), and default macro files. These macro configuration files determine how sendmail will be configured to behave on your platform.



[Blum, Chapter 11, figure 11.2]

To configure sendmail we only need to worry about the sendmail.mc file, and the database and text files in the /etc/mail structure. You should never need to directly edit the sendmail.cf file. The default platform specific configuration should have been taken care of during installation, or by the base operating system install. Below is a detailed look at some of the relevant configuration directives that we will use to build the different sendmail systems in our configurations.

THE BUILDING BLOCKS OF THE SENDMAIL.MC FILE

Some of the basic pieces include FEATURES, MAILERs, DOMAINs, DEFINE (options), and OSTYPEs. All of these types of directives tell sendmail how to behave. Below we are going to look closely at the relevant directives, why they are important, and lastly how to use them in the sample configuration.

Some of these have particular importance from a security view:

- smrsh – the secure restricted shell.
- access_db – Access control, relay control, anti-spam.
- blacklist_recipients – Anti-spam.
- dnsbl – Anti-spam.
- relay_hosts_only – Anti-spam, relay control.
- delay_checks - Anti-spam.
- confPRIVACY_FLAGS – Turning off unnecessary/dangerous functions.
- confRUN_AS_USER – Running at reduced access levels, non root.

FEATURE(smrsh, '/usr/lib/smrsh')dnl
The "smrsh" command instructs sendmail to use the /usr/lib/smrsh secure restricted shell rather than /sbin/sh (the Bourne shell).
DEFINE(confRUN_AS_USER, mail)dnl
Later versions of sendmail have the ability to run as a different user ID (UID). This feature allows the sendmail child processes to run as a different UID. There are several issues with this feature, first the main process must be started as UID root, and second it does not support local mail delivery. In our example (scenario one) we are going to use this feature on the incoming layer.

FEATURE(mailertable, 'dbm /etc/mail/mailertable')dnl
The "mailertable" feature allows us to control how mail flows between the gateway and our internal system (or any domain for that matter). For inbound email we will have an entry in this database that directs all inbound email be delivered to a specific gateway, i.e. the GroupWise GWIA or Exchange SMTP connector.
An example of mailertable database entries:

yourdomain.com	smtp:mygateway.internal.com
yourseconddomain.com	esmtp:mysecondgateway.internal.com

FEATURE(redirect)dnl
This feature turns off (rejects) the user.redirect functionality of sendmail. We don't intent to have any local account on these servers and therefore don't need it.

FEATURE(use_cw_file)dnl
This feature tells send mail to look for the /etc/mail/local-host-names txt file to determine if there are other domains or hosts to accept mail for (mail for local delivery). It will by default use the DNS name of the server (or hosts entry) to automatically make this determination. In our configurations, we don't ever want to accept mail for local delivery so we want to leave this file blank, but its good practice to leave this feature on.

FEATURE(access_db, 'dbm /etc/mail/access')dnl
The "access_db" feature allows us to control the flow of email through the sendmail system. This file is critical in controlling the flow of email between SMTP gateways. Some example entries:

Allow any IP address starting with 10 to relay mail though this gateway:

10	RELAY

Allow recipient domain to relay, when use with "relay_hosts_only" feature:

mydomain.com	RELAY

Allow any host starting with IP address of 10.1.1.x, will bypass "dnsbl" and allow a blacklisted site to send you mail:

10.1.1	OK

Reject with error message indicated, when used with feature "blacklist_recipients":

10.1.1.5	ERROR:550 Some text message
spammer.com	ERROR:550 Some text message
user@	ERROR:550 Some text message

On senders domain or IP address, send to discard mailer:

Discard.com	DISCARD
10	DISCARD

FEATURE(blacklist_recipients)dnl
The "blacklist_reciepients" directive changes the way the "access_db" directive works. In addition to using IP addresses in the databases, this allows the use of domain names (mydomain.com), email addresses (myuser@mydomain.com), and users (myuser@). This is especially useful when dealing with unwanted email or when white listing business partners mail systems.

FEATURE(relay_hosts_only)dnl
This feature "relay_hosts_only" also changes the way the "access_db" feature works. This allows us to put our own domain entry in the access database with RELAY without fear someone will source mail off our gateway. This is very important to our design when there is more than one inbound domain we accept mail from. If there is only one, we can use the "relay_entire_domain" feature (see next) to accomplish the same thing without an entry in the access database. Special Note: Do not use a domain name and RELAY in the access database without turning on this feature. This would allow a spammer to source mail from your domain using your server (send mail from someuser@yourdomain.com to anywhere)!

FEATURE(relay_entire_domain)dnl
The following feature "relay_entire_domain" is necessary in our configuration since we want to use sendmail as a pass-though. This will allow outbound email with your companies' domain name to be relayed out.

FEATURE(always_add_domain)dnl
The "always_add_domain" feature causes sendmail to add the right hand side (RHS) domain name to the source email addresses of messages where it does not exist. When enabled a message send locally from UID root would become root@yourdomain.com. This makes sure all messages originating at this gateway have a domain name.

FEATURE('delay_checks')dnl
The "delay_checks" feature is useful when used in combination with the "dnsbl" feature for SPAM control. Without enabling this feature, the log entry for the rejected email will not contain the destination users mail address. Having the destination address is very useful when troubleshooting; typically the person calling your organizations support center is the person whom cannot receive the email. This feature will put the destination users address into the reject log message allowing you to search the log for it directly. In the other case, the external entity can be asked whom they were trying to send to. Otherwise, you are stuck looking for origin SMTP servers, not a straight forward as it might seem. For example, the sender works for company abc.com but is sending you email from their home cable modem.

FEATURE('dnsbl', 'blackholes.mail-abuse.org', '"Mail rejected from host "$&{client_addr}", your email system is listed at http://www.mail-abuse.org/rbl"')dnl
The "dnsbl" feature allows sendmail to utilize DNS based black hole services to block email. Unsolicited email will be cover in more detail later. Important note: the "access_db" will override "dnsbl." This is useful for allowing a company whom is blacklisted to send you mail, i.e. local white list.

FEATURE(nullclient, 'mysmarthost.mycompany.com')dnl
The "nullclient" feature directive tells sendmail to deliver all mail to a specified host. These is useful for intermediary boxes such as virus scanners, or inbound mail systems in our configuration.

MAILER(local)dnl
MAILER(smtp)dnl
undefine('UUCP_RELAY')dnl
undefine('BITNET_RELAY')dnl
The above section defines the mailers. In our case we are going to need the local mailer to handle postmaster and abuse mail. It is important to note that "smtp" includes SMTP, ESMTP, SMTP8 mailers. Also of importance, we

definitely only want to handle SMTP and variants and not UNIX to UNIX copy (UUCP), other legacy mailers, or pieces of their functionality. There are ways to use the pathing in UUCP to bounce, or relay mail off your system or otherwise fool your access controls. We use the undefined macro to disable these [Allman][http://www.iss.net/security_center/static/210.php]

define('confSMTP_LOGIN_MSG', '$j AUTHORIZED USE ONLY! UNSOLICTED EMAIL NOT WELCOME!')dnl
The above command defines the connect banner for the SMTP service.

define('confPRIVACY_FLAGS', 'goaway restrictqrun authwarnings noexpn novrfy need mailhelo')dnl
The above command turns off some features of sendmail, and requires the use of others. This allows us to turn off most of the unneeded features of the listener. Note: The "goaway" directive includes the following directives; noreceipts, restrictmailq, restrictqrun, noetrn and nobodyreturn [Vixie, Avolio][Allman].

define('confMAX_DAEMON_CHILDREN', 200)dnl
The features sets a maximum number of child processes sendmail can launch. By default in sendmail there is no limit. Once this limit is reached the parent process stops accepting new connections until the number falls below the count.

define('confQUEUE_LA',3)dnl
This defines a system load average, that when reached will tell sendmail to stop delivering mail and simply queue it.

INCOMING LAYER CONFIGURTION
Getting back to the example configuration, we need to setup and compile the sendmail.mc file for the server that is processing the inbound layer. For this server the sendmail.mc should look something like this:

```
# sendmail.mc  INCOMING servers created 1/6/03 JM
divert(0)dnl
OSTYPE(solaris2)dnl
DOMAIN(generic)dnl
FEATURE(smrsh, '/usr/lib/smrsh')dnl
FEATURE(nullclient, esmtp:myvirusscanner.mycompany.com)dnl
FEATURE(redirect)dnl
FEATURE(use_cw_file)dnl
FEATURE(access_db, 'dbm /etc/mail/access')dnl
FEATURE(relay_hosts_only)dnl
FEATURE(blacklist_recipients)dnl
```

```
FEATURE('dnsbl', 'some.spam-blacklist.com', '"Mail rejected from host
"$&{client_addr}",your email system is listed at someblacklist.net - see:
http://someblacklist.net/bl.shtml?"$&{client_addr}')dnl
define('confRUN_AS_USER', mail)dnl
define('confTO_IDENT','0s')dnl
define('confSMTP_LOGIN_MSG', '$j AUTHORIZED USE ONLY!
UNSOLICTED EMAIL NOT WELCOME!')dnl
define('confPRIVACY_FLAGS', 'goaway restrictqrun authwarnings noexpn
novrfy needmailhelo')dnl
define('confMAX_DAEMON_CHILDREN', 200)dnl
define('confQUEUE_LA',3)dnl
define('confMAX_MESSAGE_SIZE', 4194304)dnl
```

This configuration is doing the following:

- Delivering all mail to the content security layer (all accepted mail).
- Using the secure restricted shell.
- Running child processes as user mail.
- Using "access_db" and "nullclient" to control the flow of mail.
- Turning off dangerous and unneeded commands.
- Requiring a fully qualified sender with resolvable domain names.
- Displaying a banner on connect.
- Using a DNS black list to reject SPAM.
- Setting maximum queuing and processes to prevent system overload.
- Setting the maximum message size to accept to 4MB.

Use the M4 macro engine to compile the configuration file:
Note: In Solaris you will need to add /usr/ccs/bin to your path.

```
#cd /usr/lib/mail/cf
#m4 ../m4/cf.m4 sendmail.mc > /etc/mail/sendmail.cf
```

We will need to create a user called "mail" and change some file permissions to get the "confRUN_AS_USER" option to work properly:

```
#useradd mail
#chown mail /var/spool/mqueue
```

Configure /etc/mail databases and text files including:

- access
- aliases
- mailertable
- local-host-names

The access database for the inbound and delivery layers combined should look something like this:

10	RELAY
192.168	RELAY
localhost	RELAY
mydomain.com	RELAY
myseconddomain.com	RELAY

The mailertable database is not needed on the incoming layer since we are sending all mail directly to the delivery layer.

The aliases database can be left default and local-host-names file should be blank since we don't want this server to accept mail for local delivery for any domain.

Use the "makemap" utility to build the databases:

#cd /etc/mail
#makemap –v dbm access < access
#makemap –v dbm aliases < aliases
#makemap –v dbm mailertable < mailertable
#newlaiases (to activate the new aliases database)

CONTENT SECURITY AND DELIVERY LAYERS CONFIGURTION

For the content security and delivery layers the sendmail.mc file should look something like this:

```
# sendmail.mc  CS and DELVIERY servers created 1/6/03 JM
divert(0)dnl
OSTYPE(solaris2)dnl
DOMAIN(generic)dnl
FEATURE(smrsh, '/usr/lib/smrsh')dnl
FEATURE(mailertable, 'dbm /etc/mail/mailertable')dnl
FEATURE(redirect)dnl
FEATURE(use_cw_file)dnl
FEATURE(access_db, 'dbm /etc/mail/access')dnl
FEATURE(relay_hosts_only)dnl
FEATURE(blacklist_recipients)dnl
MAILER(smtp)dnl
MAILER(local)dnl
define('confTO_IDENT','0s')dnl
define('confSMTP_LOGIN_MSG', '$j AUTHORIZED USE ONLY!
UNSOLICTED EMAIL NOT WELCOME!')dnl
define('confPRIVACY_FLAGS', 'goaway restrictqrun authwarnings noexpn
novrfy needmailhelo')dnl
define('confMAX_DAEMON_CHILDREN', 200)dnl
```

```
define('confQUEUE_LA',3)dnl
define('confMAX_MESSAGE_SIZE', 4194304)dnl
undefine('UUCP_RELAY')dnl
undefine('BITNET_RELAY')dnl
```

This configuration is doing the following:

- Using the secure restricted shell.
- Using "access_db" and "mailertable" to control the flow of mail.
- Turning off dangerous commands and unneeded mailers.
- Requiring a fully qualified sender with resolvable domain names.
- Displaying a banner on connect.
- Setting maximum queuing and processes to prevent system overload.
- Setting the maximum message size to accept to 4MB.

Use the M4 macro engine to compile the configuration file:
Note: In Solaris you will need to add /usr/ccs/bin to your path.

#cd /usr/lib/mail/cf
#m4 ../m4/cf.m4 sendmail.mc > /etc/mail/sendmail.cf

The access database for the content security and delivery layers combined should look something like this:

10	RELAY
192.168	RELAY
localhost	RELAY
mydomain.com	RELAY
myseconddomain.com	RELAY

The mailertable database for the content security and delivery layers should looks something like this:

mydomain.com	esmtp:myinternal.mailgateway.mydomain.com
myseconddomain.com	esmtp:myinternal.mailgateway.mydomain.com

The aliases database can be left default and local-host-names file should be blank since we don't want this server to accept mail for local delivery for any domain.

Use the "makemap" utility to build the databases:

#cd /etc/mail
#makemap –v dbm access < access
#makemap –v dbm aliases < aliases
#makemap –v dbm mailertable < mailertable
#newlaiases (to activate the new aliases database)

ANTIVIRUS

Many solutions to scanning email only consider incoming Internet email. With this solution we can scan incoming, outgoing, and internal to internal. Ideally we would have all internal SMTP capable hosts setup to "smarthost" all mail to our content security layer. This won't completely protect us from internal propagation (Nimda, as an example, used it own SMTP program) but it will help if the virus uses the servers built-in resources.

Scanning outbound mail is also desirable in the event we do get infected with a virus that uses email for propagation. We don't want to inflict our viruses on business partners and others that are linked by our addresses books.

UNSOLICITED BULK COMMERICAL EMAIL (SPAM)

Unsolicited bulk commercial email, commonly called "spam" is a concern for several reasons; employee productivity, confidentiality, systems resource utilization, malicious email code, and possible liability for delivering offensive material to employee's or other agents of the company and partners.

The sendmail program has several built in features to assist with spam control. Making sure senders are following the rules of Internet email and Internet domain name service is the best way to start. Do not turn on the features "accept_unqualified_senders" or "accept_unresolvable_domains" as these will allow email from bogus domains and senders [Allman]. It is important to note that RFC821 says that we should allow blank senders (return path) to prevent loops [Postal][http://www.rfc-ignorant.org/policy-dsn.php].

The "dnsbl" feature and the newer feature "enhdnsbl" are facilities that can be used to have sendmail do a special reverse DNS lookup on the sending systems IP address to determine if the sending host is on the list, and reject or accept the message accordingly. Many blacklists have been built for use with these features. Some of these lists can be subscribed to freely, others require a donation, or a fee.

These lists fall into three general categories of open relay, dialup network lists, and message content based. An open-relay is a mail gateway on the Internet that has been setup to allow anyone to "bounce" mail off it. This is normally not done intentionally but by administrative error. A third party can then use this facility to send bulk email and making it more difficult to determine who really sent the mail and possibly foiling control lists. Current examples of this type of list can be found at http://ordb.org or http://relays.osirusoft.com. These lists will do some active testing on the Internet looking for these open systems. The dialup access range based lists seek to list all dialup user ranges as bad with the logic being a dialup IP address should not need to be a mail server (sending mail directly). The other general type of list is not concerned as much with the open

relay aspect but in the message content itself. Some like http://www.spamcop.net uses user submissions (spam reports) to create lists.

These lists can be implemented in two ways depending on your systems specific requirements (zone transfer and direct query). Many black list operators, especially the pay sites, will allow you to zone-transfer the list to your own DNS server to speed up the process. This allows your sendmail system to access the list locally from your DNS server verses having to query theirs.

To turn a list on in sendmail, edit the sendmail.mc file and add:
FEATURE(dnsbl, black list server, message to return to sender)dnl
Example from above:

FEATURE('dnsbl', 'some.spam-blacklist.com', '"Mail rejected from host "$&{client_addr}",your email system is listed at someblacklist.net - see: http://someblacklist.net/bl.shtml?'$&{client_addr}')dnl

You can use multiple lists in your configuration. The only limitation is processing time.

Eventually someone who needs to send your company email is going to get blacklisted on one of the lists you use. To allow the system to send you mail without turning off the entire list add an "OK" entry to the access (/etc/mail/access) database.
1.1.1.1 OK

To activate:

#makemap –v dbm access < access

One of the features you should turn on in the layer using "dnsbl" is "delay_checks." This is very useful when troubleshooting blocked systems. From a logging standpoint it puts the local recipient of the message on the same line as the rejected message.

STARTUP AND SHUTDOWN

The process of starting up and shutting down sendmail will vary based on platform. On Solaris, the default file /etc/rc2.d/S88sendmail will work fine for our example.

To manually start sendmail issue either of the following commands:

#/usr/lib/sendmail –bd –q10m
#/etc/init.d/sendmail start

Note: the –qNNm directive indicates in minutes the length of the queue wait cycle, i.e. how long to wait before reprocessing the queue. It can also be specified in –qNh where "h" is hours.

To stop sendmail use either:

 # pkill -x -u 0 sendmail
#/etc/init.d/sendmail stop

SUMMARY

Adding a layer of SMTP infrastructure in the DMZ is an effective way for corporate environments to enhance the security and functionality of their electronic email systems.

We have looked at the benefits of adding this layer of defense-in-depth to our mail system, including security, scalability, stability, and flexibility. We have looked at three design scenarios that utilize this concept. Although we have based our sample configurations on the simplest of the three designs, we have provided enough detail to build any one of the three.

We have demonstrated how to security build and deploy sendmail. We have seen how to security configure sendmail to protect our user data, the mail server platform, prevent unauthorized use of the gateway, reduce the number of unwanted/unsolicited/offensive emails, and protect ourselves from email-based viruses.

WORKS SITED:

Vixie, Paul A., Avolio, Frederick M. Sendmail: Theory and Practice, Second Edition. Digital Press, 2002.

Blum, Richard Sendmail for Linux Sams Publishing, 2000.

Allman, Eric. "Anti-Spam Configuration Control." Sendmail 8.9.x Configuration Files. Version 8.186. Feb. 2, 1999. URL: http://www.sendmail.org/m4/anti_spam.html (Jan 11, 2002).

Allman, Eric. "Tweaking Configuration Options." Sendmail 8.9.x Configuration Files. Version 8.186. Feb. 2, 1999. URL: http://www.sendmail.org/m4/tweaking_config.html (Jan 11, 2002).

Allman, Eric. "Mailers." Sendmail 8.9.x Configuration Files. Version 8.186. Feb. 2, 1999. URL: http://www.sendmail.org/m4/mailers.html (Nov. 18, 2002).

Allman, Eric. "Features." Sendmail 8.9.x Configuration Files. Version 8.186. Feb. 2, 1999. URL: http://www.sendmail.org/m4/features.html (Nov. 18, 2002).

Unknown #1. “dsn.rfc-ignorant.org listing policy” URL: http://www.rfc-ignorant.org/policy-dsn.php (Jan. 3, 2003).

Unknown #2. “Securing Sendmail.” July 5 2000. URL: http://www.sendmail.net/000705securitygeneral.shtml (Oct 7, 2002).

Unknown #3, Internet Security Systems. “smtp-sendmail-relay (210)”. URL: http://www.iss.net/security_center/static/210.php (Dec 11,2002).

Spitzner, Lance. “Armoring Solaris.” Lance’s Security Papers. Aug. 19, 2001. URL: http://www.spitzner.net/armoring.html (Jan 7, 2002).

Spitzner, Lance. “Armoring Solaris: II.” Lance’s Security Papers. Jul. 20, 2002. URL: http://www.spitzner.net/armoring2.html (Jan 7, 2002).

Kehoe, Brendan (curator). “8.1: The Internet Worm.” Zen and the Art of the Internet. URL: http://sunland.gsfc.nasa.gov/info/guide/The_Internet_Worm.html (Dec 18, 2002).

Postal, Johnathon B. “SIMPLE MAIL TRANSFER PROTOCOL” RFC 821. Aug 21, 1982. URL: http://www.ietf.org/rfc/rfc0821.txt?number=821 (Dec 5, 2002)

Klensin, J., Freed, N., Rose, M., Stefferud, E., Crocker, D. “SMTP Service Extensions”. RFC 1869. Nov 1995. URL: http://www.ietf.org/rfc/rfc1869.txt?number=1869 (Dec 5, 2002).

Freed, N., Borenstein, N. “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies” RFC 2045. Nov 1996. URL: http://www.ietf.org/rfc/rfc2045.txt?number=2045 (Dec 5, 2002).

Freed, N., Borenstein, N. “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types” RFC 2046. Nov 1996. URL http://www.ietf.org/rfc/rfc2046.txt?number=2046 (Dec 5, 2002)

Cerf, Vinton G. “I REMEMBER IANA”. RFC 2468. Oct 17, 1998. URL http://www.ietf.org/rfc/rfc2468.txt?number=2468 (Dec 5, 2002)

Last Updated: April 21st, 2018

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

Blue Team Summit & Training 2018	Louisville, KYUS	Apr 23, 2018 - Apr 30, 2018	Live Event
SANS Riyadh April 2018	Riyadh, SA	Apr 28, 2018 - May 03, 2018	Live Event
SANS Doha 2018	Doha, QA	Apr 28, 2018 - May 03, 2018	Live Event
SANS SEC460: Enterprise Threat Beta Two	Crystal City, VAUS	Apr 30, 2018 - May 05, 2018	Live Event
Automotive Cybersecurity Summit & Training 2018	Chicago, ILUS	May 01, 2018 - May 08, 2018	Live Event
SANS SEC504 in Thai 2018	Bangkok, TH	May 07, 2018 - May 12, 2018	Live Event
SANS Security West 2018	San Diego, CAUS	May 11, 2018 - May 18, 2018	Live Event
SANS Melbourne 2018	Melbourne, AU	May 14, 2018 - May 26, 2018	Live Event
SANS Northern VA Reston Spring 2018	Reston, VAUS	May 20, 2018 - May 25, 2018	Live Event
SANS Amsterdam May 2018	Amsterdam, NL	May 28, 2018 - Jun 02, 2018	Live Event
SANS Atlanta 2018	Atlanta, GAUS	May 29, 2018 - Jun 03, 2018	Live Event
SANS London June 2018	London, GB	Jun 04, 2018 - Jun 12, 2018	Live Event
SANS Rocky Mountain 2018	Denver, COUS	Jun 04, 2018 - Jun 09, 2018	Live Event
SEC487: Open-Source Intel Beta Two	Denver, COUS	Jun 04, 2018 - Jun 09, 2018	Live Event
DFIR Summit & Training 2018	Austin, TXUS	Jun 07, 2018 - Jun 14, 2018	Live Event
Cloud INsecurity Summit - Washington DC	Crystal City, VAUS	Jun 08, 2018 - Jun 08, 2018	Live Event
SANS Milan June 2018	Milan, IT	Jun 11, 2018 - Jun 16, 2018	Live Event
Cloud INsecurity Summit - Austin	Austin, TXUS	Jun 11, 2018 - Jun 11, 2018	Live Event
SANS Oslo June 2018	Oslo, NO	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS ICS Europe Summit and Training 2018	Munich, DE	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Cyber Defence Japan 2018	Tokyo, JP	Jun 18, 2018 - Jun 30, 2018	Live Event
SANS Crystal City 2018	Arlington, VAUS	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Philippines 2018	Manila, PH	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Vancouver 2018	Vancouver, BCCA	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS Paris June 2018	Paris, FR	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS Cyber Defence Canberra 2018	Canberra, AU	Jun 25, 2018 - Jul 07, 2018	Live Event
SANS Minneapolis 2018	Minneapolis, MNUS	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS London July 2018	London, GB	Jul 02, 2018 - Jul 07, 2018	Live Event
SANS Cyber Defence Singapore 2018	Singapore, SG	Jul 09, 2018 - Jul 14, 2018	Live Event
SANS Charlotte 2018	Charlotte, NCUS	Jul 09, 2018 - Jul 14, 2018	Live Event
SANSFIRE 2018	Washington, DCUS	Jul 14, 2018 - Jul 21, 2018	Live Event
SANS Malaysia 2018	Kuala Lumpur, MY	Jul 16, 2018 - Jul 21, 2018	Live Event
SANS Seattle Spring 2018	OnlineWAUS	Apr 23, 2018 - Apr 28, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced