zimbra®
A SYNACOR PRODUCT

Postscreen for Zimbra

A Zimbra Collaboration Whitepaper

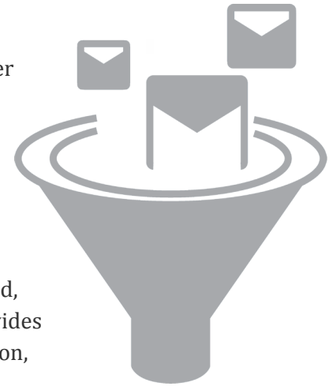
Table of Contents

Postscreen for Zimbra MTA Boosts Email Security

Starting with Zimbra Collaboration 8.7 and above, Zimbra introduces postscreen, which is an additional Anti-SPAM strategy.

Zimbra postscreen is an additional Anti-Spam strategy at the MTA level. The postscreen daemon provides protection against mail server overload. One postscreen process handles multiple inbound SMTP connections and decides which clients may talk to a Postfix SMTP server process.

In a typical scenario, good connections, bots and zombies talk with postscreen. Postscreen does basic checks and denies the connection if the message is from a bot or zombie. If the connection is not in the temporary whitelist, postscreen will pass the email to the local anti-SPAM and anti-virus engines, which can accept it or deny it as usual.



Zimbra postscreen should not be used on SMTP ports that receive mail from end-user clients (a.ka. Mail User Agent or MUAs). In a typical deployment, postscreen handles the MX service on TCP port 25, while MUA clients submit mail via the submission service on TCP port 587, which requires client authentication. Alternatively, a site could set up a dedicated, non-postscreen, “port 25” server that provides submission service and client authentication, but no MX service.

Zimbra postscreen maintains a temporary whitelist for clients that have passed a number of tests. When an SMTP client IP address is whitelisted, postscreen hands off the connection immediately to a Postfix SMTP server process. This minimizes the overhead for legitimate mail.

In a typical production setting, postscreen is configured to reject mail from clients that fail one or more tests. Zimbra postscreen logs rejected mail with the client address, helo, sender and recipient information.

Zimbra postscreen is not an SMTP proxy; this is intentional. The purpose is to keep spambots away from Postfix SMTP server processes, while minimizing overhead for legitimate traffic.

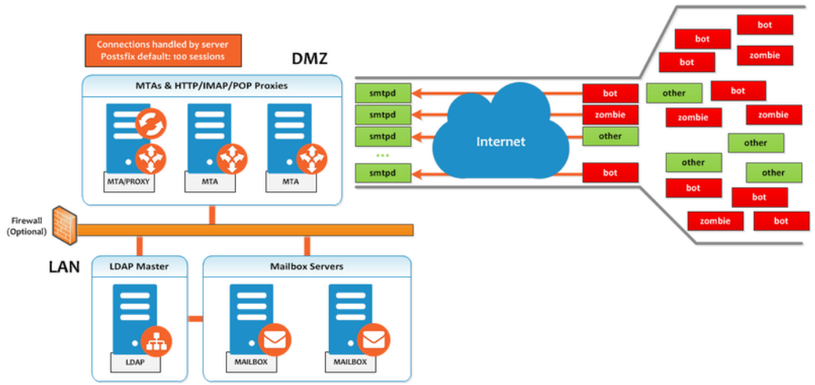
Postscreen for Zimbra:

- Provides additional anti-spam protection
- Protects agains mail server overload

How Does It Work?

Scenario without Postscreen

A typical scenario without postscreen and without other anti-spam security will suffer from the common problem where bot and zombies talk with the smtpd listeners that Zimbra offers.



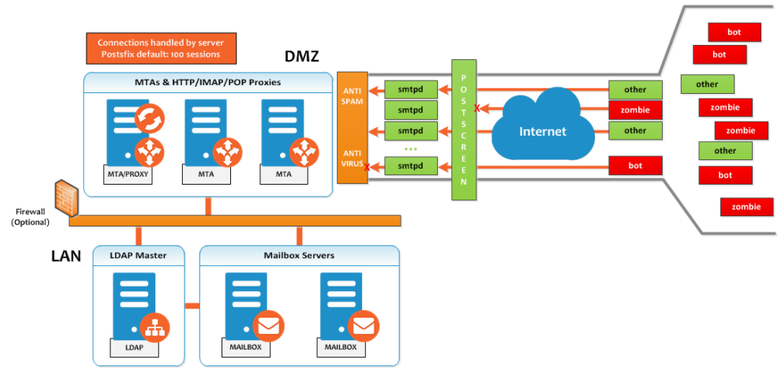
In this scenario, the good connections (called other in the diagram above) must wait until the bot or zombie finishes the communication, which sometimes can create a Timeout Error on postfix for the good connections.

```
Mar 01 19:29:54 zimbrauk postfix/smtpd[2466]: timeout after RCPT
from mail.exampl.com[60.60.60.70]
```

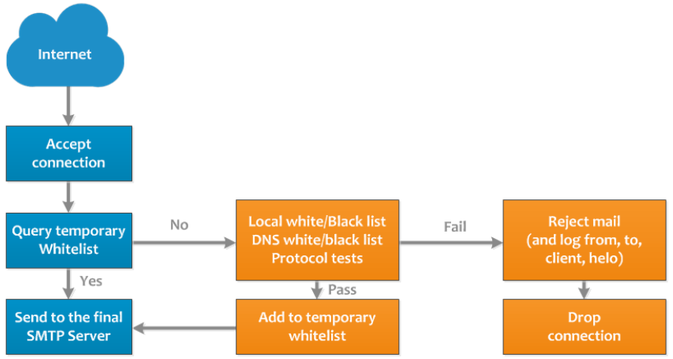

Scenario with Postscreen

A typical scenario with postscreen, the bot and zombies talk with postscreen. Postscreen does all the basic checks and can deny the connection if the message is from a bot or zombie. If the connection is not in the temporary whitelist, postscreen will pass the email to the local anti-SPAM and anti-virus engines, which can accept it or deny it as usual. You can see the mail flow in postscreen in the image below.

In this scenario, the good connections (called other in the diagram below) pass the postscreen security and talk directly with the smtp daemon, who scans the email as usual with AS/AV. All the bot or zombie emails are rejected by default.



Postscreen Workflow



Zimbra Attributes for Postscreen

For a complete list of new attirbutes for postscreen, please refer to the Zimbra Collaboration Postscreen wiki. You will also be able to link to the original postfix description help per attribute.

Please note the difference between the ignore, enforce and drop for certain attributes:

- ignore (default) - Ignore this result. Allow other tests to complete. Repeat this test the next time the client connects. This option is useful for testing and collecting statistics without blocking mail.
- enforce - Allow other tests to complete. Reject attempts to deliver mail with a 550 SMTP reply, and log the helo/sender/recipient information. Repeat this test the next time the client connects.
- drop - Drop the connection immediately with a 521 SMTP reply. Repeat this test the next time the client connects.

Enabling Postscreen

Zimbra postscreen comes enabled by default in ZCS 8.7 or above.

For a complete list of Zimbra attributes for postscreen, refer to the Zimbra Collaboration Postscreen wiki.

Configuring Postscreen: Example

Each scenario is different, so please tune the values in this example for your own environment. In this example, all values are set at the GlobalConfig level. This level is medium/high, enforcing a few attributes instead of ignore, changing them for a higher level of security.

```
zmprov mcf zimbraMtaPostscreenAccessList permit_mynetworks
zmprov mcf zimbraMtaPostscreenBareNewlineAction ignore
zmprov mcf zimbraMtaPostscreenBareNewlineEnable no
zmprov mcf zimbraMtaPostscreenBareNewlineTTL 30d
zmprov mcf zimbraMtaPostscreenBlacklistAction ignore
zmprov mcf zimbraMtaPostscreenCacheCleanupInterval 12h
zmprov mcf zimbraMtaPostscreenCacheRetentionTime 7d
zmprov mcf zimbraMtaPostscreenCommandCountLimit 20
zmprov mcf zimbraMtaPostscreenDnsblAction enforce
zmprov mcf zimbraMtaPostscreenDnsblSites 'b.barracudacentral.
org=127.0.0.2*7' zimbraMtaPostscreenDnsblSites 'dnsbl.inps.
de=127.0.0.2*7' zimbraMtaPostscreenDnsblSites 'zen.spamhaus.
org=127.0.0.[10;11]*8' zimbraMtaPostscreenDnsblSites 'zen.
spamhaus.org=127.0.0.[4..7]*6' zimbraMtaPostscreenDnsblSites
'zen.spamhaus.org=127.0.0.3*4' zimbraMtaPostscreenDnsblSites
'zen.spamhaus.org=127.0.0.2*3' zimbraMtaPostscreenDnsblSites
'list.dnswl.org=127.0.[0..255].0*-2'
zimbraMtaPostscreenDnsblSites 'list.dnswl.org=127.0.[0..255].1*-
3' zimbraMtaPostscreenDnsblSites 'list.dnswl.
org=127.0.[0..255].2*-4' zimbraMtaPostscreenDnsblSites 'list.
dnswl.org=127.0.[0..255].3*-5' zimbraMtaPostscreenDnsblSites
'bl.mailspike.net=127.0.0.2*5' zimbraMtaPostscreenDnsblSites
'bl.mailspike.net=127.0.0.[10;11;12]*4'
zimbraMtaPostscreenDnsblSites 'wl.mailspike.
net=127.0.0.[18;19;20]*-2' zimbraMtaPostscreenDnsblSites
'dnsbl.sorbs.net=127.0.0.10*8' zimbraMtaPostscreenDnsblSites
'dnsbl.sorbs.net=127.0.0.5*6' zimbraMtaPostscreenDnsblSites
'dnsbl.sorbs.net=127.0.0.7*3' zimbraMtaPostscreenDnsblSites
'dnsbl.sorbs.net=127.0.0.8*2' zimbraMtaPostscreenDnsblSites
'dnsbl.sorbs.net=127.0.0.6*2' zimbraMtaPostscreenDnsblSites
'dnsbl.sorbs.net=127.0.0.9*2'
zmprov mcf zimbraMtaPostscreenDnsblTTL 5m
zmprov mcf zimbraMtaPostscreenDnsblThreshold 8
zmprov mcf zimbraMtaPostscreenDnsblTimeout 10s
zmprov mcf zimbraMtaPostscreenDnsblWhitelistThreshold 0
zmprov mcf zimbraMtaPostscreenGreetAction enforce
zmprov mcf zimbraMtaPostscreenGreetTTL 1d
zmprov mcf zimbraMtaPostscreenNonSmtpCommandAction drop
zmprov mcf zimbraMtaPostscreenNonSmtpCommandEnable no
zmprov mcf zimbraMtaPostscreenNonSmtpCommandTTL 30d
zmprov mcf zimbraMtaPostscreenPipeliningAction enforce
zmprov mcf zimbraMtaPostscreenPipeliningEnable no
zmprov mcf zimbraMtaPostscreenPipeliningTTL 30d
zmprov mcf zimbraMtaPostscreenWatchdogTimeout 10s
zmprov mcf zimbraMtaPostscreenWhitelistInterfaces static:all
```

Testing Postscreen

You can set up DNSBLs, but leave it on ignore to test your postscreen implementation. Postscreen will log what it would have done. When you are satisfied that your implementation is correct, you can set values accordingly.

Below is a log example of the 550 error from postscreen:

```
Mar  1 02:03:26 edge01 postfix/postscreen[23154]: DNSBL rank 28
for [112.90.37.251]:20438
Mar  1 02:03:26 edge01 postfix/postscreen[23154]: CONNECT from
[10.210.0.161]:58010 to [10.210.0.174]:25
Mar  1 02:03:26 edge01 postfix/postscreen[23154]: WHITELISTED
[10.210.0.161]:58010
Mar  1 02:03:27 edge01 postfix/postscreen[23154]: NOQUEUE:
reject: RCPT from [112.90.37.251]:20438: 550 5.7.1 Service
unavailable; client [112.90.37.251] blocked using zen.spamhaus.
org; from=<hfxdgdsggfvfg@gmail.com>, to=<support@zimbra.com>,
proto=ESMTP, helo=<gmail.com>
Mar  1 02:03:27 edge01 postfix/postscreen[23154]: DISCONNECT
[112.90.37.251]:20438
```


IP Whitelist and Blacklist Using Postscreen

You can use postfix to easily whitelist or blacklist IPs by following these steps:

- Create /opt/zimbra/conf/postfix/postscreen_wblist
- Add entries. In this example, it is used as a blacklist. The IP range should be on CIDR format.

```
# Rules are evaluated in the order as specified.
# Blacklist 60.70.80.* except  60.70.80.91.
60.70.80.91/32 permit
60.70.80.0/24 deny
70.70.70.0/24 deny
```

- Set postscreen to use it:

```
zmprov mcf zimbraMtaPostscreenAccessList “permit_mynetworks,
cidr:/opt/zimbra/conf/postfix/postscreen_wblist”
zmprov mcf zimbraMtaPostscreenBlacklistAction enforce
```

- Wait for zmconfigd to pick up the change (up to 60 seconds)
- After the 60 seconds, or a manual restart of the MTA services, you will see something like this on the Log:

```
Jun 29 05:16:22 edge04e postfix/postscreen[7546]: BLACKLISTED
[70.70.70.100]:55699
```

Quick Note on for MTA on Cloud Environments

If you are using Amazon’s Elastic Load Balancer for handling SMTP traffic, including simple load-based autoscaling or load distribution that is aware of distribution across availability zones, you will need to configure as follows:

```
zmprov mcf zimbraMtaPostscreenUpstreamProxyProtocol haproxy
```

Verify that the change is in progress:

```
tail -f /var/log/zimbra.log
Jun 24 17:24:29 zre-ldap004 zmconfigd[17944]: Fetching All configs
Jun 24 17:24:29 zre-ldap004 zmconfigd[17944]: All configs fetched in
0.08 seconds
Jun 24 17:24:33 zre-ldap004 zmconfigd[17944]: Watchdog: service
antivirus status is OK.
Jun 24 17:24:33 zre-ldap004 zmconfigd[17944]: Var
zimbraMtaPostscreenUpstreamProxyProtocol changed from ‘None’ ->
‘haproxy’
Jun 24 17:24:33 zre-ldap004 zmconfigd[17944]: Rewrote: /opt/zimbra/
common/conf/tag_as_originating.re with mode 440 (0.01 sec)
Jun 24 17:24:33 zre-ldap004 zmconfigd[17944]: Rewrote: /opt/zimbra/
conf/postfix_header_checks with mode 440 (0.00 sec)
Jun 24 17:24:33 zre-ldap004 zmconfigd[17944]: Rewrote: /opt/zimbra/
common/conf/tag_as_foreign.re with mode 440 (0.01 sec)
Jun 24 17:24:33 zre-ldap004 zmconfigd[17944]: Rewrote: /opt/zimbra/
common/conf/master.cf with mode 440 (0.01 sec)
Jun 24 17:24:33 zre-ldap004 zmconfigd[17944]: Rewrote: /opt/zimbra/
conf/mta_milter_options with mode 440 (0.00 sec)
Jun 24 17:24:36 zre-ldap004 zmconfigd[17944]: All rewrite threads
completed in 2.93 sec
Jun 24 17:24:36 zre-ldap004 zmconfigd[17944]: controlProcess mta
restart (-1)
Jun 24 17:24:36 zre-ldap004 zmconfigd[17944]: CONTROL mta: bin/
zmmtactl reload norewrite
Jun 24 17:24:36 zre-ldap004 zmconfigd[17944]: mta reload initiated
from zmconfigd
Jun 24 17:24:36 zre-ldap004 saslauthd[20153]: server_exit     :
master exited: 20153
Jun 24 17:24:37 zre-ldap004 saslauthd[2925]: detach_tty      :
master pid is: 2925
Jun 24 17:24:37 zre-ldap004 saslauthd[2925]: ipc_init        :
listening on socket: /opt/zimbra/data/sasl2/state/mux
Jun 24 17:24:38 zre-ldap004 /postfix-script[2959]: refreshing the
Postfix mail system
Jun 24 17:24:38 zre-ldap004 postfix/master[20304]: reload -- version
3.1.1, configuration /opt/zimbra/common/conf
Jun 24 17:24:38 zre-ldap004 zmconfigd[17944]: All restarts completed
in 1.82 sec
```

And verify the changes by running this command:

```
postconf postscreen_upstream_proxy_protocol
postscreen_upstream_proxy_protocol = haproxy
```