SecOps tips for Zimbra

This article you will find practical tips to improve digital security and operations of Zimbra for administrators. Please use the tips responsibly and make sure to see if they are fitting for your situation and change them according to your needs. Configuration settings on this page are routinely validated by our QA team.

Install OS and Security updates as they are released

Have a procedure or policy in your organization that allows you to install security updates on Zimbra and the operating system *as they are released*.

This means you should check if security related patches for Zimbra have been released via https://wiki.zimbra.com/wiki/Security_Center

In your policy you should define the time to installation of a patch based on the CVSS Score or Zimbra Rating. For example:

Zimbra Rating	Time to install	Remarks
Low	Monthly scheduled maintenance window	
Medium	End of work-week	
High	Immediate or end of business day	Depending if mitigation is possible till the installation of the patch

- Make sure you are running a supported Operating System version
- Make sure you are running a supported Zimbra version

Zimbra does not distinguish between feature updates and security updates, so check the release note of each patch to find out if there are security related fixes.

Reboot your server after new kernel installation

Install pax OS package

All Zimbra administrators should make sure the pax package is installed on their Zimbra server. Pax is needed by Amavis to extract the contents of compressed attachments for virus scanning.

If the pax package is not installed, Amavis will fall-back to using cpio, unfortunately the fall-back is implemented poorly (by Amavis) and will allow an unauthenticated attacker to create and overwrite files on the Zimbra server, including the Zimbra webroot.

For most Ubuntu servers the pax package should already be installed as it is a dependency of

Zimbra. Due to a packaging change in CentOS, there is a high chance pax is not installed.

You should validate and install pax on all your systems as follows:

```
#Ubuntu
apt install pax

#CentOS 7 and derivatives
yum install pax

#CentOS 8 and derivatives
dnf install spax
```

Restart Zimbra using:

```
sudo su zimbra -
zmcontrol restart
```

Install a host firewall

Suggested firewall: iptables. Not recommended: firewalld (firewalld can fail leaving the system with open ports, as it has no fail-safe).

Further reading: system-config-firewall https://oracle-base.com/articles/linux/linux-firewall

Example firewall settings:

```
[root@zimbra1 ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

#block null packets
-A INPUT -p tcp --tcp-flags ALL NONE -j DROP

#block syn flood
-A INPUT -p tcp ! --syn -m state --state NEW -j DROP

#block XMAS packets
-A INPUT -p tcp --tcp-flags ALL ALL -j DROP

-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
```

```
# if you are connecting from known ip's you can comment out above and use this:
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT -s [add source ip here]

### iptables config for zimbra ###

# enable smtp
-A INPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT

# disabled http unecrypted
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
# disabled pop3
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 110 -j ACCEPT
# disabled imap
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 143 -j ACCEPT
# disabled ldap
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 389 -j ACCEPT -s [add source ip here]

#enable https
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
# disable smtp over ssl port
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 587 -j ACCEPT
# disable imap over ssl port
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 993 -j ACCEPT
# disable pop3 over ssl port
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 995 -j ACCEPT

#disable admin interface
#only use this if you cannot use a ssh tunnel
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 9071 -j ACCEPT -s [add source ip here]

-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT

[root@zimbra1 ~]# service iptables restart
```

Install a host firewall on ipv6

If you have ipv6 enabled on your host, configure iptables6 for that as well, similar to above.

Install fail2ban

Brute force attacks using SMTP authentication are common, consider installing fail2ban

https://wiki.zimbra.com/wiki/Configure_Fail2Ban_for_Zimbra_Server_with_route_instead_of_iptables_to_block_IPs

Accessing the Admin UI over a secure tunnel

Try and avoid opening up the Admin UI ports (7071,9071) on the public internet. If you can use an SSH tunnel like this:

```
ssh -L 7071:localhost:7071 user_with_low_privilege@zimbra.example.com
```

Then point your browser to: https://localhost:7071/zimbraAdmin/ (and ignore certificate warning)

Proxy Admin UI via port 9071

It is not recommended to expose the Admin UI to the Internet. Instead administrators should access Admin UI via a VPN. In any case you will need to make sure to proxy the Admin UI via Zimbra Proxy to make sure it uses the best TLS configuration. This means you should access Admin UI via the proxied port 9071, and deny access to port 7071 via a firewall. To enable this you should run as user Zimbra:

```
/opt/zimbra/libexec/zmproxyconfig -e -w -C -H `zmhostname`
zmproxyctl restart
```

SSH disable password authentication

Make sure to set `PasswordAuthentication no` in `/etc/ssh/sshd_config` and check the config file for duplicates of `PasswordAuthentication`. You may also want to set `GSSAPIAuthentication no`. Run `systemctl restart sshd` and make sure you can no longer log-in using passwords by going to a system that does not have your SSH private key.

You can also set up your firewall to only allow incoming connections to port 22 from known IP's in case you have a fixed IP.

Further reading: https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-2

Have a centralized logging server

Make sure to have reliable logs in case a hack occurs, see:

- https://github.com/zimbra/elastic-stack#installing-the-centralized-log-server

Disable imap/pop access per user

You can disable access via the firewall for all users. You can also disable imap and pop via the CoS settings. See: - https://zimbra.github.io/zimbra-9/adminguide.html#_email_messaging_features

Then you have the option to enable it for specific users.

Reason: IMAP is a common attack vector used to brute force account passwords, you can use

ActiveSync protocol or web based email.

Disable smtp authentication

This will disable the use of your Zimbra as a relay for your users, meaning the users can only send outgoing mail via ActiveSync or the WebUI.

Reason: SMTP is a common attack vector used to brute force account passwords, you can use ActiveSync protocol or web based email.

You can configure this in the Admin UI via Home → Configure → Global Settings → MTA disable authentication.

Further reading: https://imanudin.net/2018/06/13/zimbra-tips-how-to-restrict-sasl-login-access/

Verify Zimbra Postfix MTA trusted networks

If possible have no trusted network, such as:

```
zmprov ms zimbra.example.com zimbraMtaMyNetworks "127.0.0.0/8 [::1]/128 x.x.x.x/x [xxxx:xxxx:xxxx::x]/x"
zmmtactl restart
```

Where x.x.x.x/x is the ipv4 adress of the local machine, not a subnet and [xxxx:xxxx:xxxx::x]/x is the ipv6 address of the local machine.

Further reading: https://zimbra.github.io/zimbra-9/adminguide.html#setting_up_trusted_netorks

Use 2-factor authentication

Install enable and force your users to use 2FA.

For the Network Edition of Zimbra:

- https://wiki.zimbra.com/wiki/Zimbra_Two-factor_authentication

Make sure to disable the Trusted Devices feature. For example if you use the *default* Class of Service use:

```
zmprov mc default zimbraFeatureTrustedDevicesEnabled FALSE
# Verify what Classes of Service are available on your system by running:
zmprov gac
```

For the Open Source edition of Zimbra:

- https://gallery.zetalliance.org/extend/items/view/maldua-zimbra-ose-2fa-extension

Have backups and test them

Implement a procedure in your organization that deals with backup and restore specifically. Define a schedule for backup creation and restore testing. Consider the use of physical external backup media that is stored both locally and in a remote location. So you can restore off-line and have a good idea of restore time. Cloud backup's can be to slow to restore from if you have to do a full server restore.

If you use encryption for your backup media, make sure you have access to the decryption key. So that you can actually restore your backup in catastrophic events.

1. Make snapshots of your entire Zimbra server(s)
2. Make per user account backups
3. Check the retention of Zimbra NG backup and increase it if possible
4. Use a versioning system such as git to keep track of your configuration
5. Have an internal wiki to document your infrastructure, configuration and procedures

Further reading:

- https://wiki.zimbra.com/wiki/Zimbra_NG_Modules/Zimbra_NG_Backup/How_Backup_NG_Works
- https://wiki.zimbra.com/wiki/Zimbra_NG_Modules/Zimbra_NG_Backup/Disaster_Recovery
- https://wiki.zimbra.com/wiki/Zimbra_NG_Modules/Zimbra_NG_Backup/ExternalBackup

Disable indexing by search engines your log-in page

By default Zimbra allows the log-in page to be indexed in search engines on the public Internet. This is nice if you are an ISP, not so nice if you are a business.

To disable this:

```
zmprov mcf zimbraMailKeepOutWebCrawlers TRUE
zmprov mcf +zimbraResponseHeader "X-Robots-Tag: noindex"
zmmailboxdctl restart
```

Disable spoofing by local (authenticated) users

In Zimbra any user that has access to an account can send email on behalf of other users on the server. This can be done via SMTP if you have not disabled SMTP access as mentioned above.

You can force a match between FROM address and username via this guide:

- https://wiki.zimbra.com/wiki/Enforcing_a_match_between_FROM_address_and_sasl_username_8.5

Manage blocked attachments

Zimbra allows blocking incoming and outgoing attachments based on file type. By default this feature is not enabled, you can enable it in the Admin UI → Configure → Global Settings → Attachments.

It is suggested to add at least Windows executable file types. Example of a configuration:

```
zmprov gacf | grep Blocked | grep -v Common
zimbraAttachmentsBlocked: TRUE
zimbraMtaBlockedExtension: asd
zimbraMtaBlockedExtension: bat
zimbraMtaBlockedExtension: com
zimbraMtaBlockedExtension: exe
zimbraMtaBlockedExtension: hta
zimbraMtaBlockedExtension: js
zimbraMtaBlockedExtension: jse
zimbraMtaBlockedExtension: lnk
zimbraMtaBlockedExtension: pif
zimbraMtaBlockedExtension: scr
zimbraMtaBlockedExtension: shm
zimbraMtaBlockedExtension: shs
zimbraMtaBlockedExtension: vbe
zimbraMtaBlockedExtension: vbs
zimbraMtaBlockedExtension: vbx
zimbraMtaBlockedExtension: wsf
zimbraMtaBlockedExtension: wsh
zimbraMtaBlockedExtensionWarnAdmin: TRUE
zimbraMtaBlockedExtensionWarnRecipient: TRUE
```

Further reading:

- https://zimbra.github.io/zimbra-9/adminguide.html#_blocking_email_attachments_by_file_type

Configure Zimbra anti-spam

Set-up RBL's and review and implement the information at:

- https://www.missioncriticalemail.com/2019/03/21/zimbra-anti-spam-best-practices-2019/

Disable proxy servlet

Zimbra has a Java servlet that is used by some Zimlets to proxy requests from the user's web-browser via the Zimbra server to 3rd party servers. This is used by for example the Webex Zimlet. Many deployments do actually not use any Zimlet that uses the proxy servlet. The proxy servlet is a security issue as potentially it can be used to have the Zimbra server make requests to itself and bypass the firewall. Do not use wildcards in `zimbraProxyAllowedDomains` and empty it entirely if you

do not use it:

```
zmprov mc default zimbraProxyAllowedDomains ""
```

Do this for all your CoS'es, you can find them with `zmprov gac`.

Admin accounts

- Create a separate admin account with a strong password that you do not use on daily basis.
- Know your admins, and disable stale admin accounts.

If you want even more security, you can disable your admin account via the command-line and enable it only when needed.

Session time

By default Zimbra user session time is 3 days. This means that if the user leaves a laptop open with a signed-on Zimbra, the user is not asked to log-in again for 3 days.

Consider:

- decreasing zimbraAuthTokenLifetime
- decreasing zimbraAdminAuthTokenLifetime
- configuring zimbraMailIdleSessionTimeout (default is disabled)

This can be configured via the CoS settings, further reading:

- https://wiki.zimbra.com/wiki/Change_user_and_admin_web_console_session_idle_time_out

Get the last logon timestamp

You can configure Zimbra to store the time when a successful log-in occurred on an account. Originally this feature was only meant to find stale accounts and the timestamp is only updated once every 3 days.

You can have it updated once every second by configuring it like:

```
zmprov mcf zimbraLastLogonTimestampFrequency 1s
```

Please be advised that you must use SSDB for this to work, further reading:

- https://wiki.zimbra.com/wiki/Ssdb

Install and enable haveged

Further reading:

- https://www.digitalocean.com/community/tutorials/how-to-setup-additional-entropy-for-cloud-servers-using-haveged

Install and enable chronyd

Futher reading:

- https://www.tecmint.com/install-chrony-in-centos-ubuntu-linux/

Pre-authentication

If you use pre-authentication, use a SOAP implementation. The pre-authentication REST API does not support logging the originating IP.

Here is an example PHP Script that implements a SOAP pre-authentication. This particular implementation can be dropped onto a SimpleSamlPHP IDP server so that it uses SimpleSamlPHP to validate if the user is logged in. A copy can be found at: https://github.com/Zimbra-Community/zimbra-tools/blob/master/pre-auth-soap-saml.php

```
<?php

require_once('/var/www/simplesaml/lib/_autoload.php');

$as = new SimpleSAML_Auth_Simple('default-sp');
$as->requireAuth();

if($as->isAuthenticated()) {
    $attributes = $as->getAttributes();
    $email = $attributes['mail'][0];
    preauth($email);
}
else {
    header("Location: URL TO YOUR LOGIN PAGE HERE");
}

function hmac_sha1($key, $data)
{
    // Adjust key to exactly 64 bytes
    if (strlen($key) > 64) {
        $key = str_pad(sha1($key, true), 64, chr(0));
    }
    if (strlen($key) < 64) {
        $key = str_pad($key, 64, chr(0));
    }

    // Outter and Inner pad
    $opad = str_repeat(chr(0x5C), 64);
    $ipad = str_repeat(chr(0x36), 64);

    // Xor key with opad & ipad
    for ($i = 0; $i < strlen($key); $i++) {
        $opad[$i] = $opad[$i] ^ $key[$i];
        $ipad[$i] = $ipad[$i] ^ $key[$i];
    }
```

```
    return sha1($opad.sha1($ipad.$data, true));
}

function preauth($email)
{
   header("Cache-Control: no-cache, must-revalidate"); // HTTP/1.1
   header("Expires: Sat, 26 Jul 1997 05:00:00 GMT"); // Date in the past

   $domain = "https://zimbra.example.com";
   $preAuthKey = "PUT YOUR PREAUTH KEY HERE";

   $time = round(microtime(true) * 1000);
   $input_xml='<?xml version="1.0" encoding="utf-8"?>' .
               '<soapenv:Envelope ' .
                   'xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ' .
                   'xmlns:api="http://127.0.0.1/Integrics/Enswitch/API" ' .
                   'xmlns:xsd="http://www.w3.org/2001/XMLSchema" ' .
                   'xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">' .
                   '<soapenv:Body>' .
                   '<AuthRequest xmlns="urn:zimbraAccount">' .
                   '<account>'.$email.'</account>' .
                   '<preauth timestamp="'.$time.'"
expires="0">'.hmac_sha1($preAuthKey,$email.'|name|0|'.$time).'</preauth>' .
                   '</AuthRequest>' .
                   '</soapenv:Body>' .
               '</soapenv:Envelope>';

       //setting the curl parameters.
        $ch = curl_init();
        $ip = $_SERVER['REMOTE_ADDR'];
        curl_setopt($ch, CURLOPT_URL, $domain."/service/soap/preauth");
        curl_setopt($ch, CURLOPT_POSTFIELDS, $input_xml);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 300);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
        curl_setopt($ch, CURLOPT_HTTPHEADER, array("X-Forwarded-For: $ip"));
        $data = curl_exec($ch);
        curl_close($ch);

        $token = preg_match("/<authToken>.*?<\/authToken>/",$data,$matches);

        if(token)
        {
           if($matches[0])
           {
              $matches[0] = substr($matches[0], 11);//remove <authToken>
              $matches[0] = substr($matches[0], 0, strlen($matches[0])-12);//remove </authToken>
              header ("Location: ".$domain."/service/preauth?authtoken=$matches[0]");
           }
           else
           {
              header ("Location: ".$domain);
           }
        }
        else
        {
           header ("Location: ".$domain);
        }
}

?>
```

Further reading:

- https://wiki.zimbra.com/wiki/Preauth

Disable authFallBackToLocal

If you are using external authentication, for example Active Directory or SAML, make sure to disable the use of local Zimbra passwords. Please be advised that global admin accounts can always use the Zimbra internal LDAP authentication so set a passphrase or long password on global admin accounts.

To disable it on a domain run:

```
zmprov md example.com zimbraAuthFallbackToLocal FALSE
```

Disable alias login

By default users can log-in to Zimbra using their account name but also any of their alias addresses. To allow only log-in with the account name run:

```
zmlocalconfig -e alias_login_enabled=false
```

Disable sharing in CoS

In the Admin UI go to Configure → Class of service → Features and verify all the features related to sharing and disable what you can.

Disable external IMAP/POP in CoS

In the Admin UI go to Configure → Class of service → Features and disable external IMAP/POP access.

Account management

Have or create a procedure in your organization so that new users get correct access rights, but also so that you can disable accounts for users that are leaving the organization.

Implement automated account creating and removal based on your procedure. Remove stale accounts and also verify no one is using the External Accounts feature in Zimbra and if needed automatically purge External Accounts.

Do not use weak ciphers in TLS certificates

This wiki page describes how to configure Zimbra with strong TLS. It assumes you are using Zimbra proxy and deny traffic to non proxied ports via a host firewall.

Further reading:

- https://wiki.zimbra.com/wiki/Cipher_suites

Log the correct origination IP

Check your logs and see if they log the correct origination IP and not only that of the proxy server.

If you only see internal IP's in the log, you need to configure Zimbra to log the IP from the X-Forwarded-For header. https://wiki.zimbra.com/wiki/Log_Files#Logging_the_Originating_IP

For IPv6 configuration of originating IP see: https://wiki.zimbra.com/wiki/Configuring_for_IP_V6

```
zmlocalconfig zimbra_http_originating_ip_header
zimbra_http_originating_ip_header = X-Forwarded-For

zmprov mcf +zimbraMailTrustedIP 127.0.0.1
zmprov mcf +zimbraMailTrustedIP <proxy ip here>
zmprov mcf +zimbraMailTrustedIP <more proxy here>
zmcontrol restart
```

Further reading:

- https://wiki.zimbra.com/wiki/Log_Files#Logging_the_Originating_IP

zimbraPrefShortEmailAddress

By default Zimbra only lists the name of the sender of an email. For example: `CEO'' you can configure Zimbra to display the entire email address so that it shows` CEO" ceo@example.com. You can do so like this:

```
zmprov mc default zimbraPrefShortEmailAddress FALSE
```

Do this for all your CoS'es, you can find them with `zmprov gac`. Not yet supported on Modern UI.

Setting the `SameSite` cookie attribute

By default value will be Strict but user can change it to Lax, None and ""(empty) for no `SameSite` attribute.

```
zmlocalconfig -e zimbra_same_site_cookie="Strict"
```

Further reading: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite

Prevent Host header injection vulnerability in Zimbra

This is an old issue but Zimbra installations can have a very long life span, in addition it is a good precaution to validate your configuration, just in case. Zimbra Proxy has the ability to strictly enforce which values are allowed in the Host header passed in by the client.

This is *enabled by default* on ***new installations*** but left *disabled* for ***upgrades*** from previous versions unless toggled during the installation.

The functionality may be altered by setting the `zimbraReverseProxyStrictServerNameEnabled` boolean configuration option followed by restarting the proxy server.

- TRUE – strict server name enforcement enabled
- FALSE – strict server name enforcement disabled

```
zmprov mcf zimbraReverseProxyStrictServerNameEnabled TRUE
```

When the strict server name functionality is enabled, additional valid server names may be specified using the `zimbraVirtualHostName` and `zimbraVirtualIPAddress` configuration items at the domain level.

```
zmprov md example.com zimbraVirtualHostName mail.example.com zimbraVirtualIPAddress 1.2.3.4
```

Only one virtual ip address is needed per domain although more than one is acceptable.

In case you have pointed multiple DNS domain names to your Zimbra server, all these domains must be configured as Zimbra Virtual Hosts. If you set `zimbraReverseProxyStrictServerNameEnabled` to `true`, Zimbra will show an error 400 page for any domains not configured in Zimbra. It will also prevent others from making rogue reverse proxies on domains out of your control.

Without changing anything you can validate your configuration using:

```
zmprov gacf | grep -i zimbraReverseProxyStrictServerNameEnabled
zmprov gs `zmhostname` | grep -i zimbraReverseProxyStrictServerNameEnabled
```

Prevent Zimbra from sending X-Mailer

```
zmprov mcf zimbraSmtpSendAddMailer FALSE
```

Zimbra installation integrity check

A script that allows Zimbra administrators to create checksums of all the files in a Zimbra installation. The output of the script can be used to identify unintended changes and newly created files. Such changes can for example be caused by hackers. https://wiki.zimbra.com/wiki/Integrity_check

Set up a notification for TLS certificate expiration

Also when you are NOT using Zimbra to serve our TLS to your users!

https://wiki.zimbra.com/wiki/Notification_for_certificate_expiration