zimbra®
A SYNACOR PRODUCT

Zimbra Security and PCI DSS

A Zimbra Collaboration Whitepaper

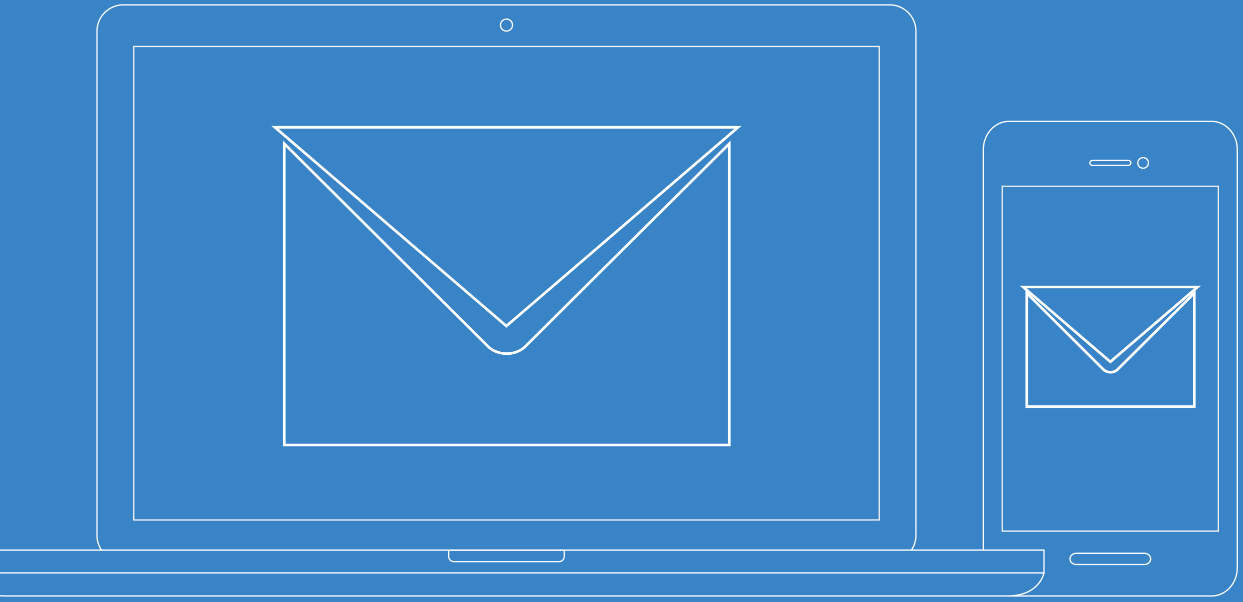


Table of Contents

Overview

Emails are accessible to most people anywhere and anytime. However, when it comes to handling sensitive data like credit card information, you need to rethink your messaging options.

The Payment Card Industry Data Security Standard (PCI DSS) is an information security framework intended to help merchants and service providers protect credit and debit card transactions from data breaches.

This whitepaper describes the PCI DSS framework and how you should structure your Zimbra configuration to achieve compliance.

Background

The purpose of the PCI DSS is to protect cardholder data (CHD) and sensitive authentication data (SAD) from unauthorized access and loss. CHD consists of the Primary Account Number (PAN), cardholder name, expiration date, and service code. SAD includes the full track data (magnetic-stripe data or equivalent on a chip), CAV2/CVC2/CVV2/CID, and PIN blocks.

Email messages typically pass through several servers before reaching the recipient. The messages in these emails are readable during transit if they travel unprotected.

According to the PCI requirements, companies are required to protect cardholder data even during transit. However, sending sensitive information such as cardholder data using your standard email means that the data is vulnerable.

Requirement 4.1 dictates that you should not transmit unencrypted credit card data over open, public networks. **Requirement 4.2** also states that you should not transmit unencrypted PANS via messaging technologies such as email.

In this case, the subject in question is email communication of sensitive data. Some solutions include:

1. **End-to-end encryption**: Email communications are widely considered private and secure, which they are not. To ensure privacy and security, companies can use end-to-end encryption, which means that only the recipient can decrypt the message. Full end-to-end encryption also means that even service providers cannot read the messages.

2. **Educate and train your employees**: In addition to the security measures and technologies that you implement, you also need to train your employees to maintain compliance. Even when using an end-to-end encrypted email service, you need an encryption key that is provided to the user's account. If your employee mishandles this encryption key, a cyber-criminal can use the key to sift through your emails. Train your employees to safeguard their encryption keys.

3. **Phishing prevention**: Cybercriminals understand that small businesses hold valuable information and have struggling IT departments due to limited budgets. They use the weakest link (humans) in a secure system to gain critical access. Big companies can invest in phishing prevention technologies, but smaller businesses can opt to focus on more cost-effective strategies such as employee education.

If emailing credit card information, be PCI compliant by:

- Changing your process and system to be PCI compliant by encrypting email. Do not send clear text credit cards via unencrypted email.
- Training your employees to forbid sending or receiving customer card data.
- Ensuring policies state that unencrypted data (credit card numbers, national ID numbers, etc.) cannot be sent via email or other end-user technologies.

What happens if you receive email with personal information by accident?

- Inform the sender to stop. Educate them about the dangers of using email to send credit card information. Make sure you don't include the original email in your response.
- Talk to your IT department about the best way to dispose of this message securely. (Many servers have journals/archives for audit and compliance or long term backup.)
- Be sure there is training for employees, so they know how to handle this situation.

TLS Parameters

Transport Layer Security (TLS) encrypts data sent over the internet to ensure that eavesdroppers and hackers are unable to see what you transmit, which is particularly useful for private and sensitive information such as passwords, credit card numbers, and personal correspondence. (Further reading: https://www.internetsociety.org/deploy360/tls/basics)

The level of security provided by TLS is most affected by the protocol version (i.e. 1.2, 1.3, etc.) and the allowed cipher suites. Most TLS clients and servers support multiple alternatives, so they have to negotiate when establishing a secure connection to select a common TLS version and cipher suite.

TLS Protocol Version

Concerning TLS version support, the NIST guidelines defined in **NIST SP 800-52r2** state the following:

> Servers that support government-only applications shall be configured to use TLS 1.2 and should be configured to use TLS 1.3 as well. These servers should not be configured to use TLS 1.1 and shall not use TLS 1.0, SSL 3.0, or SSL 2.0.

NIST guidelines are specific to the US government. SP 800-52r2 specifies a variety of acceptable cipher suites for TLS 1.2 and earlier. The standard does not require support for any particular cipher suites but offers guidance on choosing stronger ones.

For small scale systems, it is recommended that only a subset of these algorithms be used. Allowing more cipher suites widens the attack surface to your server. For more information on cipher suites, refer to this link https://www.ssl.com/guide/tls-standards-compliance/.

Zimbra Security and PCI Compliance

The Zimbra MTA and Nginx services are internet facing services that protect and manage client connections to your internal services. Ensure that you prevent man-in-the-middle attacks and enhance email confidentiality by tweaking the configuration as per your security requirements. Here are some of the recommendations.

Generate ssl_ciphers

Encryption is always evolving. It is recommended to use Mozilla's SSL Config generator (https://sslconfig.mozilla.org) to get the list of ssl_ciphers suited for your environment.

Select `Intermediate` and `Nginx` as the Zimbra proxy is based on Nginx. Based on your Zimbra server version, select the appropriate nginx and OpenSSL versions (`rpm -qa | grep -e zimbra-openssl -e zimbra-nginx`). The tool also reports the oldest supported clients that work with this configuration: Firefox 27, Android 4.4.2, Chrome 31, Edge, IE 11 on Windows 7, Java 8u31, OpenSSL 1.0.1, Opera 20, and Safari 9.

From the generated config file copy the value from ssl_ciphers:

```
ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
```

Server Configuration

As a zimbra user, run the following commands:

```
zmprov mcf zimbraMtaSmtpdTlsCiphers high
zmprov mcf zimbraMtaSmtpdTlsProtocols '!SSLv2,!SSLv3'
zmprov mcf zimbraMtaSmtpdTlsMandatoryCiphers high
zmprov mcf zimbraMtaSmtpdTlsExcludeCiphers 'aNULL,MD5,DES'
zmprov -l mcf zimbraReverseProxySSLCiphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384' ①
zmprov mcf zimbraReverseProxySSLProtocols TLSv1.2
zmprov mcf +zimbraReverseProxySSLProtocols TLSv1.3
```

① - Based on the ssl_ciphers generated in the above section.

`zmconfigd` will also ensure that postfix is updated.

Also configure Zimbra mailbox to allow the use of `TLSv1.3`. Open `/opt/zimbra/conf/localconfig.xml` in a text editor and find the line `mailboxd_java_options`. Set `TLSv1.2,TLSv1.3` in `https.protocols` and `jdk.tls.client.protocols`.

```
<key name="mailboxd_java_options">
  <value>
    -server -Dhttps.protocols=TLSv1.2,TLSv1.3
    -Djdk.tls.client.protocols=TLSv1.2,TLSv1.3
    -Djava.awt.headless=true -Dsun.net.inetaddr.ttl=${networkaddress_cache_ttl}
    -Dorg.apache.jasper.compiler.disablejsr199=true
    -XX:+UseG1GC -XX:SoftRefLRUPolicyMSPerMB=1 -XX:+UnlockExperimentalVMOptions
    -XX:G1NewSizePercent=15 -XX:G1MaxNewSizePercent=45
    -XX:-OmitStackTraceInFastThrow -verbose:gc
    -Djava.net.preferIPv4Stack=true
    -Xlog:gc*=info,safepoint=info:file=/opt/zimbra/log/gc.log:time:filecount=
20,filesize=10m
  </value>
</key>
```

Restart the proxy and mailbox services:

```
zmproxyctl restart
zmmailboxdctl restart
```

As of writing this documents, the following packages are now GA:

- OpenSSL 1.1.1h support for TLS 1.3
- OpenSSL 1.1.1h with FIPS module support
- Postfix 3.5.6 support for TLSv1.3
- Nginx 1.19.0 support for TLSv1.3

Zimbra 9.0 “Kepler” Patch 13 and 8.8.15 “James Prescott Joule” Patch 20 introduced these features. If you are running a version older than the specified one, you will not have support for TLS 1.3.

Other Security Considerations

DH Parameters

Generating DH parameters improves key exchange and mitigates against Logjam attack. To understand more about Diffie-Hellman (DH) key exchange parameters, read this article: https://weakdh.org/.

Run as `zimbra` user:

```
wget https://raw.githubusercontent.com/internetstandards/dhe_groups/
master/ffdhe4096.pem -O /etc/ffdhe4096.pem

su - zimbra

zmprov mcf zimbraSSLDHParam /etc/ffdhe4096.pem
```

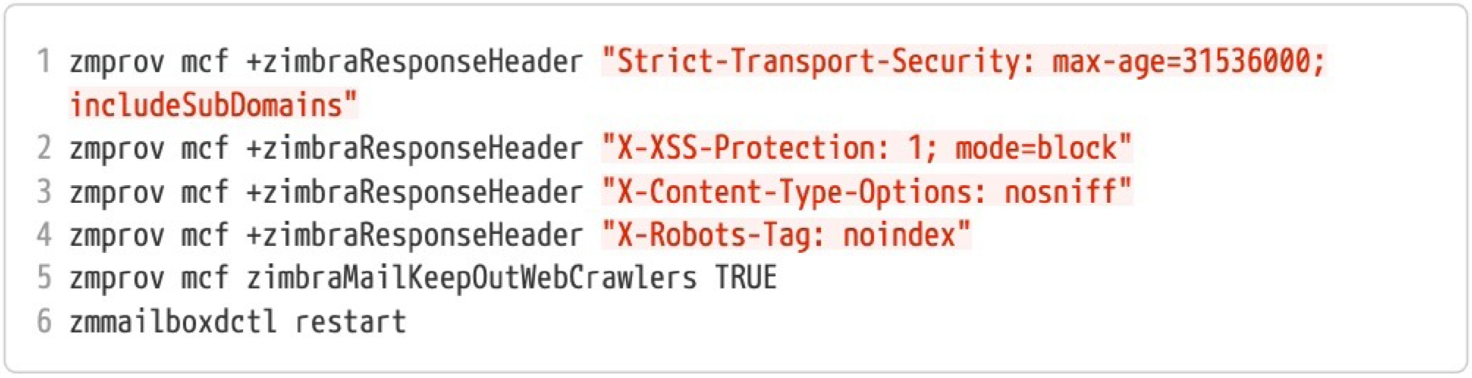
Restart all Zimbra services!

To improve performance of the random number generator, install haveged. This will help with both speed and reliability of the dhparam operation. More information here - https://www.digitalocean.com/community/tutorials/how-tosetup-additional-entropy-for-cloud-servers-using-haveged

Additional HTTP Headers

The following headers will:

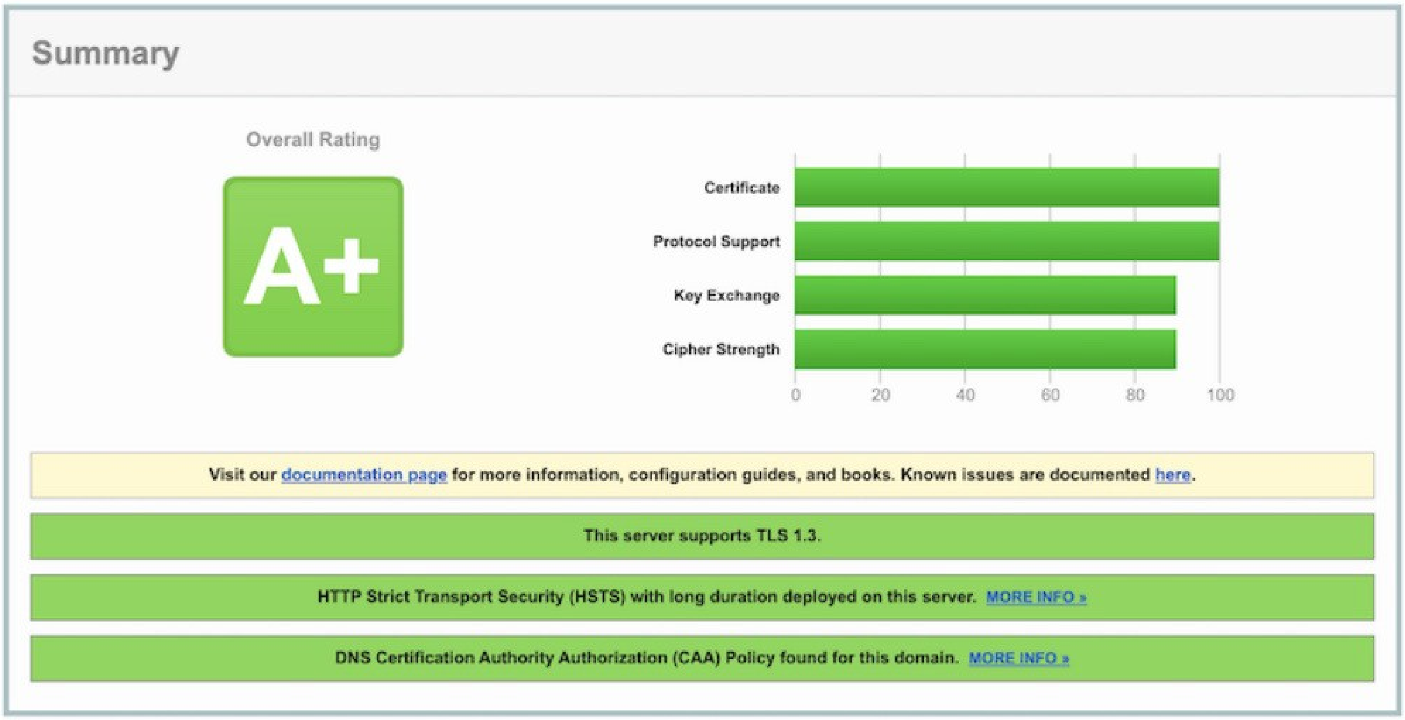
- Enable HTTP Strict Transport Security (HSTS)
- Disable search indexing of your server by Google, etc.



```
zmprov mcf +zimbraResponseHeader "Strict-Transport-Security: max-age=31536000; includeSubDomains"
zmprov mcf +zimbraResponseHeader "X-XSS-Protection: 1; mode=block"
zmprov mcf +zimbraResponseHeader "X-Content-Type-Options: nosniff"
zmprov mcf +zimbraResponseHeader "X-Robots-Tag: noindex"
zmprov mcf zimbraMailKeepOutWebCrawlers TRUE
zmmailboxdctl restart
```

Validation

Validate your settings online using SSL Labs. Go to https://www.ssllabs.com/ssltest/analyze.html and enter the domain name of your Zimbra server. If you followed the steps in this article, you should receive an A+ score, and there should be no mention of weak ciphers in the report.



Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This server supports TLS 1.3.

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. MORE INFO »

DNS Certification Authority Authorization (CAA) Policy found for this domain. MORE INFO »