



Zimbra SSL SNI for HTTPS

A Zimbra Collaboration Whitepaper

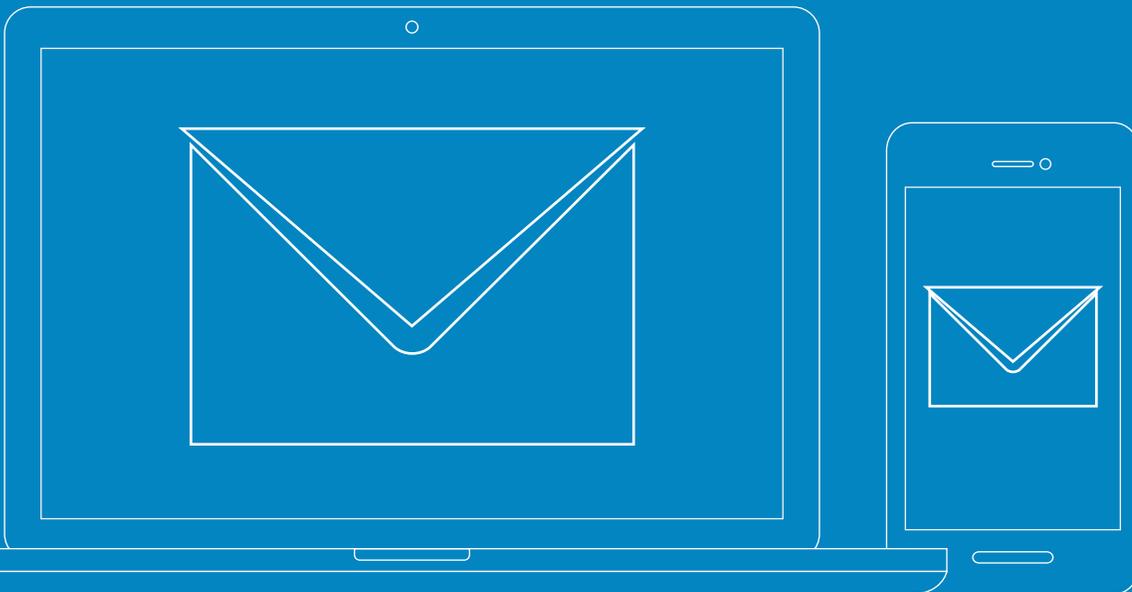


Table of Contents

Zimbra SSL SNI Reduces SSL Configuration Complexity	3
Getting Started	4
<i>Prerequisites</i>	<i>4</i>
<i>Browser Support for SNI</i>	<i>4</i>
<i>SSL SNI Workflow</i>	<i>5</i>
Configuring the IP Address per Domain	5
<i>Verifying and Preparing the Certificates</i>	<i>6</i>
<i>Deploying the Certificate(s) on the Domain</i>	<i>7</i>
<i>Proxy Check</i>	<i>8</i>
<i>Rewrite and Restart Proxy</i>	<i>8</i>
<i>Testing</i>	<i>8</i>
<i>Troubleshooting</i>	<i>8</i>

Zimbra SSL SNI Reduces SSL Configuration Complexity

Zimbra SSL SNI (Server Name Indication) allows the proxy server to present multiple certificates on the same IPv4 address and TCP port number, allowing multiple secure (HTTPS) domains to be served on the same IP address without using the same multi-SAN certificate.

Zimbra SSL SNI is excellent for Business Service Providers who host anywhere from five, up to thousands of domains. With Zimbra SSL SNI, they can protect their users without being worried about the number of (already limited) IPv4 addresses per domain.



Server Name Indication (SNI) is an extension to the TLS computer networking protocol by which a client indicates which hostname it is attempting to connect to at the start of the handshaking process. This allows a server to present multiple certificates on the same IP address and TCP port number and hence allows multiple secure (HTTPS) websites (or any other Service over TLS) to be served off the same IP address without requiring all those sites to use the same certificate. It is the conceptual equivalent to HTTP/1.1 name-based virtual hosting, but for HTTPS. The desired hostname is not encrypted, so an eavesdropper can see which site is being requested.

To make SNI useful, as with any protocol, the vast majority of visitors must use web browsers that implement it. Users whose browsers do not implement SNI are presented with a default certificate and hence are likely to receive certificate warnings. Source:Wikipedia

Zimbra SSL SNI:

- Allows the proxy server to present multiple certificates on the same IPv4 address & TCP port number
- Allows multiple secure HTTPS domains to be served on the same IP address without using the same multi-SAN certificate

Getting Started

Zimbra Collaboration support of SSL SNI requires and uses features of the proxy service, which is required beginning with Zimbra Collaboration 8.7.

Prerequisites

- Zimbra proxy service must be installed and enabled on the server. In a multi server environment, these steps should be performed on the proxy node.
- You should have a signed certificate + matching key pair and the trusted chain certs from your CA (Certificate Authority). This is a common issue, so please, make sure you check your files before deploying them.

You can bind Multiple SSL Certificates to just one ipv4 address, which will pair to the respective domain names. For example:

```
1.1.1.1 => example.com
1.1.1.1=> otherdomain.com
```

You could have another IPv4 address with another group of SSL Certificates. You can even have different types of SSL Certificates:

```
3.3.3.3 => yetanotherdomain.com (A Comodo Wildcard SSL Certificate)
3.3.3.3 => thisisanotherdomain.com (A free Let's Encrypt SSL Certificate)
3.3.3.3 => customer001.net (A RapidSSL Certificate)
etc.
```

Browser Support for SNI

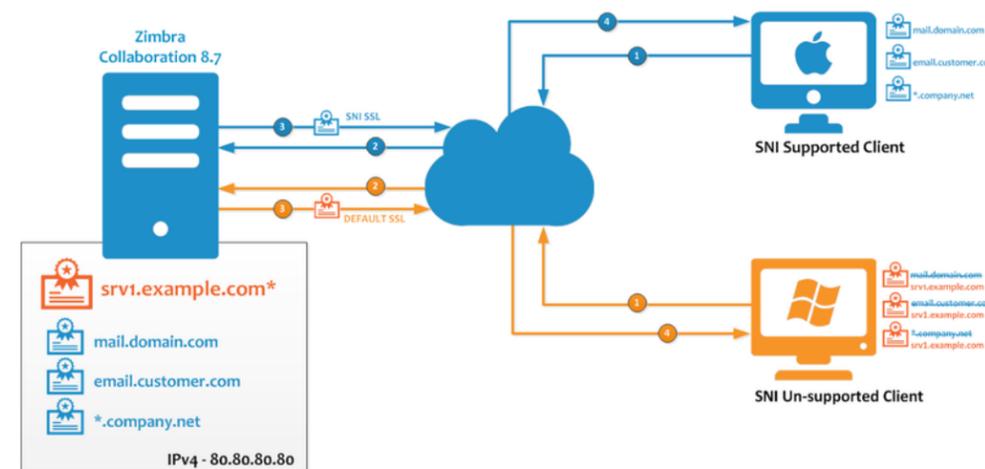
For a list of browsers that support SNI, refer to the [Zimbra Multiple SSL Certificates, Server Name Indication \(SNI\) for HTTPS](#) wiki.

Note, however, that it is the responsibility of the web browser software to support the application of SNI.

Zimbra SSL SNI requires and uses features of the proxy service. The proxy service is required beginning with Zimbra Collaboration 8.7

SSL SNI Workflow

Below is a diagram of the SSL SNI workflow.



Configuring the IP Address per Domain

Follow these steps to configure the IP address per domain:

1. Add the new domain, in this case example.com.
2. Set `zimbraVirtualHostName` to `mail.example.com` and `zimbraVirtualIPAddress` to `1.2.3.4`.
3. Make sure the `zimbraVirtualHostName` is set to the name that will be used to access the domain (URL) and the SSL certificate is signed for the same name.

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

NOTE: If the server is behind a firewall and NAT'ed with an external address, make sure external requests for "mail.example.com" hit the aliased IP address and not the actual local IP address of server.

Verifying and Preparing Certificates

You should have three files received from the CA (might vary depending on the CA):

- The server (domain) certificate
- Two chain certs

You should also have an existing key file, which was used to generate the csr.

1. Save the example.com certificate, key and chain files to a directory /tmp/example.com. You can receive single or multiple chain certs from your CA. Here we have two chain certs from the CA: example.com.root.crt and example.com.intermediate.crt.

```
ls /tmp/example.com
example.com.key
example.com.crt
example.com.root.crt
example.com.intermediate.crt
```

2. Add the chain certs to a single file called example.com_ca.crt

```
cat example.com.root.crt example.com.intermediate.crt >>
example.com_ca.crt
```

3. Confirm that the key and certificate matches and chain certs completes the trust. As the zimbra user:

```
/opt/zimbra/bin/zmcertmgr verifycrt comm /tmp/example.com/
example.com.key /tmp/example.com/example.com.crt /tmp/example.
com/example.com_ca.crt
```

Check the output; it should say something like what is shown below. If it does not, make sure you have the correct key and chain cert files.

```
** Verifying '/tmp/example.com.crt' against '/tmp/example.com.key'
Certificate '/tmp/example.com.crt' and private key '/tmp/example.
com.key' match.
** Verifying '/tmp/example.com.crt' against '/tmp/example.
com_ca.crt'
Valid certificate chain: /tmp/example.com.crt: OK
```

Deploying the Certificate(s) on the Domain

1. Add the domain certificate and chain files to a single file called example.com.bundle

```
cat example.com.crt example.com_ca.crt >> example.com.bundle
```

2. Run the following command as the zimbra user to save the certificates and key in LDAP:

```
/opt/zimbra/libexec/zmdomaincertmgr savecrt example.com
example.com.bundle example.com.key
** Saving domain config key zimbraSSLCertificate...done.
** Saving domain config key zimbraSSLPrivateKey...done.
```

The syntax is:

```
/opt/zimbra/libexec/zmdomaincertmgr savecrt <domainname>
<certificate with chain certs> <keyfile>
```

3. Run the following command as the zimbra user to deploy the domain certificate. This will save the certificate and key as /opt/zimbra/conf/domaincerts/example.com:

```
/opt/zimbra/libexec/zmdomaincertmgr deploycrt
** Deploying cert for example.com...done.
```

Proxy Check

Run these commands on the proxy hosts or on the server if it's a single-server deployment.

- `zimbraReverseProxySNIEnabled` should be set to `TRUE` in server and global config

```
zmprov mcf zimbraReverseProxySNIEnabled TRUE
```

Rewrite and Restart Proxy

Restart the proxy to rewrite the changes to proxy config.

```
zmpoxyctl restart
```

Once the restart is successful, try to access the domain using the URL, which is set in `zimbraVirtualHostName` over https. Also check the certificate loaded in the browser. In this case, the URL will be `https://example.com`

Testing

You can now go to a web browser and check that for each different `zimbraVirtualHostName` you see a different SSL certificate and that its details are correct for that virtualhostname.

Troubleshooting

- If you do not see the correct domain cert by accessing the domain with its `zimbraVirtualHostName` (example.com), make sure that the https connection from the internet/intranet is going to the server's local IP address, which is defined in `zimbraVirtualIPAddress`. Also make sure you have activated `zimbraReverseProxySNIEnabled` to `TRUE`.
- If you are using multiple proxy servers or adding new proxy servers, make sure you copy all the contents of `/opt/zimbra/conf/domaincerts/` to all the proxy servers. Otherwise the proxy service