SSL Quick Reference

Get Certificate from remote host

Ever wondered when your SMTP SSL Certificates are up for renewal? What DNS entries your certificates have? A quick and dirty way of doing it from the command line was shown at

- https://serverfault.com/questions/131627/how-to-inspect-remote-smtp-servers-tls-certificate#131628
- https://stackoverflow.com/questions/13127352/how-to-check-subject-alternative-names-for-a-ssl-tls-certificate

Note: the discussions covered other things, and are well worth a 5 minute read.

This is a quick and dirty that will get the certificate (and a lot of other stuff), but the certificate is in its MIME encoded format.

```
printf 'quit\n' | \
openssl s_client -connect smtp.example.com:25 -starttls smtp
```

This basically makes a connection to smtp.example.com on port 25, issuing a starttls, then sends the *quit* command which logs out. The openssl command retrieves the the entire conversation, which includes the certificate, and displays it on the

You can do the same thing for other ports, like 587 for submission. If you want to test the SSL port (465), just remove the *-starttls smtp* from the command:

```
printf 'quit\n' | \
openssl s_client -connect smtp.example.com:465
```

If you want to test an IMAP server, you need to send it a different logout (the first line). To log out of it, you need *a1 logout* followed by a line return, so

```
printf 'a1 logout\n' | \
openssl s_client -connect mail.example.com:143 -starttls imap
```

Again, connecting to imaps (port 993), you just don't do the starttls

```
printf 'a1 logout\n' | \
openssl s_client -connect mail.example.com:143 -starttls imap
```

And, finally, to look at a web site certificate, use port 443, and simply a line return, but you need to put in the server name on systems which have more than one web site (virtual hosting). Do that with the *-servername* flag.

```
printf "\n" | \
openssl s_client -showcerts -servername web.example.com -connect
```

```
web.example.com:443
```

All the above is well and good, but it would be nice to decode the certificate, wouldn't it? Well, openssl has a command that will allow you to inspect a certificate using the *openssl x509* subcommand. For additional information, see *man openssl-x509*. We want the -noout flag to keep our dump clean (prevents the output of the encoded version of the certificate)

Dump the certificate

Turning the certificate into something a human can read is done with the command *-text* flag, so let's pipe the output of the previous command to that.

```
printf 'quit\n' | \
openssl s_client -connect smtp.example.com:25 -starttls smtp | \
openssl x509 -text -noout
```

If you want to find what names the certificate is valid for, they are on a line which contains the text DNS, so grepping the output of the above will give you what you need without reading the whole thing.

```
printf 'quit\n' | \
openssl s_client -connect smtp.example.com:25 -starttls smtp | \
openssl x509 -text -noout | \
grep DNS
```

Get Dates

You could use *grep* to find the expiration date of a certificate

```
printf 'quit\n' | \
openssl s_client -connect smtp.example.com:25 -starttls smtp | \
openssl x509 -text -noout | \
grep 'Not After :'
```

But, the openssl x509 has a special flag for that, *-dates*, so it is simpler to write it as

```
printf 'quit\n' | \
openssl s_client -connect smtp.example.com:25 -starttls smtp | \
openssl x509 -dates -noout
```

Other

Again, *man openssl-x509* gives you more than I'm showing here under the Display Options section, but just a brief list of some interesting flags. -serial - the serial number of the certificate -subject - Subject Name -issuer - Issuer Name -startdate - beginning date of certificate (notBefore) -enddate - expiry date of certificate (notAfter)

Links

- https://serverfault.com/questions/131627/how-to-inspect-remote-smtp-servers-tls-certificate#131628
- https://stackoverflow.com/questions/13127352/how-to-check-subject-alternative-names-for-a-ssl-tls-certificate

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/quickreference/ssl

Last update: **2024/03/04 22:03**