CA/BROWSER FORUM

INFORMATION FOR SITE OWNERS AND ADMINISTRATORS

Here you'll find helpful information about the installation and use of SSL/TLS Certificates. If you are not already familiar with SSL/TLS certificates, then you should read this post first – Information for users.

How to select your CA?
It is crucial to choose the most suitable Certification Authority for the sake of your certificate. You should take care of the reputation of the CA together with its convenience and certificate prices while choosing the best one for you.

What is a CSR?
A CSR or Certificate Signing request is a block of encrypted data that is generated on the server that the certificate is going to be used on. It contains information that will be needed to generate your certificate such as your organization name, common name (domain name), locality, and country. It also contains the public key that will be included in your certificate. A private key is usually generated at the same time that you create the CSR.

Why do I need a CSR?
A certificate authority can use a CSR to create your SSL certificate, but it does not need your private key. You need to keep your private key secret. The certificate created with a particular CSR will only work with the private key that was generated with it. Hence if you lose the private key, the certificate will no longer work.

What are the contents of a CSR?
Common Name (CN): The fully qualified domain name (FQDN) of your server.
Organization (O): The legal name of your organization.
Organizational Unit (OU): The division of your organization handling the certificate.
City/Locality (L): The city where your organization is located.
State(S): The state where your organization is located.
Country (C): The two-letter ISO code for the country where your organization is location.
Email address: An email address used to contact your organization.
Public Key: The public key that will go into the certificate.

An example:
CN: *.cabforum.org
O: CAB Forum
OU: Research and Development
L: Sunnyvale
S: California
C: US

What is the format of a CSR?
Most CSRs are created in the Base-64 encoded PEM format. This format includes the "———BEGIN CERTIFICATE REQUEST———" and the "———END CERTIFICATE REQUEST———" lines at the beginning and end of the CSR.

How does one create a CSR?
It depends on the type of the web server that you will use the certificate. Thus please refer to the vendor instructions to complete the process.
If you are familiar with OpenSSL, you can use the following command to generate a CSR and private key:
openssl req -new -keyout server.key -out server.csr

What other issues should I focus on?
The issues faced by system administrators installing and maintaining SSL/TLS certificates and keys can range from simple to complex. The problems faced do not end there, one has to be aware of the quality of protection provided by SSL which depends on more than just the length of the key, but also how that key is used. This area of the website is specifically intended to address issues faced by system administrators. Therefore, we will skip over some of the rudimentary issues and focus on deployment-level guidance.

- Obtain your certificates from a reliable Certification Authority -You've already decided that you need a reliable CA. You can use resources to assist you found at sites like – – SSL Shopper –
- Protect your Private Key

to search type and hit enter

RECENT NEWS

- Ballot 212 – Canonicalise formal name of the Baseline Requirements September 1, 2017
- Ballot 210 – Misc. Changes to the NCSSR August 31, 2017
- 2017-08-03 Minutes August 3, 2017
- 2017-07-20 Minutes July 20, 2017
- Ballot 204 – Forbid DTPs from doing Domain/IP Ownership July 11, 2017
- Ballot 205 – Membership-Related Clarifications July 6, 2017
- Ballot 192 – Notary Revision June 28, 2017
- 2017-06-21 F2F Minutes Meeting 41 in Berlin June 21, 2017
- Ballot 203 – Formation of Network Security Working Group June 19, 2017

PAST PROCEEDINGS

Past Proceedings Select Month

BY CATEGORY

By Category Select Category

- Make sure your certificate doesn't expire
- Ensure Domain Name Coverage
- Ensure Certificate Chain – some CAs provide such a utility
- Use tools like those listed here [internal link]
- Make sure you are using the most current version of your server distribution and the most current SSL library.
- Disable SSL v2
- Consider disabling SSL v.3.1
- Don't serve mixed HTTP and HTTPS content
- Disable Insecure Renegotiation
- Use persistent connections
- Encrypt 100% of your traffic
- Ensure that secure cookies are used
- Use the do-not-cache http header for sensitive data
- Use Secure Protocols and Cipher-Suites
- Enable Session Resumption
- Implement HSTS

Finally, here are some configuration hints for a few common server platforms:

Apache SSLProtocol -ALL +SSLv3 +TLSv1 SSLCipherSuite ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:RC4:HIGH:!MD5:!aNULL:!EDH SSLHonorCipherOrder on

Nginx ssl_protocols SSLv3 TLSv1 TLSv1.1 TLSv1.2; ssl_ciphers ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:RC4:HIGH:!MD5:!aNULL:!EDH; ssl_prefer_server_ciphers on;