Wildcard Subject Alternate Name SSL/TLS Certificates

Jan 12, 2014 • Category: Security

Managing hundreds or thousands of servers for SSL/TLS can be a challenge due to the potential number of certificates involved. To address this, I recently looked into combining two common management features of certificates, wildcard domain names and subject alternative names (SANs) into a “Wildcard SAN” certificate. Eventually I found that these certificates are in use but knowledge of them does not appear to be widespread.

The conclusion is that wildcard SAN certificates are supported by public and private CAs, are in use at major websites (Google and Yahoo) and appear to be safe for SMTP with some known limitations.

Background

In SSL/TLS, domain name verification occurs by matching the FQDN of the system with the name specified in the certificate. The certificate name can be in two locations, either the Subject or the Subject Alternative Name (`subjectAltName`) extension. When present in the Subject, the name that is used is the Common Name (CN) component of the X.500 Distinguished Name (DN). A second place that is often checked is the Subject Alternative Name (SAN) extension which can contain a list of DNS names, IP addresses, email addresses or URIs.

Both wildcard domains and subject alternative names are techniques to enable certificates to authenticate more than one domain name. This is often useful as it is common for a system to have more than one domain name.

- **Wildcard Domains:** Wildcard domains such as `*.example.com` are useful when protecting multiple services on one domain such as `www.example.com`, `mail.example.com`, `mx.example.com`, `ftp.example.com`, `blog.example.com`, etc. however it has several limitations:
  - **Non-zero length subdomain:** first is that many sites will use a combination of `www.example.com` and `example.com` and a `*.example.com` wildcard will not match the latter.
  - **Only flat subdomain support:** wildcards will not support multiple subdomains, for example `*.m.example.com` will not be matched by `*.example.com`.
  - **Only one domain:** finally, `*.example.com` will not support an entirely different subdomain such as `foobar.com`
- **Subject Alternative Name:** Using the X.509 `subjectAltName` extension has been useful to address some of the limiations of wildcard domains, namely they can contain multiple FQDNs of all types so names with differing numbers of subdomains and entirely different domains can

be suppored. To make SANs even more useful, the goal of this effort was to validate the support for using wildcard domain names in the SAN.

IETF Standards Support

Before starting, the first place to check was support in the X.509 PKI standards and IETF RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile does indicate that wildcard SANs may be used in certificates but are not defined within the RFC:

the semantics of subject alternative names that include wildcard characters (e.g., as a placeholder for a set of names) are not addressed by this specification. Applications with specific requirements MAY use such names, but they must define the semantics.

In the Lab - OpenSSL

To try this in the lab, we create a CSR using OpenSSL by creating a config file to be referenced by the `openssl req` command which can generate a key pair and Certificate Signing Request (CSR) with the WSANs included as shown below:

create OpenSSL req.cfg

```
[req]
req_extensions = v3_req
[v3_req]
subjectAltName = @alt_names
[alt_names]
DNS.1 = *.example.com
DNS.2 = *.m.example.com
DNS.3 = example.com
```

Now generate the keypair and CSR:

```
$ openssl req -new -newkey rsa:2048 -sha256 -nodes -out keypair.csr -keyout keypair.key -c
```

Once the CSR is available, use it to make a certificate request from a private CA to test support such as Microsoft Certificate Authority.

Finally, use the certificate in an application to verify successful SSL/TLS connections. For example, using the Apache web server, we can reference the key and certificate in the conf file:

Apache conf

```
LoadModule ssl_module modules/mod_ssl.so
Listen 443
<VirtualHost *:443>
   ServerName www.example.com
   SSLEngine on
   SSLCertificateFile /path/to/keypair.cert
   SSLCertificateKeyFile /path/to/keypair.key
</VirtualHost>
```

Finally, connect a web browser to the web server and see if the certificate validates, first importing and trusting the private CA root certificate of course.

In the Wild - Google and Yahoo!

Now that it has been established that certificates may have wildcard SANs and they can be issued, it made sense to see if these certificates were used in the wild. Investigating public CA websites indicated that most websites offered either wildcard CN certificates or explicit FQDN SAN certificates but not a combination of wildcard SAN certificates.

Undeterred, I checked to see if anyone was using these in the wild. Information was thin but I did find a single post referencing Google on StackOverflow for YouTube. Pulling up their certificate and then Yahoo!'s indicated that these two services make widespread use of wildcard SAN certificates.

Google Wildcard SAN Certificate

Examing the Google certificate provided some good insight in that:

- multiple different domains are supported
- multiple levels of subdomains are supported
- popular websites are supported
- many SANs can be supported

This indicated popular browser support, however, it did not indicate popular issuance of such certificates as the certificate is not signed directly by a public CA but is signed by the `Google Internet Authority G2` Certificate Authority, a subordinate CA under GeoTrust. Some Internet reports have indicated that subordinate CA certificates also cost in the range of $150,000 to set up and $75,000 / year to maintain which makes it unavaialble as a mainstream solution and there are technical constraints as well.

Google cert

Yahoo! Wildcard SAN Certificate

Finding the Google certificate was a strong indicator that these certificates are used by relying applications, however, we still need to see if public CAs will offer them. Moving on to Yahoo! we see that Yahoo! also uses a wildcard SAN certificate and this one is signed directly by DigiCert. From the Yahoo! certificate we learn that:

- at least one public CA, DigiCert, offers these certificates
- a mix of non-wildcard and wildcard SANs can be supported

Yahoo! cert

Public Certificate Authority Offerings

Knowing that WSAN certificates are in the wild and offered by at least one CA enabled me to reach out directly to two public CAs and inquire about this feature even if it was not listed on their websites:

- **DigiCert:** The first CA I reached out to was DigiCert because they supplied the Yahoo! certificate. To purchase a WSAN certifiate, you need to purchase individual wildcard certificates and then contact DigiCert to have those combined into a single certificate.
- **GlobalSign:** I also reached out to GlobalSign as I have a colleague there and I was informed that they also support WSAN certificates but as an unpublished offering due to customer demand.

Use Cases

TLS/SSL certificates are used for a variety of purposes and for this exercise, I investigated both HTTPS and SMTP. It appears WSAN certificates are safe to use for HTTPS with web browsers and may be safe for SMTP.

HTTPS

Given the widespread use of WSAN certificates by Google and Yahoo! on their popular websites, it seems reasonable to say that these certificates are supported by common web browsers. Mobile use still needs to be investigated.

SMTP

SMTP over TLS is defined by IETF RFC 3207. It appears that some mail servers have issues with wildcard certificates. While Sendmail is known not to support SAN, representatives from public CAs and my professional experience have indicated no issues, possibly given the level of TLS name verification current in use.

- ** Exchange: ** Exchange 2007 and later requires certificates that have explicit internal FQDNs.
- ** Sendmail: ** Sendmail does not currently support name validation using SANs.

Conclusion

From the above, we can conclude:

- wildcard SAN (WSAN) certificates are supported by IETF RFC 3280
- WSAN certs are in widespread use for HTTPS
- WSAN certs can be used for SMTP
- WSAN CSRs can be generated by OpenSSL
- Public CAs (DigiCert, GlobalSign) sign WSAN certificates
- Micrsoft CA signs WSAN certificates
- popular websites use WSAN certificates
- multiple different domains are supported
- multiple levels of subdomains are supported
- many SANs can be supported within the SAN extension
- a mix of non-wildcard and wildcard SANs can be supported

This was an useful exercise for me from an operations and certifiate management perspective. If you have experience with these certificates, please provide a note below.

Author: **John Wang**