Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 12, No. 4, December : 2023

SSL ASSURANCE SUITE: EVALUATING SSL CERTIFICATE AND PUBLIC KEY INFRASTRUCTURE

Varsha Laad Research Scholar
Dr, Atul Dattarya Newase Professor**,**
Research Cell Dr. A. P. J. Abdul Kalam University, Indore, India

Abstract:
These days, a lot of web-based applications—particularly those related to banking and e-commerce—need the transmission of sensitive information, including credit card numbers, as well as personal data, and these transactions must happen entirely online. These high-security transactions are commonly carried out by users on websites they visit using web browsers. Because of this, the browser is among the most fundamental pieces of online software. SSL certificates, that are used for server authentication, guarantee the protection of the connection between the user and the website. Certain requirements must be met for certificates issued by Certificate Authorities (CAs) that have successfully completed worldwide audits. The Baseline Requirements (BR) paper issued by the Certificate Authority/Browser (CA/B) Forum, which is recognized as the authority in the WEB Public Key Infrastructure (WEB PKI) ecosystem, defines the requirements for certificate issuing. By alone, issuing the certificates in fulfillment with the specified standards is insufficient to create a secure SSL connection. Validating certificates, which offer a secure connection and confirm the legitimacy of the website, is the responsibility of the online browsers that people use the most. A thorough SSL certificate public key infrastructure (SSL Test Suite) was developed for this study in order to evaluate how web browsers behave when faced with certificates that don't meet BR criteria. The intended use of the test suite is to efficiently investigate web browsers' certificate validation behaviors.
Keywords: Security, Public Key infrastructure, SSL, Certificate Authority/Browser, Baseline Requirements, SSL Test Suite

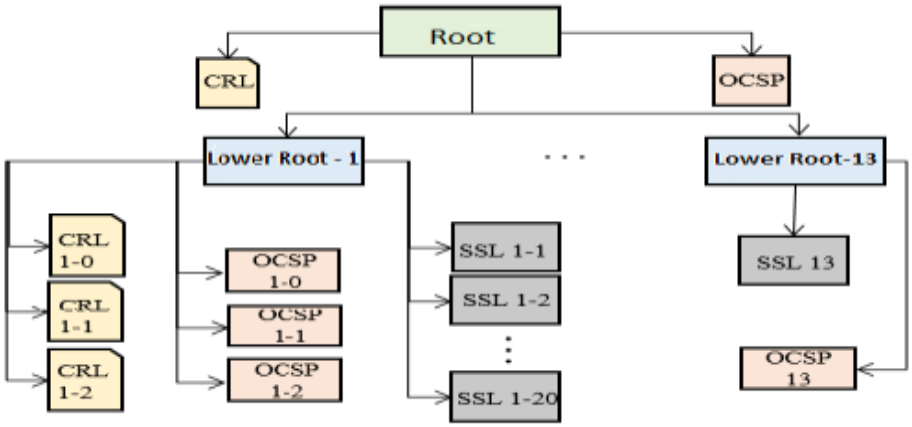
Introduction:

The Internet Connects billions of people and numerous organizations every day, spanning from routine activities to transactions that require the highest level of privacy and integrity assurance. Many advantages enjoyed by internet users, such as the ability to conduct electronic public services and banking transactions, as well as making purchases independent of time and place, rely on HTTPS connections. HTTPS utilizes the SSL/TLS protocol to secure communication over the open network. The public key infrastructure used in this protocol plays an important role on mechanisms related to privacy, integration, authentication and non-repudiation functions through SSL certificates**[1].** These certificates, in X.509 format, electronically link the cryptographic key pair used in server identity authentication to the identity of the server owner. Certificate Authorities (CAs), trusted for their cryptographic signatures, are responsible for the top level of the public key infrastructure hierarchy.
The CA/B Forum, a voluntary organization comprising CAs, web browsers, and other PKI elements, publishes technical documents containing rules for SSL certificate production. CAs aspiring to be internationally recognized as trustworthy must adhere to the limitations defined in the BR document during certificate production. Although the document is frequently updated to maintain the highest level of security provided by SSL certificates, studies have revealed numerous certificates in the ecosystem that question the consistency of the public key infrastructure.
The basis of all trusted communication for clients rests upon a small and curated set of root certificates from trusted certificate authorities (CAs**) [2].** The security of the ecosystem, with erroneous certificates, relies not only on CAs but also on applications that undertake the verification of certificates. Applications are expected to carefully review the given certificate chains and accept only genuine ones. This is especially true for web browsers that use the X.509 public key infrastructure.. However, research has shown weaknesses in the certificate verification systems of applications,

including web browsers.
The impact of certificate verification on security has become a focal point among researchers, leading to the exploration of security vulnerabilities in applications, browsers, and libraries in this field. The security assurance expected from SSL/TLS handshake critically hinges on the premise that communication peers, particularly the clients, correctly perform the prescribed validation of the server-provided X.509 certificate chain. **[3]** The literature includes studies that examine the certificate verification behaviors of applications by implementing a man-in-the-middle attack on the handshake protocol used with SSL applications. Other studies explore the certificate verification behaviors of applications by producing certificates with different methods. The frankencerts study, using certificates created by randomly combining certificate fields gathered from the internet, has led to the identification of inconsistencies among applications and flaws in their behavior
While these studies vary in certificate production methods, they all evaluate the certificate verification approaches of SSL applications and web browsers within the scope of RFC 5280 and X.509 standards. Certificate validation usually follows RFC 5280—the widely accepted international X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer, or service identity contained within the certificate**[4].** Studies focusing on certificate restrictions tailored for SSL in the BR document are limited to comparing the state of the WEB PKI ecosystem before and after the publication of the BR or evaluating the compliance of certificates in the ecosystem with the BR document
In this study, SSL Test Suite, , a test SSL public key infrastructure was established to analyze the ability of web browsers' certificate verification mechanisms to detect certificates that violate BR document-specific restrictions at a level higher than RFC 5280 and X.509 standards. SSL Test Suite consists of PKI components that implement scenarios that may cause a security breach if the restrictions specified in the BR document are not applied. Since the test scenarios are designed to comply with only one rule from the rule set in each case, the behavior of different browsers against certificates with obvious errors can be analyzed with the test suite.

SSL Test Suite
The SSL Test Suite is a comprehensive test public key infrastructure containing valid and erroneous scenarios for certificates, created in accordance with the BR (Baseline Requirements) specifications. It includes test scenarios for SSL and sub-root certificates based on BR version 1.8.0. The primary criteria for these test scenarios are that the certificates adhere to the X.509 ASN.1 structure and trigger only a single error in each test case. Root certificates are excluded from potential discrepancies as they undergo rigorous scrutiny in the process of being included in the trusted root store of web browsers.
The SSL Test Suite certificate hierarchy has been generated from a valid root certificate in compliance with BR (Baseline Requirements) constraints. Thirteen different sub-root certificates were created from the root certificate, and SSL certificates were then generated with 32 different scenarios applied to the sub-root certificates. Additionally, components related to revocation control mechanisms have been included in the structure, creating a comprehensive test PKI (Public Key Infrastructure) model. The structure of the modeled test hierarchy is provided in Figure 1.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 12, No. 4, December : 2023



A. Certificate Verification

In the existing model of the hierarchical WEB PKI structure, the concept of a certificate chain allows tracking the path from a website's certificate to the authority that issued it. The certificate chain, in its simplest form, consists of root, sub-root, and end-entity certificates. Certificate verification is a lengthy and critical process that involves checks for each certificate in the chain, including validity date verification, mathematical verification of the signature, revocation control, structural checks, and algorithm checks.

While the BR document does not provide technical details regarding certificate verification, the certificate profiles defined here play a significant role in applying the controls during the certificate verification stage.

B. CA/B BR Requirements – Scenario Creation

When creating scenarios for SSL and sub-root certificates, requirements ensuring reliable certificate verification, as specified in the BR document, were identified. These requirements, related to certificate profiles and revocation control, were transformed into scenarios covering potential erroneous situations. The created scenarios are grouped into version control, validity period control, revocation controls, extension controls, and algorithm controls according to the certificate verification steps.

When creating test scenarios, one of the critical points is to have only one erroneous certificate in the chain for each test case. This means that in scenarios testing end-entity certificates, only SSL certificates are faulty in the certificate chain, and in scenarios testing sub-root certificates, only sub-root certificates are faulty. This way, even if the browser does not provide detailed error messages, the certificate triggering the error can be identified.

1) Version Control: Extensions playing a crucial role in the execution of certificate verification steps were introduced with version 3 certificates . BR Section 7.1.1 states that an SSL certificate needs to be of type X.509 v3. In this context, an SSL certificate with version 2 was produced. This check was excluded for sub-root certificates.

2) Validity Period Control: In cryptography, every public-private key pair has a specific lifespan. Since the certificate binds the public key and the end entity, the certificate also has a lifespan. Each X.509 certificate is associated with a validity period, and the actual use time of it should be within that scope**[5].** If the certificate exceeds its validity period, it should not be accepted as valid by browsers. According to BR Section 6.3.2, SSL certificates issued after September 1, 2020, must not have a validity period exceeding 398 days. In this context, an SSL certificate with a valid lifetime that exceeds the allowed period was produced. This scenario was not applied to sub-root certificates since there is no constraint regarding the validity period of sub-root certificates in the document.

3) Revocation Controls: Certificates can be revoked for reasons such as suspicion of compromise of the private key, changes in the information enclosed in the certificate, or the termination of domain ownership. Successful revocation of a certificate requires not only action from the certificate owner and certificate authorities but also requires clients, especially web browsers, to query revocation and

comply with it. The CAs must continuously keep track of the revoked certificates and make certificate revocation information public and (as much as possible) available in time**.**

Revocation control in the public key infrastructure is performed by two methods: Certificate Revocation Lists (CRL) and the Online Certificate Status Protocol (OCSP). Clients can access these services using the access points contained in the certificate. Certificate-specific CRL access information is found in the CRL Distribution Point (CDP) extension of the certificate, while OCSP access information is found in the Authority Information Access (AIA) extension.

Since revocation control is an important step in the public key infrastructure, the acceptability and security of the data should be analyzed well when processing revocation data. In order to ensure that web browsers perform revocation control in line with the tested scenario, a revocation control mechanism targeting only the scenario has been built into the certificate. For example; In the scenario intended to test web browsers' response to a particular invalid OCSP responder, the invalid OCSP responder and a structurally valid SSL certificate are created. The feature of this certificate is that it has the targeted OCSP server address as the OCSP address in the AIA plugin and does not have a CRL address or the CRL address cannot be accessed. The goal is to force the application validating the SSL certificate to use the targeted OCSP server. In this case, what is expected is that the application will catch the invalid OCSP response and warn the user about this situation. PKI restrictions are respected on all other components in the hierarchy for this scenario

Scenarios for processing certificate revocation data include CRL revocation check for SSL and subroot certificates, CRL validity period and signature check, OCSP revocation check, and OCSP response validity period checks.

4) Plugin Controls:

The scenarios listed below have been developed for the plugins that are part of the certificate.

a) Key Usage Plugin: In this plugin, the basic usage purposes of the certificate, such as using it for electronic signature, CRL or other certificate signing, are specified. According to BR Section 7.1.2.2.e, the Key Usage plugin must be included in the sub-root certificates. It is indicated by checking the key Cert Sign bit that the subroot certificate will be used for end-user certificate generation. Based on this point, sub-root certificates were created that did not contain the Key Usage plugin and did not include the key Cert Sign bit.

b) Extended Key Usage (EKU) Plugin: This plug-in specifies the specific purposes for which the certificate can be used, other than the basic purposes that can be specified in the Key Usage plug-in. The feature that indicates that the certificate will be used for server authentication purposes is provided by checking the "Server Authentication" bit in this plugin. According to BR Section 7.1.2.3.f, the EKU plugin must be present in SSL certificates. This plugin may include server authentication, client authentication, or both. However, it cannot be used for any other purpose other than these values. In this context, the code Signing bit was marked in the EKU plugin and SSL certificates without the EKU plugin were produced.

c) Basic Constraints Plugin: Whether the certificate is authorized to generate another certificate; Even if it is authorized, the maximum chain length of the certificates produced under it is specified in this plugin. According to BR Section 7.1.2.2.d, this extension should be present in sub-root certificates and the CA bit must be marked. Sub-root certificates that do not contain this plugin and those that do contain the CA bit are not marked are included in the hierarchy.

d) Subject Alternative Name (SAN) Plugin: The SAN plugin with Subject Common Name specifies the domain name information for which the certificate is valid. According to BR Section 7.1.4.2, SSL certificates must have a SAN field, and the domain name specified in the Subject Common Name field must also be included under the SAN plugin. SSL certificates can be valid for a single domain name or can be used for multiple domain names. For example; The “*” character makes the certificate valid for all subdomains. However, there are restrictions on the position of this character, called "wildcard" in the document. SSL certificates have been produced that do not meet these restrictions.

5) Algorithm and Key Controls: The BR document contains restrictions on the cryptographic keys

that can be used for both SSL certificates and CA certificates. According to BR Section 7.1.3, the public key of end user and sub-root certificates must be at least Elliptic Curve 256 bits or RSA 2048 bits. Additionally, the use of the SHA-1 hash algorithm in signing is prohibited because it is insecure. However, the use of known weak keys (see Debian Weak Key) is also prohibited. In this context, SSL and sub-root certificates that do not meet cryptographic requirements have been produced.

Conclusion

The SSL Test Suite was developed to analyze the performance of web browsers during SSL certificate verification. The scenarios in the suite focus on various aspects, including version control, validity period control, revocation controls, extension controls, and algorithm controls, to assess the capability of browsers to detect certificates violating the constraints specified in the BR document.

By providing a comprehensive testing environment, the SSL Test Suite enables the evaluation of web browsers' compliance with the BR requirements, allowing for the identification of potential security vulnerabilities. The structured approach of the suite, with scenarios covering different aspects of certificate verification, contributes to a thorough examination of browser behavior in response to non-compliant certificates.

Understanding the limitations and potential risks associated with SSL certificate verification is crucial for enhancing the overall security of the internet ecosystem. The SSL Test Suite serves as a important tool for researchers, developers, and security professionals to assess and improve the robustness of web browsers in handling SSL certificates, ultimately contributing to a safer online experience for users.

References

[1] S. Cristiano and F. F. Liu, "On splitting public keys for the public key infrastructure," *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, Hong Kong, China, 2005, pp. 112-115, doi: 10.1109/EEE.2005.96.

[2] E. Oakes, J. Kline, A. Cahn, K. Funkhouser and P. Barford, "A Residential Client-side Perspective on SSL Certificates," *2019 Network Traffic Measurement and Analysis Conference (TMA)*, Paris, France, 2019, pp. 185-192, doi: 10.23919/TMA.2019.8784633.

[3] S. Y. Chau *et al*., "SymCerts: Practical Symbolic Execution for Exposing Noncompliance in X.509 Certificate Validation Implementations," *2017 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, 2017, pp. 503-520, doi: 10.1109/SP.2017.40.

[4] J. Zhu, C. Wan, P. Nie, Y. Chen and Z. Su, "Guided, Deep Testing of X.509 Certificate Validation via Coverage Transfer Graphs," *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Adelaide, SA, Australia, 2020, pp. 243-254, doi: 10.1109/ICSME46990.2020.00032.

[5] P. Fu, Z. Li, G. Xiong, Z. Cao and C. Kang, "SSL/TLS Security Exploration Through X.509 Certificate's Life Cycle Measurement," *2018 IEEE Symposium on Computers and Communications (ISCC)*, Natal, Brazil, 2018, pp. 00652-00655, doi: 10.1109/ISCC.2018.8538533.

[6] D. G. Berbecaru and A. Lioy, "An Evaluation of X.509 Certificate Revocation and Related Privacy Issues in the Web PKI Ecosystem," in *IEEE Access*, vol. 11, pp. 79156-79175, 2023, doi: 10.1109/ACCESS.2023.3299357.