SSL Survey HTTP Rating Guide

version 1.0 (5 July 2010)

Copyright © 2010 Qualys SSL Labs (www.ssllabs.com)

Abstract

The Secure Sockets Layer (SSL) protocol is a standard for encrypted network communication. We feel that there is surprisingly little attention paid to how SSL is configured, given its widespread usage. SSL is relatively easy to use, but it does have its traps. This guide aims to establish a straightforward assessment methodology, allowing administrators to assess SSL server configuration confidently without the need to become SSL experts.

Q QUALYS® **SSL LABS**

www.ssllabs.com

Methodology Overview

Our approach consists of three steps:

1. We first look at a certificate to verify that it is valid and trusted. A server that fails this step is always assigned a zero score.

2. We inspect server configuration in three categories:

   a. Protocol support

   b. Key exchange support

   c. Cipher support

3. The final score—a number between 0 and 100—is a combination of the scores achieved in the individual categories. A zero score in any category forces the total score to zero.

Because small differences between configurations are sometimes less important, we also assign letter grades to servers. Letter grades are generally more useful—it is instantly clear that a server given an A is well configured. Table 1 shows how a numerical score is translated into a letter grade.

Table 1. Letter grade translation

Numerical Score	Grade
score >= 80	A
score >= 65	B
score >= 50	C
score >= 35	D
score >= 20	E
score < 20	F

Our methodology is purposefully designed to be straightforward and practical. Advanced users may prefer to dig deep into the finer details of SSL (e.g., which ciphers are better than others), but we feel that such an approach, although intellectually challenging, would make this guide less useful in practice.

What This Guide Does Not Cover

Our immediate goal is to focus on those configuration problems whose presence can be determined remotely and without manual assessment. It is only a fully automated approach that makes it possible to perform a large-scale assessment of SSL configuration practices. Our aim is to scan all SSL servers on the public Internet.

In focusing on automation, we have decided not to look for certain problems. We will list those problems in this guide, and hopefully find ways to enhance our automation to include them in a future version of this guide. Some of those problems are listed here:

Certificate quality

Three certificate types are currently in use: domain-validated, organisation-validated and extended-validation (EV) certificates. This guide requires a certificate to be correct, but does not go beyond this basic requirement. The domain-validated and organisation-validated certificates are generally treated in the same way by the current generation of browser software, and thus offer similar assurance to users. EV certificates are treated significantly better and, generally, they are

recommended for high-value web sites. Without a reliable way to determine the purpose of a web site, however, there is little that this guide can do to assess whether a certificate used on an arbitrary site is suitable for the purpose of the site.

Session hijacking issues in web applications

There are several ways in which web applications can subvert SSL and make it less effective. For example, session cookies that are not marked as secure can be retrieved by a determined attacker, leading to session hijacking and thus application compromise. Such problems are not the fault of SSL, but they do affect its practical applications nevertheless. Detecting web application–specific problems is non-trivial to perform in an automated fashion, and this version of the guide does not attempt to do it. We leave this problem for the consideration in the future. In the meantime, to remove any doubt that might exist about the seriousness of the above-mentioned issues, we will state that *any application that incorrectly implements session token propagation should be awarded a zero score.*

What Should My Score Be?

We don't know. In order to tell you whether you've configured your SSL server correctly, we would need to know what your site does. Because different web sites have different needs, it is not possible for us to choose any one configuration and say that it works for everyone. But we can do two things. First, we can give you some general configuration advice and tell you what you should never do. Second, we can give you some general guidance using examples of what other web sites do. If that's what you are interested in, skip to the end of this document for more information.

Is SSL Enough?

No. A non-trivial web site cannot be secure if it does not implement SSL, but SSL is not enough. SSL deals with only one aspect of security, and that is the security of the communication channel between a web site and its users. SSL does not and cannot address a number of possible security issues that may exist on a web site. View SSL as a foundation on which to build, but the foundation alone is not enough.

Acknowledgements

The first version of this guide was written by Ivan Ristic `[http://blog.ivanristic.com]`, and subsequently enhanced by the contributions from the following individuals, listed in alphabetical order: Christian Bockermann, Christian Folini, Robert Hansen, Ofer Shezaf and Colin Watson.

Certificate Inspection

Server certificate is often the weakest point of an SSL server configuration. A certificate that is not trusted (i.e., is not ultimately signed by a well-known certificate authority) fails to prevent man-in-the-middle (MITM) attacks and renders SSL effectively useless. A certificate that is incorrect in some other way (e.g., a certificate that has expired) erodes trust and, in the long term, jeopardises the security of the Internet as a whole.

For these reasons, any of the following certificate issues immediately result in a zero score:

- Domain name mismatch

- Certificate not yet valid
- Certificate expired
- Use of a self-signed certificate
- Use of a certificate that is not trusted (unknown CA or some other validation error)
- Use of a revoked certificate

> **Note**
>
> Some organisations create their own (private) CA certificates, a practice that is entirely legitimate, provided such CA certificates are distributed, in a safe manner (e.g., through the use of customised browsers) to all those who need it. Without the access to such certificates we may not be able to verify that a site we are inspecting has a trusted certificate, but we believe that such sites will be relatively rare. Such issues can be considered on a case-by-case basis.

Scoring

SSL is a complex hybrid protocol with support for many features across several phases of operation. To account for the complexity, we rate the configuration of an SSL server in three categories, as displayed in Table 2. We calculate the final score as a combination of the scores in the individual categories, as described in the "Methodology Overview" section.

Table 2. Criteria categories

Category	Score
Protocol support	30%
Key exchange	30%
Cipher strength	40%

The sections that follow explain the rating system for each of the categories.

Protocol Support

First, we look at the protocols supported by an SSL server. For example, both SSL 2.0 and SSL 3.0 have known weaknesses. Because a server can support several protocols, we use the following algorithm to arrive to the final score:

1. Start with the score of the best protocol.
2. Add the score of the worst protocol.
3. Divide the total by 2.

Table 3. Protocol support rating guide

Protocol	Score
SSL 2.0	20%
SSL 3.0	80%
TLS 1.0	90%
TLS 1.1	95%
TLS 1.2	100%

Key Exchange

The key exchange phase serves two functions. One is to perform authentication, allowing at least one party to verify the identity of the other party. The other is to ensure the safe generation and exchange of the secret keys that will be used during the remainder of the session. The weaknesses in the key exchange phase affect the session in two ways:

- Key exchange without authentication allows an active attacker to perform a MITM attack, gaining access to the complete communication channel.

- Public cryptography is used during key exchange: the stronger the server's private key, the more difficult it is to break the key exchange phase. A weak key, or an exchange procedure that uses only a part of the key (the so-called exportable key exchanges), can result in a weak key exchange phase that makes the per-session secret keys easier to compromise.

Table 4. Key exchange rating guide

Key exchange aspect	Score
Weak key (Debian OpenSSL flaw)	0%
Anonymous key exchange (no authentication)	0%
Key length < 512 bits	20%
Exportable key exchange (limited to 512 bits)	40%
Key length < 1024 bits (e.g., 512)	40%
Key length < 2048 bits (e.g., 1024)	80%
Key length < 4096 bits (e.g., 2048)	90%
Key length >= 4096 bits (e.g., 4096)	100%

Cipher Strength

To break a communication session, an attacker can attempt to break the symmetric cipher used for the bulk of the communication. A stronger cipher allows for stronger encryption and thus increases the effort needed to break it. Because a server can support ciphers of varying strengths, we arrived at a scoring system that penalises the use of weak ciphers. To calculate the score for this categor,y we follow this algorithm:

1. Start with the score of the strongest cipher.

2. Add the score of the weakest cipher.

3. Divide the total by 2.

Table 5. Cipher strength rating guide

Cipher strength	Score
0 bits (no encryption)	0%
< 128 bits (e.g., 40, 56)	20%
< 256 bits (e.g., 128, 168)	80%
>= 256 bits (e.g., 256)	100%

SSL Configuration Advice

This section offers advice on how to configure an SSL server correctly. Actually, given that there is no single correct configuration for every possible use of SSL, we aim merely to give some reasonable advice that works for the majority of users. Your individual needs may vary. As a rule of thumb, the higher the value of your site, the stricter the configuration should be. Although this section tells you what a typical SSL configuration should be, the rating guide will give you some idea of how you can improve upon it if necessary.

Minimal Configuration Requirements

This section lists the minimal requirements every SSL server must fulfil:

1. Use a private key that is at least 1024 bits long.

   a. Do not reuse keys across certificates.

   b. Generate a new key for every certificate you request.

2. Make sure the certificate is valid for all the domain names that will be required. For example, make sure the certificate works for `www.example.com` as well as `example.com` (without the `www` part).

3. Purchase a valid server certificate with an assurance level that is appropriate for your needs:

   a. Domain-validated certificates are cheap and suitable for most uses.

   b. Organisation-validated certificates do not seem to have an advantage that would justify their higher cost.

   c. High-value web sites (e.g., stores and financial institutions) should use EV certificates.

4. Use TLS v1.0 or better.

5. Use strong cipher suites:

   a. Do not use anonymous key exchanges.

   b. Do not use crippled (export) key exchanges.

   c. Do not use ciphers that are weaker than 128 bits.

   d. Support cipher suites that offer 256-bit encryption or better.

6. Ensure that SSL is not subverted at the application level:

   a. Session cookies *must* be marked as secure.

b. No part of a site should be available as plain text (no encryption).

   i. On port 80, there should be only a redirection back to the SSL-protected site.

   ii. Your server must be configured not to serve on port 80 (unprotected) any of the resources that are otherwise delivered over SSL.

   iii. There is no danger in having a plain-text part of your web site, provided there is no functionality on it.

   iv. If you *must* have some functionality on the plain-text part of your web site and require sessions (thus breaking rule 1), then you need to use two separate session mechanisms in parallel, making sure they do not overlap. *We advise against this approach, as it is very easy to implement such a scenario incorrectly and thereby compromise security.*

c. No page should mix protected (SSL) with unprotected (non-SSL) content.

Configuration Guidance

For some guidance as to how strong a server's SSL configuration should be, please refer to Table 6.

Table 6. SSL configuration guidance

Site type	Cert	Key size	Client cert	Cipher	Key ex	Protocol
Sites that do not implement authentication and where all content is public	-	-	-	-	-	-
Sites that implement authentication	DV	1024	-	128	-	TLS v1.0+
E-commerce web sites	EV	1024	-	128	-	TLS v1.0+
Internet banking and similar high-value sites	EV	2048	Desirable	256	-	TLS v1.0+
Sites that contain highly sensitive data that must remain secret for years	EV	4096	Yes	256	Ephemeral Diffie-Hellman only.	TLS v1.0+

> **Note**
>
> Despite what Table 6 says, we could argue that even the simplest of web sites need SSL certificates. Plain-text communication is subject to interception and modification, which means that your users could be receiving your site's pages altered. They could be seeing a completely different web site for all you know. SSL, even with the cheapest certificate, provides some assurance that your site will not be modified in transit.

Performance Considerations

Performance does not feature in our methodology, but some consideration of it is unavoidable. The impact of SSL processing will generally not affect smaller web sites in any way, but larger sites will likely be able to measure it. As a rule of thumb, your SSL configuration should be such that you achieve minimal desired security over the duration of the key. Configurations that are stronger than necessary may make you feel good, but may also be costly, either through the higher cost of hardware or through the lower performance of your web site. Choosing a key that is longer than necessary is most likely to hurt you.

For Consideration in Future Releases

We have left a number of open issues for future consideration:

Perfect forward secrecy bonus points
Some key exchange mechanisms allow for perfect forward secrecy. Perfect forward secrecy is a property that ensures that short-term session keys cannot be compromised after a compromise of a long-term secret key (which was used in the generation of the short-term keys). We should consider giving bonus points to those servers that allow only the key exchange mechanisms that allow for forward secrecy.

Differences between encryption algorithms
Should the choice of encryption algorithms affect server score? For example, it is said that RC4 is not as strong as once thought, so its availability might warrant negative points.

Default ciphers
In SSL v3 and later, clients provide servers with a list of cipher suites they are willing to use. We want to determine whether servers typically select the first mutually acceptable cipher suite for a session, or if there is a logic behind the suite selection. If the latter, then we might want to consider giving a better grade to the servers that make better choices.

SSL v2 support
Should servers that support SSL v2 be awarded a zero score in the protocol category? A zero score for the entire test?

About SSL

The Secure Sockets Layer (SSL) protocol is a standard for encrypted network communication. It was conceived at Netscape in 1994; version 2.0 was the first public release. SSL was later upgraded to 3.0, and, with further minor improvements, standardised under the name TLS (*Transport Layer Security*). TLS v1.2, the most recent version, is defined by RFC 5246 [`http://www.ietf.org/rfc/rfc5246.txt`].

About SSL Labs

SSL Labs [`https://www.ssllabs.com`] is a computer security research organisation that, unsurprisingly, focuses on the SSL standard. Our aim is to discuss the rarely mentioned aspects of SSL, promote its correct usage, and generally inspire everyone to do their part to promote security. Unlike in some other areas (e.g., application security), security is relatively easy to achieve when it comes to SSL. Thus we believe that there are no excuses for a lack of security.

About Qualys

Qualys, Inc. is the leading provider of on demand IT security risk and compliance management solutions delivered as a service. Qualys' Software-as-a-Service solutions are deployed in a matter of hours anywhere in the world, providing customers an immediate and continuous view of their security and compliance postures. The QualysGuard¨ service is used today by more than 3,500 organizations in 85

countries, including 40 of the Fortune Global 100 and performs more than 200 million IP audits per year. Qualys has the largest vulnerability management deployment in the world at a Fortune Global 50 company. Qualys has established strategic agreements with leading managed service providers and consulting organizations including BT, Etisalat, Fujitsu, IBM, I(TS)2, LAC, SecureWorks, Symantec, Tata Communications, TELUS and VeriSign.