SHA-1 Versus SHA-2

Overview

Most of the documentation out there on the transition from SHA-1 certificates to SHA-2 certificates will tell you three things:

- Breaking SHA-1 is not yet practical but will be in a matter of years.
- It is important to start transitioning to SHA-2 as soon as possible.
- Much of your legacy software and infrastructure may not support SHA-2 yet.

This paper will provide you information to help you make informed decisions about how to analyze your systems and transition from SHA-1 to SHA-2 in an orderly way. Rather than waiting a few years when SHA-1 collisions start being actively exploited to compromise systems, Trustwave strongly recommends you begin your SHA-1 transition planning immediately.

The first section provides some technical background to help you understand how and when SHA-1 based systems are likely to be attacked. The second section describes the various places certificates are used, and what is going on in the industry. The last section contains real-world considerations that may need to be taken into account as part of planning a transition to SHA-2.

The Problem with SHA-1

Cryptographic Hash Functions

SHA-1 is a cryptographic hash function used in a variety of places in modern cryptosystems (including SSL/TLS), having replaced MD5 as the secure hash function of choice when a number of security flaws were discovered in MD5. However SHA-1 is now starting to show its age, and is being replaced by the SHA-2 family of hash functions. High and medium security environments have already abandoned SHA-1, for example, NIST has banned the use of SHA-1 effective December 31, 2013.

Unlike SHA-1, which is a 160-bit hash function, there are six SHA-2 hash functions, with a variety internal block sizes and output sizes. The most commonly used SHA-2 hash functions are SHA-256 and SHA-512, with the other four being based on the same functions with different initial values and truncated outputs. In most environments, SHA-256 provides sufficient security and is the SHA-2 hash function that Trustwave recommends transitioning to.

Digital signatures use asymmetric cryptographic operations to provide proof that a message was signed by someone in possession of the corresponding private key. However, asymmetric cryptographic operations are computationally expensive, both in terms of the key size and the length of the input. Because of this, digital signatures, including the signatures used in digital certificates, sign a hash of the message instead of the message itself. As long as the hash function is a “secure” hash function, this is

sufficient: it is computationally impractical for the attacker to create another message that has the same hash; therefore he is unable to take the signature and attach it to a new message of his choice.

Breaking this property requires finding two messages that share the same hash. This can be done by repeatedly altering a non-critical field in each message until a message from the first set of messages has the same hash as a message from the second set of messages. Because of the birthday paradox, this happens when the number of messages is approximately the square root of the total number of possible hashes. Therefore, when considering collision resistance, a hash function has an equivalent strength of at most half the number of bits in the hash, and possibly fewer. Since SHA-1 produces a 160 bit hash, the strength is at most 80 bits. The best current cryptanalysis of SHA-1 uses clever math tricks to reduce that to about 60 bits of effective strength, and future cryptographic advances will continue to reduce the strength even further.

Algorithm	Approximate Strength
MD5	<40
SHA-1	~60
SHA-256	128
SHA-512	256

Once the attacker has an "innocent" message and a "forged" message that share the same hash, he simply has to get the "innocent" message signed, and then can transfer the signature to the "forged" message, and it will validate correctly. This attack was used to create forged MD5 certificates, most famously back in 2012 by the Flame malware to create fraudulent Microsoft certificates that could be used to fool Windows Update.

How Long Do I Have?

Of course, finding a collision still requires a substantial amount of CPU time. Exactly how worried one should be depends on a number of variables, including the value to the attacker of being able to produce a forged certificate for a particular identity. For high value targets, such attacks may become practical within the next few years, and because available CPU power increases exponentially every year, even moderate or low value targets are vulnerable within the foreseeable future.

A back of the envelope calculation of the cost to produce a SHA-1 collision comes from an analysis by Jesse Walker (https://www.schneier.com/blog/archives/2012/10/when_will_we_se.html):

Year	Cost
2015	Less than $700,000
2018	Less than $173,000
2021	Less than $43,000

Actual costs may be substantially less if there are further improvements to collision techniques. Given the high profile of SHA-1, and its extensive use as the most widely supported secure hash algorithm for much of the last two decades, such improvements cannot be ruled out.

So the use of SHA-1, especially in combination with timeframes larger than a few years is becoming extremely risky. If a certificate is issued today, it is typically valid for up to three years, so customers need to be considering the hardware that will be available in 2018 when determining the appropriate strength of certificate to use. Also, it is important to remember that an attacker may not have to actually pay the costs above, as they may be stealing CPU and GPU cycles from compromised machines that are part of a botnet. Currently, there typically are more productive uses of compromised systems than attacking SHA-1, but that may change in the not too distant future.

Because of this, certificate authorities, web browsers, and other entities that rely on digital certificates are working together to eliminate the usage of SHA-1 in publicly trusted certificates within the next few years.

The Problem of Inertia

While everyone will have to upgrade to SHA-256 certificates sooner or later, for low security websites, it is not unreasonable to conclude that there is no urgent need to upgrade to SHA-256. After all, the likelihood of a man-in-the-middle attack against a low value website seems fairly low. For many sites, it may be reasonable to simply plan to transition to SHA-256 certificates as the existing certificates expire. However, unless there is a compatibility reason not to do so (see below), new certificates that are being deployed should be SHA-256 certificates. The lack of a compelling need is the reason why SHA-1 is still widely used for SSL/TLS certificates, despite the fact that security experts have been calling for a transition to SHA-256 for many years already. The important thing to understand is that a transition from SHA-1 to SHA-256 is inevitable. When it will happen is just a question of scheduling and resources, and organizations need to start planning for the transition sooner rather than later, to ensure there is enough time to transition smoothly.

Upcoming Browser UI Degradation

The slow rate at which SHA-256 is being adopted by SSL-protected websites is a significant problem for browser vendors. On the public internet, browsers cannot start producing security warnings for SHA-1 until significant majority of web sites are using SHA-256 certificates, otherwise users will just ignore the warnings. This produces a bit of a chicken and the egg problem: browsers don't want to warn about weak certificates because users would get warnings everywhere and just ignore them, while websites have little incentive to use better certificates if browsers accept existing certificates. In order to provide adequate security to end users, browsers have started announcing plans to slowly produce more and more severe warnings for SHA-1 certificates, based on the expiration date of the certificate. It is up to websites to make sure they have SHA-256 in place by the appropriate deadlines in order to avoid browser warnings, which is why it is important that newly deployed certificates are SHA-256 certificates.

The Biggest Problems

Because of the risks of continuing to trust SHA-1, users of cryptographic certificates are being strongly encouraged to move to SHA-256 certificates within the next few years. In principle, this is a very straightforward procedure: simply re-issue the existing certificates as SHA-256 and install them on the

appropriate servers. Certificate authorities are offering free SHA-256 certificates for holders of existing SHA-1 certificates in order to make the transition as painless as possible.

The relevant standards for publicly trusted certificates are being updated to require that SHA-256 certificates will no longer be issued after the end of 2015. In higher security environments, it is worth considering configuring SSL/TLS clients to no longer accept SHA-1 as a valid hash algorithm after the SHA-256 transition is complete. However, as usual in the security space, there are a number of other practical problems that need to be considered.

Legacy Software

In many environments, transitioning to SHA-256 is complicated by the presence of legacy software, operating systems, or devices that predate the existence of SHA-256, and hence do not support the algorithm. In most cases, the solution is to move to a newer version of the software, operating system, or device, as SHA-256 support has generally been added to such systems within the last five years or so, if it wasn't already available. Identifying those components, getting them upgraded, and testing that everything will continue to work after transitioning to SHA-256 certificates is by far the most complicated part of any SHA-256 transition plan.

Non-browser use of TLS

It is also important to remember that TLS is not only used by web servers and browsers. The protocol is intended for use with arbitrary data transportation protocols, and is widely used in other contexts as well (email, Microsoft Exchange, LDAP, file transport, etc.). If these connections are not upgraded as well, they may continue to function correctly, however over time they will become less and less secure. They also may unexpectedly stop working in the future as OS and software vendors drop support for insecure cryptographic functions. It is important that all usages of SHA-1 in your environment have been identified and upgraded.

This can be complicated, because many products do not document the fact that they communicate internally using web services, let alone what cryptographic algorithms are used to protect those internal communications. It may be necessary to upgrade to the latest version of the software in order to get SHA-25656 support, if it is even available at all.

Custom software solutions

The most significant challenge is the use of SHA-1 within custom software, whether developed internally or by contractors. Software development, especially for cryptographic software, has significant lead times and in many cases requires validation against various compliance requirements. Furthermore, it is not easy to analyze such solutions to determine if they use SHA-1 internally, or even worse, MD5. Any cryptographic software needs to be regularly evaluated and upgraded if necessary in order to determine whether it is in compliance with the latest best practices.

Client-auth certificates

Many VPNs and some high-end security solutions use mutually authenticated TLS, where the client's certificate is also checked for validity, in addition to the server's certificate. It is important these certificates are updated as well.

Embedded hardware devices

Increasingly, small hardware devices also have web-based management systems, or communicate via TLS. In many cases, upgrading the firmware is difficult or impossible. If SHA-1 is in use on these devices, an analysis must be done to determine the risk posed by such devices. Given the rate at which acceptable standards for network and web security are changing as well as the rate device capabilities are improving, it may be necessary to replace these devices sooner than originally planned to stay up-to-date with current security standards like SHA-256.

Putting it all together

Since browsers and the internet are used for everything from cute cat videos to online banking, and SSL/TLS is used even more widely as a protection mechanism for virtually any kind of network connection (LDAP, email, VPNs, web services, and so on) transitioning from SHA-1 to SHA-256 can get complicated quickly. The amount of work necessary depends on how extensively SSL/TLS certificates are used. The transition is also complicated by the fact that SSL/TLS is a communications protocol, and the transition may need to be coordinated with external third parties, who may have their own transition timeline (or no transition plan at all).

As noted in the last section, it may be difficult to even identify all the locations where SHA-1 is used within a complex organization. The SHA-1 to SHA-256 transition is a good opportunity to analyze and document the usage of certificates for authentication within your infrastructure, and proper communications security in general. In addition to helping assure a smooth transition, the analysis may also identify other unsecured or inadequately secured connections or data flows. Unless data is adequately protected along EVERY step of its journey throughout the system, attackers will have the opportunity to access the data. This includes communications with cloud-based backup systems, which are increasingly being targeted by attackers.

If certificates are only being used to protect connections to a website, then transitioning to SHA-256 is technically very easy: simply re-issue the site's certificate as SHA-256, and deploy it on the web server. However it is important to be aware that older, unpatched operating systems (e.g. Windows XP before SP3, Windows 2000, and so on) and ancient browsers will no longer be able to connect to the server.

For more complicated environments, the transition may be more complicated and may require coordinating the transition with vendors and external parties. Like most upgrades, the transition will go smoother if there is a carefully thought out plan of action. Communications protected by SHA-1 certificates will eventually be vulnerable to interception, and the time to start thinking about transitioning to SHA-256 is now.