

Factsheet FS-2009-01

Vulnerabilities in the internet PKI caused by use of MD5

On 30 December 2008 a group of researchers at the 'Chaos Communication Congress', an annual security conference held in Berlin, gave a practical demonstration that the 'Public Key Infrastructure' (PKI) on the internet has some serious weak spots. They demonstrated¹ that they had been successful in creating a rogue certificate that was trusted by all common browsers.

In this factsheet you can read a short explanation of the research, the risks in the short term and the actions that need to be undertaken both by yourself and by other concerned parties. We conclude this factsheet with two paragraphs with background information on digital signatures, hashes, collisions and the shelf life of cryptography in general.

The research in brief

This research has an impact on the use of certificates to set up secure connections between a browser and a website. This concerns an 'HTTPS' or 'SSL/TLS' connection. Such a connection can be recognised within browsers by means of a padlock or in some cases by a coloured address bar.

The research demonstrates that when a certificate has been signed (or appears to have been signed) by a competent authority (a Certificate Authority or CA for short), this no longer offers any guarantee that this certificate was also verified by this authority. This means that there is no longer any guarantee that the certificate actually belongs to the correct party. In other words: in the past you had a large degree of certainty² with a secure connection that you were dealing with the right website (after checking the certificate). Now it has been demonstrated that this security can be compromised by a practical attack.

Because there are still CAs that sign certificates with MD5, an obsolete cryptographic method, the researchers succeeded in creating a rogue certificate that appeared to have been signed by an official Certificate Authority (a root CA). As a result this rogue certificate is automatically trusted by all common browsers. What is even worse is that the rogue certificate itself can act as a Certificate Authority.

As a result, the researchers are in a position to create and sign a certificate themselves for any random web server in the world, which cannot be distinguished from a real one and which is trusted automatically. They can therefore impersonate any random other party without the visitor to a website being able to discover this on the basis of the certificate.

An overview of the facts:

- > Researchers have created a rogue certificate that can be used to impersonate any website in the world.
- > This rogue certificate is automatically trusted by all the common browsers.
- > The researchers achieved this by making use of weaknesses in MD5, a cryptographic hash function.
- > Even though the weaknesses of MD5 have been known for many years, it is still used to sign certificates on the internet.
- > If you still make use of certificates that are signed using MD5, then you need to replace it as soon as possible.

Certificates and secure connections

A certificate is the basis of two functionalities of a secure connection. These functionalities are most of all important for applications where personal or financial data are transmitted to another party, for example in the case of telebanking, e-government services and online shopping.

1. The encryption of data exchanged between the browser and the website. The result is that data can no longer be read by third parties.
2. The possibility of monitoring whether a connection has really been made to the correct website.

The research to which we refer in this factsheet has an impact on the second function of a certificate.

¹ A detailed and readable description of the research, the results and recommendations can be found on <http://www.win.tue.nl/hashclash/rogue-ca/>. Video and audio versions of the presentation are also available. The research itself is available from <http://eprint.iacr.org/2009/111>.

² Of course there is no such thing as 100% security. A PKI consists of technical *and* organisational measures, therefore a lack of organisational measures may also result in incorrectly issued certificates. Eddy Nigg recently demonstrated this. He requested a certificate for the domain mozilla.com without any problems; see <https://blog.startcom.org/?p=145>

What are the risks in the short term?

The researchers admit themselves that it is unlikely that another person is going to be able to implement such an attack on the internet PKI in the short term. GOVCERT.NL has also not detected any attacks at the time of writing and more or less discounts that there are already rogue certificates in circulation at this time.

In order to carry out such an attack one needs specialised knowledge of the weaknesses in MD5, the obsolete cryptographic method still used by some CAs for signing certificates. In addition, the researchers themselves developed methods to create a rogue certificate in a short time.

They believe that the CAs in question that still make use of obsolete digital signatures will have enough time to move over to new methods.

The researchers have published their methods³ and have taken some measures to prevent the certificate they have created from being misused.

If it was not already clear following the previously publicised attacks in 2005 and 2007, there is not the slightest doubt now that it is irresponsible to continue to use MD5.

Protection against vulnerability and the actions you can undertake yourself

The researchers demonstrated with their research that the internet PKI contains weak spots because some CAs still make use of obsolete means of creating a digital signature. As a consequence the entire PKI is at risk, not just those persons who have dealings with the CAs in question. The following analogy will clarify this to a certain extent: if it turned out that it was very easy to obtain a real passport in a certain municipality in the Netherlands under false pretences, then that would be a problem not only for the residents of that one municipality but would also undermine confidence in the value of every passport for everyone who came into contact with passports.

The foregoing makes it clear that individual end users and owners of certificates can do very little to protect themselves against this vulnerability, let alone solving these. In an ideal situation the following would now happen:

1. Every CA that still makes use of MD5 would stop doing so as quickly as possible and would migrate to a better hash function⁴.
2. Everyone who still has a certificate that has been signed with MD5 will replace this as soon as possible (see also: 'Replace MD5 ... but with what?' on the following page).
3. If the above two requirements are met (or that much earlier as is considered necessary), the browser vendors can withdraw support for MD5.

The most important parties in the above process are the CAs and the browser vendors. CAs bear a responsibility to make use of sensible cryptographic methods on the basis of their task as a trusted organisation that is permitted to sign certificates within a PKI. It is necessary to evaluate on a regular basis whether a cryptographic method is (still) reliable. It is the case that cryptographic methods that are reliable now may not be reliable for various reasons at a later time.

Browser vendors can exercise a great deal of indirect influence on CAs, because they determine which certificates - and therefore also the *type* of certificates - they include and trust in their browsers as

³ Whether software vulnerabilities should be published or not and how this should be done is a subject of heated discussion within the security community. You can read about how the researchers dealt with this in the context of the research at <http://www.phreedom.org/blog/2009/verisign-and-responsible-disclosure/>

⁴ Verisign, the owner of one of the root CAs that still used MD5, has in the meanwhile made it known that it has given up MD5 and that customers can have their old MD5 certificates replaced free of charge. See https://blogs.verisign.com/ssl-blog/2008/12/on_md5_vulnerabilities_and_mit.php

What is a Public Key Infrastructure?

A Public Key Infrastructure (PKI) is a collection of organisational and technical resources with which you can reliably perform the following.

1. Determine the identity of another party.
2. Encrypt information.
3. Sign information.

Examples of technical aspects:

- > Cryptographic methods used for encryption and signing.
- > The certificates used to establish the owners and purposes of encryption keys.

Examples of organisational aspects:

- > The process for issuing a certificate by a competent government body.
- > Determining the identity of the applicant.
- > Monitoring the internal processes of issuing bodies.

Look at <http://www.pki-overheid.nl/over-pki-overheid> for more information about PKIs (destination link is in Dutch).

standard. In this way they can serve as an extra motivation. If browser makers stop supporting MD5 (the weak hash function), then this would have immediate consequences for the certificates signed using MD5 that are still in circulation. These will stop working or generate warning messages, depending on the choices made by the browser vendors.

All this does not mean that you should not undertake a number of actions yourself:

1. As an end user there is almost nothing you can do to reduce the risks of this proven threat. At this time there are still so many certificates with a MD5 signature in circulation that rejecting such certificates completely is not really a practical solution. There is an extension in circulation for Firefox⁵ that alerts the user to signatures based on MD5, but in practice this normally generates false positives. This is only an option for home users with expert knowledge.
2. It is important within organisations to keep track of which and what type of certificates are in use, even if you have your own internal PKI with a root certificate. This includes certificates from your official websites, your internal websites, client certificates and other solutions that make use of SSL certificates, such as SSL VPNs.
3. If you are still signing certificates internally on the basis of MD5 then make plans to phase this out. If you make use of certificates that are signed on the basis of MD5 then you need to replace these as soon as possible with certificates signed on the basis of a more recent algorithm. You can read more in the following paragraph about your options.

Replace MD5 ... but with what?

The researchers' motivation in publishing this research was to demonstrate that it has for a long time been irresponsible to use MD5 to sign certificates and they have been very successful in this. You therefore need to migrate now, but the question is: "To what, if MD5 is no longer satisfactory?"

The successor to MD5 is SHA-1, but an even newer variant has been available for some time, namely SHA-2 (a collective name for SHA-224, SHA-256, SHA-384 and SHA-512). It has also been demonstrated that SHA-1 has some weak spots⁶ and it is expected that SHA-1 will in the not too distant future disappear as a result of practical attacks. The US National Institute of Standards and Technology (NIST) goes even further. It requires American government organisations to abandon SHA-1 and move over to a SHA-2-variant⁷ before the end of 2010.

You have two options at the moment:

1. Transition to SHA-1. This is the easiest option. SHA-1 is supported by practically all CAs and all software. It is therefore relatively easy and cheap to make the transition. The disadvantage of this option is that SHA-1 already includes known vulnerabilities. Although this is not yet a practical threat, the strength of SHA-1 may soon come under pressure. If this is the case then it will be necessary to make a new transition, which will involve fresh costs.
2. Transition to SHA-2. This option is less simple. Support for SHA-2 is far from being a matter of course for all CAs and all software. Before you migrate to SHA-2 you will need to find out if you are also going to have to upgrade your software. This of course entails additional costs. Moreover, the use of such a certificate can also create a problem for some visitors to your website if their browser does not support SHA-2. There is also an advantage to migrating to SHA-2. It is by far the most future-proof option at this time because SHA-2 is expected to last another ten years before any practical attacks will be possible.

⁵ The extension, *SSL Blacklist*, can be found on <http://www.codefromthe70s.org/sslblacklist.aspx>. This extension was first developed in connection with the vulnerability in OpenSSL on Debian. For more information on that vulnerability see GOVCERT.NL factsheet FS-2008-04 at <http://www.govcert.nl/download.html?f=111> (Dutch only).

⁶ You can read the position of NIST regarding SHA-1 on <http://csrc.nist.gov/groups/ST/hash/statement.html>

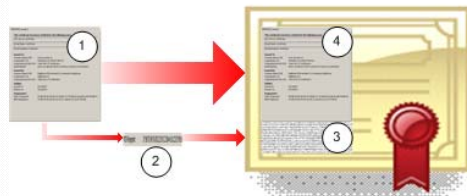
⁷ See http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html

BACKGROUND INFORMATION

Digital signatures, hashes and collisions

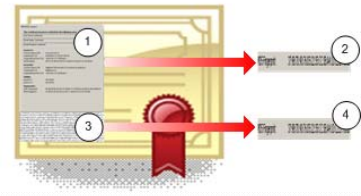
A digital signature on a certificate is created by means of a 'hash function'. Well known examples of hash functions are MD4, MD5 and different variants of SHA. A hash function calculates a short character string (the hash) from a set of data. The set of data can be all kinds of things: a document but also an e-mail or the source code for a program. You can see the role a hash plays in signing a certificate in the following box.

Placing and verifying a digital signature



Signing

A signed certificate is created as follows. The Certificate Authority (CA) receives data from the applicant (1). The CA calculates a hash from this (2). The CA encrypts the hash (3). This constitutes the signed certificate together with the original data (4).

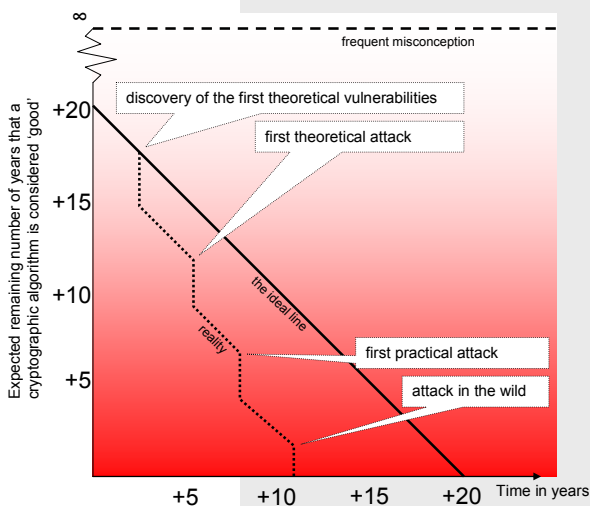


Monitoring

A browser verifies a certificate as follows. The browser calculates a hash (2) from the original data on the certificate (1). After this the browser decrypts the hash that was encrypted by the CA (3). If both hashes (2 and 4) are identical then the certificate has not been modified and can therefore be trusted.

Because a hash is a short character string, there will always be several sets of data that generate the same hash. A good hash function therefore has the following characteristics:

- Every modification to the original set of data must (following recalculation) result in a significantly different hash.
- It must not be possible to derive the original set of data (the pre-image) from the hash.
- It is very difficult to create a second set (the second pre-image) oneself from an existing set of data that will generate the same hash. This is not yet possible with MD5, but the current attack is coming close to this.
- It is very difficult to create two sets oneself which generate the same hash (a collision). This has been possible for quite some time with MD5⁸.



The half-life of encryption

Good encryption encrypts data in such a way that it can only be deciphered with the correct key. A 'brute force' attack, which is simply calculating all possible keys and selecting the correct one from these, is not feasible in practice because it takes too much time.

It would be incorrect to believe that one can generate a good product or good service for an indefinite period on the basis of a 'good' cryptographic algorithm. The reality is in fact different.

There are two factors as a result of which a good cryptographic algorithm will nevertheless lose its value eventually. In the first place the calculation power of computers increases every year, so that a brute force attack will become practically feasible at a certain moment. Secondly, it is very difficult to create a good cryptographic algorithm. Small errors in the algorithm can have far-reaching consequences as regards its robustness. Therefore it is very important to subject an algorithm to extensive peer review (checks by others). Through this process errors almost always

come to light which the original designer has failed to see. But even then it is highly unlikely that the algorithm will be fault-free. The diagram clearly shows that every weakness shortens the lifespan of an algorithm.

⁸ The results of research into collisions in MD5 are maintained on <http://www.win.tue.nl/hashclash/>. Software that can be used to create MD5 collisions can be found on this page as well.