The death of MD5 (and some SSL certificates)

BROKEN CHAIN OF TRUST

Researchers set out to compromise MD5 in an effort to convince people to stop using it. We explain how the attack worked and what this means for you. **BY KURT SEIFRIED**

Message Digest algorithm 5 (MD5 for short) is a one-way cryptographic hashing function. Put in its simplest terms, it takes input, mangles it, and generates a 128-bit value (usually expressed as a 32-character hexadecimal number such as *76ffd163bd23504cfeb873a9c027b2ed*). The same input (e.g., *password*) will always have the same output (for example, *5f4dcc3b5aa765d61d8327deb882cf99*). So why use MD5? When cryptographically signing data (such as email or SSL certificates), it is much more efficient to sign a cryptographic signature of the data rather than the entire block of data itself (128 bits of data compared with a kilobyte or more for an SSL certificate).

MD5 is widely used. For example, many Linux distributions use it by default to hash password values in the */etc/shadow* password file, numerous SSL certificate authorities support it, and many application vendors use it rather than stronger algorithms such as SHA-1 or SHA-256 (a hashing algorithm similar in functionality to MD5).

The Trade-off

Like any security issue, a continuum of choices generally ranges from a combination of "cheap, easy, insecure, and computationally inexpensive" to "expensive, difficult, secure, and computationally expensive." In the case of MD5, it falls somewhere in the middle, not so much because of any conscious choices to cut corners, but largely because of its age (it was invented in 1991).

The largest flaw with MD5 is its limited hash size: At 128 bits, it is significantly smaller than many modern hashing algorithms such as SHA-1 (160 bits) or SHA-256 (256 bits). This limited hash size allows attackers to conduct what is known as a "birthday attack." In cryptographic terms, a birthday attack occurs when two different inputs (e.g., two different but validly formed SSL certificate requests) have the same output after being passed through a hashing function such as MD5. Because MD5 only has 2^{128} possible outputs, and there are obviously more than that many possible inputs (e.g., 100 standard ASCII characters represent 2^{800} possible inputs) [1]. Even something as simple as a date stamp and a serial number can easily represent over 2^{128} potential inputs.

Realistically, the only thing preventing someone from attacking MD5 is the amount of computational power needed and the resistance of the algorithm to various types of attacks.

Unfortunately several weaknesses were found in MD5 (some as far back as 1993), and computational power got very cheap much faster than anyone expected, even taking Moore's law into account. So armed with several known weaknesses in MD5, a group of researchers set out to attack and compromise it in a way that would finally demonstrate the weaknesses in a conclusive manner (and hopefully convince people to stop using it).

One of the most public uses of MD5 is in SSL certificate signing; a small group of certificate authorities (such as Thawte, RapidSSL, RSA, and VeriSign Japan) still use MD5, making them vulnerable to this attack. Now all that was needed was for an attacker to create two certificate requests: one a standard and legitimate request for a secure website and the other a certificate with signing authority allowing one to use it to create signed certificates at will.

How the Attack Worked

In a nutshell, the researchers found a certificate authority that issued certificates in a way that allows the

attacker to control the data placed in the certificate by the certificate authority. It's no good for you to create two certificates that have matching MD5 signatures if the certificate authority adds a timestamp and random serial number, thus changing the MD5 signature for the certificate. The vulnerable certificate authority used sequential serial numbers (for example, 1001, 1002, 1003) and timestamps that were exactly six seconds in the future from the time the user submitted the certificate request to their website.

Now all the researchers had to do was find sufficiently cheap computing hardware so that they could calculate a pair of certificates in a reasonable amount of time.

Fortunately, the PlayStation contains a specialized chip called the "Cell" processor that is uniquely suited to calculating a birthday attack, and with a mere 200 machines (about US$ 80,000 at retail prices) the researchers were able to calculate the initial data needed to find a matching set of certificates in 10 hours. Further computation was needed to generate the certificates, which was done on a quad core system (in other words, not a very expensive machine).

Ultimately the researchers were able to carry out a successful attack that gave them a certificate that could be used to sign other certificates. Fortunately, because they are the good guys, they had the certificate dates set to 2004 so that it was expired and raised a warning when encountered.

What This Means for You

Although this attack requires a relatively modest budget (approximately $100,000 for hardware), the technological sophistication needed is quite high. Additionally, only a handful of certificate authorities were affected by this problem because the vast majority stopped using MD5 some years ago (when someone finds a theoretical weakness in a security system, a practical exploit is often not far behind).

Although this type of attack is the holy grail of bad guys abusing the web (using it, they can pretend to be your bank or an online store), it is unlikely you will see an attacker creating and using a signing certificate to impersonate websites. The main reason is that there are much easier ways to impersonate a secure website.

Fake SSL Certificates

The bad news is attackers have a much simpler way to get an SSL certificate for an arbitrary site: They can simply buy one. In one case, someone was able to buy a certificate for Mozilla.org from an SSL reseller that did no checks to ensure that the individual was allowed to get certificates or even was affiliated at all with Mozilla.org [2].

In other cases, attackers have been able to get a certificate "Issued in minutes" (to quote RapidSSL.com) with fake requests by faxing in orders on official looking letterhead from organizations for which they want the SSL certificates. To put it simply, they claim to verify your information securely somehow in a few minutes (realistically, some simply query the WHOIS information for your domain and email the contacts listed, giving them a chance to cancel the certificate order). With CCL certificate authorities selling certificates to virtually anyone requesting them with only minimal oversight, the system can be abused easily by attackers.

How to Protect Yourself

Unfortunately, you can't do much to protect yourself. Even if you enable certificate revocation checking in your web browser, if attackers use the MD5 method to create a fake authority certificate, they can simply leave out the certificate revocation information (meaning your browser can't check to see whether it has been revoked or not!). Disabling the root certificates for the authorities that still support MD5 will break a large number of websites, some of which you might want to use (which is largely why, so far, Firefox has not blocked the use of the Comodo SSL authority).

If you want to do this for yourself, the instructions are: Go to the *Advanced | Encryption* settings tab in Firefox and click on *View Certificates*, then select *Authorities*. Now search for the certificate you want to disable (manually, because there is no search function) and select it, then all you have to do is select *Edit* and uncheck the box for *This certificate can identify web sites*.

In the future, websites that use this certificate authority to get their website certificates will show up as not signed by a trusted authority, and you'll get the Firefox warning. The process is just as, or more, convoluted in other web browsers. Oh, and most of the certificate authorities in your web browser have no descriptive information. Some of them don't even have valid websites (because they have gone out of business and sold their signing certificates to other companies). Web browsers could make significant improvements in this area.

Conclusion

The good news is that the organizations responsible for technical standards such as MD5 and SHA-1 have not been sitting still. The American National Institute of Standards and Technology (NIST) is holding a competition to develop and choose a new hashing algorithm that should be good for several decades of use [3]. Web browser vendors have also not been standing still. The advent of new "Extended Verification" certificates place much stricter controls on how certificates are issued and to whom (although one could argue this should have been done all along).

Unfortunately, without education, most users will not be able to tell the difference between a website with a "standard" certificate and one with "Extended Verification," although in most cases, the browser places the company name on a green background in the address bar. ■

INFO

[1] Creating a rogue CA certificate: *http://www.phreedom.org/research/rogue-ca/*

[2] mozilla.dev.tech.crypto: *http://groups.google.com/group/mozilla.dev.tech.crypto/browse_thread/thread/9c0cc829204487bf?pli=1*

[3] Cryptographic hash algorithm competition: *http://csrc.nist.gov/groups/ST/hash/sha-3/index.html*

THE AUTHOR

Kurt Seifried is an Information Security Consultant specializing in Linux and networks since 1996. He often wonders how it is that technology works on a large scale but often fails on a small scale.