Why You Shouldn’t Completely Trust Files Signed with Digital Certificates

By Andrey Ladikov

Published: 2015-01-29 · Archived: 2026-04-05 14:32:41 UTC

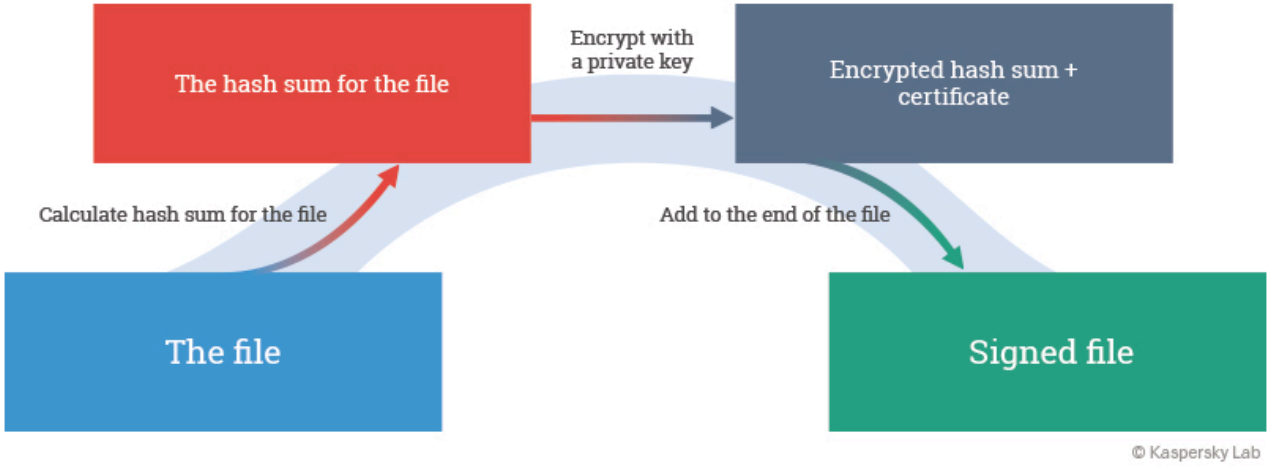
A digital certificate with a file is always seen as a token of its security. For users, a digital certificate is an indication that the file does not contain malicious code. Many system administrators develop their corporate security policies by allowing users to launch only those files that are signed with a digital certificate. In addition, some antivirus scanners automatically consider a file to be secure if it is signed with a valid digital certificate.

However, users’ absolute trust in files signed with digital certificates encourages cybercriminals to search for various ways to have their malicious files signed with the same trusted digital certificates to help use them in their criminal schemes.

This article looks into the main threats associated with signed files, and suggests practical methods of mitigating the risks associated with launching them.

Creating digital signatures for files

Before we explore the threats associated with using digital certificates, let us first look into the process when a file is signed with a digital certificate:



1. 1 The software developer compiles the file.
2. 2 A hash sum (MD5, SHA1, or SHA2) is calculated for the file.
3. 3 That hash sum is encrypted with the software developer’s private key.
4. 4 The obtained encrypted block of data and the digital certificate are added to the end of the file.

The digital certificate contains the software developer’s public key, which can be used to decrypt the message and check the file’s integrity. It also contains information with which the software developers’ authenticity can be

checked.

The authenticity of the file's manufacturer is confirmed with the help of the Certification Authority (CA). This entity certifies to other users that the public key that decrypts the hash sum and checks the file's integrity does indeed belong to the developer in question. To do so, the CA signs the developer's certificate and thus testifies that the unique pair of public and private keys belongs to that particular developer. A certificate from the CA testifying that the file is authentic is also added to the end of the file alongside the developer's certificate.

CA certificates are verified by no one other than these entities. For Windows to trust the certificates issued by a certain CA, that CA's certificate must be placed into the operating system's storage of certificates. The certificates of the most authoritative CAs have undergone an audit and are automatically included into the storage and are delivered to users along with Windows updates. Certificates issued by other CAs can be added to the storage at the discretion of the user.

The use of trusted certificates by cybercriminals

Now let's look at attacks that can be carried out at each stage of signing a file. We are not interested in theoretical attacks based on the weaknesses of the encryption algorithms used to sign the file, but will concentrate instead on the attack methods most often used by cybercriminals in practice.

Planting malicious code at the file compilation stage

In many large software companies, files are signed automatically immediately after the file compilation is complete. File compilation is done centrally on a dedicated Build server.

If cybercriminals gain access to a software manufacturer's corporate network, they can use the corporate Build server to compile a malicious file on it, so it automatically gets signed with the company's digital signature. As a result of this attack, cybercriminals obtain a malicious file signed with a valid digital certificate.

In practice this type of attack is quite rare because large software manufacturers have adequate security in place to protect their Build servers. Nevertheless, there have been identified cases when targeted attacks were successfully conducted and malicious files were signed with a trusted company's certificate.

Stealing a private key

Sometimes, cybercriminals succeed in penetrating a corporate network and gaining access to a private key used to sign files. With that key, they can sign any malicious file and pass it off as a file produced by a legal software manufacturer.

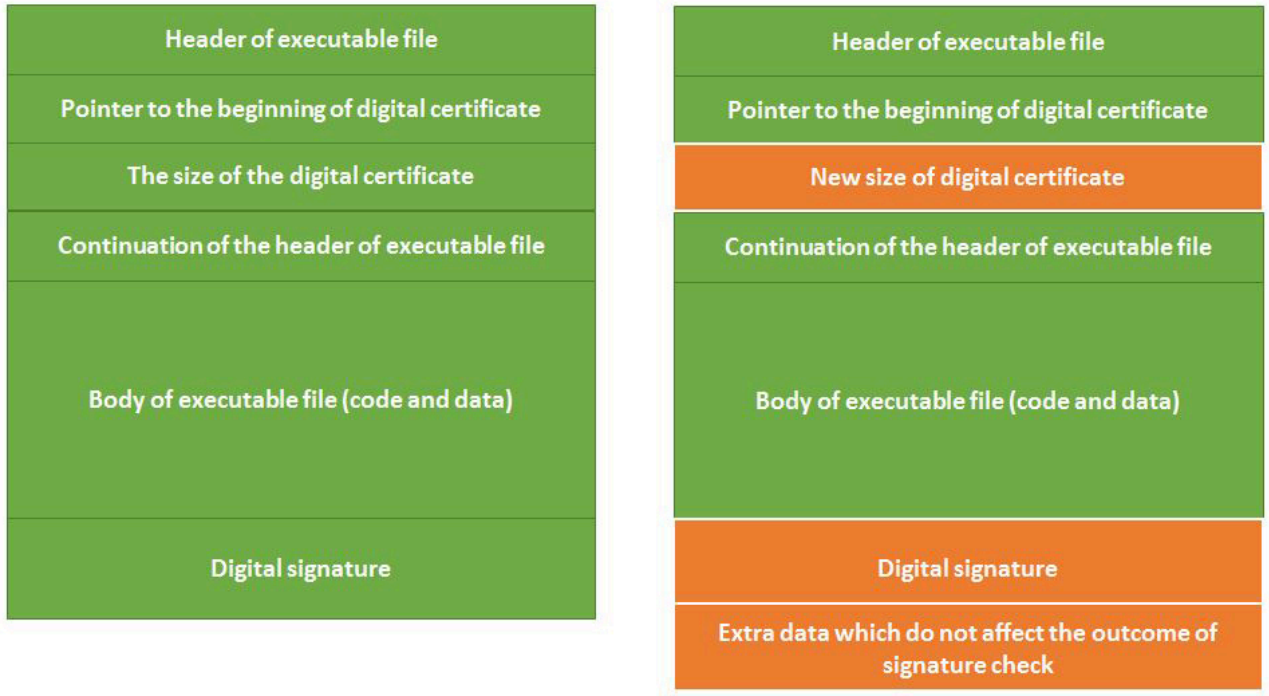
One way to steal a private key is to use specialized malware created specifically for this purpose.

After stealing a private key, the cybercriminal either uses it or sells it to someone else to use. The more famous the software manufacturer from which the key was stolen, the more valuable the key will be among cybercriminals. Software from well-known manufacturers does not attract any suspicion from users and security administrators on corporate networks.

At the same time, large software manufacturer companies keep their private keys in dedicated, well-protected hardware modules, which makes it much more difficult to steal them. As a result, private keys are typically stolen from smaller companies or private software manufacturers who do not pay enough attention to security.

Vulnerabilities in the algorithms that check executable file signatures

For an operating system to know which part of the file is supposed to contain the information about the presence of a digital certificate, the header of each signed executable file includes 8 bytes of data that contain information about the location and the size of the digital certificate. These 8 bytes are ignored when checking the file's signature. If a block of data is added to the end of the file's signature, and the size of the signature is increased by an appropriate amount, these changes also will also have no effect on the outcome of the signature check. This makes it possible to gain extra space in a signed file where data can be added without affecting the outcome of a signature check.



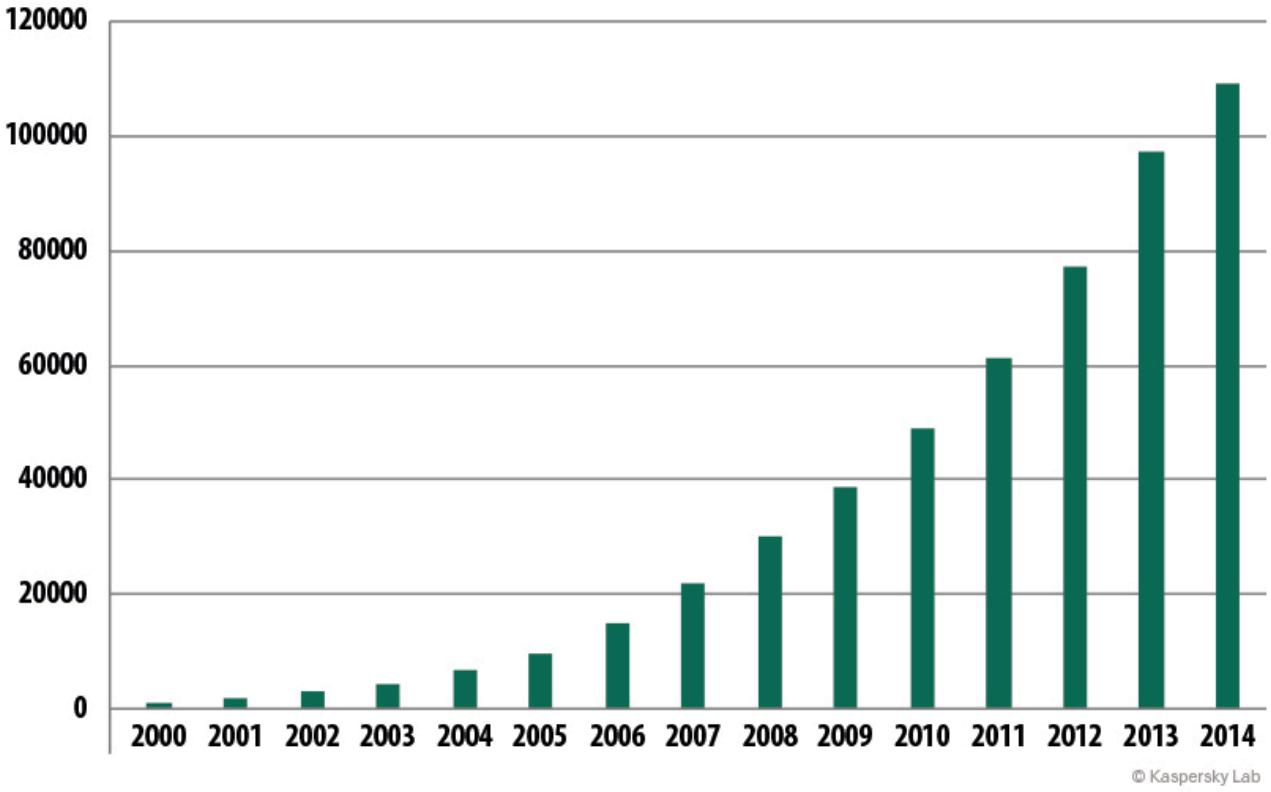
This algorithm is used actively in legal web installers: software developers who create these web installers modify the size of the digital signature to make room for an additional block of data, so that the digital certificate block includes a link to a file for that installer to download from the software developer's page and install on the users' system. This is a practical approach for software developers because the installer does not have to be re-signed each time the link to the software distribution kit is changed: it is enough to simply change the link stored in the digital signature block.

Cybercriminals, in turn, can use this algorithm for their own purposes. A cybercriminal takes a web installer for legal software, and changes the link so a different distribution kit to be downloaded. The installer then downloads and installs malware on the user's system. After that, the cybercriminal uploads the modified installer to software distribution sites.

To fix this vulnerability, Microsoft released a security update that enforces a rigorous check of each file's digital certificates. However, this update does not apply automatically because many software developers use the above algorithm in their installers, and their software programs would be considered unsigned if this update was applied across the board. The user can enable this update manually, if required.

The use of legally obtained certificates

A few years ago, digital certificates were actively used by large software manufacturers that were legally registered companies. Today, certificates are used increasingly often by individual software developers and small companies. The graph below shows how the number of certificates with which to sign software code known to Kaspersky Lab changed over time. As can be seen, the number of certificates is steadily growing year on year.



The number of certificates verified by CAs and known to Kaspersky Lab

The procedure of purchasing a certificate to sign executable code is quite simple: individuals must present their passport details, and companies must present their registration details. Some certificate-issuing CAs make no further checks into the activities of the companies seeking to purchase the certificate. All a CA does is it issues a certificate entitling the client to sign executable files, and verifies that the certificate has indeed been issued to the specific person or company.

This enables cybercriminals to legally purchase a certificate to sign their malicious and/or potentially unwanted software.

It is companies manufacturing potentially unwanted software that most often purchase certificates. On the one hand these companies do not manufacture malware programs, so they can legally purchase a digital certificate to

sign their software. On the other hand, they produce software annoys users. In fact, they get their software signed with digital certificates precisely to encourage users to trust them.

Untrusted certificates

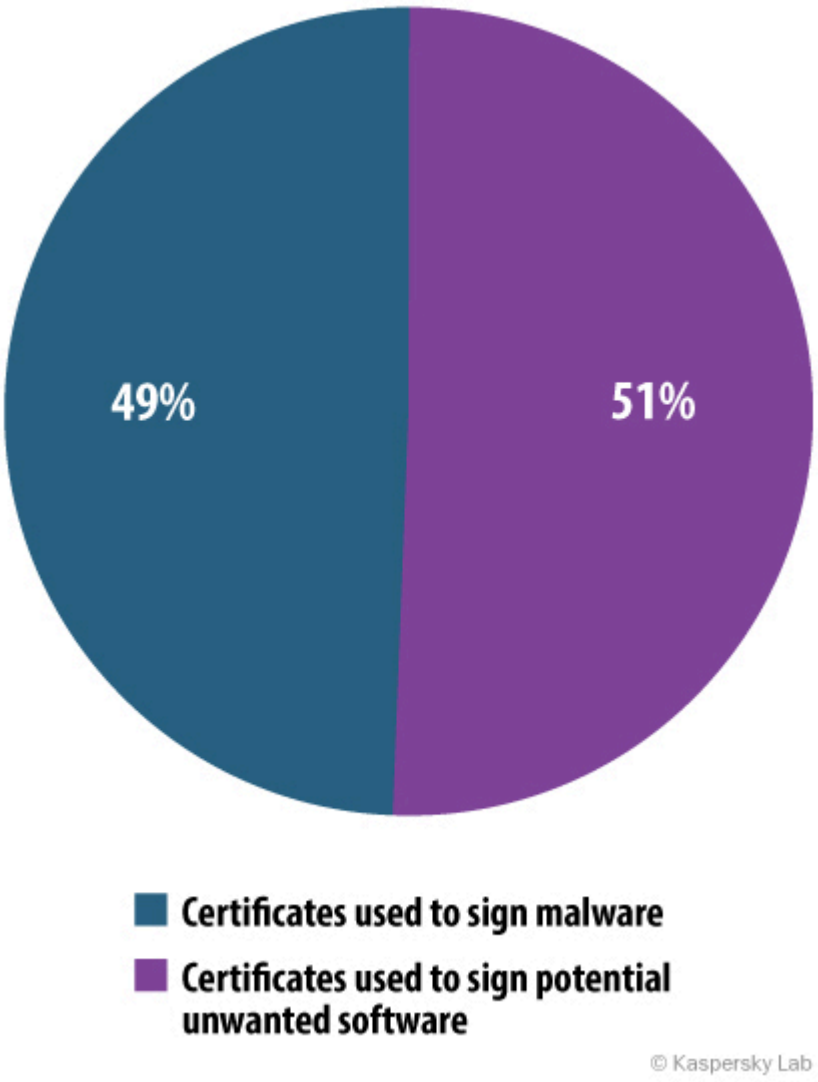
In all cases described above, be it stealing a private key, compromising a company's infrastructure and signing a file with that company's digital certificate, or purchasing a certificate with the intent of signing malware with it, the end result is the same: a trusted certificate is used to sign a malicious file.

Therefore, these certificates cannot be considered trusted in spite of the fact that their authenticity has been verified by a CA, as they were (or continue to be) used to sign malicious files. We will hereafter describe these certificates as 'untrusted'.

If a private key is stolen from a software developer, or a company's infrastructure is compromised and a trusted certificate is used to sign a malicious file, the CAs cease verifying the trustworthiness of the certificate that was earlier issued by them (a process also known as recalling the certificate). The speed of the CA's reaction depends on how soon it becomes known that the certificate has been used by somebody other than the legitimate developer.

However, when a certificate was purchased to sign potentially unwanted software, the CAs do not always recall the certificate. As a result the certificate could remain valid and be used to sign potentially dangerous software.

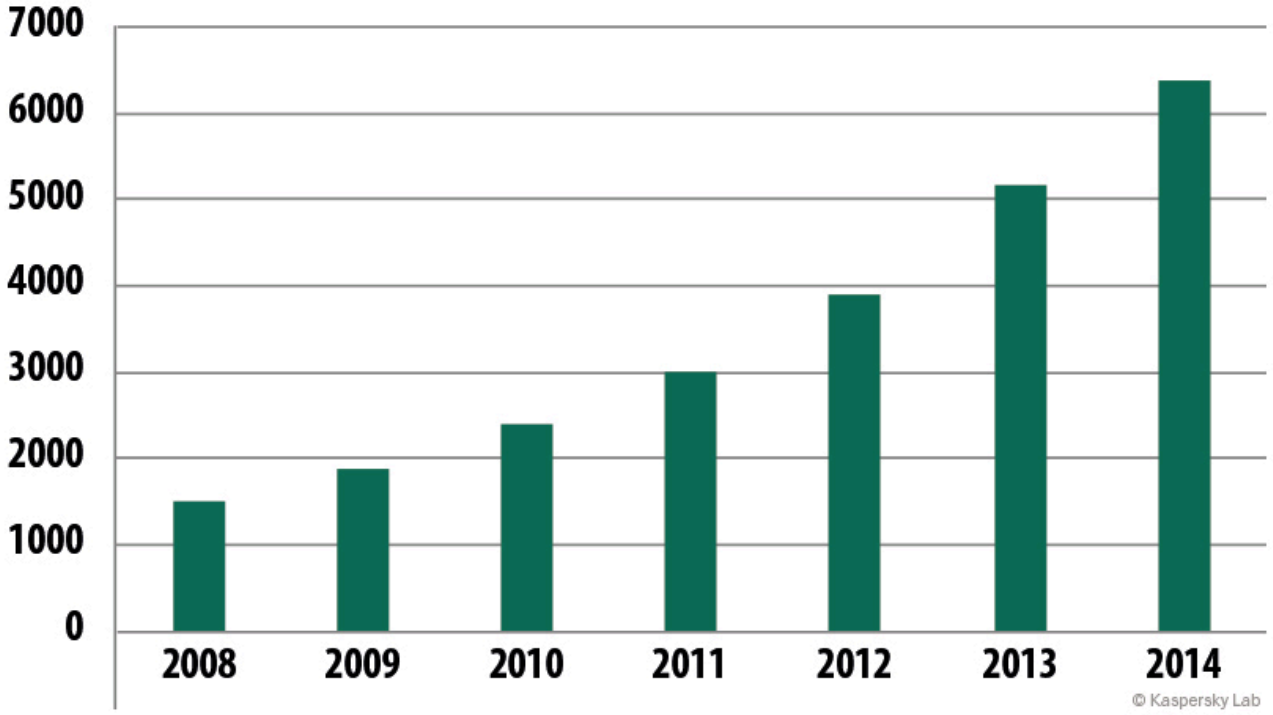
The following chart shows the proportions of untrusted certificates used to sign malware and potentially unwanted software (Kaspersky Lab data).



Breakdown of untrusted certificate numbers by their type

Methods of protection against launching software programs signed with untrusted certificates

We have discussed the most popular cybercriminals techniques to get files signed with digital certificates. Recently we have seen an increasingly significant problem concerning malicious and potentially unwanted files being signed with digital certificates. In 2008, 1,500 certificates were later used to sign malware; in 2014, there were more than 6,000 of these cases.



The number of untrusted certificates known to Kaspersky Lab

Given the growing number of threats associated with malicious files signed with digital certificates, users and administrator can no longer risk placing blind faith in signed files and just allow them to be launched simply because they have a digital certificate.

Here are a few practical tips to reduce your chances of launching a new malware program that has a valid digital certificate and hasn't yet reached your anti-virus databases:

1. 1
    1. 1.1 Only allow the launch of software programs signed by a reputable manufacturer.

You can substantially reduce the risk of infection on your computer by disabling the launch of all software programs signed with digital certificates belonging to unknown software manufacturers. As described above, certificates are most often stolen from smaller software companies.

1. 1
    1. 1.1 Only allow programs to be launched after they are identified by their unique digital signature attributes.

Several certificates issued to the same company may be distributed under the same name. If one of these certificates is stolen from a reputable company, a check that automatically trusts well-known publishers would allow a file signed with a stolen certificate.

To prevent this from happening, before allowing programs signed with known certificates to launch, it is necessary to check other attributes as well as the certificate name. These attributes might be the serial number or

certificate fingertip (hash sum). Serial numbers are only unique within the range of certificates issued by a single CA, so we recommend checking this along with the company that issued the certificate in the first place.

1. 1
    1. 1.1 Activate the MS13-098 security update.

For experienced users and system administrators, it is advisable to enable update MS13-098 – it fixes an error which enables the inclusion of additional data in a signed file without tampering with the file's signature. To read more about how to activate this update, follow this link to Microsoft Security Center.

1. 1
    1. 1.1 Do not install certificates from unknown CAs into your security storage.

It is not a good idea to install root certificates from unknown CAs into your storage. If you do so, any files signed with a certificate confirmed by that specific CA will subsequently be considered trusted.

1. 1
    1. 1.1 Use a trusted certificates database from a security software manufacturer.

Some security software manufacturers, including Kaspersky Lab, include a database of trusted and untrusted certificates in their products; this database is updated on a regular basis along with the anti-virus databases. With this database, you will receive prompt updates about as-yet unrecalled certificates used to sign malware and/or potentially unwanted software. Files signed with untrusted certificates from this database require enhanced monitoring by the security product.

The database of trusted certificates includes certificates from reputable software publishers that were used to sign trusted software programs. If a certificate is listed in this database, it is a strong indicator that corporate application control can allow the application to launch.

If this kind of database is included in a security product it will help make the administrator's job easier, sparing them the need to create and maintain an in-house database of trusted certificates.

The number of digital certificates used to sign malware and/or potentially unwanted software is doubling every year on average. That is why it is vital that companies exercise ever greater control over signed files with the help of security product tools, and follow the above security policies.
