CAN WE TRUST A TRUSTEE? AN IN-DEPTH LOOK INTO THE DIGITALLY SIGNED MALWARE INDUSTRY

Adrian-Stefan Popescu & Gheorghe Jescu
Bitdefender, Romania

Email {apopescu, gjescu}@bitdefender.com

ABSTRACT

Over the last couple of years, the *Windows* operating system has implemented a growing number of user notifications before a file is executed, ranging from messages confirming the execution of downloaded applications to alerts for files that are not digitally signed. An increasing number of developers are using certificates issued by Certificate Authorities (CAs) to create a more trustworthy environment for users. Although certificates should be used by legitimate developers only, we are seeing an increasing number of malware files that are digitally signed with trusted certificates. This evasion technique is successful not only against the operating system, but also against security vendors that are creating additional filters for trustworthy files.

This paper presents an analysis of different methods for using a certificate to digitally sign malware files, using either stolen certificates that were originally issued to a trusted IT company, or certificates that have been issued to certain developers who use them with malicious intent. In the context of certificates being issued by a trusted CA, we wonder if there is a possibility that a potentially unwanted behaviour was intended from the beginning.

Finally, this paper tries to raise awareness about possible selection issues at the CA level. Has an in-depth analysis been completed on the companies that request certificates or the files that will be signed? What should happen when a certificate is explicitly revoked for malicious behaviour?

1. INTRODUCTION

Modern operating systems have improved their architecture so that they can make better use of digitally signed files. Having all operating system files digitally signed should reduce the incidence of malware attacks. Unfortunately, malware has started to use digital signatures as well. Some of these digital certificates have been revoked by the authorities that issued them, but many of them are still valid and used by malware to trick the user into thinking that the file was created by a reliable source. This also creates a window of opportunity for some unwanted applications (adware, keyloggers, etc.) to increase their credibility as trusted applications. In the past eight months, the volume of digitally signed malware has increased. The chart in Figure 1 shows this trend.

This paper intends to take a look at the situation. We will discuss the multi-layered process of digitally signing a file, followed by possible vulnerabilities that can be exploited. We will explore some of the benefits to the malware author of digitally signing the file. Another section is dedicated to observations and cases of vulnerabilities used. Finally, we will present conclusions and ask questions regarding the safe usage of digital certificates.

2. STEPS IN CREATING A FILE WITH A DIGITAL SIGNATURE

A company that produces software for end clients and wants to create not only software, but also an entire environment of trust between it and the client, should use digital signatures. We will consider executables for the *Windows* operating system (only) due to its widespread use. First, the company must choose a Certificate Authority (CA). This should be one of the companies that are considered 'trusted' by *Windows*, so that no alerts are generated when applications are downloaded and executed. There are 52 different trusted CA names that are associated with only 22 companies. The most widely used are from the *Symantec Group* (*VeriSign* and *Thawte*), *Comodo Group*, *Go Daddy* and *GlobalSign*. Usually, a company will take into account not only the name and experience of the CA, but also the price of a certificate. Although there are cases where digital certificates are free of charge, such as *CACert*, not all of these are considered trustworthy by *Windows*.

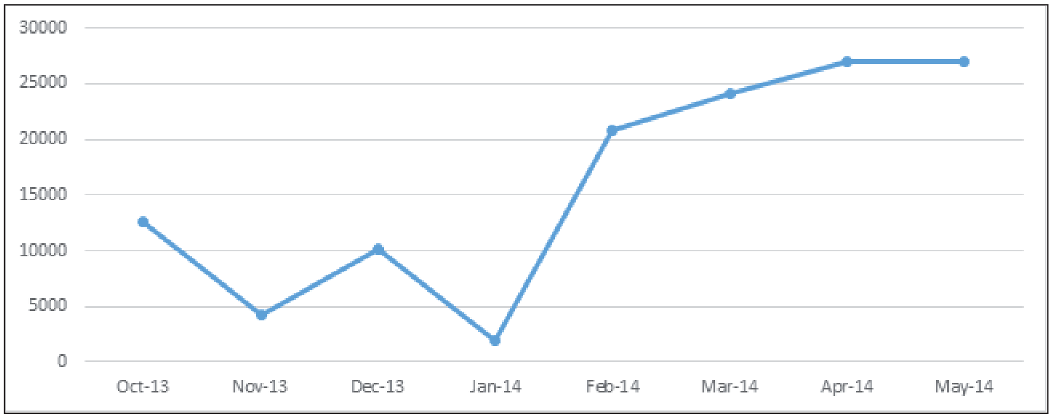


Figure 1: Number of similar digitally signed malware files.

Each CA has its own set of requirements and information that must be supplied before a certificate is issued. Usually the information requested regards a company's registration number, current address, contacts and service bills. These are needed for a CA to confirm that the company really exists and that all the information is correct. Also, the CA requests that the data is real and no pseudonyms are used. Some CAs mention that the process will involve a step in which the company's registration number and information will be validated using a third-party channel. Although almost all CAs have a methodology of good practice when a certificate is issued, it is doubtful as to whether the receiver of the certificate has indeed made all the arrangement to secure the private key.

After all the information has been checked and the CA is certain that the applicant company has no malicious intentions and that all the information is real, a certificate is issued and sent to the contact. From this point, the certificate requester is responsible for signing files and for protecting the integrity and safety of the certificate and private key.

Almost all CAs specify that the subscriber shall not use the certificate in association with any unlawful or suspicious activities. If such a situation arises, or if the exclusive control over the certificate is lost, the certificate will be revoked and listed on the Certificate Revocation List (CRL). This list is published either each time a new certificate is revoked or periodically (when no certificate has been revoked).

Usually, the CA requests that a certificate is installed on a single system and used by either a single person or a very limited number of persons. Also, the system on which the certificate is installed should not be connected to the Internet, but only to a single secure network, and a service to generate a digital signature is created for the software developers. As stated among good practice methods, a file should not be signed with the private key until the point of its deployment to the public. Until that moment, a test certificate should be used.

3. POSSIBLE VULNERABILITIES IN THE MULTI-LAYER PROCESS OF CODE SIGNING

In this section, we classify the possible vulnerabilities based on the layer exploited.

3.1 The use of different social engineering techniques to trick a CA into generating a valid certificate

a. Behave as if the request comes from a large and well-known company. It is possible that some well-known companies, within which multiple certificates are issued to multiple persons, would have a less constrained validation process. One such example is the famous case of the certificate wrongly issued to 'Microsoft Corporation' in 2001 by *VeriSign* [1]. This generated a discussion about the availability of the revocation lists and how these should be checked and administrated. The revocation problem could not be solved without a patch because the location for the CRL (certificate revocation list) was not embedded in the digital signature and no hard-coded location for the *VeriSign* revocation list was available. There was also a problem with *Internet Explorer*, which could not check automatically for revocation at that time.

b. Use information collected from the Internet to act as if you are a certain person or representative of a company. This scenario is common thanks to the large amount of information that can usually be collected about a person or a company through Internet searches and similar methods. It is possible that some information is not real, but very similar to the real information.

c. Use information about entities that don't exist or from companies registered in countries where information validation is difficult. Some CAs specify that the validation process for companies that are registered in countries with laws that make it hard for them to check the authenticity of the information submitted, will be more thorough. Although a large number of certificates are issued for companies based in countries with difficult validation, these certificates are used more for aggressive toolbars or potentially unwanted applications. Also, some of these are issued to companies that have doubtful names or no longer exist.

d. Offering misleading or incomplete information about the software that will be developed, or using the fine line between potentially unwanted application and malware to hide the real scope of the software. One of the reasons for the revocation of a certificate is the use of the certificate for digital signing of harmful code [2]. It is unclear where this line is set. If an application can download files from a third-party website, is it considered not harmful until it downloads a malware file? It is very unlikely that a company will state in the certificate request form that it generates aggressive advertisement or bundle downloads, thus harming its image or credibility.

3.2 Malware with digital-certificate-stealing abilities

Some of the certificates that malware authors use to increase their level of trust are real certificates, released to trusted companies or trusted users. They steal them by infecting the computer on which the trusted company/user keeps their private key or certificate. *Windows* systems keep a store of digital certificates and offer a specific set of APIs to access them. By using APIs such as CertOpenSystemStoreA, PFXExportCertStoreEX, CertCloseStore, CertEnumCertificatesInStore and CertDuplicateCertificateContext, one can obtain all the necessary information one needs to create a digital signature using the private key that is stored on that computer or the certificate itself. We have discovered a large number of malware samples that use these APIs to gather certificates and private keys from systems. Not only do they steal the keys, but some of them delete the keys from the system after they get a copy of them. Figure 2 shows an example of the flow of APIs used by a piece of malware.

We have studied malware with this kind of behaviour since the beginning of 2013. We can see from the chart in Figure 3 that since the beginning of 2014, the number of malware samples

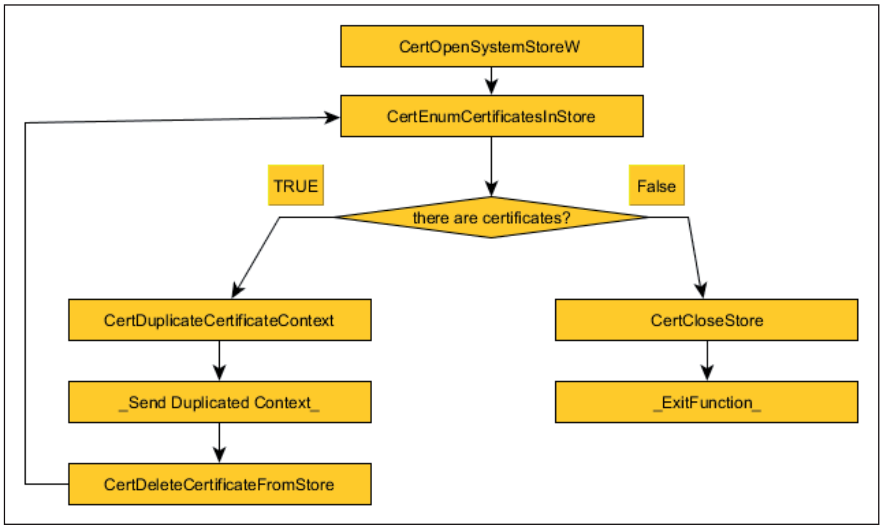

CertOpenSystemStoreW
CertEnumCertificatesInStore
TRUE
False
there are certificates?
CertDuplicateCertificateContext
Send Duplicated Context
CertDeleteCertificateFromStore
CertCloseStore
ExitFunction

Figure 2: Digital certificate handling by malware.

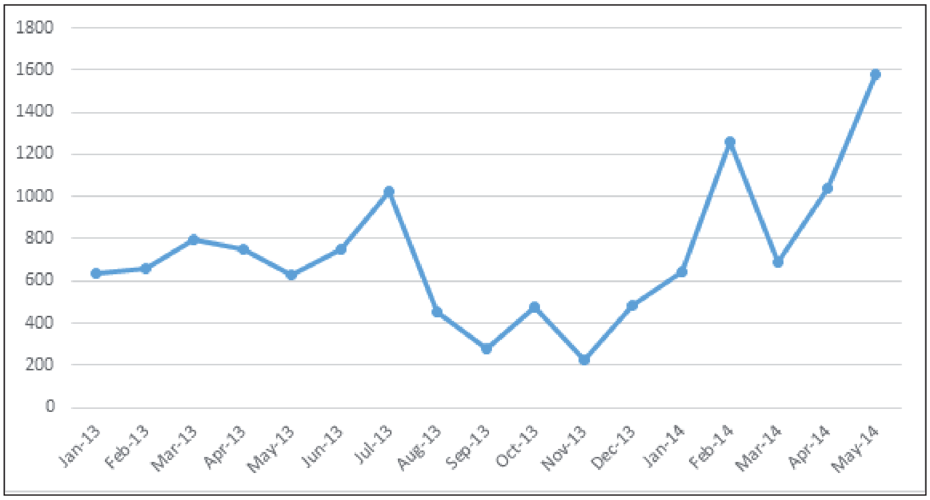


Figure 3: Number of similar malware files with certificate-stealing capabilities.

with digital-certificate-stealing capabilities has increased. This fact led us to the conclusion that this kind of malware is becoming more popular as the request for stolen certificates grows.

3.3 Cryptographic attack and MD5 or SHA-1 forgery

In code signing, the two hashes used are MD5 and SHA-1. There are well demonstrated weaknesses of MD5, including the way two files with the same MD5 can be created [3], having a different behaviour but containing slightly different bytes. This method can be used by malware authors to create files with malicious behaviour, but the files must not be too different from the legitimate file, for this to be generated in a feasible time. At CRYPTO 2005 [4], a mathematical approach of creating SHA-1 collisions with a complexity of less than 269 was presented. The complexity of creating a malware file with the same hash as a clean digitally signed file is still high, but considering the large botnets controlled by malicious authors nowadays, it is possible for one to be generated. Although we didn't encounter such a case, we considered that it is computationally possible to create such a file.

4. WHY WOULD YOU USE A DIGITAL CERTIFICATE TO SIGN A MALWARE FILE?

4.1 Evasion of detection experiment

Most anti-virus products choose to avoid scanning files that are digitally signed. We conducted an experiment with 10 well-known anti-virus products. We created a digital certificate of our own, took some malware that was detected by all of the products, and signed the malware with that certificate.

The certificate was registered on the operating system on which the anti-virus products were installed. As we expected, some of the malware files were no longer detected, even though the digital certificate has been created by us. Every anti-virus product that we tested lost detection for at least one of the files. The biggest loss recorded was of 37% for one of the products. On average, 10% of the tested files were no longer detected.

We then conducted the same experiment, with the same anti-virus products and the same number of files, but this time instead of malware files, we used adware components and other possibly unwanted applications. Again, every anti-virus product that we tested lost detection for at least one of the files. The biggest loss recorded was of 25% for one of the products. On average, 20% of the tested files were no longer detected.

Lately, we have discovered a new type of malware that uses digital signatures to avoid anti-virus detections. The idea is to use a digitally signed file that acts as an interpreter and executes a malicious script. The most common scenario is to have a RarSfx archive that contains the AutoIt interpreter (which is digitally signed) and a malicious AutoIt script in its raw form. Upon execution, the RarSfx archive drops the files and executes the AutoIt interpreter with the malware script as its parameter. Since the AutoIt interpreter is digitally signed, some security products will ignore this file/process.

4.2 More trustworthy potentially unwanted applications

Another reason for digitally signing malware files can be found by looking at the situation from an adware author's point of view. Having a digitally signed file can increase your credibility as an honest software developer in the minds of average users and inside companies. As an example, we took a look at a digitally signed piece of adware, Adware.Mplug. In a two-month period, we discovered over 2,000 different samples, all digitally signed, that infected more than 147,000 different systems. These systems were all over the world, from Europe (France >31,000, Germany >5,000) to North America (USA >11,000, Canada >5,000), South America (Brazil >3,000, Colombia >400), Asia (Japan >3,500, India >2,700), Australia (>1,700), and even Africa (Egypt >800, South Africa >890).

As you can see in Figure 4, there is no exception regarding continents. People from all over the world tend to trust this kind of application, especially if it is digitally signed.

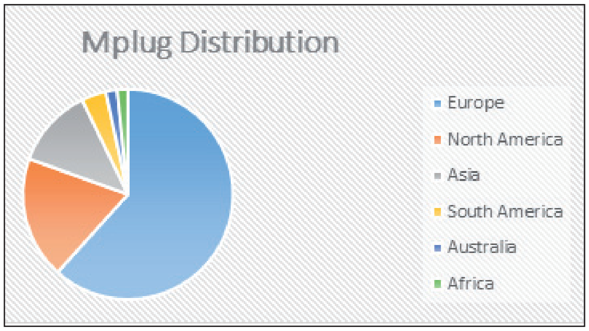
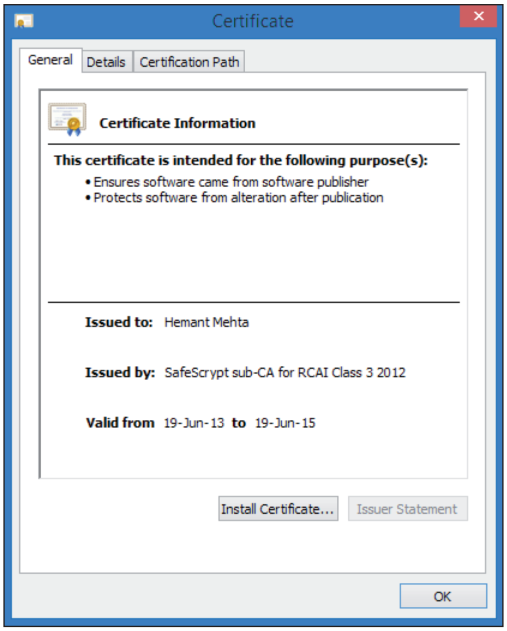


Figure 4: Distribution of MPlug adware.

5. OBSERVATIONS AND CASES OF USED VULNERABILITIES

In our research, we encountered some interesting cases and carried out a more in-depth analysis.

a) In a large number of cases, we observed that a lot of details about a certain company/person from a company can be found through chosen Internet searches. For example, we encountered malware files that had valid signatures issued for 'Hemant Mehta' from the organization 'Brass Copper And Alloy India Ltd'. The certificate was issued by 'Sify Technologies Limited' through 'SafeScryptsub-CA for RCAI Class 3 2012'. The certificate was still valid in June 2014, even though we found malware files signed with it starting in February 2014.



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures software came from software publisher
- Protects software from alteration after publication

Issued to: Hemant Mehta

Issued by: SafeScrypt sub-CA for RCAI Class 3 2012

Valid from 19-Jun-13 to 19-Jun-15

Figure 5: Malware using certificate issued to Hemant Mehta.

We easily discovered that Mr Hemant Mehta has a managerial function and a public email address on the company's web page (h******@hex*****.com) and his telephone number is listed on another website. We noticed that another email address was used in the certificate instead of the one stated earlier. This could be an indication that a possible attacker used a compromised email address from hex*****.com and the name and information of the general manager. We consider that this sort of information regarding a company's management should not be able to be discovered so easily.

b) Some of the abused digital certificates are used only in one type of malware, but this is true only for a small portion of them. We have found the same certificates used for two, three or even more than three types of malware. We believe that this could be an indication that:

1. The person/group that has stolen or socially engineered that certificate is the author of more than one type or malware.

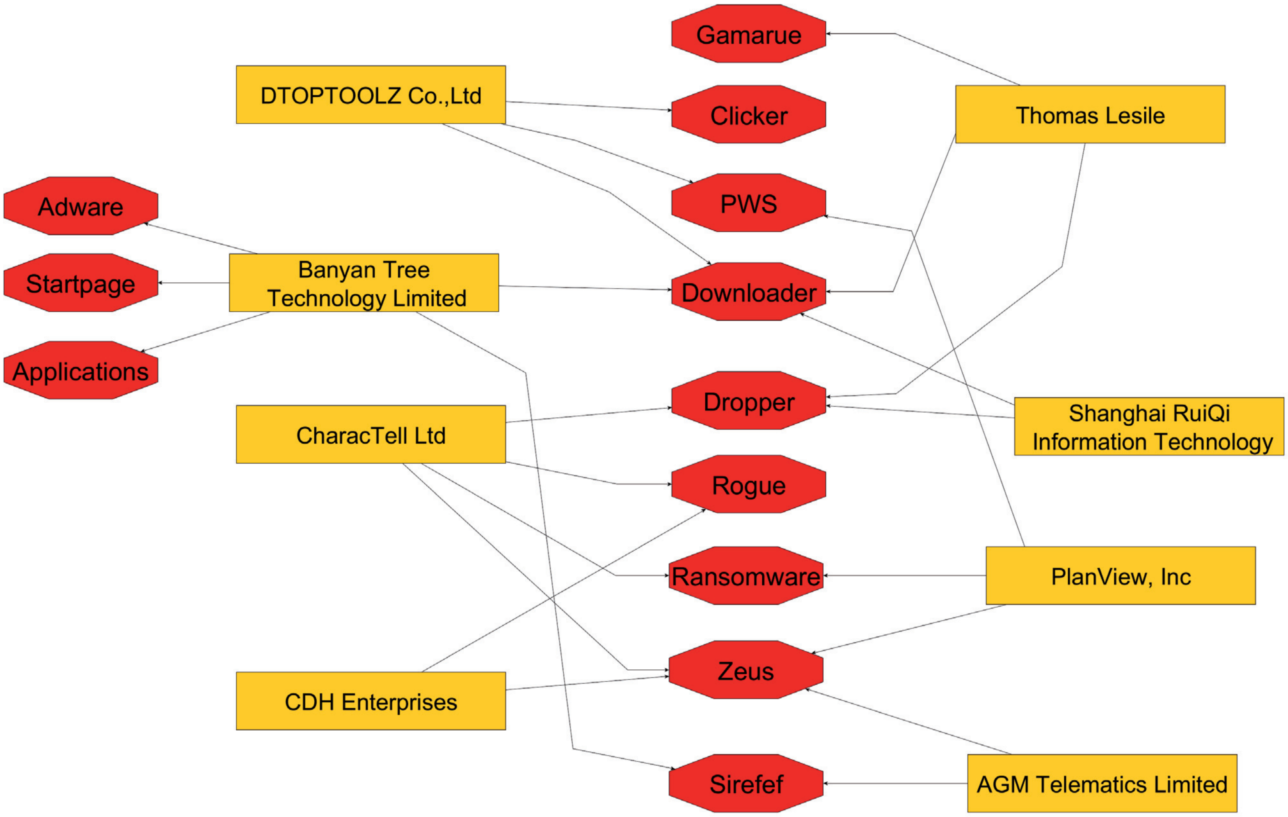


Figure 6: Malware used certificates in connection with malware families.

2. The person/group that has stolen or socially engineered that certificate sold it to more than one malware author.

Figure 6 shows an example of the second situation, which we believe is more likely. As you can see, several types of malware used one or more digital certificates. This could mean that there is a black market for these kinds of goods – a virtual place where you can buy digital certificates.

c) Very similar names were used to issue three certificates by three different CAs: 'John W. Richard' from *Comodo Group*, 'William Richard John' from *StartCom*, and 'JOHN WILLIAM RICHARD' from *Thawte*. All of them have now been revoked. The certificates were issued in consecutive months and were used to sign very similar files. From the certificate information, we can see that all three are from the US, (Eugene, OR), and one of them has a description that uses a large random word. When searching for the entire name we find that it is associated with medical doctors, high status persons from the UK (e.g. Lords, Barons, Earls) and military officers. Considering this, it is possible that this was a result of exploiting vulnerabilities in automated verifications of the CAs.

d) In our research we noticed that the largest number of stolen certificates used for signing malware files were issued by *VeriSign* (more than 37%). This is probably the result of the methodology of the signing process after the certificate has been issued.

Also, it is possible that a larger number of attacks on the issuing methodology are made against *VeriSign* because of the wide use of the issued certificates.

e) At the time of writing this paper, 24% of the certificates we discovered as being used by malware have not been revoked by the CA. Malicious activities regarding some of them were first encountered over six months ago. In one of these cases, the certificate was issued in November 2012 with validity until February 2015. The first malware file signed with it was encountered on 25 October 2013, and at that moment the file was detected by four anti-virus products. After a week, the file was detected by more than 10 anti-virus products, and in June 2014 by 34 of them. Although there is a very high detection rate for this file, the certificate has not been revoked or marked as used by malware in any way.

6. RELATED WORK

Previous articles regarding digital certificates and their use for malicious purposes have focused on signed malware executables [5], or on web threats and software configuration threats [6]. Also, the vulnerability of CAs that use MD5 hashing was discussed at Black Hat 2009 [7].

CONCLUSIONS

Based on the data presented in this paper, we ask ourselves the following questions:

- Is the CA periodically checking the signed files and the activity of that certified company or individual developer?
- If an old version of a legitimate digitally signed application that has a known vulnerability is exploited by a malware author, shouldn't it be revoked directly by *Windows*? In this case, the activity of the Snake malware [8] could have been prevented (a vulnerable VirtualBox driver was used).
- The price for a new certificate after revocation due to losing exclusive access should increase significantly as the number of stolen certificates grows. This could make the buyers increase the security of the system responsible for keeping their certificates safe. However, if this happens and the certificate belonging to a small company or an individual developer is compromised, will they be able to afford to renew the certificate and revoke the old one, or will they be forced to go to a cheaper CA?
- In a world where digital certificates are stolen and used to sign malware, shouldn't there be a special type of revocation, so that the operating system can directly block an execution and alert the user to the fact that the serial number of the certificate was used to sign files with malicious intent?

REFERENCES

[1] Microsoft Security Bulletin MS01-017. http://technet.microsoft.com/en-us/library/security/ms01-017.

[2] Comodo Code Signing Certificate Subscriber Agreement. http://www.comodo.com/repository/docs/code-signing-subscriber-agreement.pdf.

[3] Wang, X.; Yu, H. How to Break MD5 and Other Hash Functions. CRYPTO, 2005.

[4] Win, Y. L.; Wang, X.; Yu, H. Finding Collisions in the Full SHA-1.

[5] Niemela, J. Its Signed, therefore its Clean, right? CARO, 2010.

[6] Wood, M. 'Want my autograph?':The use and abuse of digital signatures by malware. Proceedings of the Virus Bulletin International Conference 2010.

[7] Moxie Marlinspike. Breaking SSL With Null Characters. Black Hat 2009.

[8] BAE Systems Applied Intelligence. Snake Campaign and Cyber Espionage Toolkit. http://info.baesystemsdetica.com/rs/baesystems/images/snake whitepaper.pdf.