'WANT MY AUTOGRAPH?': THE USE AND ABUSE OF DIGITAL SIGNATURES BY MALWARE

Mike Wood
Sophos Inc., 580 Granville St., Vancouver BC, Canada

Email mike.wood@sophos.com

ABSTRACT

Encryption has always been a part of malware, from basic ROT13 string encoding to multi-layered packing algorithms. However, malware authors have discovered ways to exploit the existing strengths and weaknesses of public key cryptography in addition to their home-grown crypto. With the many layers that make up the Public Key Infrastructure (PKI) – certificate issuance, verification, revocation and all of the protocols and software that go in between – scammers have several weaknesses at their fingertips to abuse the overall system. Cheap SSL certificates with automated issuance procedures facilitate the fast and anonymous set-up of rogue e-commerce sites. Moreover, malware authors are able to pass their trojans off as binaries from a legitimate source, using valid or invalid signatures, as most users simply click through the related security warnings. Making matters worse, much of the endpoint software consuming digitally signed content has its own weaknesses, including off-by-default certificate revocation checking mechanisms. In addition to abuse, malware authors are also exploiting the strengths of public key cryptography for uses including secure botnet command and control. This paper discusses these abuses of digital signatures and possible approaches to turn the criminals' investment in their fraudulent reputation into additional protection mechanisms.

INTRODUCTION

The principle of a signature serves a very useful purpose – it is a definitive mark – a guarantee that an object has been created, approved or validated by some entity. Digital signatures by design serve the same purpose – to authoritatively bind an identity to some data, be it a web address, executable software or otherwise. While these signatures fundamentally rely on public key cryptography mathematics, there are many file formats, network protocols, software components and business entities which make up the PKI to support the consumption of digital signatures. As with any complex multi-layered system, individual components as well as interactions between components can have weaknesses – which criminals are happy to exploit.

Firstly, malware is exploiting the presence of digital signatures as a social-engineering tactic to defraud the end-user. Both SSL certificates and *Microsoft Authenticode* code-signing signatures are being abused. The automated identity vetting used when issuing free trial or low-cost domain-validated SSL certificates eases the set-up of a rogue e-commerce site with a certificate trusted by most major browsers. Many malware binaries are being signed using rogue certificates – either with test signatures from *Microsoft* SDK tools or rogue self-signed certificates masquerading as certificates from a legitimate corporation. Some malware authors are even obtaining code-signing certificates from legitimate certificate authorities in the name of legally registered corporations.

The strengths of public key cryptography itself are also being abused to prevent the hijacking of botnet command and control. While researchers have been able to successfully infiltrate certain botnets for short periods of time [1, 2], the Conficker botnet continues to stave off a benevolent takeover by security researchers through its use of a custom public-key-based authentication protocol used for payload downloads and C&C rendezvous.

In the end however, as malware authors turn to abusing the strengths and weaknesses of public key cryptography, so too can their definitive mark be turned against them to derive better defences.

ABUSE OF SOFTWARE DIGITAL SIGNATURES

This section discusses the abuse of digital signatures on software, either as a means to bolster the reputation of fraudulent software publishers or to implement strong security protections in malware.

Abuse of Microsoft Authenticode

Microsoft is placing heavier emphasis on code signing. For *Windows Vista x64*, *Windows 7* and onward, the operating system will require all kernel-mode software (i.e. drivers) to have a valid digital signature [3]. *Authenticode* is the *Microsoft* standard format used by *Windows* systems to authenticate Portable Executable (PE) files.

Authenticode background

The purpose of *Authenticode* is twofold: to verify software came from a particular software publisher, and to verify the software has not been tampered with. *Microsoft* provides detailed documentation on the *Authenticode* signature formats and authentication procedures [4]. The following is a very brief outline to provide the necessary context for the discussion of abuses that follows.

The software publisher's *Authenticode* signature of a PE file contains three critical components: a one-way hash (digest) of the software, the software publisher's encryption of the digest using their private key, and the software publisher's x509 certificate containing the public key needed to verify the signature on the file. The signature may optionally contain a signed timestamp indicating when the software was signed.

A software publisher may obtain a code-signing certificate by purchasing one from a Certificate Authority (CA). In this case, the publisher's x509 certificate additionally contains a signature by the CA on the certificate itself to bind the identity (i.e. the Distinguished Name) of the publisher to the embedded public key. This establishes a certificate chain from the CA to the publisher. Code-signing certificates can also be generated using many freely available software tools, including *Microsoft* SDK's MakeCert.exe and the *OpenSSL* suite.

To have a valid *Authenticode* signature, the signature must have a certificate chain which terminates with a certificate installed as a trusted root authority. Several CA certificates are installed by default on *Windows* to bootstrap the chain of trust. Certificates generated manually using *MakeCert* or *OpenSSL* tools will fail to verify unless the root certificate is installed manually (e.g. *MakeCert* generates a certificate signed by a 'Root Agency' CA which is not trusted on *Windows* by default). In addition, a valid *Authenticode* signature requires that either the software publisher's certificate is not expired or the signature contains a signed timestamp within the validity period of the signing certificate, which is signed using a certificate chain which also terminates in a trusted certificate authority.

Authenticode abuse

Initially, it is interesting to examine the use of digital signatures in malware over time. The following plots highlight the use of *Authenticode* signatures from the start of 2008 up to April 2010 for analyst-classified malware samples at *Sophos* (i.e. PE files which have been manually examined and classified as malicious by a virus analyst). While this approach certainly does not encompass all digital signature use across all malware, it highlights a variety of abuses and provides direction for further exploration.

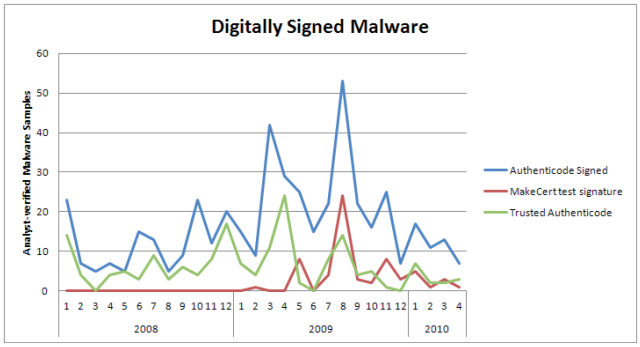

Figure 1: Distribution of valid and invalid Authenticode signatures on analyst-classified malware.

Figure 1 shows the distribution of valid and invalid *Authenticode* signatures on analyst-classified malware. The 'Authenticode Signed' line represents all samples which contain an *Authenticode* signature, valid or invalid, while the 'Trusted Authenticode' line represents only samples for which the default *Authenticode* policy returned success on a *Windows XP* machine (i.e. signtool /pa). In addition, the 'MakeCert Test Signature' line represents those samples signed by a certificate produced with the *MS* SDK MakeCert.exe tool.

The gap between 'Authenticode Signed' and 'Trusted Authenticode' increases as time moves forward from 2008, suggesting malware authors as a whole may be purchasing (or getting access to) fewer code-signing certificates from legitimate CAs. Moreover, the 'MakeCert Test Signature' line increased dramatically over time, going from no use in 2008 to a large spike mid 2009, tapering off in late 2009 but with new samples still in early 2010. Recall that signatures generated by such test certificates do not successfully verify on *Windows* by default. These two trends together could mean malware developers will spend more time developing runtime armour to disable certificate security checks, which are discussed later.

Figure 2 highlights the distribution in the types of errors seen for the malware samples which had a digital signature but which failed to verify against the default *Authenticode* policy.

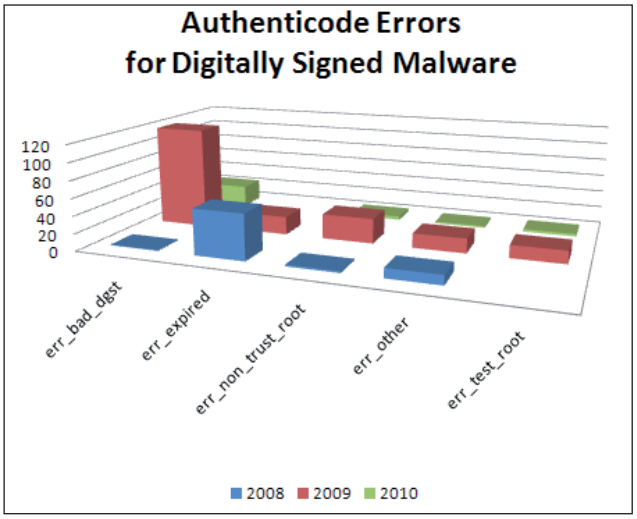

Figure 2: Authenticode errors seen in digitally signed malware.

The first noticeable spike occurs in 2008 for 'err_expired' which indicates that the *Authenticode* signature expired. The samples are all 'dialler trojans' – programs that typically dial a premium-rate phone line, normally with the intent of gaining access to pornographic material. The *Authenticode* signatures fail to verify since the signing certificate is beyond its validity period and the signature was made without a timestamp.

The next major spike occurs in 2009 for 'err_bad_dgst' which indicates that the hash value embedded in the *Authenticode* signature fails to match the one dynamically computed during verification. This suggests the signature has been tampered with or that a valid *Authenticode* signature has simply been copied from another file. This is common amongst the family of PcClient malware, which typically contain an invalid signature purporting to come from *Microsoft*. Typos in the issuer field, such as 'Microsoft Root Authorit' instead of the expected 'Microsoft Root Authority', strongly hint at such manual tampering.

The 'err_non_trust_root' represents signatures for which the cryptographic hash calculation and public key verified, but whose certificate chain terminated in a non-trusted certificate – not including those generated by the *MakeCert* utility. While these samples could be signed using a legitimate certificate from a CA not pre-installed on *Windows*, it is more likely the case these signatures are that of a rogue CA, generated by the malware authors themselves. This behaviour is common to the AlvaBrig or MultiBanker family of trojans, which masquerade as 'Symantec Corporation' software. Note: these signatures may include a timestamp from a legitimate CA, such as *Comodo Time Stamping Signer*, which is a freely available online service.

Figure 3 shows the distribution of samples for which the *Authenticode* signature verified by a loose categorization of

malware class – meaning the signature computation was correct and the certificate chain terminated with a trusted pre-installed CA certificate.

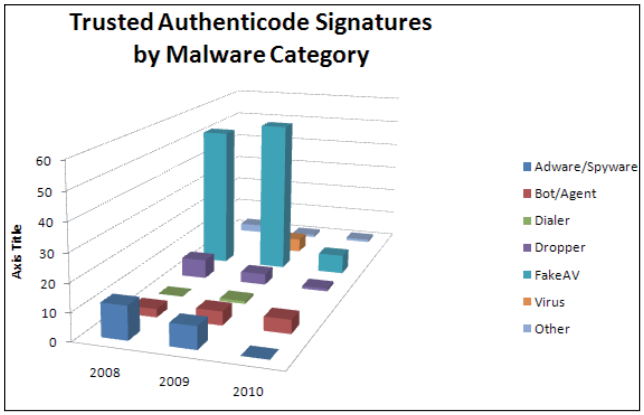

Figure 3: Trusted Authenticode signatures by malware category.

Figure 3 demonstrates how rogue security software (fake AV) dominates the malware signed with code-signing certificates from legitimate CAs. There are several different 'vendors' of said fake AV software, including 'AntiSpyware LLC', 'AntiSpywareSolutionPro Inc.', 'digiweb corporation', 'Pc Utility Inc.' and 'Fast Click Corp.' to name a few. Looking deeper at the specific case of Fast Click Corporation, the certificate is registered to 'Fast Click Corp.' and is valid from 13 July 2009 for one year. However, the current WHOIS data for the fast-click-corporation.com domain, which is listed as an alternative name in the certificate, shows the domain was created on 4 February 2010 – more than six months after the certificate was issued. Trawling earlier WHOIS data shows said domain was originally registered on 5 June 2009. The first confirmed malware sample bearing this signature was seen on 22 October 2009. The Fast Click Corp. certificate was issued by The USERTRUST Network on 13 July 2009 and was revoked on 24 November 2009. Notably, in order to pass the business validation check for the code certificate issuance process, Fast Click Corp. is registered as an offshore corporation in Panama by a law firm specializing in helping clients 'take advantage of Panama's flexible corporation law' [5]. Other CAs have been abused as well, including *Thawte* for AntiSpywareSolutionPro Inc. (incorporated in Belize) which was issued on 25 April 2008 and revoked on 6 January 2009.

Spyware and adware also contribute significantly to the totals for signed malware. These software distributors may have longer-lived certificates than their fake AV counterparts, though the line between malicious and potentially unwanted applications in the rogue security wares arena is more blurred than ever. On the one hand, you have spyware/adware organizations such as 'Favorit Network' apparently using an affiliate software-bundling program. Under this guise their software is often bundled with that of innocuous online games, the organization's network space AS48445 (FAVN) has been known to host malicious content. Favorit Network has a code-signing certificate from *Thawte* which is valid for two years from 11 February 2009 to 11 February 2011 and is not (presently) revoked.

Non-malicious online gambling software can sometimes be seen distributed or linked via spam, as in the case of 'Skill on Net ltd'. In this example, the software publisher also uses an affiliate program, paying a share of the profits to an affiliate ID embedded within the signed game software itself. Skill on Net has a *Thawte* certificate valid from 18 February 2009 to 18 February 2011 and is not (presently) revoked.

It is interesting to notice the virus category spike in the autumn of 2009. One would not expect a virus author to include code signing as part of its file infection mechanism, particularly as that would require the private key to be embedded in the viral code! Looking into the samples, they are all W32/Induc-A infections, which infect a library module (SysConst.dcu) in the Delphi Development environment that subsequently infects every executable it compiles.

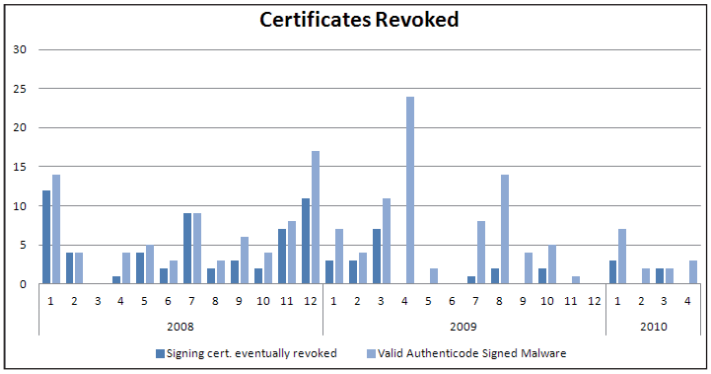

Figure 4: Certificates revoked.

Lastly, Figure 4 pairs the number of malware samples with a valid *Authenticode* signature with the number of samples with a valid *Authenticode* signature whose certificate was *eventually* revoked. Notice the majority of signed malware from 2008 has the signing certificate revoked, whereas the percentage steadily decreases into 2009 and 2010. This demonstrates perhaps the most troubling trend regarding *Authenticode* signatures – that certificates abused to sign malware are now *less* likely to be revoked than before.

Malicious Authenticode configuration

Several recent downloader and fake AV trojans have taken to abusing *Authenticode* in the reverse manner by disabling the certificate checks that are typically enabled by default. These settings disable the security dialogs for executables downloaded via *Internet Explorer*.

The malicious configuration consists of four registry entries. The first two include:

HKCU\Software\Microsoft\Internet Explorer\Download
CheckExeSignatures
no

HKCU\Software\Microsoft\Internet Explorer\Download
RunInvalidSignatures
0x00000001

which disables the publisher dialog upon download and allows potentially expired or revoked applications to proceed without warning. The second two include:

HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Associations
LowRiskFileTypes
.exe

HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Attachments
SaveZoneInformation
0x1

which marks executable files as a low risk and strips their zone information, so that EXE files downloaded from the Internet are treated with the same risk level as a locally stored text file.

There are additional certificate-handling registry entries that can be set as part of a group policy, found under \SOFTWARE\Policies\Microsoft\SystemCertificates, though malware abusing such registry settings did not surface during this investigation. As for the kernel-mode code signing for *Vista x64* and later versions of *Windows*, there are mechanisms for developers to load test versions of their drivers, which may in future be exploited by malware.

Botnet command and control

The Conficker worm is the only botnet to date to use public key cryptography as a means to securely deliver command and control (C&C) actions as well as to update to newer versions. Researchers have been successful at hijacking other botnet operations, such as Torpig/Mebroot [1] and Waledac [2]. These hijacks were possible as both botnets rely on obfuscated C&C rather than authenticated C&C – Torpig using simple XOR encryption and Waledac using multiple encoding layers, both over plain HTTP. Conficker remains immune to such a takeover, as C&C actions as well as binary updates are checked for a signature by the malware author's private key before they are executed – if the signature check fails, the command is simply ignored.

Although the signature check is one of Conficker's strengths, it can also be a weakness. Just as the malware itself can authenticate a C&C command or check that a binary update has been signed by the malware author, so too can systems tracking botnet activity. Knowing the non-standard protocol for Conficker's digital signature check [6] allows an automated system to classify a payload as malicious with complete certainty (as long as the classification includes the signature data).

WEB-BASED PKI ABUSE

This section discusses web threats exploiting digital signatures.

Rogue software payment sites

Rogue security software, aka scareware or fake AV, is one of the largest and fastest growing threats on the Internet. A victim is first duped with phony warnings of a trojan or virus infection on their computer, either via malicious software or directly from web pages designed to look like the user's desktop. Unwitting users can end up at one of these pages by visiting a compromised website or malicious redirection stemming from search engine optimization poisoning [7]. In the end, the phony warnings are all part of a ruse to extort money from the victim, conning them into paying to clean up the bogus infection.

As with any con or social engineering attack, the details are what count. Web page warnings are carefully crafted to reflect near carbon-copies of *Windows Explorer*. Rogue software is decaled with multiple dialogs, progress bars, flashy images and detection meters, though these details are all simply to get the victim to the payment site. To instil a greater confidence in the victim, many rogue payment processing websites are using HTTPS fully equipped with a certificate from a CA.

Genuine SSL certificates were used by a rogue payment site since as early as 2008 [8], where the makers of rogue xp-antivirus acquired an *Equifax* certificate – though the site was taken down within a short period of time. Unlike the landing pages or compromised site redirectors which can have uptimes as short as a matter of hours [9], payment processing sites tend to have a much longer lifespan – with some sites registered in mid 2009 still remaining active today. The sites are typically hosted with known 'dirty networks' to ensure their longevity. Ensuring this longevity is especially important, as many payment sites support multiple brands of fake AV and affiliate programs to drive traffic to their payment site [10].

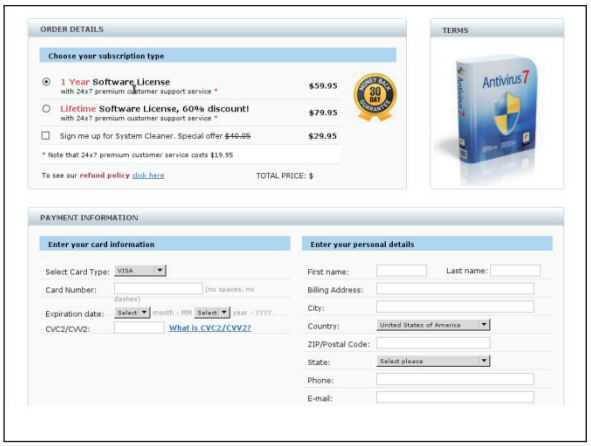

Figure 5: 'Certified Secure Payments'.

Figure 6: 'Secure Billing System'.

Though it is certainly clear the payment site webmasters themselves are behind some of the maliciously generated traffic, as a number of redirections target the payment site with an empty affiliate ID parameter.

Harvesting the payment sites can pose a challenge. Landing pages may use HTTP or JavaScript redirection and the payment sites may check HTTP headers for a referrer-known affiliate domain. However, examining the operation for a small cross-section of payment sites highlights a number of patterns and idiosyncrasies of SSL certificate use.

The group behind the 'Certified Secure Payments' or 'Fast Easy Payments' service (Figure 5) exclusively use *Equifax RapidSSL* certificates with a validity period of one year. Over the observation period, the service had at least nine domains hosted on two separate known dirty networks ATECH-SAGADE (91.188.59.0/24) and Eventis Host (195.5.161.0/24). Brands of fake AV sold include 'Antivirus Plus' and 'Antivirus 7'.

The group behind the 'Secure Billing System' (Figure 6) also exclusively use *GeoTrust RapidSSL* certificates with a one-year validity period. At least 11 domains are associated with the service, with each presently active site initially serving a correct Domain Validated certificate. However, at least two instances of certificate re-use were observed for domains comprising this payment service. For example, the pcsbilling.com certificate was issued on 7 December 2009 and revoked on 11 March 2010, though the site remains active and serves the SSL certificate for softpayb.com. Brands of fake AV sold include 'PC Antivirus' and 'SW Protector'.

On the contrary, the 'LSS Payment Inc.' service (Figure 7) was using *Thawte SSL* 123 certificates in late 2009 (e.g. thesecurebill.com) – again all domain-control certificates, one per domain, each valid for one year. However, recent sites for the same payment service, which sells fake AV brands including 'Internet Antivirus Pro' and 'Ghost Antivirus', are abusing *Comodo Free SSL* certificates, valid only for three months.

Other rogue payment sites are set up to use cheap SSL certificates and advertise non-fake AV related products as a front (i.e. no hidden HTTP link structure required). The 'Alyarica Ltd' and 'Green Cart' sites directly advertise software such as 'MyBackupMaster' and 'Eco Green Computer', though the

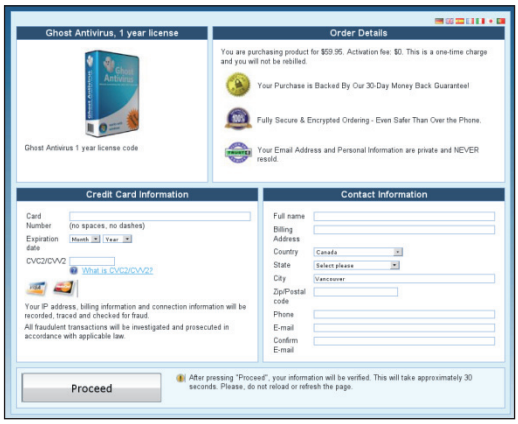

Figure 7: 'LSS Payment Inc.'

services are hosted alongside known fake AV (netname: GIBIBITS-LTD-966647 and ExpertsChallenge C IWEB-CL-T151-360CL-452). Alyarica uses two domains sharing a single *Thawte* SSL 123 certificate (valid only for one) while Green Cart uses a *Comodo PositiveSSL* certificate – both valid for only one year.

Table 1 highlights the distribution of SSL certificates used in such fraudulent payment sites across CAs. The 'Certificates' column represents the number of unique certificates issued, while the 'Domains' column represents the number of unique domains seen to use said certificates. The difference between these two columns indicates the amount of certificate re-use. Notably, only three of the 25 certificates were revoked by the CA at the time of writing.

Certificate authorities	Certificates	Domains	Certs. revoked
Comodo CA Limited	3	3	
Equifax	14	17	3
GLOBE HOSTING CERTIFICATION AUTHORITY	1	1	
GoDaddy.com, Inc.	1	2	
Thawte Consulting (Pty) Ltd.	1	1	
Thawte Consulting cc	5	8	
Total	**25**	**32**	**3**

Table 1: Distribution of SSL certificates used in fraudulent payment sites across CAs.

Interestingly, there does not appear to be a mark-up on the price for rogue AV software sold via SSL-enabled payment sites. The investigation turned up several SSL and non-SSL enabled payment services though the phony licence prices all appeared to fall roughly within the $50–$90 range.

Phishing with certificates as bait

Digital certificates have also been used as the bait in phishing attacks, primarily those that target online banking credentials. In a notable targeted attack in early 2009, the fraudsters were able to trick an *Experi-Metal Inc*. employee to provide their *Comercia* banking details, including all two-factor authentication data, to a rogue site [11]. The phishing attack posed as a certificate update from *Comercia*, which the bank had indeed been sending to customers on a yearly basis. The Zeus (aka Zbot) malware is also commonly distributed via spam messages with one notable campaign from October 2009 posing as a digital certificate upgrade, using the subject 'Please install digital certificate software' to dupe recipients into opening the attachment.

CERTIFICATE HANDLING FLAWS

This section explores the flaws with certificate handling in some of the legitimate components that make up the PKI, including

certificate authority practices, web browsers and their combined interactions.

Certificate authorities

All of the abused SSL certificates were Domain Validated which implies they were issued through an automated identity vetting procedure. The certificate requestor is authenticated by an automated system simply by providing either a phone number or email address. The purchase process is even advertised by the CA as being issued in 'Minutes: Turbo-Fast!' – ideal for the fraudster to set up their secure site quickly and anonymously.

Domain-validated certificate	*Certificate cost*	*Identity vetting procedure*	*Supported revocation protocols*
Thawte SSL 123	$149, 1yr	Automated	OCSP and CRL
Equifax RapidSSL	$69, 1yr	Automated	CRL
Globe Hosting SSL Standard	$16, 1yr	Automated	OCSP and CRL
Comodo FreeSSL	Free, 1–3 months	Automated	OCSP and CRL

Table 2: Domain-validated certificates.

Some certificates carry warranties with a maximum payout of up to $10,000 and are payable to site visitors who incur a financial loss via an online credit card transaction while using the site. However, the CA is only held liable in the event the CA failed to follow their own policies for identity vetting as documented in their Certificate Practice Statement.

While code-signing certificates do require additional authentication, the two-step process is also vulnerable to abuse. The CA first verifies that the business is legally registered in its country of operation, and then makes a verification phone call to the certificate requestor by looking up the business's phone number via an online phone directory. While more costly for the fraudsters, this process is vulnerable to abuse by offshore business registration.

CA subscriber agreements do at least contain provisions to revoke certificates being used for malicious purposes. However, the fraud departments of the two most abused CAs discussed provided no response to emails reporting this certificate abuse, and at the time of writing the certificates reported remain valid (i.e. not revoked).

Web browser certificate handling

Even if certificates are revoked, some of the most common software with which users will encounter certificates is less likely to be configured to properly warn the user about revoked certificates. This section looks at the certificate handling of the two most popular web browsers – *Internet Explorer* and *Firefox* – at roughly 60% and 25% of market share [12]. Only the latest versions of each browser are considered to reflect the most up-to-date scenarios – this means *Internet Explorer 8.0.6* and *Firefox 3.6.3*.

Both browsers support the two certificate revocation methods: Certificate Revocation Lists (CRL) and Online Certificate Status Protocol (OCSP). However, each browser's default settings can result in different scenarios for the malicious certificate abuses observed.

Internet Explorer certificate options are found in the 'Tools > Internet Options > Advanced' window under the heading 'Security'. There are a number of checkboxes which specify the certificate handling preferences:

- 'Check for server certificate revocation' – off by default.

  When enabled, *IE8* will check the CRL and/or OCSP provider embedded in a server's SSL certificate, depending on which revocation methods are supported by the certificate. This is not enabled by default.

- 'Warn about certificate address mis-match' – on by default

  When enabled, *IE8* will present a warning to the user if the domain in the certificate does not match the one visited, e.g. http://example.com vs. http://www.example.com. This is enabled by default.

- 'Check for publisher's certificate revocation' – on by default

  When enabled, *IE8* will check the CRL and/or OCSP provider embedded in a software publisher's code-signing certificate, depending on which revocation methods are supported by the certificate. This is enabled by default.

- 'Check for signatures on downloaded programs' – on by default.

- 'Allow software to run or install even if the signature is invalid' – off by default.

  The last two options correspond to the same CheckExeSignatures and RunInvalidSignatures registry entries discussed earlier, which malicious downloader trojans are disabling to streamline further malware downloads.

Firefox SSL certificate options are found in the 'Edit > Preferences > Advanced > Encryption' menu tab. The 'Verification' dialog provides settings for OCSP providers, which can be enabled to check only the certificate's embedded OCSP URI (if present) or to use a designated OCSP server for all certificates. There is an additional option to treat a certificate as invalid if the OCSP server connection fails, which is off by default. Checking the certificate's embedded OCSP URI is set as the default. The 'Revocation Lists' dialog shows which CRLs have been imported into *Firefox* and facilitates adding new CRLs to the list. When a CRL is added, it can be configured to update automatically (the default) or be updated manually through the dialog. The CRL list is empty by default and does not auto-populate as HTTPS sites are visited, though an imported CRL is configured to auto-update by default.

In summary, neither *IE8* nor *Firefox* certificate options are set to safe defaults on a fresh installation. Both browsers fall short of supporting SSL certificate revocation – *IE8* being off by default

and *Firefox* only having OCSP enabled, which is ineffective for many of the cheaper Domain Validated SSL certificates that do not contain an OCSP extension. In particular, note the most abused CA for rogue payment sites is *Equifax RapidSSL* which does not support OCSP. Perhaps the fraudsters behind those services have recognized the gap in the delivery of certificate revocation information to end-users – without OCSP and automatic download of CRLs (the default for both browsers) even when a CA revokes the certificate, users will not be warned.

LESSONS LEARNED

This section highlights some of the lessons we have already learned, but particularly in a PKI abuse context.

Safe defaults

The need for safe defaults is well demonstrated in a recent case of *Symbian* mobile phone malware. Software cannot be installed on recent versions of *Symbian* phone operating systems unless it contains a valid digital signature. But having made it through *Symbian*'s Express signing process, the MerogoSMS worm could be installed directly onto phones and continue to spread via text messages [13]. Although the publisher's certificate was revoked less than five days after the malware was reported, the *Symbian* phones are not configured to check for certificate revocation by default. As such, the malware remains unfettered and so can continue spreading to phones that have not been otherwise locked down by their owner.

Users are their own worst enemy

With safe default configurations in place, users can be presented with more meaningful warnings regarding the invalidity of a certificate. However, that still places a burden on the user to make a decision. Research shows that users are not able to effectively parse URLs [14] and will thus be vulnerable to wildcard certificates spoofing legitimate domains. Furthermore, users seldom act appropriately when presented with security warnings and often immediately accept a digital certificate with little to no scrutiny [15]. As such, malware authors are free to continue abusing invalid certificates or bogus *Authenticode* signatures under the expectation that the mere presence of a signature is enough to gain users' trust.

Bad guys like it cheap and anonymous

Domain-validated SSL certificates have received significant negative attention with regards to the abuse which the automated-issuance process facilitates. The protection gained by adding non-automated identity checking is best exemplified with the simple policy change for .cn TLD domain registration to require a paper application [16]. Figure 8 shows the number of spam messages seen using .cn domains over the three-month span surrounding the policy change date of 14 December 2009, which highlights the significant drop in new spammy domains.

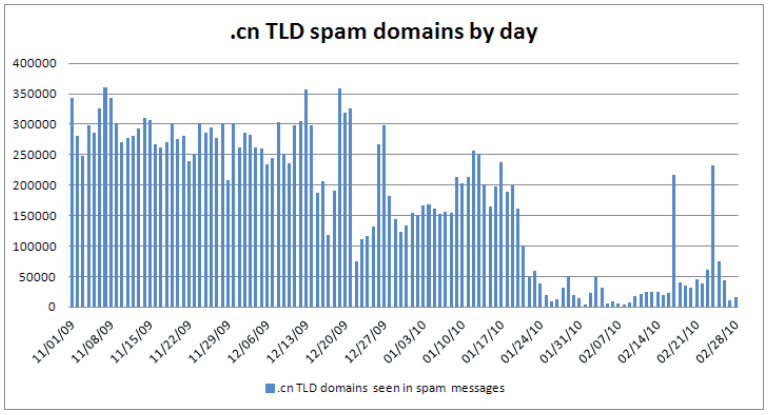

Figure 8: Number of spam messages using .cn domains.

RELATED WORK

The most directly relevant work [17] focuses specifically on signed malware executables, in contrast to this paper which includes web threats and software configuration threats as well. Note that [17] was presented at CARO 2010 a mere week before the submission deadline for this paper, suggesting abuse of digital signatures is indeed an evolving threat.

Some vulnerabilities in certificate-handling software make it possible for fraudsters' malicious domains to masquerade as legitimate ones over SSL. For example, an erroneous subject name parsing routine would allow an attacker to insert a null character into the domain name to spoof virtually any domain (i.e. 'sophos.com\x00mymalwarehost.biz' would be displayed as 'sophos.com') [18]. CAs using MD5 hash digests were vulnerable to attackers using MD5 collisions to generate cryptographically correct signed certificates for arbitrary subjects [19]. Additionally, a network-based man-in-the-middle attack is discussed in [20] which subverts HTTPS sessions using EV certificates. In contrast to these publications which focus on unintended vulnerabilities, this paper focuses on digital signature abuse which makes use of the intended behaviour – and thus inherent weaknesses of the PKI.

CONCLUSIONS

The AV industry is in a strong position to impact the level of security achieved through digital certificates, from both a software implementation and a deployment perspective. In future, CAs may well improve identity vetting processes to pre-empt the issuance of certificates to fraudsters or may drastically reduce lag time for certificate revocation. However, there may always be the risk that a malicious organization will remain innocuous for an initial incubation period. Balancing the trade off between how deep to investigate certificate subjects and the security the investigation provides may not be in the primary interests of the CA – that being selling certificates. Moreover, even if the CA processes improve, the security settings for many of the software components that consume digital certificates are either not enabled, provide warning messages that are not simple to act upon, or run the risk that the settings have been disabled by the malware itself.

In contrast, AV software is in an advantageous position on a number of these aspects. Most AV software is configured to auto-update, so new data published on malicious certificates can be consumed in real time. Heuristic malware detections based on rogue or test certificate signatures could proactively detect unseen samples. Furthermore, AV software must contain logic to thwart hostile code, which can defend against and alert the user to any runtime disablement of certificate checks. The haphazard use of SSL certificates by malicious webmasters can additionally improve automated blocklisting systems: being able to automatically blacklist domains sharing the same known malicious certificate, as well as avoiding HTTP header or anti-emulation JavaScript web content armour since the SSL handshake happens prior to the data exchange.

At its core, AV software is all about reputation management: reputation of software, networks, web content and URLs. Certificate management can serve as a key piece of data to build up or detract from a publisher's reputation and ultimately help manage the user's decision making with such a complex and multi-layered threat.

REFERENCES

[1] Stone-Gross, B. et al. Your Botnet is My Botnet: Analysis of a Botnet Takeover. Proceedings of the 16th ACM conference on Computer and Communications Security, 2009. pp.635–647.

[2] Sinclair, G.; Nunnery, C.; Kang, B.B.H. The Waledac Protocol: The How and Why. MALWARE 2009: The Fourth Annual Conference on Malicious and Unwanted Software, 2009.

[3] Microsoft. Kernel Mode Code Signing Walkthrough. [Online 25 July 2007.] http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/KMCS_Walkthrough.doc.

[4] Windows Authenticode Portable Executable Signature Format. [Online 21 March 2008.] http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/Authenticode_PE.docx.

[5] Tapia, Linares & Alfaro. Offshore Services. Tapia, Linares & Alfaro Attorneys At Law. [Online] [Cited: 25 May 2010.] http://www.talial.com/en/offshore.html.

[6] Porras, P.; Saïdi, H.; Yegneswaran, V. A Foray into Conficker's Logic and Rendezvous Points. USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2009.

[7] Howard, F.; Komili, O. Poisoned search results: How hackers have automated search engine poisoning attacks to distribute malware. Sophos, 2010.

[8] Armin, J. McColo – Cyber Crime USA 2008 Version 2.0. HOSTEXPLOIT.COM. [Online] 2008. [Cited 20 May 2010.] http://hostexploit.com/downloads/Hostexploit%20Cyber%20Crime%20USA%20v%202.0%201108.pdf.

[9] Rajab, M.A. et al. The Nocebo Effect on the Web: An Analysis of Fake Anti-Virus Distribution. 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2010.

[10] Samosseiko, D. The Partnerka: What is it and Why Should You Care? Proceedings of the 19th Virus Bulletin International Conference 2009.

[11] Krebs, B. Krebs On Security. [Online] February 2010. [Cited: 20 May 2010.] http://krebsonsecurity.com/2010/02/comerica-phish-foiled-2-factor-protection/.

[12] Net Applications. Browser Market Share. [Online] May 2010. http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0.

[13] Hypponen, M. Trojan.MergoSMS. F-secure.com. [Online] 22 March 2010. http://www.f-secure.com/weblog/archives/00001912.html.

[14] Sheng, S. et al. Anti-Phishing Phil: The design and evaluation of a game that teaches people not to fall for phish. Proceedings of the 3rd symposium on Usable privacy and security, 2007.

[15] Dhamija, R.; Tygar, J.D.; Hearst, M. Why phishing works. ACM, 2006. Proceedings of the SIGCHI conference on Human Factors in computing systems. p.590.

[16] CNNIC. China Internet Network Information Center. The Notification about further enhancement of auditing domain name registration information. [Online] 11 December 2009. [Cited: 20 May 2010.] http://www.cnnic.net.cn/html/Dir/2009/12/12/5750.htm.

[17] Niemela, J. It's Signed, therefore it's Clean, right? CARO, 2010.

[18] Marlinspike, M. Breaking SSL With Null Characters. Black Hat 2009.

[19] Sotirov, A. et al. MD5 considered harmful today. [Online] 30 December 2008. [Cited: 12 May 2010.] http://www.win.tue.nl/hashclash/rogue-ca/.

[20] Zusman, M.; Sotirov, A. Attacking Extended Validation SSL. Black Hat 2009.

[21] Ragan, S. Criminals using Comodo to attempt legitimacy. [Online] 26 May 2009. [Cited: 13 May 2010.] http://www.thetechherald.com/article.php/200922/3750/Criminals-using-Comodo-to-attempt-legitimacy.

[22] Thawte. THAWTE STARTER PKI PROGRAM AGREEMENT. [Online] [Cited: 20 May 2010.] http://www.thawte.com/assets/documents/repository/agreements/thawte_SPKI_End_User_Subscriber_Agreement.pdf.

[23] Comodo. Comodo Certificate Subscriber Agreement. [Online] [Cited: 20 May 2010.] https://support.comodo.com/index.php?_m=downloads&_a=downloadfile&downloaditemid=69.