The Broken Shield: Measuring Revocation Effectiveness in the Windows Code-Signing PKI

Doowon Kim¹, Bum Jun Kwon¹, Kristián Kozák², Christopher Gates³, and Tudor Dumitraş¹

¹University of Maryland, College Park, ²Masaryk University, ³Symantec
Why is the Code Signing PKI required?

- Nature of software distributed over the Internet
  - Unidentifiable software authors (publishers)
  - May be tampered
Why is the Code Signing PKI required?

- Code signing PKI helps establish ...
  - **Authenticity** of publisher
  - **Integrity** of software
Abuse and Primary Defense

• Abuse cases
  – Stuxnet
  – Black Market
  – Etc.

• Primary defense: Revocation
  – Compromised certificates must be revoked
  – To make them no longer valid

Motivation

- In our prior work, we found that 2/3 compromised certificates are not revoked

- Why are the most not revoked yet?

- Furthermore, do CAs properly understand the code signing PKI and revoke compromised certificates without any mistakes?

We measure the effectiveness of revocations

How to Revoke Potentially Compromised Certificates?

We identify **three steps** required:

1. Promptly discovery compromised certificates
2. Invalidate all signed malware when revoking
3. Disseminate revocation information for clients
Step #1: Discover Compromised Certificates

Security companies

Discover

Compromised

Discover

Compromised

Collaborate

CAs
Step #1: Discover Compromised Certificates

RQ1) How **promptly** do CAs discover and revoke compromised certificates after they appear in the wild?

→ We found **delays of 5.6 months** to revoke compromised certificates
Step #2: Invalidate All Signed Malware

Code Signing PKI

CAs

Revoke

Compromised
Step #2: Invalidate All Signed Malware

- One-to-many relationship
  - A certificate is used to sign numerous samples
  - C.f., TLS, one-to-one relationship

Code Signing PKI

TLS

The Broken Shield: Measuring Revocation Effectiveness in the Windows Code-Signing PKI
Step #2: Invalidate All Signed Malware

Code Signing PKI

RQ2) Do CAs properly revoke them and invalidate all malwares?
Step #2: Invalidate All Signed Malware

RQ2) Do CAs properly revoke them and invalidate all malwares?
→ We found that CAs improperly revoke 5% compromised certificates and 5% signed malware are still valid

→ More critical and challenging than TLS
Step #3: Disseminate Revocation Information

- Always-available for clients
- Must not remove expired certificates in CRLs
Trusted Timestamping

• Trusted creation timestamp of a program

• Extend trust in the program beyond expiration date

Code signing PKI

Sample #1

\[ t_i \]: issue date

\[ t_e \]: expiration date

Valid

now
Trusted Timestamping

- Trusted creation timestamp of a program
- Extend trust in the program beyond expiration date
- Must care about even expired certificates
Step #3: Disseminate Revocation Information

RQ3) Do CAs properly maintain revocation information and disseminate it?
Step #3: Disseminate Revocation Information

RQ3) Do CAs properly maintain revocation information and disseminate it?

- We found that CAs removed 278 certificates from CRLs and improperly maintain infrastructures
- **More critical** and **more challenging** than TLS
Contributions

• We identified the effective revocation process
  1. Discover compromised certificates
  2. Invalidate all signed malware when revoking
  3. Properly disseminate revocation information

• We measured the effective revocation process and showed that revocation in the code signing PKI is more critical and more challenging than TLS
Outline

• Data collection

• Results: Effectiveness of revocation process
  – Discovery of compromised certificates
  – Invalidation of all signed malware
  – Dissemination of revocation information
Data Collection: Challenges

• No large corpus of code singing certificates
  – TLS: Censys.io, IPv4 scanning, Alexa 1M domains, etc

• Unable to know when certificates are revoked
  – Revocation date: The date that determines the validity of signed sample
  – C.f., TLS: The date at which the revocation took place
Data Collection

Data sources
(Symantec, MalSign, WINE...)
Data Collection

Data sources
(Symantec, MalSign, WINE...)

Collect Certificates

Code signing certificates
Data Collection

Data sources
(Symantec, MalSign, WINE...)

Collect Certificates

Code signing certificates

CRLs

OCSP points
Data Collection

Data sources
(Symantec, MalSign, WINE...)

Collect Certificates

Code signing certificates

CRLs

OCSP points

Revocation publication date collection system

Removed Certs. checker

Reachability checker

Response checker

The Broken Shield: Measuring Revocation Effectiveness in the Windows Code-Signing PKI
Outline

• Data collection

• Results: Effectiveness of revocation process
  – Discovery of compromised certificates
  – Invalidation of all signed malware
  – Dissemination of revocation information
Step #1: Discover Compromised Certificates

- Collaborate with security companies to promptly discover compromised certificates

Security companies

- McAfee
- Reversing Labs
- Symantec
- VirusTotal

CAs

Compromised

Discover

Collaborate
Step #1: Discover Compromised Certificates

- Collaborate with security companies to promptly discover compromised certificates
- Promptly start investigations and revoke them
  - Revocation delay should be as short as possible
Revocation Delay: Definition

- Revocation delay: $t_p - t_d$
- $t_d$: the earliest detection dates of signed malware
  - E.g., the earliest submission date of VirusTotal
- $t_p$: the dates when revoked serial numbers are added to CRLs (aka revocation publication date)
Revocation Delay: Result

- Delay \((t_p - t_d)\): from 1 day to 1,553 days (4.25 years)
- Average delay: 171.4 days (5.6 months)
- Compromised certificates not promptly revoked

⇒ Clients remain exposed to this threat for 5 months
Estimation of Compromised Certificates

• Estimate the # of abused certificates in the wild
  – Used the mark-recapture methodology
  – Due to no corpus of code signing certificates to cover all code signing certificates in the wild

\[ N = \frac{n_1 \times n_2}{p} \]

  P: Intersection of two samples
  N1: sample #1
  N2: sample #2

• Population:
  – n1: VirusTotal hunting data set
  – n2: Symantec telemetry data set
Discovery of Compromised Certificates

- Estimated compromised certificates are **2.74X larger** than actually observed

- Even large security companies **cannot cover most of compromised certificates** in the wild
  - A cause of **long revocation delay**
Outline

• Data collection

• Effectiveness of revocation process
  – Discovery of compromised certificates
  – Invalidation of all signed malware
  – Dissemination of revocation information
Role in the Second Step

- CAs should decide the *effective revocation dates* \((t_r)\) to invalidate all malware signed with the compromised certificate
What is the Effective Revocation Dates \((t_r)\)?

- Revocation will be made dependent on a specific date, **effective revocation date** \((t_r)\)

- It determines the validity of signed samples
  - Depending on \(t_r\) signed samples become valid or invalid
What is the Effective Revocation Dates ($t_r$)?

- Revocation will be made dependent on an effective revocation date ($t_r$)

\[
\begin{align*}
\text{Sample #1} & \quad t_i \quad t_{b1} \quad t_{b2} \quad t_e \\
\text{Sample #2} & \quad \text{Valid} \\
\end{align*}
\]

- $t_i$: issue date
- $t_e$: expiration date
What is the Effective Revocation Dates ($t_r$)?

- Revocation will be made dependent on an effective revocation date ($t_r$)

Sample #1 Sample #2

$t_i$ $t_{b1}$ $t_r$ $t_{b2}$ $t_e$ now

$t_i$: issue date
$t_e$: expiration date

Valid
What is the Effective Revocation Dates ($t_r$)?

- Revocation will be made dependent on an effective revocation date ($t_r$)

\[ t_i \] : issue date

\[ t_e \] : expiration date

Sample #1

Sample #2

Valid

Invalid

$now$
What is the Effective Revocation Dates ($t_r$)?
What is the Effective Revocation Dates ($t_r$)?

Effective revocation date
Security Threat

- What if sample signed before $t_r$ are malware?
  - Clients are exposed to the security threat

\[ t_i: \text{issue date} \]
\[ t_e: \text{expiration date} \]
Two Types of Revocation

• Soft revocation: $t_i < t_r < t_e$
  – Invalidate only samples signed after $t_r$
  – But security threats exist

• Hard revocation: $t_r = t_i$
  – No security threats, but invalidate all benign samples
## Trends of Revocation Policy by CAs

<table>
<thead>
<tr>
<th>CA</th>
<th>&lt; $t_i$</th>
<th>= $t_i$</th>
<th>≤ $t_e$</th>
<th>&gt; $t_e$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comodo</td>
<td>0</td>
<td>426</td>
<td>1,437</td>
<td>17</td>
<td>1,880</td>
</tr>
<tr>
<td>Thawte</td>
<td>0</td>
<td>74</td>
<td>1,055</td>
<td>39</td>
<td>1,168</td>
</tr>
<tr>
<td>Go Daddy</td>
<td>2</td>
<td>14</td>
<td>672</td>
<td>18</td>
<td>706</td>
</tr>
<tr>
<td>VeriSign</td>
<td>2</td>
<td>59</td>
<td>430</td>
<td>51</td>
<td>542</td>
</tr>
<tr>
<td>DigiCert</td>
<td>1</td>
<td>161</td>
<td>323</td>
<td>3</td>
<td>488</td>
</tr>
<tr>
<td>Starfield</td>
<td>0</td>
<td>3</td>
<td>153</td>
<td>2</td>
<td>158</td>
</tr>
<tr>
<td>Symantec</td>
<td>0</td>
<td>33</td>
<td>89</td>
<td>1</td>
<td>123</td>
</tr>
<tr>
<td>WoSign</td>
<td>0</td>
<td>57</td>
<td>17</td>
<td>0</td>
<td>74</td>
</tr>
<tr>
<td>StartCom</td>
<td>0</td>
<td>0</td>
<td>47</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>Certum</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>96</td>
<td>117</td>
<td>1</td>
<td>214</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5</td>
<td>924</td>
<td>4,349</td>
<td>132</td>
<td>5,410</td>
</tr>
</tbody>
</table>

- The majority is soft revocation (83%)
### Trends of Revocation Policy by CAs

<table>
<thead>
<tr>
<th></th>
<th>&lt; $t_i$</th>
<th>= $t_i$</th>
<th>$\leq t_e$</th>
<th>&gt; $t_e$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comodo</td>
<td>0</td>
<td>426</td>
<td>1,437</td>
<td>17</td>
<td>1,880</td>
</tr>
<tr>
<td>Thawte</td>
<td>0</td>
<td>74</td>
<td>1,055</td>
<td>39</td>
<td>1,168</td>
</tr>
<tr>
<td>Go Daddy</td>
<td>2</td>
<td>14</td>
<td>672</td>
<td>18</td>
<td>706</td>
</tr>
<tr>
<td>VeriSign</td>
<td>2</td>
<td>59</td>
<td>430</td>
<td>51</td>
<td>542</td>
</tr>
<tr>
<td>DigiCert</td>
<td>1</td>
<td>161</td>
<td>323</td>
<td>3</td>
<td>488</td>
</tr>
<tr>
<td>Starfield</td>
<td>0</td>
<td>3</td>
<td>153</td>
<td>2</td>
<td>158</td>
</tr>
<tr>
<td>Symantec</td>
<td>0</td>
<td>33</td>
<td>89</td>
<td>1</td>
<td>123</td>
</tr>
<tr>
<td>WoSign</td>
<td>0</td>
<td>57</td>
<td>17</td>
<td>0</td>
<td>74</td>
</tr>
<tr>
<td>StartCom</td>
<td>0</td>
<td>0</td>
<td>47</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>Certum</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>96</td>
<td>117</td>
<td>1</td>
<td>214</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5</td>
<td>924</td>
<td>4,349</td>
<td>132</td>
<td>5,410</td>
</tr>
</tbody>
</table>

- The majority is soft revocation (83%)
- 132 (2.5%) certificates are set to after expiration date
  - Ineffective revocation
  - All signed samples still valid
Ineffective Revocation Date Setting

• 1,022 certificates, revoked out of 45,613 certificates

• Soft revocation: 891 (87%) certificates

• Wrong effective revocation date: 45 (5%) certificates
  – 4,716 malware signed with the 45 certificates
  – 250 (5%) signed malware is still valid

→ Clients remain exposed to the security threat
Outline

• Data collection
• Effectiveness of revocation process
  – Discovery of compromised certificates
  – Invalidation of all signed malware
  – Dissemination of revocation information
1. Specify CRLs and OCSP points in certificates
2. Responsible for expired certificates
3. Maintain infrastructure to be always-available for clients
Enforcement in Windows

- *Soft-fail* policy for checking revocation status
  - Windows believes a certificate is valid unless revocation status information is available
#1. Certificates without CRLs and OCSP Points

- 788 certificates (0.5% out 144k): no CRLs and OCSP
  - 86% of them were issued by Thawte before 2003
  - All of them already expired
  - However, if malware is signed with the certificates and trust-timestamped, the malware can be still valid

➤ Clients have no means to check the status
#1. Certificates without CRLs and OCSP Points
#2. Unreachable CRLs and OCSP Server

- 13 CRLs (6% out of 215) are unreachable
  - 5 CRLs: HTTP 404 Not Found error
    - They moved the CRLs file to another place
    - One CRL domain is taken by a domain reseller

- 15 OCSP points
  - Bad hostname, timeout, forbidden, & method not allowed
#2. Unreachable CRLs and OCSP Server

![Certificate Window](image)

- Certificate Policies
- CRL Distribution Points
- Authority Information Access
- Key Usage
- Basic Constraints
- Thumbprint
- Extended Error Information

Fields:

Revocation Status: The revocation process failed.
#2. Unreachable CRLs and OCSP Server

2018 Copyright. All Rights Reserved.

The Sponsored Listings displayed above are served automatically by a third party. Neither the service provider nor the domain owner maintain any relationship with the advertisers. In trademark issues please contact the domain owner directly (contact information can be found in whois).

Privacy Policy
#2. Unreachable CRLs and OCSP Server
#2. Unreachable CRLs and OCSP Server

- 13 CRLs (6% out of 215) are unreachable
  - 5 CRLs: HTTP 404 Not Found error
    - They moved the CRLs file to another place
    - One CRL domain is taken by a domain reseller

- 15 OCSP points
  - Bad hostname, timeout, forbidden, & method not allowed

➔ Programs signed with the certificates can still be valid
  - due to trust timestamping and soft-fail policy
#3-1. Transient Revoked Certificates in CRLs

• Recall: CAs, responsible for even expired certificates

• But, 278 revoked certificates **removed** from 18 CRLs

• Contacted the all CAs
  – A CA started investigations and found the flaw
  – And fixed the flaw thanks to our study and replied ...
    • “Thank you ... we were **removing** certificates from the CRL that had **expired** ... We've modified our system to now exclude Code Signing, which means that **once revoked**, the certificate should **remain** on the CRL **indefinitely**.”

➤ Even CAs **misunderstand** the code signing PKI
#3-2. Inconsistent Responses from CRLs and OCSP

- Responses from CRLs and OCSP should be consistent
  - E.g., if one is found in CRLs, the response from OCSP for the certificate indicates that “revoked”

- 19 certificates have inconsistent responses
  - All certificates were issued by Go Daddy and StartField

➤ CAs improperly maintain OCSP and CRLs servers
Conclusion

• The primary defense against abuse is revocation

• Revocation in code signing PKI is more critical and more challenging than TLS

• Hard to discover compromised certificates & samples

• Erroneously setting effective revocation dates
  – Makes malware valid although the certificate is revoked

• Improper dissemination of revocation information
  – Makes signed malware valid due to the soft-fail policy
Data Release

- Our data sets are available at signedmalware.org
  - CRLs for code signing certificates
  - Revocation publication dates
Thank you!

Doowon Kim
dooowon@cs.umd.edu
http://signedmalware.org