



Defense Guided by Experience

Analyzing the MD5 collision in Flame

Alex Sotirov

Co-Founder and Chief Scientist

Trail of Bits, Inc

Overview of Flame

- Discovered sometime in 2012
- Active since at least 2010
- Complex malware
 - almost 20MB in size
 - multiple components
- Very limited targeted attacks



Iran
189

Israel
Palestine
98

Sudan
32

Syria
30

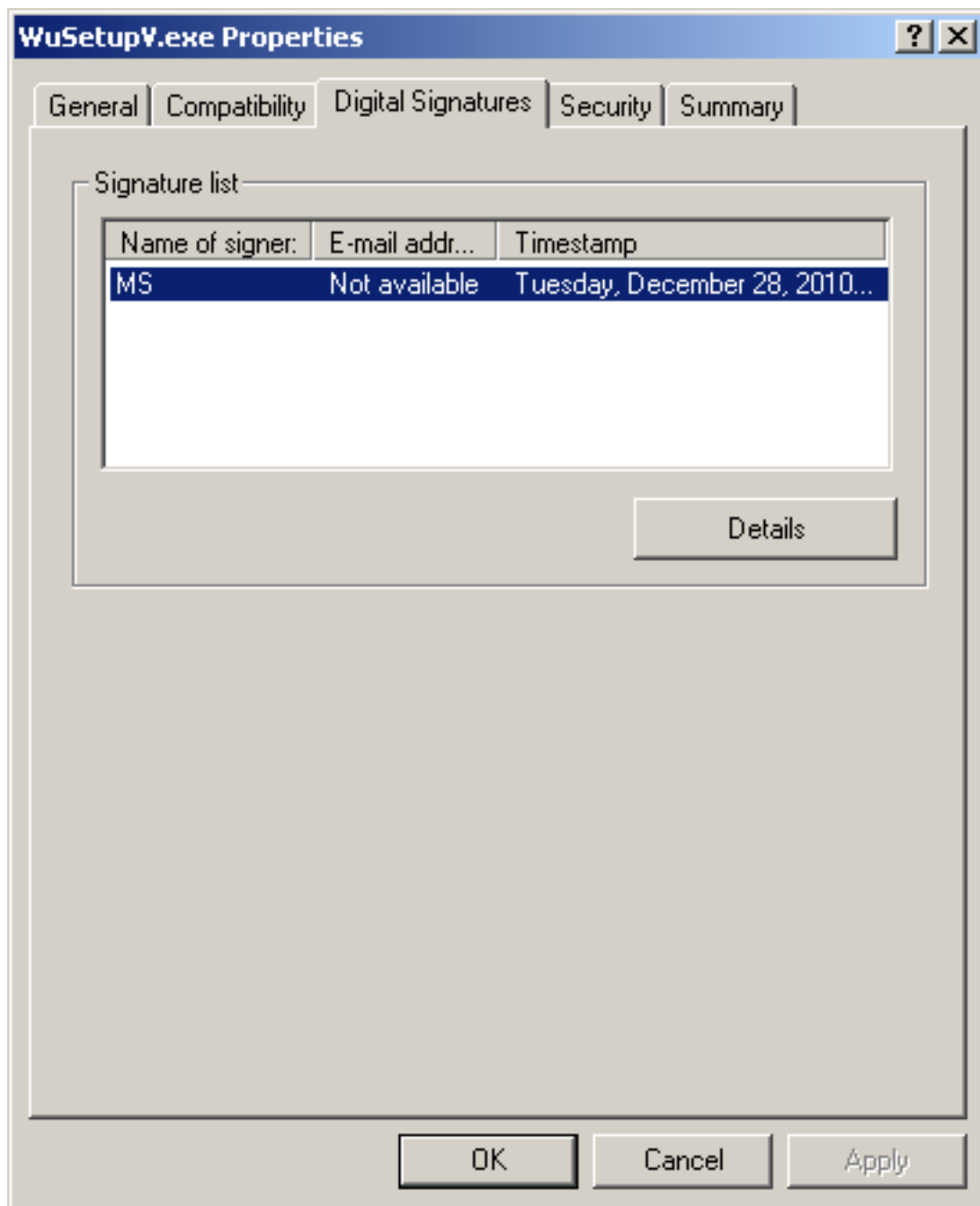
Lebanon
18

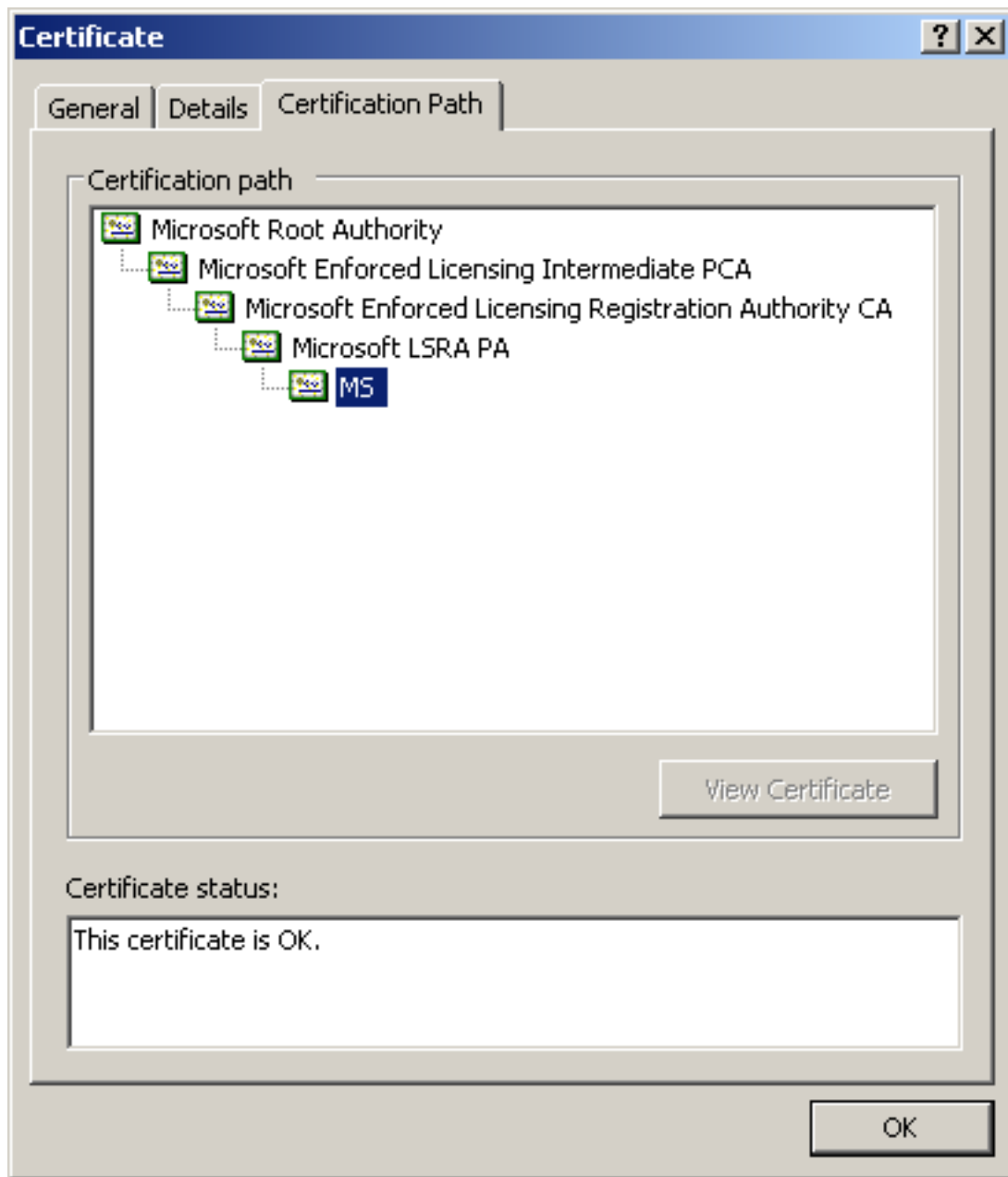
Saudi
Arabia
10

Egypt
5

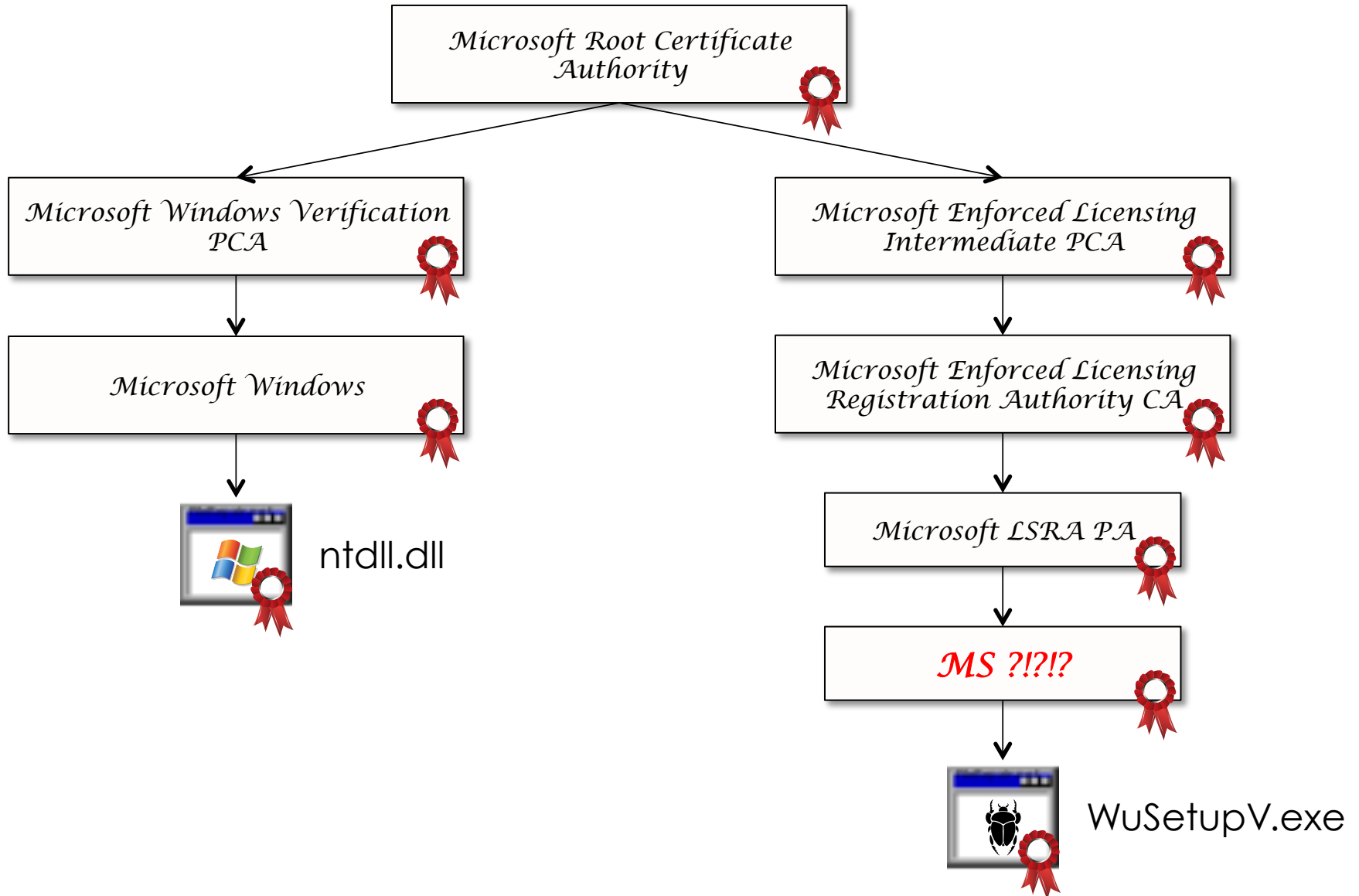
Flame propagation

- Flame registers itself as a proxy server for `update.microsoft.com` and other domains
 - WPAD (Web Proxy Auto-Discovery Protocol)
 - local network only
- Man-in-the-middle on Windows Update
 - SSL spoofing is not needed, Windows Update falls back to plaintext HTTP
 - serves a fake update signed with a Microsoft code-signing certificate





Certificate hierarchy





Part II

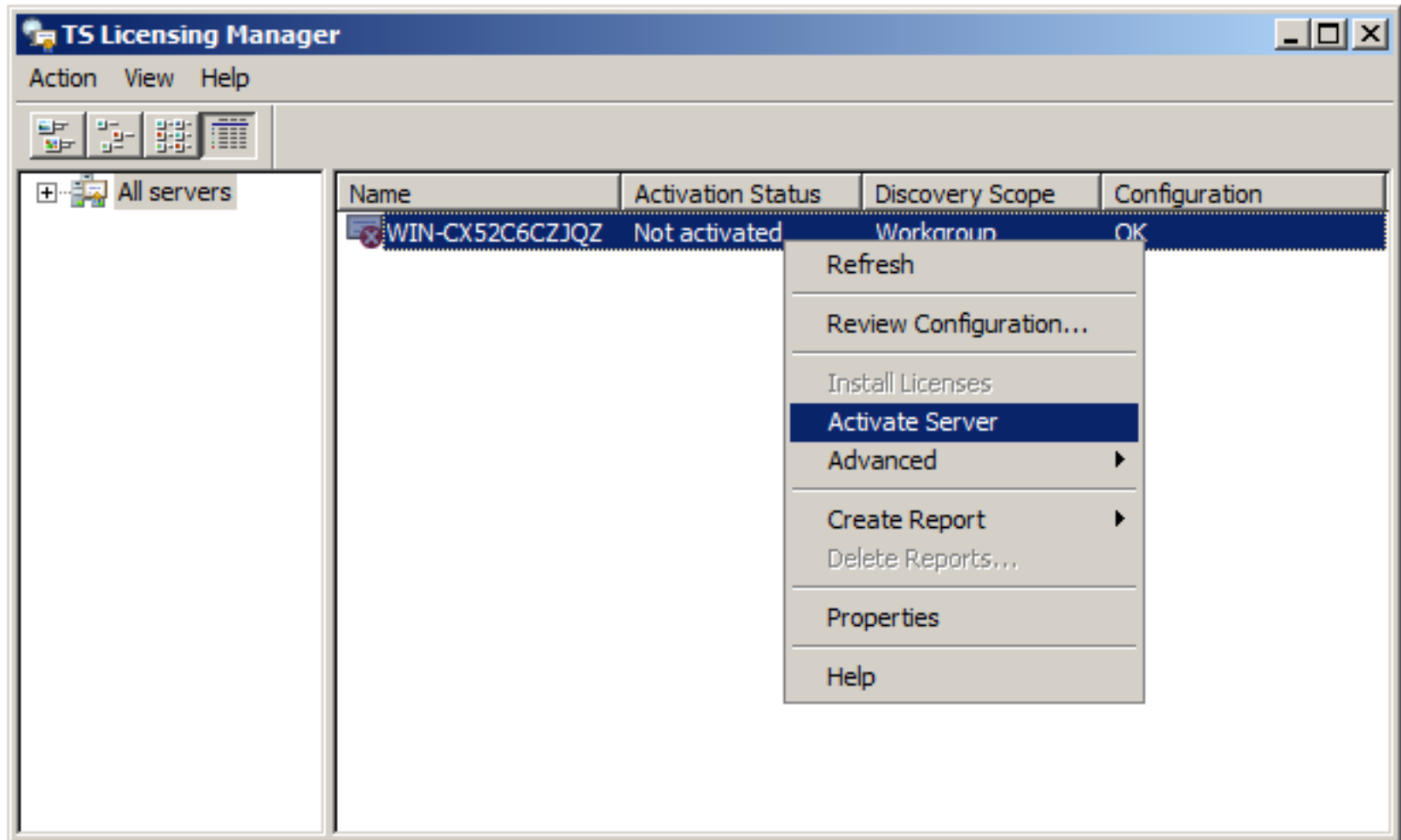
Terminal Services Licensing



Terminal Services Licensing

- License management system for Terminal Services clients
- Based on X.509 certificates, signed by a Microsoft certificate authority
- The license server receives a signed certificate during the activation process
- Fully automated process

License Server activation




License Server activation

Activate Server Wizard

Connection Method

Select the most appropriate connection method.



The connection method selected for license server activation will also be used to contact the Microsoft Clearinghouse when Terminal Services client access licenses (TS CALs) are installed.

To change the connection method after activation, go to the Connection Method tab of the license server's Properties dialog box.

Connection method:

Automatic connection (recommended)

Description:

This is the recommended method. The license server will automatically exchange the required information with the Microsoft Clearinghouse over the Internet.

Requirements:

The computer must be able to connect to the Internet by using a Secure Sockets Layer (SSL) connection.

< Back

Next >

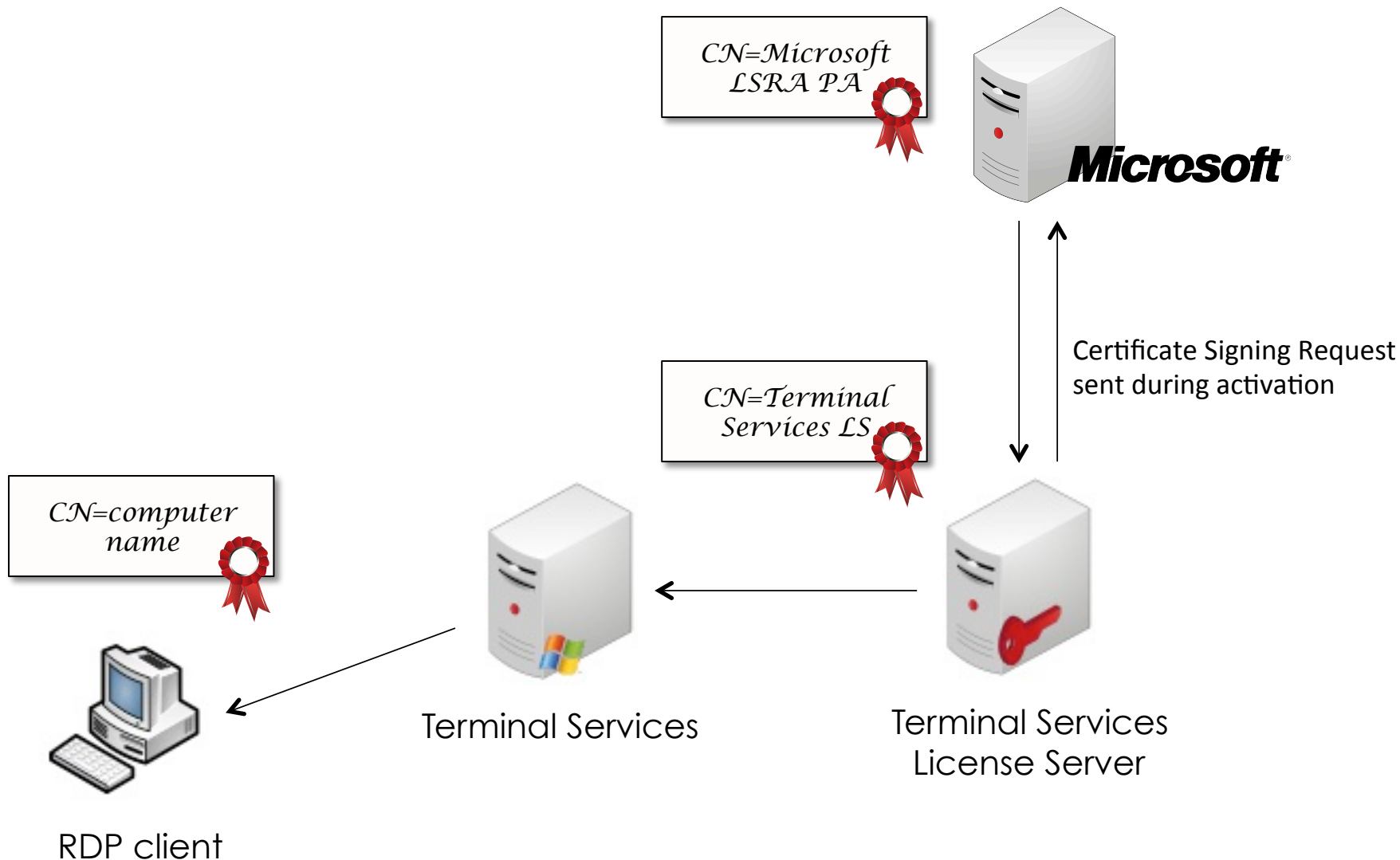
Cancel



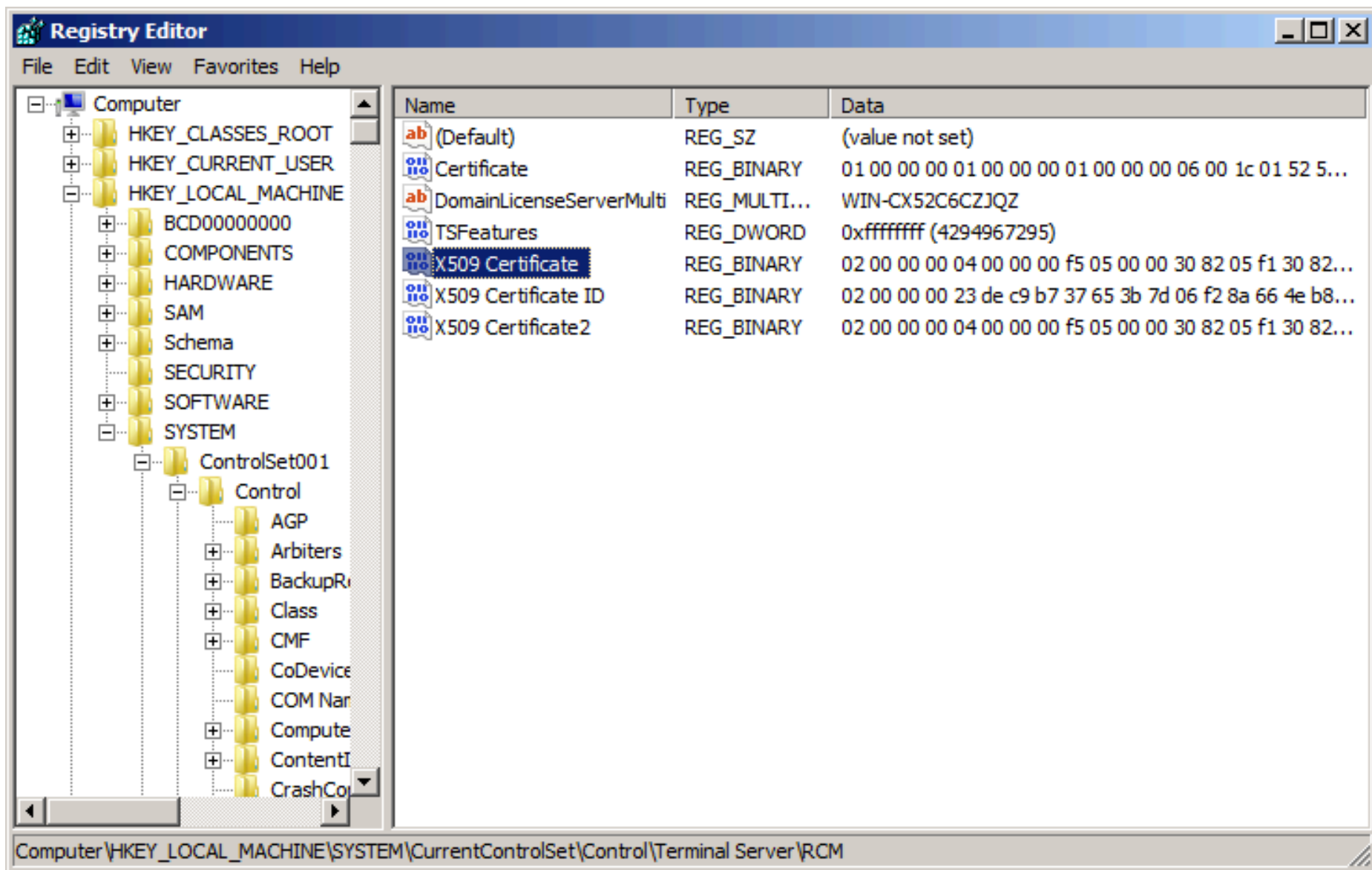
License Server activation

1. License Server generates a private key
2. License Server creates an X.509 Certificate Signing Request containing:
 - user information entered in the activation wizard
 - machine id ?
 - public key
3. Microsoft activation server returns a certificate signed by the Microsoft LSRA PA certificate authority containing:
 - subject CN=Terminal Services LS
 - public key
 - MD5 signature
4. The certificate is stored in HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\RCM\X509 Certificate

Terminal Services Licensing



Terminal Services certificate



Finding old certificates

- The Microsoft LSRA PA certificate authority was replaced after Flame became public
- New certificates are issued from a different PKI root and are signed with SHA-1
- Since the certificates are stored in the registry, we can find a few registry dumps containing certificates from 2010-2011 with a simple Google search



"31 32 30 32 31 39 32 31 34 38 33 39 5a"

Search

3 results (0.02 seconds)

- Web**
[30:82:1f:b0:06:09:2a:86:48:86:f7:0d:01:07:02:a0:82:1f:a1:30:82:1f ...
dev.bitradius.com/key1.txt](#)
- Images**
[... 74:69:6f:6e:20:41:75:74:68:6f:72:69:74:79:20:43:41:30:1e:17:0d:31:30:30:32:
31:39:32:31:34:38:33:39:5a:17:0d:31:32:30:32:31:39:32:31:34:38:33:39:5a:30:81 ...](#)
- Maps**
- Videos**
[Windows Registry Editor Version 5.00 \[HKEY_LOCAL_MACHINE
swirski.net/wtsfix2.reg](#)
- News**
[... 79,20,43,41,30,1e,17,0d,31,30,30,32,31,39,32,31,34,38,33,39,5a,17,0d,31,32,\
30,32,31,39,32,31,34,38,33,39,5a,30,81,80,31,13,30,11,06,0a,09,92,26,89,93 ...](#)
- Shopping**
- More**
[Windows Registry Editor Version 5.00 \[HKEY_LOCAL_MACHINE ...
swirski.net/wtsfix.reg](#)
- [... 79,20,43,41,30,1e,17,0d,31,30,30,32,31,39,32,31,34,38,33,39,5a,17,0d,31,32,\
30,32,31,39,32,31,34,38,33,39,5a,30,81,80,31,13,30,11,06,0a,09,92,26,89,93 ...](#)
- New York, NY**
[Change location](#)

Certificate properties

- Subject is CN=Terminal Services LS
- All certificates issued by Microsoft LSRA PA were valid until Feb 19, 2012
- No other identifying information
- No Extended Key Usage restrictions
 - inherited from the CA certificate, which allows code signing
- Microsoft Hydra X.509 extension
 - not supported by Crypto API
 - certificate fails validation and cannot be used for code-signing on Vista and Windows 7

Everyone can sign code!

- Everybody with an activated Terminal Server could also sign code as Microsoft and spoof Windows Update on XP
- On Vista and Windows 7, the certificate fails to validate because of the Hydra extension
- MD5 collisions was necessary to remove the extension and allow the attack to work on all versions of Windows



Part III

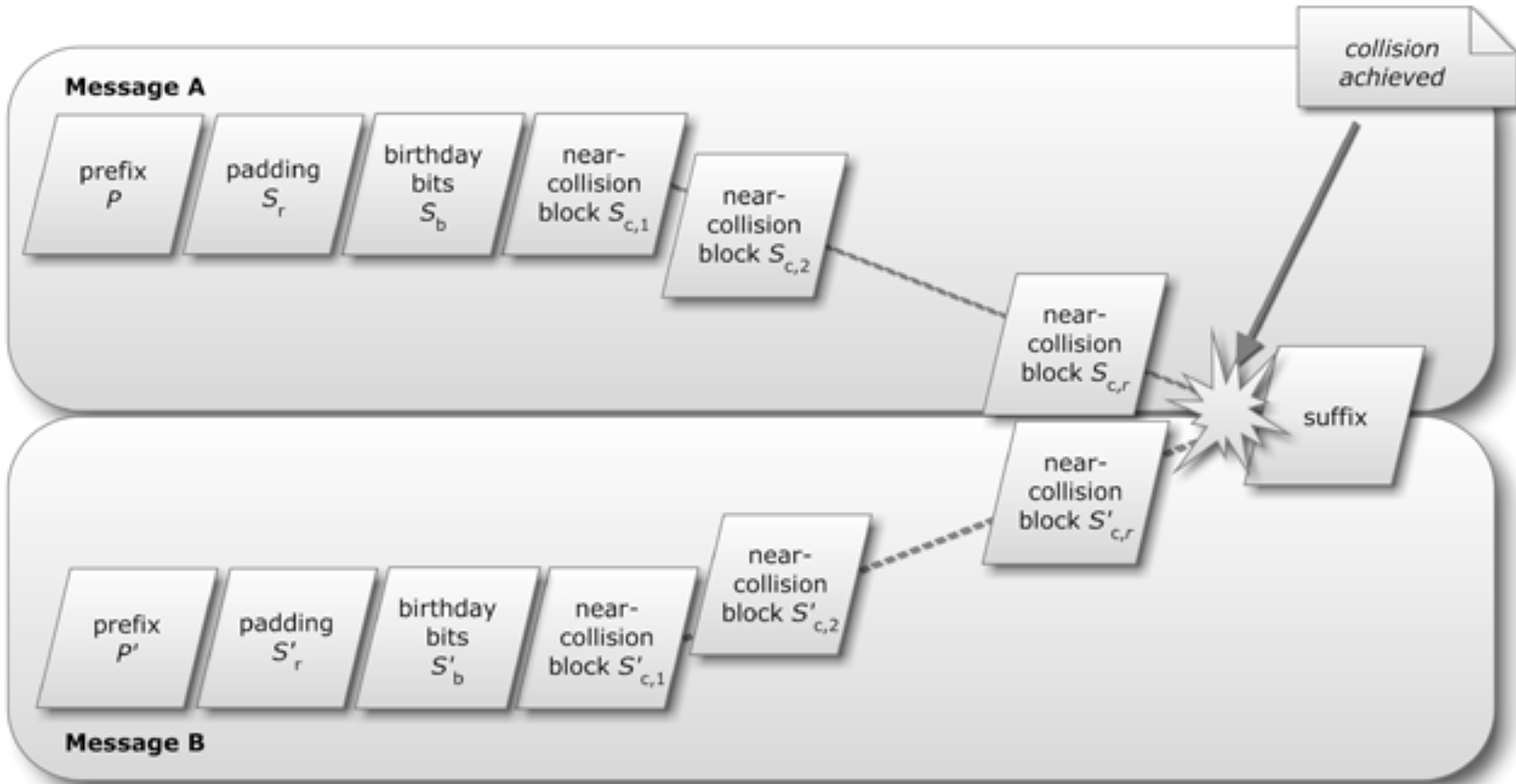
Background on MD5 collisions

MD5 hash algorithm

- Hash function designed in 1991
- Known to have weaknesses since 1993
- First demonstrated collisions in 2004
- Despite demonstrated attacks, remained in wide use until recently

- Classical collisions
 - insert specially computed blocks in a file to produce two files with different contents and matching MD5 hashes
 - limited control over the collisions blocks
- Chosen-prefix collisions
 - first demonstrated by Marc Stevens at Technische Universiteit Eindhoven in 2006
 - append specially computed blocks to two different files to make their hashes match
 - arbitrary prefixes before the collisions block

Chosen-prefix MD5 collisions



RapidSSL attack in 2008

- Collaboration of hackers and academics led by Alex Sotirov and Marc Stevens
- Demonstrated a practical MD5 collision attack against the RapidSSL CA:
 - resulted in a rogue SSL certificate authority trusted by all browsers
 - allows man-in-the-middle attacks on SSL
- Presented at the CCC in 2008
- Authors worked with CAs to discontinue all use of MD5 signatures

RapidSSL collision generation

- About 2 days on a cluster of 200 PS3s
- Equivalent to about \$20k on Amazon EC2



Generating a rogue certificate

1. Predict the contents of the real certificate that will be issued by the CA
 - most fields have fixed values or are controlled by us
 - we need to predict the serial number and validity period, which are set by the CA
2. Build a rogue certificate with arbitrary contents
3. Generate RSA public key containing collision blocks that make the MD5 hashes of the two certificates match
4. Get signed certificate for a domain we own from the certificate authority
5. Copy signature to the rogue certificate

Colliding SSL certificates

serial number	chosen prefix (difference)	serial number
validity period		validity period
real cert domain name		rogue cert domain name
real cert RSA key	collision bits (computed)	real cert RSA key
X.509 extensions	identical bytes (copied from real cert)	X.509 extensions
signature		signature

- The contents of the real certificate must be known before we can generate the collision blocks
- Collision generation takes about 2 days
- How do we predict the serial number and validity period of our certificate two days before it is issued?



Part IV

MD5 collision in Flame

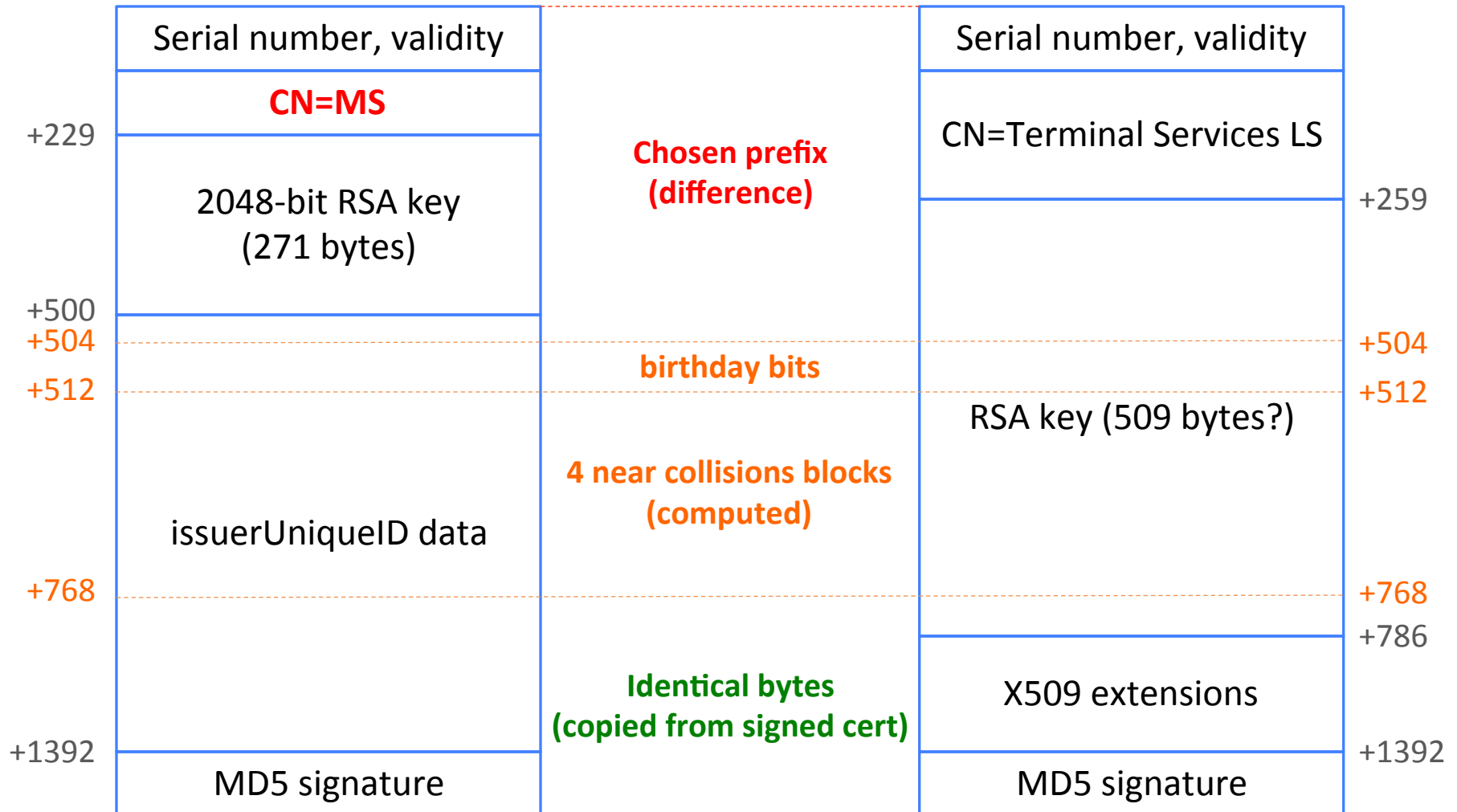
Flame certificate properties

- Fields entirely controlled by the attacker:
 - serial number 7038
 - validity from Feb 19, 2010 to Feb 19, 2012
 - subject CN=MS
 - 2048-bit RSA key
- Non-standard issuerUniqueID field:
 - ignored by Crypto API on Windows
 - contains the birthday bits and near collision blocks generated by the attacker
 - the length of the field also covers the X.509 extensions from the real certificate, thus hiding them from Crypto API

Colliding certificates

Flame certificate

Certificate signed by Microsoft



Cryptographic complexity

- 64 birthday bits, 4 near collision blocks
- Similar complexity to the RapidSSL attack for a single collision attempt
- About \$20k on Amazon EC2 in 2008, or cheaper if you have a large cluster

- Predicting the validity period
 - fully automated CA operation
 - validity period determined by time of request
 - attacker need to get the certificate issued in a 1-second window
- Predicting the serial number
 - serial number based on a sequential certificate number and the current time
 - attacker needs to get the certificate issued in a 1-millisecond window
 - significantly more difficult

Predicting the serial number

- Sample serial numbers from the Microsoft LSRA PA certificate authority:

Feb	23	19:21:36	2010	GMT	14:51:5b:02:00:00:00:00:00:08
Jul	19	13:41:52	2010	GMT	33:f3:59:ca:00:00:00:05:25:e0
Jan	9	20:48:22	2011	GMT	47:67:04:39:00:00:00:0e:a2:e3

- Serial number format:
 - number of milliseconds since boot (4 bytes)
 - CA index (fixed 2 byte value)
 - sequential certificate number (4 bytes)

Predicting the serial number

- Sequential certificate number
 - each certificate gives the attacker its current value and increments it by one
 - attacker can increment it to an arbitrary number by getting more certificates
- Number of milliseconds since boot
 - each certificate discloses its current value
 - incremented each millisecond until the system is rebooted
 - attacker needs to get certificate at the right time to match the predicted serial number

Predicting the serial number

- Sources of timing variability
 - system load
 - packet jitter
- Large number of attempts required to get the certificate issued at the right moment
 - significantly more costly than the RapidSSL attack, likely 10-100x
 - did the attackers have a much faster collision generation algorithm or a larger cluster?
 - were they located close to the target server to minimize packet jitter?

Cryptographic forensics

- The tool used for the RapidSSL attack was open-sourced in 2009
- Did the Flame authors use it?

hashclash

Framework for MD5 & SHA-1 Differential Path Construction and Chosen-Prefix Collisions for MD5



Project Home

[Downloads](#)

[Wiki](#)

[Issues](#)

[Source](#)

Summary [People](#)

Project Information

Recommend this on Google

[Project feeds](#)

Code license

[GNU GPL v3](#)

Labels

md5, collision, differentialpath,
framework, chosen-prefix,
hashclash, birthdaysearch



Members

[marc.ste...@cwi.nl](#)

HashClash



News

2010-11-08 Added SHA-1 programs:

- `diffpathanalysis_sha1` to analyze disturbance vectors, generate message rounds 2,3,4 (requires CUDA to run).

Cryptographic forensics

The bit differences in the near collision blocks can be used to determine what technique produced them:

Using our forensic tool, we have indeed verified that a chosen-prefix collision attack against MD5 has been used for Flame. More interestingly, the results have shown that not our published chosen-prefix collision attack was used, but an entirely new and unknown variant. This has led to our conclusion that the design of Flame is partly based on world-class cryptanalysis.

Marc Stevens, CWI.nl



Remaining Questions

- Was the collision generated with the open-source HashClash tool or developed independently?

- Flame Authenticode Dumps
<http://blog.didierstevens.com/2012/06/06/flame-authenticode-dumps-kb2718704/>
- RapidSSL attack
<http://www.win.tue.nl/hashclash/rogue-ca/>
- Flame malware collision attack explained
<http://blogs.technet.com/b/srd/archive/2012/06/06/more-information-about-the-digital-certificates-used-to-sign-the-flame-malware.aspx>
- Marc Stevens' PhD thesis
<http://marc-stevens.nl/research/papers/PhD%20Thesis%20Marc%20Stevens%20-%20Attacks%20on%20Hash%20Functions%20and%20Applications.pdf>
- CWI cryptanalyst discovers new cryptographic attack variant in Flame spy malware
<http://www.cwi.nl/news/2012/cwi-cryptanalyst-discovers-new-cryptographic-attack-variant-in-flame-spy-malware>
- MSRC 2718704 and Nested EKU enforcement
<http://rmhrisk.wpengine.com/?p=57>
- Analyzing Flame's replication pattern
http://threatpost.com/en_us/blogs/snack-attack-analyzing-flames-replication-pattern-060712
- Microsoft Certificate Services serial numbers
<http://blacktip.wordpress.com/2010/06/24/serial-killer/>