Repository Of Signed Code

Levente Buttyán
Laboratory of Cryptography and System Security (CrySyS Lab)
Budapest University of Technology and Economics
www.crysys.hu

this is joint work with D. Papp, B. Kócsó, T. Holczer, and B. Bencsáth

* work funded by the Office of Naval Research Global (ONRG)
Motivation

- modern operating systems require digital signature on system software before it is installed
  - drivers, OS updates, ...

- advanced attackers (APTs) started to use malware signed with compromised keys or fake certificates
  - kernel drivers used by Stuxnet and Duqu were signed with compromised keys of otherwise legitimate hardware manufacturers
  - Flame appeared to be a signed Windows update; certificate chain contained a fake certificate that looked like a valid Microsoft certificate
Motivation

- more recent examples
  - Winnti (2011, 2013)
    - in 2011, the group infected players of a popular online game via a malicious game update signed with the possibly compromised code signing key of a South-Korean game vendor
    - attacks against South Korean social networks Cyworld and Nate in 2011 used a Trojan that was digitally signed using a certificate stolen from a Japanese gaming company
    - a digital certificate of the same company was used in 2013 in Trojans deployed against Tibetan and Uyghur activists
    - successful cyber espionage attacks on companies such as Apple, Facebook, Twitter and Microsoft in 2013
    - attackers returned in 2015 and used a dropper that was signed with a stolen and still valid code signing certificate belonging to Taiwanese electronics maker Acer

- problem: standard signature verification procedure does not allow for detecting key compromise and fake certificates
Objectives

- augment the standard signature verification workflow with additional services that help to detect malicious software
  - provide reputation information on signers and signed code
    - Is this a known signed software?
    - What do we know about it? (e.g., Virus Total score)
    - How many other users have requested information about this software?
    - Is this software has a known signer?
    - What do we know about pieces of software it signed before?
  - notify key owner when a new object signed with a specific key is seen
    - this makes it possible to detect key compromise and fake certificates relatively quickly

- build a system that provides the necessary infrastructure and mechanisms for these additional services
Use case: Checking signer reputation

com.harvesters.linkupwow

ivan
Use case: Checking signer reputation

com.harvesters.linkupwow

1/47
Use case: Checking signer reputation

what else has ivan signed?

- com.androidemu.harvemm1
- com.androidemu.harvespmxd
- com.androidemu.harvedragon3
- com.harvesters.linkupwow
- ...
Use case: Checking signer reputation

- com.androidemu.harvemm1
- com.androidemu.harvespmxd
- com.androidemu.harvedragon3
- com.harvesters.linkupwow
- ...

23/55
23/51
23/54
23/54
22/50
...

ivan
Use case: Alerting key owners
Use case: Alerting key owners
Use case: Alerting key owners

never seen before
### Approach

- **develop a large database that can store millions of signed objects**
  - Portable Executable (PE) files
  - Java Archives and Android Packages (JAR/APK)
  - public key certificates

- **provide services built on top of the database**
  - simple queries for file hashes
  - complex queries based on object attributes
  - visualization of relationships between signed software and certificates
  - alerting users when the system encounters an object matching some pre-registered criteria

- **provide a web based and a programmatic (JSON) interface to the services**

- **collect signed software and certificates massively**
  - proactive crawling of public sites and repositories
  - allow for uploading objects by users
System architecture

1. **SQL DB**
   - meta data
   - alert info
   - session info

2. **No-SQL DB**
   - attributes
   - ref to file
   - relationships

3. **Alert Engine**

4. **Query Engine**

5. **Upload Interface (Web)**

6. **Feed Interface (files)**

7. **Alert Reg Interface (Web)**

8. **Search Interface (Web)**

9. **JSON Interface**

10. **Preprocessing Modules (per object type)**

11. **Uploads (single)**

12. **Search (Web)**

13. **Alerts**

14. **Feeds (bulk)**

15. **Query (Web)**

16. **JSON (Web)**

17. **Alerts**

18. **Connection to external malware DB**
### ROSCO DBs

- **Hadoop cluster of 6 nodes**
  - 1 name node, 5 data nodes
  - 100TB total disk space
  - ~33TB effective capacity

- **HBase database**
  - open source, no-SQL, distributed DB
  - tables for object attributes and relationships between objects

- **regular SQL database**
  - meta-data of objects
  - alert filters
  - user and session data
Object types collected

- **X.509 public key certificates**
  - millions of certificates collected (~60 million) by
    - acquiring available collections (e.g., SSL Observatory) and using ZMap
    - extracting certificates from signed software

- **signed Portable Executables (exe, dll)**
  - thousands of files collected by
    - crawling public software repos (e.g., SourceForge)
    - browsing OS distributions
    - filtering malware feeds

- **signed Java Archives (jar) and Android Packages (apk)**
  - thousands of files collected by crawling third party app stores
Pre-processing modules

- each object type has its own pre-processing module that parses the object and inserts appropriate data in the DBs.

- parsing process may invoke other pre-processing modules:
  - e.g., PE file may have certificates embedded, which are passed to the pre-processing module responsible for certificates.

- duplicates are checked before inserting data into the DB:
  - crawlers may return objects that have already been stored.
  - in case of duplicates, only meta-data is updated.

- relationships to already stored objects are identified when inserting a new object:
  - is the new object signed with a known public key?
  - if the new object is a certificate, does it contain a known public key?
Relationships between objects

- can be represented by a directed graph
  - three types of nodes: certificate, public key, signed software
  - two types of edges:
    - certificate → public key: certificate contains the public key
    - public key → signed object: public key verifies signature

- example:
Relationships between objects

Graph representation of connected signed objects

Nodes
- Public key
- Certificate
- PE
- JAR
- APK

Edges
- Public key → SO: Completely verified
- Public key ← SO: Contained
- Public key ← SO: Self signed

Details

<table>
<thead>
<tr>
<th>ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Hash: 51a97ad597e4b9443bda0a97be6244f0ff49e4512ca6f4d8ec5f66a20ab4146a</td>
</tr>
<tr>
<td></td>
<td>Vendor: Sun Microsystems Inc.</td>
</tr>
<tr>
<td></td>
<td>Package name: com.harvesters.linkupwow</td>
</tr>
<tr>
<td></td>
<td>Filename: com.harvesters.linkupwow_093124.apk</td>
</tr>
<tr>
<td>1</td>
<td>Hash: 14e6754180c7e3165413cb09b6c2ba03746f0e4af5b614018b934c6615c42f</td>
</tr>
<tr>
<td></td>
<td>Type: RSA</td>
</tr>
<tr>
<td></td>
<td>Length: 1024</td>
</tr>
<tr>
<td>10</td>
<td>Hash: 14e220d9a0cb6ea5dc17bb80c67f6906af25d26f72f58b6c6645eb9e77eb16c</td>
</tr>
<tr>
<td></td>
<td>Issuer CN: iven</td>
</tr>
<tr>
<td></td>
<td>Subject CN: iven</td>
</tr>
<tr>
<td></td>
<td>Valid from: 2011-04-16 11:28:46</td>
</tr>
<tr>
<td></td>
<td>Valid to: 2066-01-17 11:28:46</td>
</tr>
<tr>
<td></td>
<td>Issuer C: ZH</td>
</tr>
<tr>
<td></td>
<td>Subject C: ZH</td>
</tr>
</tbody>
</table>
the same features are also available via the JSON interface of ROSCO!
Search options
Certificate search

X.509 certificate fields and extensions
Certificate search

Results - All result: 6182

- E7051650A758A4820B2B614CB2A185A867320575E69ADCF258EDB1437B215832
- A048C4C84FA0B046E9DC4F2CA4D389FDC2008CDBF8D859B03C1BFED18898
- 666057354045624C7444AD00FAE3852A0BD3228FD7A04145E92CB2EC20FE26E
- A54EAF0C02BC35E91FA513A99D3119E015B125403CE311102238D69ED62CB7A4
- 2347AB242719DF0EAB91E230A508EAD604ECF27A4C176F84AB1574A5C4502
- 348207703C80C189750324885AB728691F6E2514E79EAA264C18D5C4E76065
- 1FA2353C597D5D6EE6115E876B37341EAAB5A3EF9A3D52061DC42956E70BBA
- 15F16D132D4AA6D7855D909E9D3484FC36554399C1BE2507B119D57FAEE4EB
- 7EF2F1B63F747B3D3D9A4F4519CF65D3DFF28715509586437644C3B37816D426
- 278AC217F30D90EC81087C741EA2E406E0363D1395D0C565DE409C6A2DAB6A911
- 1B517B585CFDE60C0002B519C33C8DE3485BFF759BD0C2D18C143F85913375
- D605D19F4379670338698C83602D7216C180E95C1E68672B5DAD556D9228
- 2B8ADBE565C07E22AFD322C86701B8675467C297D0F1623FP86472C3610FFC
- 61540F87AA541C894206DA78CE6EFF65069913223E857F9E261D4A81598BB
- 028D0428000C07500423F5D7020825212084C0411200FB22C562
## Certificate search

### Metadata

<table>
<thead>
<tr>
<th>Last viewed</th>
<th>2015-09-29 06:27:30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uploaded at</td>
<td>2014-12-02 18:54:24</td>
</tr>
<tr>
<td>Uploaded from</td>
<td>sslobservatory</td>
</tr>
<tr>
<td>Uploaded at</td>
<td>2015-09-17 06:23:31</td>
</tr>
<tr>
<td>Uploaded from</td>
<td>sslobservatory</td>
</tr>
</tbody>
</table>

| Queried counter | 34                   |

### Certificate

<table>
<thead>
<tr>
<th>Signed Object SHA1</th>
<th>3A82B1B23E3498D8296C15BDD0205DFCDEC98278</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signed Object MD5</td>
<td>F3D3CBB2CBE094F6FA93BEC1D082B9CF</td>
</tr>
<tr>
<td>Version</td>
<td>3</td>
</tr>
<tr>
<td><strong>Serial number</strong></td>
<td>12345678</td>
</tr>
<tr>
<td>Valid from</td>
<td>2005-07-07 12:57:15</td>
</tr>
<tr>
<td>Valid to</td>
<td>2007-05-22 16:41:47</td>
</tr>
<tr>
<td>Issuer CN</td>
<td>admin.starkingnet.hu</td>
</tr>
</tbody>
</table>
Certificate search

Serial Number: 12345678

Issuer C (country):
HU

Prefix search is case sensitive
• suggested when you exact
Not prefix search is not case sensitive
• suggested when you not exact

Timeout for searches:
60 sec
Not given parameter: NULL
Malformed parameter: MALFORMED

Search

Results - All result: 8

• 4FA0C6A5D7B40020ED9CF860BC5A6BDCE35376809D1ED043A4CEC7A18763F05
• 7EF2F1B63F747B3D3D9A4F4519CF65D3DFP287155059586437644C3B37816D426
• AB7A7543B2ED3C07D6FDF7D43246D8E3C55AEEBBB19DE0ED8957F6C6035E016A
• 225F3BA987D21FF8BBC2C49EAB39C88A456443AE13615BD5358BF0F87B2EE26
• 687BDC88E17EE452FF3A021C502353D26498C0B18FA0AD730C39E5DF2BDE5DF
• 86A6D8A4B642F096308C0AAC24B5FE7537CDDB47A7863CDE345FE09248D72284
• 9C51C34588892884699EE10410497296A55D52B944464285BF04C18F503E4018
• 72961F38BED425C63209D06BA504CD65F9AB4168D065DFBEA0B2D965D6C65967
## PE search

<table>
<thead>
<tr>
<th>Type</th>
<th>Filename</th>
<th>Timestamp</th>
<th>Min OS version</th>
<th>Potential Malware</th>
</tr>
</thead>
</table>

Prefix search is case sensitive
- suggested when you exactly know what to search
Not prefix search is not case sensitive
- suggested when you not exactly know what to search

Timeout for searches:

60 sec
Not given parameter: NULL
Malformed parameter: MALFORMED

Search
PE search

Type: EXE

Results - All result: 3811

- 0484DEFCEB264AF7FF71548F53F6274D29CA74EF0520D96F079E58585793B106
- 82F14FC1A9175A842CE3F1204906E27A64680CA199C63BB983866A9A99F754CEE
- F291EAF4B561C80A63346DB6E3E8D4AC6A7F3B51D4E65893A377D24E68AD6143
- A8B6504711179C2D54B306D3FEC17C3CD1BB369F64041CDA0AD3FB7924A4BBCF
- 6858F7CDAC133077BE5DFC831FDFA8F587BE01A144639CE533E47D3F18BDDFC
- 1D42C869EACC2925B183B8E8CCFD537C48D9C0C1B2877A8B656D0C62C3F5E78E
- 7738DB32095B3F27FE8BA6D3A9015D1A51F94E692BDC0213E2160C89A350
- 785A4222BEDA2E2AE35E579E27AF944D53FC111E22D0C364E58628B1CB3519
- 15721AFB2B245E9C90F7C647D07ED756463DB979D0270025A23F2DC214AD2492
- C3AAD1CFB2F0355BA557B2A50C728E29DB7A13095F8B541EF3B0D90F47C8C3
- 0BFCF478FA7A705B4762BE9EE4B5282722FFB67163C2CDCFC18CFF39561A104C
- 6A3FD1EE1B4CFD67ED231C97406806FC18F837CC7C8579E1E6009F5E8FF10F7
- 11F7D7F4E51D5C11CEBDD17F6DFFCABF4DBB0ACB79A93E2304D480E0964426F54
- 6FA58E9C27F6AC0D423B2A9ECDDB9385A4814CB97766D8180E38A4A46E727BA
binary was obtained from a malware feed (via bulk upload) or VT score > 33% (a script regularly checks the VT score of all stored binaries)
Public key search

Prefix search is case sensitive
- suggested when you exactly know what to search
Not prefix search is not case sensitive
- suggested when you not exactly know what to search

Timeout for searches:
60 sec
Not given parameter: NULL
Malformed parameter: MALFORMED

Search
Public key search

Public key

Length:
768

Results

All result: 51075

- 94D6A51FB54510609143A3B089220C5F94FF59B80DFB3656949191D882F8D296
- DEA37448C6C9976B47E55835498AC73BE9865507B9A031753E6F5E8B944503E3
- 30F69AAB30B1F113AD364300C53E343AF9D9BE07D10087900B35F3D68C10FD1
- 717ECB67DB141295F9404AC8A66B8E8B3513EC52DAE7DD0EA682BFFA4D9F5D
- 3D119640ED32C38629997AED357194EC31425430B9EE02E374774C76183BC8C6
- 0C22E2BD6DA5E853808779A4B9D06C9FD67EB5603A59AE8CCAC00185002DF2
- C4030ABF2BD4361E25EA5DD75A46E8DA93E1C3D27BFD94E944A5DCAF96092BC8
- 99B139EB6B2FD68A0554E5B98E9827B647C5B1EE14811FC91015044DA5618749
- 3065B312638415500688D14B97F6581AB7DFD99E14F8803F6A0FA9A3F7B4F22
- 04A6055B9F39194CA0CC66899F5CE8819E1F8F95A4CBDC325639CB120196CFF1
- 37F52D388F357A6AB5C140C94BDC4AFD7A9A9F4998BB485B7531F85522D285E3
- 2580F33CD7D26E02DDE14C20A3E527EC5938EFC8F2PF2FAE0E415E9968EF8
- 74F467C7C2BD6301258A42CE0DF8A8248B291CA3C12BD1CFFD1DBE529706F87F0
- EC3620F6BC4100465585B813DB1B5CFE5BBD2C6E4B8907D7C16F673642160
## Public key search

<table>
<thead>
<tr>
<th>Type</th>
<th>RSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>768</td>
</tr>
<tr>
<td>RSA modulus</td>
<td>C7599A86C45E3A2E55CD4486A9373322635208902D25ADC83BC3B32D434B3B929DAECB31754F55663EDF3F82B91B8F25C0856DED631A41763DAF0FA429EE3AC3DBC9DD737F3772341FDD94734C28D4A4B462475D45E2B484DE4397CC4341B6ED</td>
</tr>
<tr>
<td>RSA exponent</td>
<td></td>
</tr>
</tbody>
</table>

**Public key:**

94D6A51FB5451060914A3B089220C5F94FF59B80DFB3656949191D882F8D296

[Jump to graph view]
Public key search – graph view
### String alerts

<table>
<thead>
<tr>
<th>Active</th>
<th>Name</th>
<th>Type</th>
<th>Field</th>
<th>Keyword</th>
<th>Email</th>
<th>Notify</th>
<th>RSS</th>
<th>Matched</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>Microsec cert</td>
<td>CERT</td>
<td>Issuer CN (common name)</td>
<td>Microsec</td>
<td><a href="mailto:butlyan@crysys.hu">butlyan@crysys.hu</a></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Add alert**

### Signed alerts

<table>
<thead>
<tr>
<th>Active</th>
<th>Name</th>
<th>Type</th>
<th>Email</th>
<th>Notify</th>
<th>RSS</th>
<th>Matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>Name</td>
<td>CERT upload</td>
<td><a href="mailto:rosco@crysys.hu">rosco@crysys.hu</a></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

**Save alert**
Why should anyone use ROSCO?

- **end-user**
  - ROSCO helps identifying potentially malicious software before it is installed

- **singing party (CA or software maker)**
  - ROSCO helps detecting key compromise and fake certificates

- **software platform operators (e.g., operating system providers and global software service providers)**
  - they are also signing parties
  - providing data to ROSCO helps to maintain trust in their platform

- **security companies**
  - ROSCO can be an additional source of information
    - on end-user behavior (what applications they install?)
    - on attack campaigns and trends in signing malicious code

- **regulators and authorities**
  - ROSCO can help them to derive statistics that can serve as an input when defining global defense strategies and coordination mechanisms
Potential limitations

- central database operated by a single entity
  - needs to be trusted (→ independent academic research lab)
  - single point of failure (→ only extends current PKI, not replaces it)

- database must be fed with new data all the time
  - new signed objects (code and certificates)
  - regular update of "potential malware" flags

- users should learn about ROSCO and be motivated to use it
  - average user may not understand how ROSCO differs from Virus Total, Google’s Certificate Transparency, or Microsoft SmartScreen’s Filter

- signing parties should learn about ROSCO and be motivated to use it
  - usefulness of the alert service depends on the upload rate of new content and the overall coverage of ROSCO
Related work

- Virus Total
  - also allows for identifying potentially malicious software
  - based on a completely different approach
    - scanning submitted file with AV products
  - does not detect new malware immediately
    - ROSCO can identify fresh malware based on signer information
  - however, unlike ROSCO, VT also works for unsigned software

→ ROSCO complements the services provided by Virus Total
Related work

- **Windows SmartScreen**
  - a feature that helps to detect phishing websites and protects the user from installing malware
    - checks the visited sites against a dynamic list of reported phishing sites
    - checks files downloaded from the web against a black list of reported malicious software and a white list of well-known applications
  - only works on Windows
  - details are not public
    - are digital signatures used to reduce false positives?
    - does it use any other reputation information?

![Windows protected your PC](image.png)
Related work

- **Google Certificate Transparency**
  - makes it possible to detect certificates that have been mistakenly issued or maliciously acquired
  - based on three components
    - **Certificate Logs**
      - publicly auditable, append-only records of certificates
    - **Monitors**
      - periodically contact all of the log servers and watch for suspicious certificates
    - **Auditors**
      - verify that a particular certificate appears in a log
  - similar concept but focuses only on SSL/TLS certificates
Future plans

- acquire more data
  - continue crawling
  - develop collector apps
    - browser plug-in
    - mobile app
  - collaboration
  - build and run a Monitor for Certificate Transparency

- search for interesting anomalies and statistics in the DB

- open ROSCO for public non-commercial use
Interested in trying out?

or send an e-mail to: rosco-vb2015@crysys.hu

please send feedback to: rosco-feedback@crysys.hu
Laboratory of Cryptography and System Security (CrySyS Lab)  
Budapest University of Technology and Economics  
www.crysys.hu

contact:

Levente Buttyán, PhD  
Associate Professor, Head of the CrySyS Lab  
buttyan@crysys.hu