Yandex

Managing digital code signing in an engineering company

Eldar Zaitov, Evgeny Sidorov

What the talk is about?

- Code signing
- How to keep signing keys secret in a big company
- What advantages can be taken

What is Yandex?

- Yandex is one of the largest internet companies in Europe, operating Russia’s most popular search engine
- over 4000 software developers
- over 40 applications with millions of users

Intro

Code Signing

- The app isn’t tampered with (integrity)
- Created by those it claims to come from (authenticity)
- Verifier checks the signature
- Verifier decides if it trusts the publisher

Challenges

- Signing identities must be kept secret
- 3rd parties and former employees mustn’t have access to signing identities
- Signing identities access management - lots of developers need to sign their apps

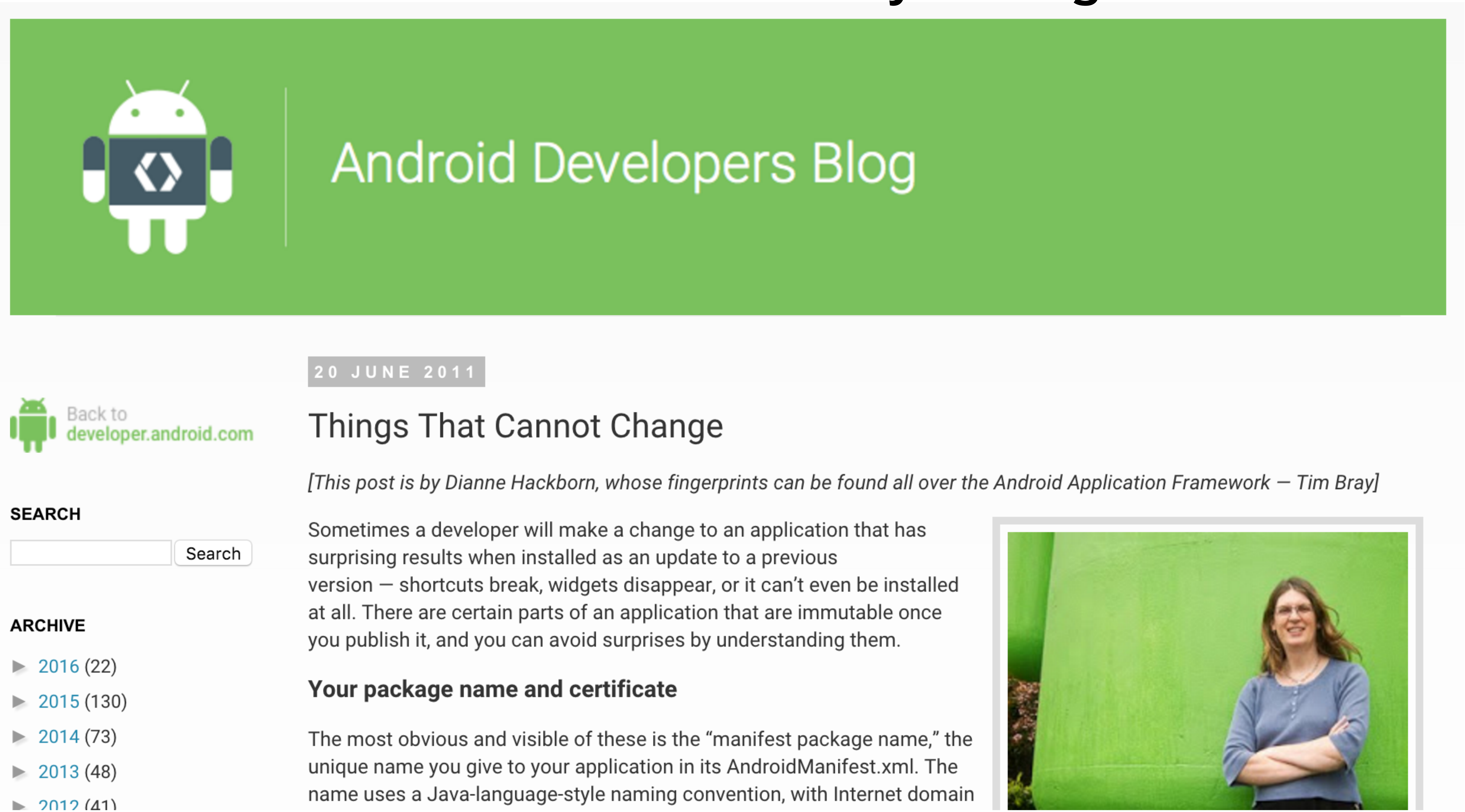
Challenges

- Usability
- Continuous integration

Key leakage

- Stuxnet (two companies signing keys were compromised)
- Nokia’s signing keys leak http://bit.ly/T7unFj

Android: certificate can’t be easily changed



Android Developers Blog

Back to developer.android.com

SEARCH

Search

ARCHIVE

- ▶ 2016 (22)
- ▶ 2015 (130)
- ▶ 2014 (73)
- ▶ 2013 (48)
- ▶ 2012 (41)

20 JUNE 2011

Things That Cannot Change

[This post is by Dianne Hackborn, whose fingerprints can be found all over the Android Application Framework — Tim Bray]

Sometimes a developer will make a change to an application that has surprising results when installed as an update to a previous version — shortcuts break, widgets disappear, or it can’t even be installed at all. There are certain parts of an application that are immutable once you publish it, and you can avoid surprises by understanding them.

Your package name and certificate

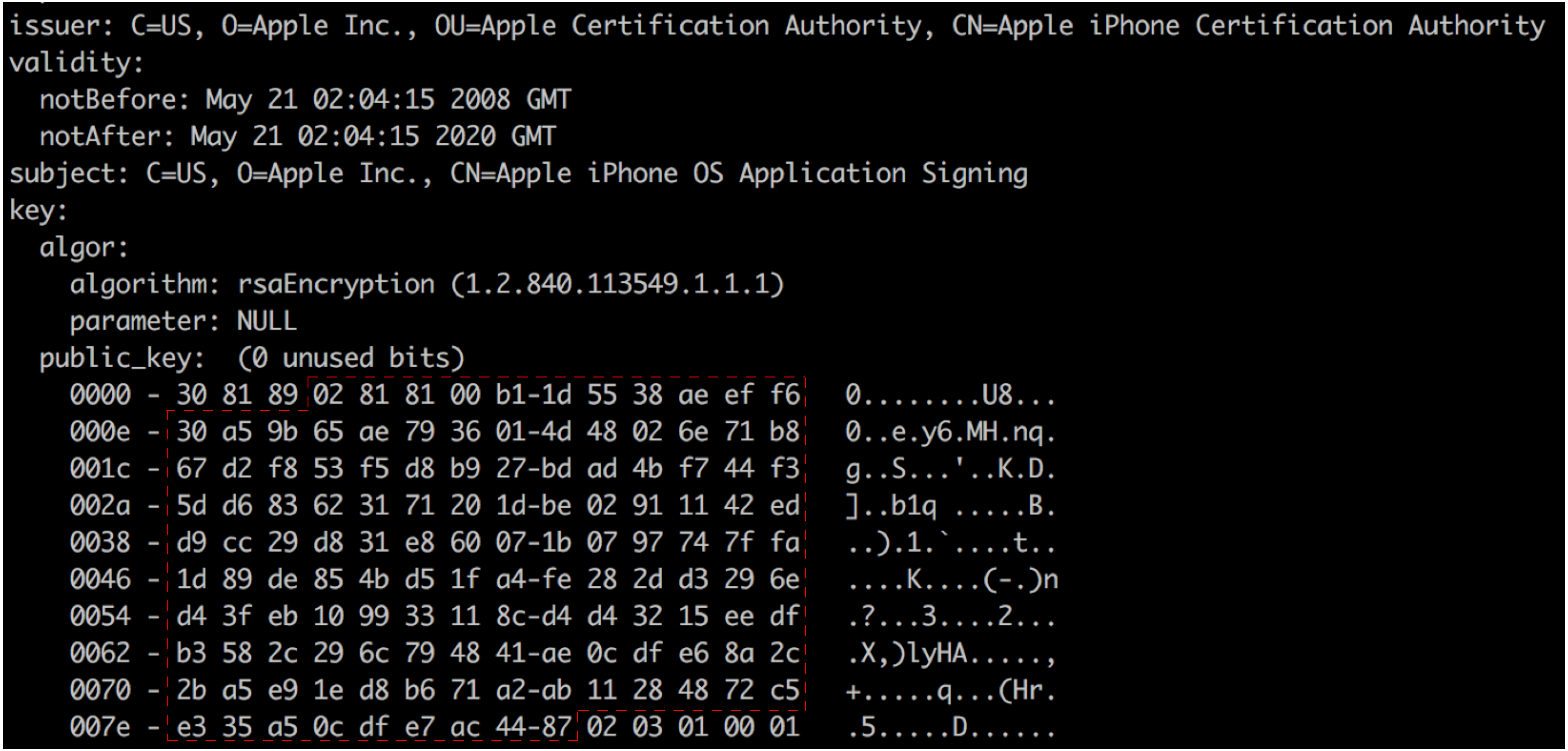
The most obvious and visible of these is the “manifest package name,” the unique name you give to your application in its AndroidManifest.xml. The name uses a Java-language-style naming convention, with Internet domain

Apple code signing

iOS code signing

- AppStore identities (for uploading to AppStore)
- Enterprise code signing (AdHoc and In House)
- In House key can be used to resign some existing apps

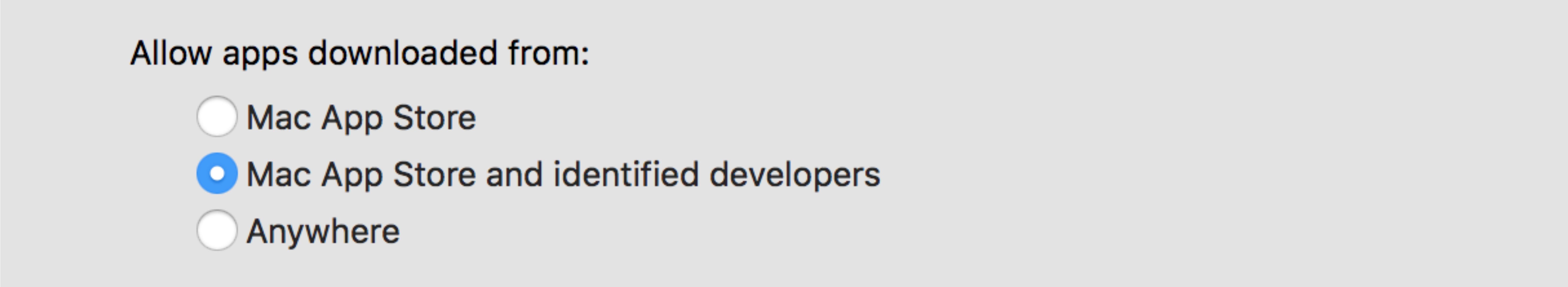
AppStore code signing



```
issuer: C=US, O=Apple Inc., OU=Apple Certification Authority, CN=Apple iPhone Certification Authority
validity:
  notBefore: May 21 02:04:15 2008 GMT
  notAfter: May 21 02:04:15 2020 GMT
subject: C=US, O=Apple Inc., CN=Apple iPhone OS Application Signing
key:
  algor:
    algorithm: rsaEncryption (1.2.840.113549.1.1.1)
    parameter: NULL
  public_key:  (0 unused bits)
    0000 - 30 81 89 02 81 81 00 b1-1d 55 38 ae ef f6   0........U8...
    000e - 30 a5 9b 65 ae 79 36 01-4d 48 02 6e 71 b8   0..e.y6.MH.nq.
    001c - 67 d2 f8 53 f5 d8 b9 27-bd ad 4b f7 44 f3   g..S...'..K.D.
    002a - 5d d6 83 62 31 71 20 1d-be 02 91 11 42 ed   ]..b1q .....B.
    0038 - d9 cc 29 d8 31 e8 60 07-1b 07 97 74 7f fa   ..).1.`....t..
    0046 - 1d 89 de 85 4b d5 1f a4-fe 28 2d d3 29 6e   ....K....(-.)n
    0054 - d4 3f eb 10 99 33 11 8c-d4 d4 32 15 ee df   .?...3....2...
    0062 - b3 58 2c 29 6c 79 48 41-ae 0c df e6 8a 2c   .X,)lyHA.....,
    0070 - 2b a5 e9 1e d8 b6 71 a2-ab 11 28 48 72 c5   +.....q...(Hr.
    007e - e3 35 a5 0c df e7 ac 44-87 02 03 01 00 01   .5.....D......
```

Mac OS Code signing

- Kernel Extensions (kexts) in OS X must be signed
- Gatekeeper (though there are other ways to bypass it)



Allow apps downloaded from:

○ Mac App Store
● Mac App Store and identified developers
○ Anywhere

Apple Code Signing

- LC_CODE_SIGNATURE section in Mach-O
- SHA-1 and RSA
- CodeDirectory - hashes of code chunks
- Requirements - additional rules for signature verification (csreq)
- http://bit.ly/2awvCfz

Android code signing

Android Code signing

- Shows that apps are from the same author
- Used to establish trust relationships among apps
- Android doesn’t use PKI for code signing

Trust among applications

- Permissions can be declared with «signature» protection level
- Shared user id
- Some apps check hash of other apps certificates to decide to trust them or not

Android certificate change

- Available since Android 5.0
- Requires an update of all your apps at the same time

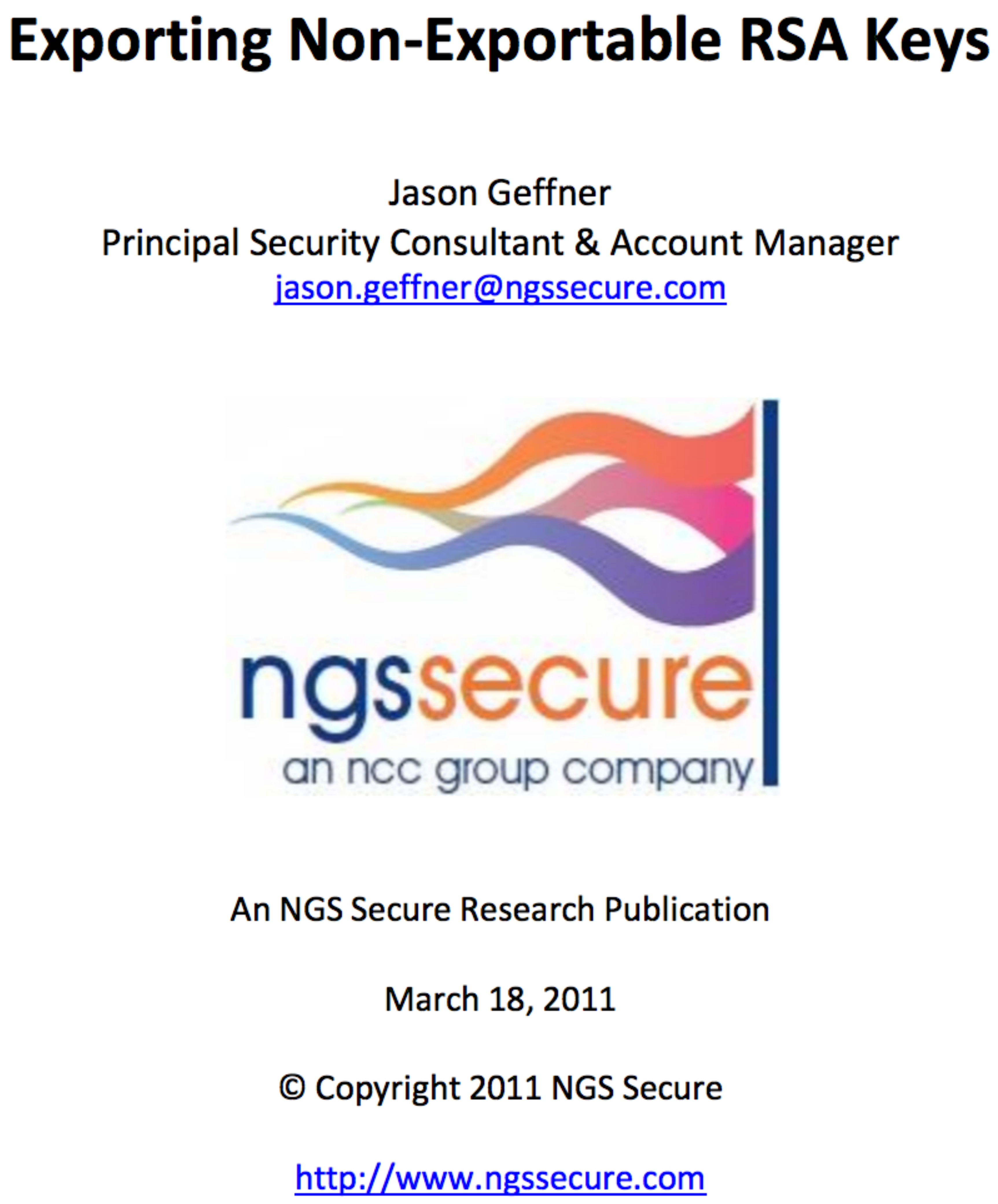
MS Authenticode

MS Authenticode

- Used to establish reputation in SmartScreen
- Used in UAC (User Account Control)
- The same key can sign drivers (up to Win 8.1, UMDF drivers in Win10 if MS Cross Cert added)

MS Authenticode

- Easy to steal from build agents (using mimikatz for example)
- Signing malware drops key reputation from Smart Screen



Exporting Non-Exportable RSA Keys

Jason Geffner
Principal Security Consultant & Account Manager
jason.geffner@ngssecure.com

An NGS Secure Research Publication

March 18, 2011

© Copyright 2011 NGS Secure

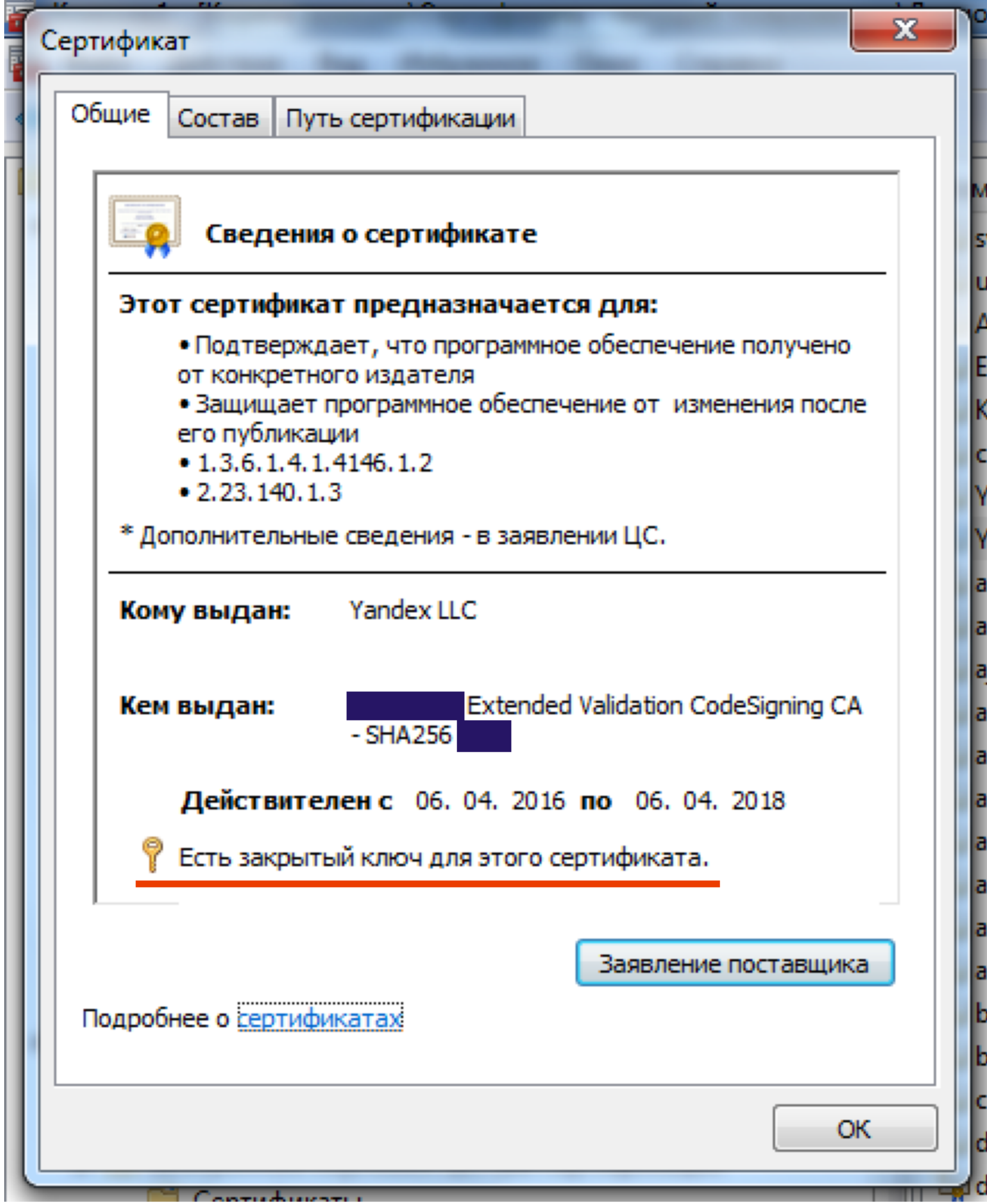
http://www.ngssecure.com

EV Certificates

- Stored inside hardware token
- Never leave hardware device
- Can’t be stolen by malware
- Trusted by default by SmartScreen

EV Certificates

- Bad issuing process - private key was left in the system
- Marked as «non exportable» though can be stolen by malware or insiders



Сведения о сертификате

Этот сертификат предназначается для:

- Подтверждает, что программное обеспечение получено от конкретного издателя
- Защищает программное обеспечение от изменения после его публикации
- 1.3.6.1.4.1.4146.1.2
- 2.23.140.1.3

* Дополнительные сведения - в заявлении ЦС.

Кому выдан: Yandex LLC

Кем выдан: [redacted] Extended Validation CodeSigning CA - SHA256 [redacted]

Действителен с 06. 04. 2016 **по** 06. 04. 2018

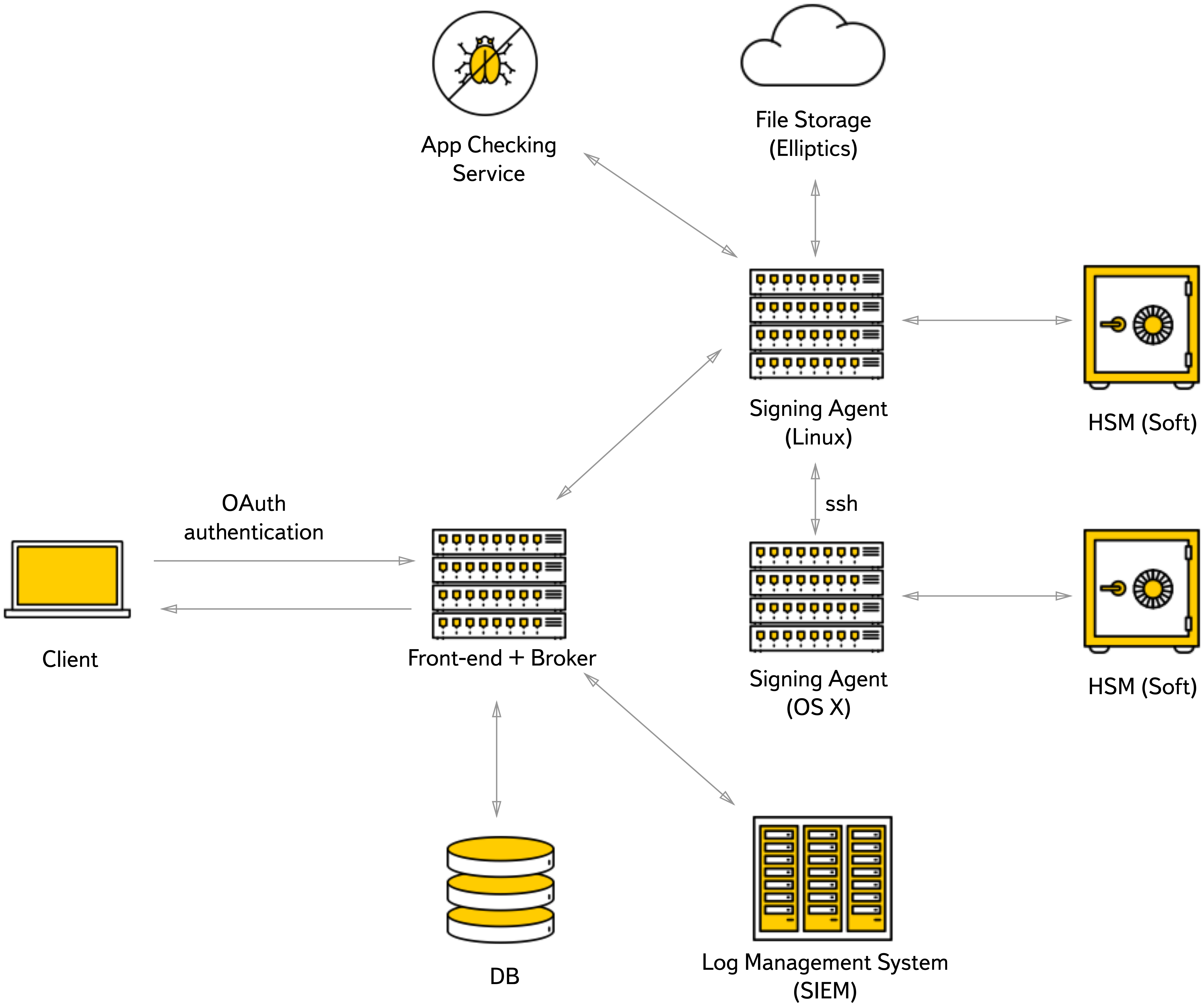
Есть закрытый ключ для этого сертификата.

Подробнее о сертификатах

Code signing as a service

Basic idea: The ПODPNSATOЯ

- Keep all signing identities in one protected place
- Sign binaries on demand
- Manage permission to sign with particular identities


App Checking Service
File Storage (Elliptics)
Signing Agent (Linux)
HSM (Soft)
OAuth authentication
ssh
Client
Front-end + Broker
Signing Agent (OS X)
HSM (Soft)
DB
Log Management System (SIEM)

Signing part implementation

- ‘jarsign’ for Android apps (no surprises)
- ‘osslsigncode’ is used to sign Windows binaries (incl. drivers and MSI packages)
- ‘osslsigncode’ supports ‘dual sign’ including MSI packages
- All ‘provisioning profiles’ for iOS are stored on signing agents

Signing part implementation

- Build process time increases by 2-3 min for Win (30 Gb of Yandex.Browser binaries), a few seconds for other platforms
- Two ‘signing agents’ are enough in the majority of cases
- Sometimes time stamp servers don’t respond - be ready for that

Build process integration

- The fastest way - replace code sign binary with a client script
- A custom plugin for grade to sign APK files
- https://github.com/openbakery/gradle-xcodePlugin to build iOS and Mac apps
- A custom plugin for gradle-xcodePlugin iOS and Mac apps build system

Pros

- Signing identities can’t be stolen
- ‘Malware signing’ case can be easily investigated
- 3rd party & former employees don’t have access to signing identities
- Developers can’t avoid signing, the service can be turned into a security checkpoint

Cons

- Malware still can be signed if OAuth token is leaked and an attacker has access to internal network
- Network becomes a bottleneck
- Single point of compromise

Security checkpoint

Security control: general checks

- The files being signed can be briefly analyzed
- Can query AV services before signing
- In doubt the signing process can be turned into manual approve mode
- Security team will be notified and will look into the incident

Security checks: Android apk

- Decompiles APK files and analyzes sources
- Finds custom TrustManager classes, ’ALLOW_ALL_HOSTNAME_VERIFIER’ usage
- Analyzes AndroidManifest.xml (unprotected custom permissions, exported activities, content providers etc.)
- Finds Reflection usage, PendingIntents etc.
- Any other checks you want…

Security checks: iOS apps

- List of URLs missing ‘https’ schema
- Keys, passwords in Info.plist etc.
- ‘Application Transport Security’ settings
- Apple’s ‘Private API’ usage
- …

Other ideas

- Use commits signing for the most sensitive apps (drivers etc.)
- Data transmitted over network can be reduced for windows apps - but we lose the ability to analyse the code
- We need one open source lib to sign them all

Conclusions

Code signing as a service

- Helps to keep identities secret
- Prevents private key leakage
- Can be turned into a security checkpoint for mobile and desktop apps

Thank you!
Questions?

Contacts

Evgeny Sidorov (e-sidorov@yandex-team.ru) @SidorovEvgenij

Eldar Zaitov (ezaitov@yandex-team.ru) @kyprizel