RSAConference2016

San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

SESSION ID: HT-F01

Breaking Closed Systems with Code-Signing and Mitigation Techniques

Gavin Hill

Director of Threat Intelligence
Venafi

#RSAC

Learning Objectives

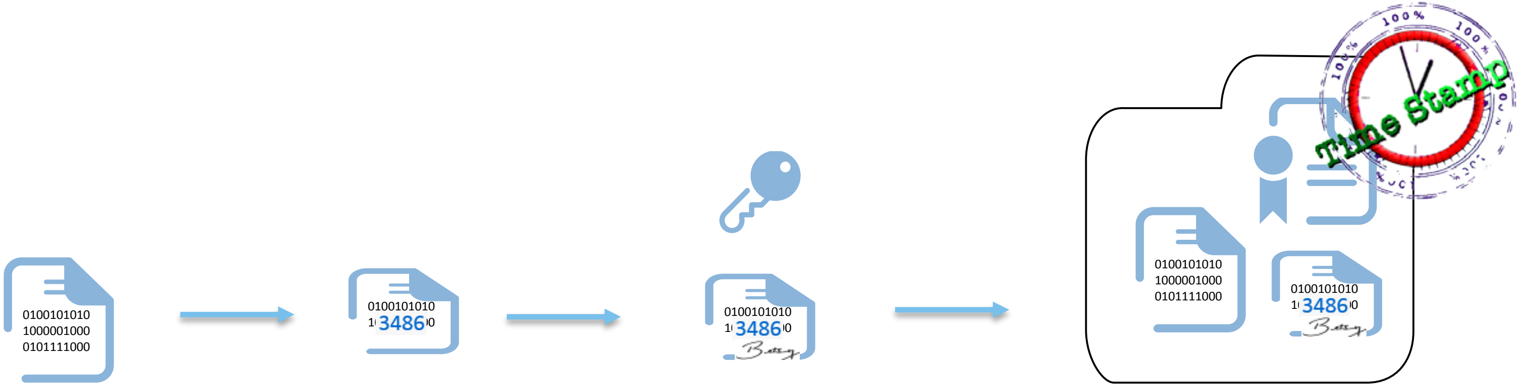
- Code Signing Overview
  - Common use cases (today & tomorrow)
  - Comparing open systems with closed systems
- Threat Landscape
  - Underground market (Theft & Services)
  - Bypassing security controls
  - The Carbon problem
- Mitigating Code Signing abuse

Why Code Signing?

Can I trust the code?

Has the code been tampered with since it was signed?

Code Signing Process



Hash of code created with hashing algorithm

Private key used to sign hash

Package bundled together with certificate

Common Use Cases

App Publishing

Google play

Apple Store

Software distribution

Container Security

Software upgrades

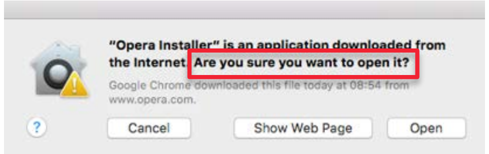
File distribution

Execution of scripts
- Start / Stop services
- Deploy code

Open Systems

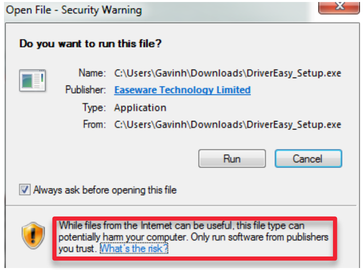
- ✓ Software issuers are trusted by default with a vetting process
- ✓ Users are given the choice to trust a publisher or not

Mac OS



"Opera Installer" is an application downloaded from the Internet. Are you sure you want to open it?

Google Chrome downloaded this file today at 08:54 from www.opera.com.



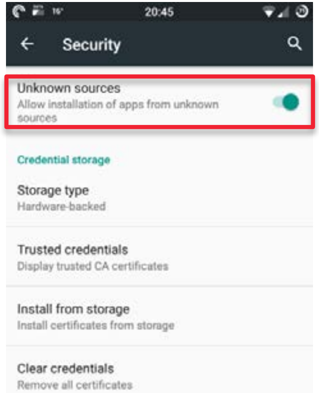
Open File - Security Warning

Do you want to run this file?

Name: C:\Users\Gavinh\Downloads\DriverEasy_Setup.exe
Publisher: Easeware Technology Limited
Type: Application
From: C:\Users\Gavinh\Downloads\DriverEasy_Setup.exe

☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?



Unknown sources ●
Allow installation of apps from unknown sources

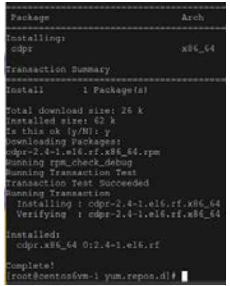
Credential storage

Storage type
Hardware-backed

Trusted credentials
Display trusted CA certificates

Install from storage
Install certificates from storage

Clear credentials
Remove all certificates



```
 Package                                    Arch
Installing:
 cdpr                                       x86_64

Transaction Summary
Install       1 Package(s)

Total download size: 26 k
Installed size: 62 k
Is this ok [y/N]: y
Downloading Packages:
cdpr-2.4-1.el6.rf.x86_64.rpm
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : cdpr-2.4-1.el6.rf.x86_64
  Verifying  : cdpr-2.4-1.el6.rf.x86_64

Installed:
  cdpr.x86_64 0:2.4-1.el6.rf

Complete!
[root@centos6vm-1 yum.repos.d]#
```

Certificate automatically accepted without user warning

Closed Systems

- ✓ **Publisher certs are not trusted, only manufacturer**
- ✓ **Doesn’t provide ways to sideload apps**

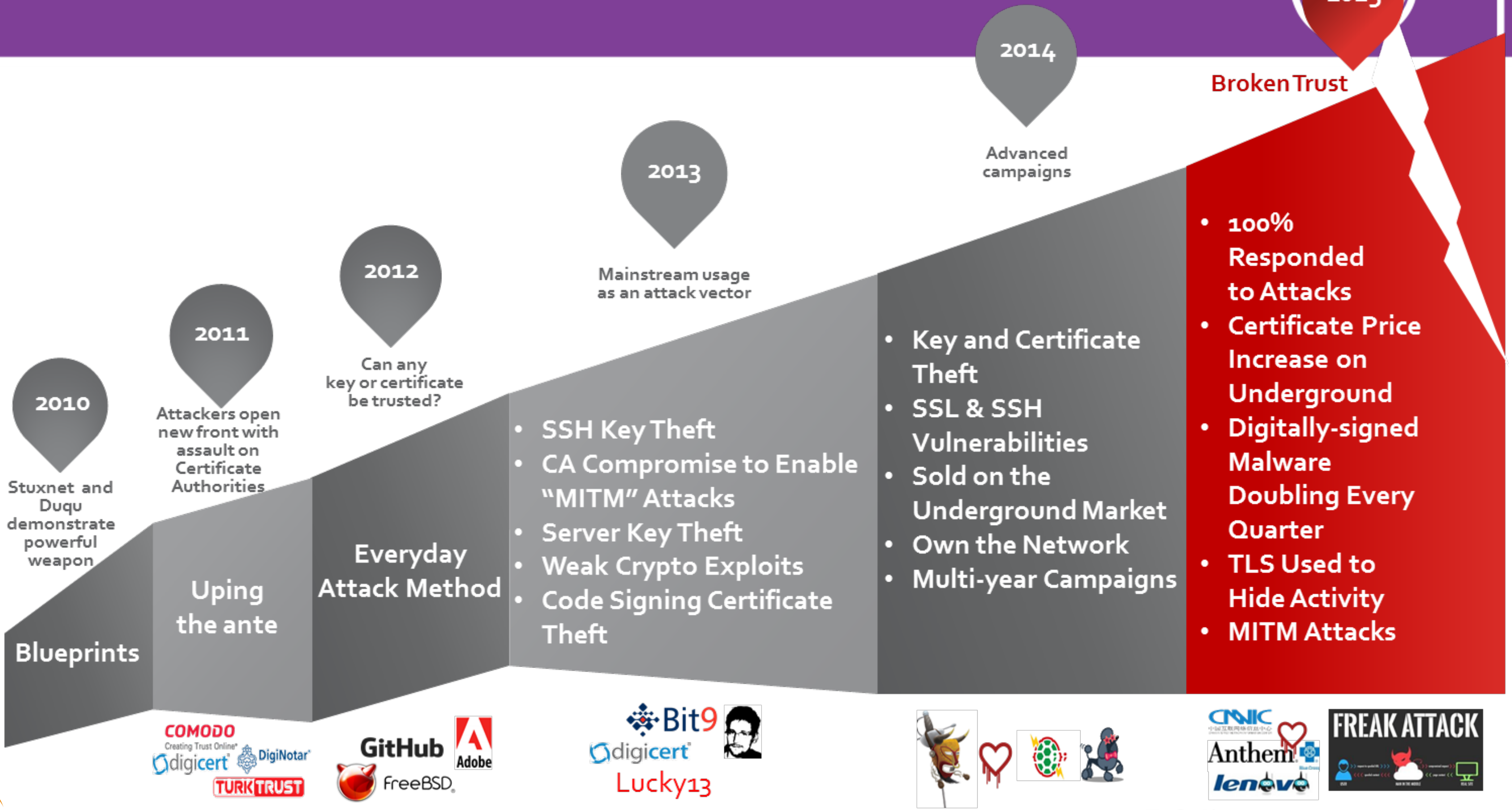
Legally DMCA prohibits breaking any signature schema

Hackers do it anyway!

- Tesla hack -> Weak encryption
- GM/Chrysler -> Firmware vulnerabilities to bypass validation
- iOS -> Buffer overflow to root / jailbreak devices
- Weak hashing or key length

Threat Landscape

Rise of Attacks on Trust



2010 – Stuxnet and Duqu demonstrate powerful weapon

Blueprints

2011 – Attackers open new front with assault on Certificate Authorities

Uping the ante

2012 – Can any key or certificate be trusted?

Everyday Attack Method

2013 – Mainstream usage as an attack vector

- SSH Key Theft
- CA Compromise to Enable "MITM" Attacks
- Server Key Theft
- Weak Crypto Exploits
- Code Signing Certificate Theft

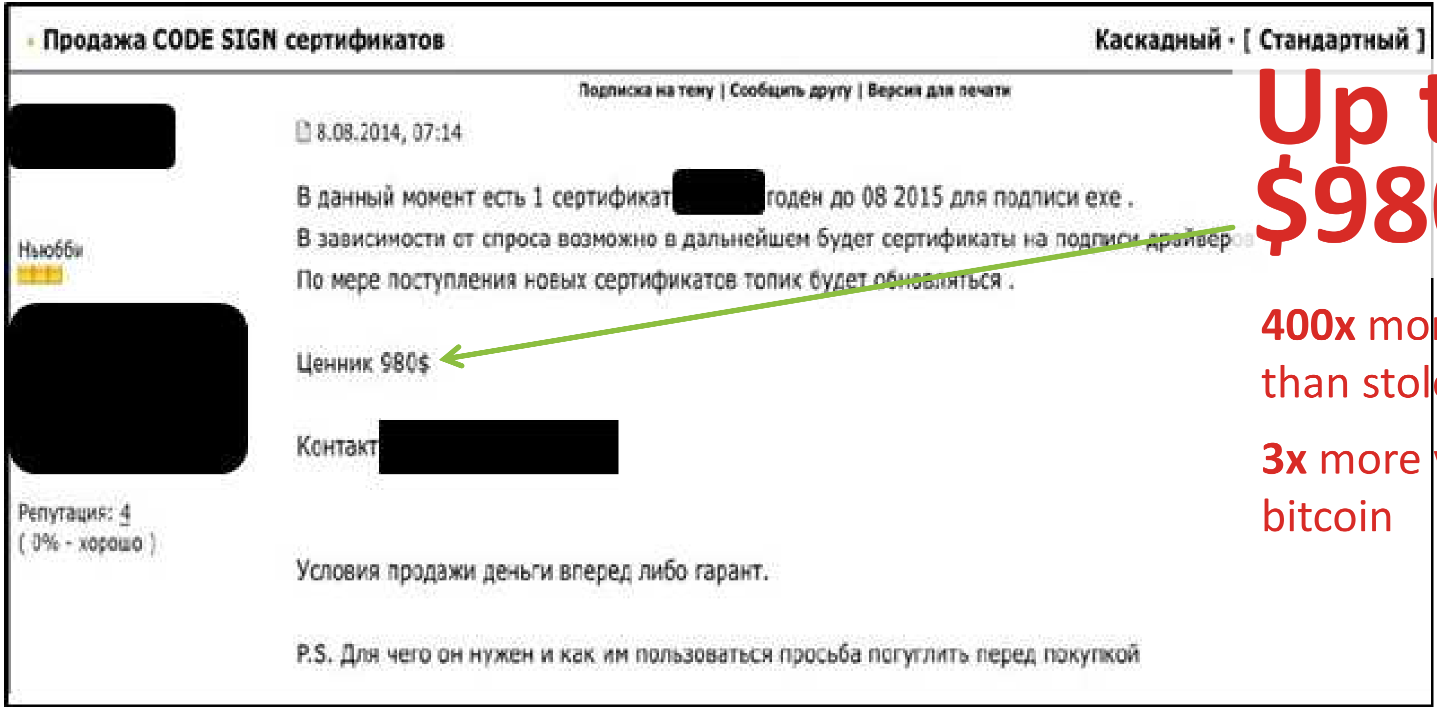
2014 – Advanced campaigns

- Key and Certificate Theft
- SSL & SSH Vulnerabilities
- Sold on the Underground Market
- Own the Network
- Multi-year Campaigns

2015 – Broken Trust

- 100% Responded to Attacks
- Certificate Price Increase on Underground
- Digitally-signed Malware Doubling Every Quarter
- TLS Used to Hide Activity
- MITM Attacks

Marketplace for Stolen Certificates



Продажа CODE SIGN сертификатов

Каскадный · [Стандартный]

Подписка на тему | Сообщить другу | Версия для печати

[redacted]

Ньюбби

[redacted]

Репутация: 4
(0% - хорошо)

8.08.2014, 07:14

В данный момент есть 1 сертификат [redacted] годен до 08 2015 для подписи exe .
В зависимости от спроса возможно в дальнейшем будет сертификаты на подписи драйвер
По мере поступления новых сертификатов топик будет обновляться .

Ценник 980$

Контакт [redacted]

Условия продажи деньги вперед либо гарант.

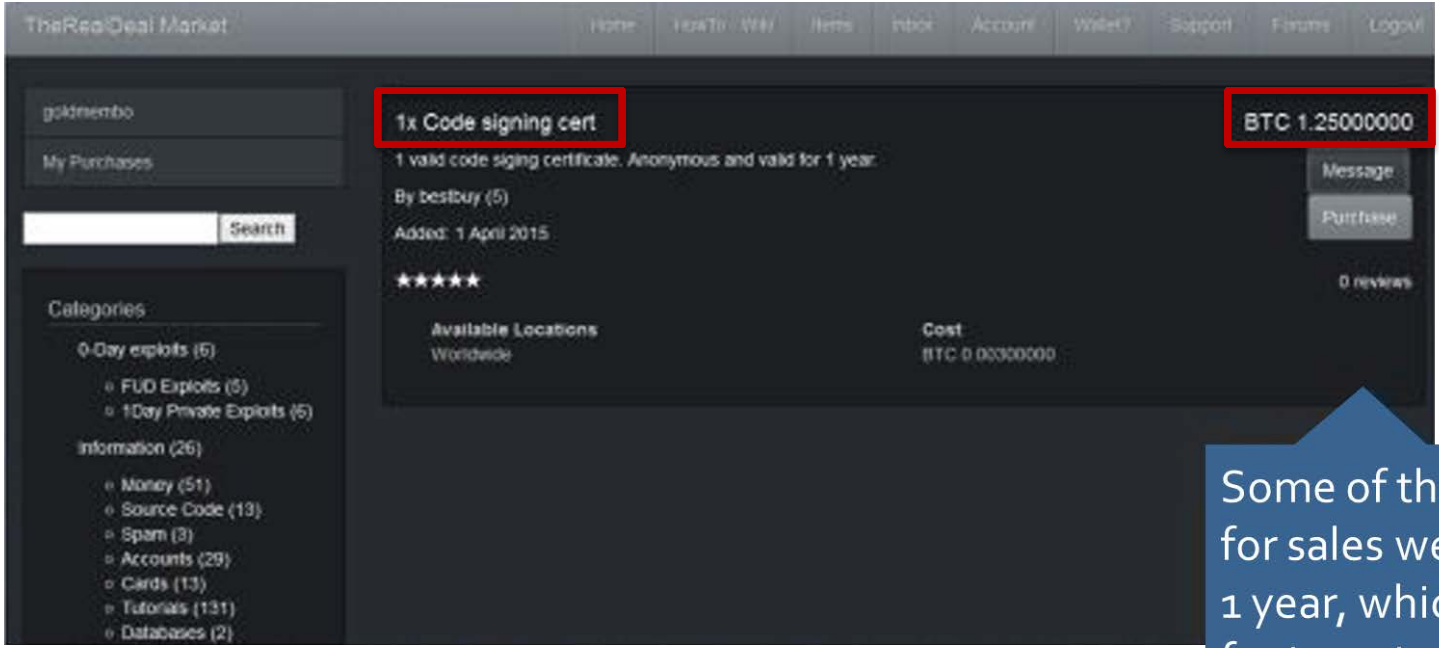
P.S. Для чего он нужен и как им пользоваться просьба погуглить перед покупкой

Up to $980/ea

400x more valuable than stolen credit card

3x more valuable than bitcoin

Underground Certificates-as-a-service (CaaS)



Categories

- 0-Day exploits (6)
  - FUD Exploits (5)
  - 1Day Private Exploits (6)
- Information (26)
  - Money (51)
  - Source Code (13)
  - Spam (3)
  - Accounts (29)
  - Cards (13)
  - Tutorials (131)
  - Databases (2)

1x Code signing cert

BTC 1.25000000

1 valid code siging certificate. Anonymous and valid for 1 year.

By bestbuy (5)

Added: 1 April 2015

★★★★★

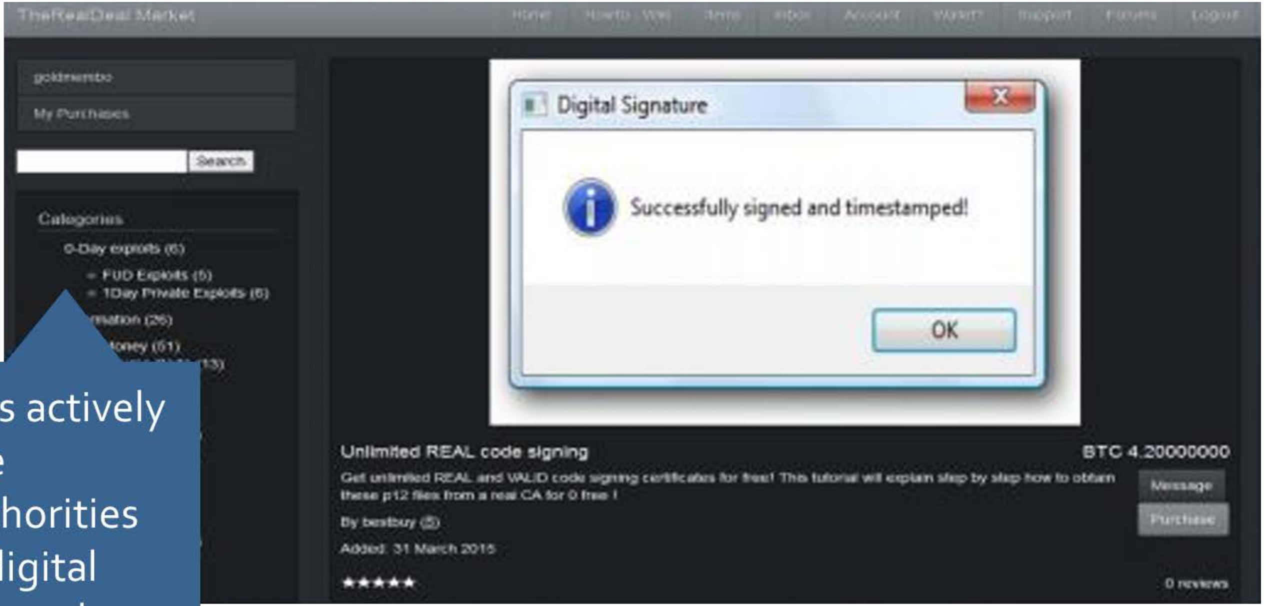
0 reviews

Available Locations	Cost
Worldwide	BTC 0.00300000

Some of the certificates for sales were issued for 1 year, which is enough for targeted APT

InfoArmor: GovRAT

Underground Certificates-as-a-service (CaaS)



Successfully signed and timestamped!

Unlimited REAL code signing

BTC 4.20000000

Get unlimited REAL and VALID code signing certificates for free! This tutorial will explain step by step how to obtain these p12 files from a real CA for 0 free !

By bestbuy (5)

Added: 31 March 2015

★★★★★

0 reviews

The bad actors actively use legitimate certificate authorities (CA) to issue digital certificates for malware

InfoArmor: GovRAT

Blind Trust in Signed Code

Domain Validated (DV) Certificate

- Easily acquired
- Inexpensive or free
- Very little validation performed

Extended Validation (EV) Certificate

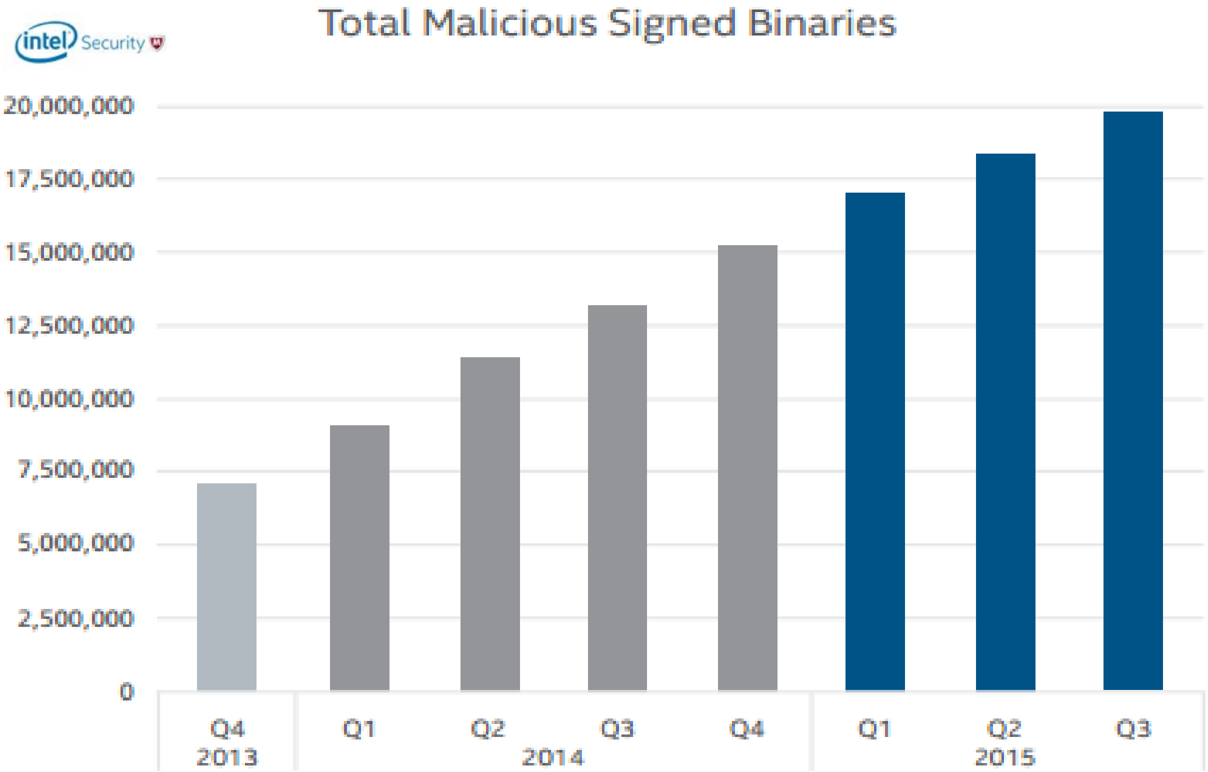
- Rigorous process to acquire
- Expensive
- Extensive validation

"Programs signed by an EV code signing certificate can immediately establish reputati[illegible] reputation [illegible] prior reputation exists for [illegible] publisher." Microsoft

Are we setting ourselves up for failure?

Ref: https://blogs.msdn.microsoft.com/ie/2012/08/14/microsoft-smartscreen-extended-validation-ev-code-signing-certificates/)

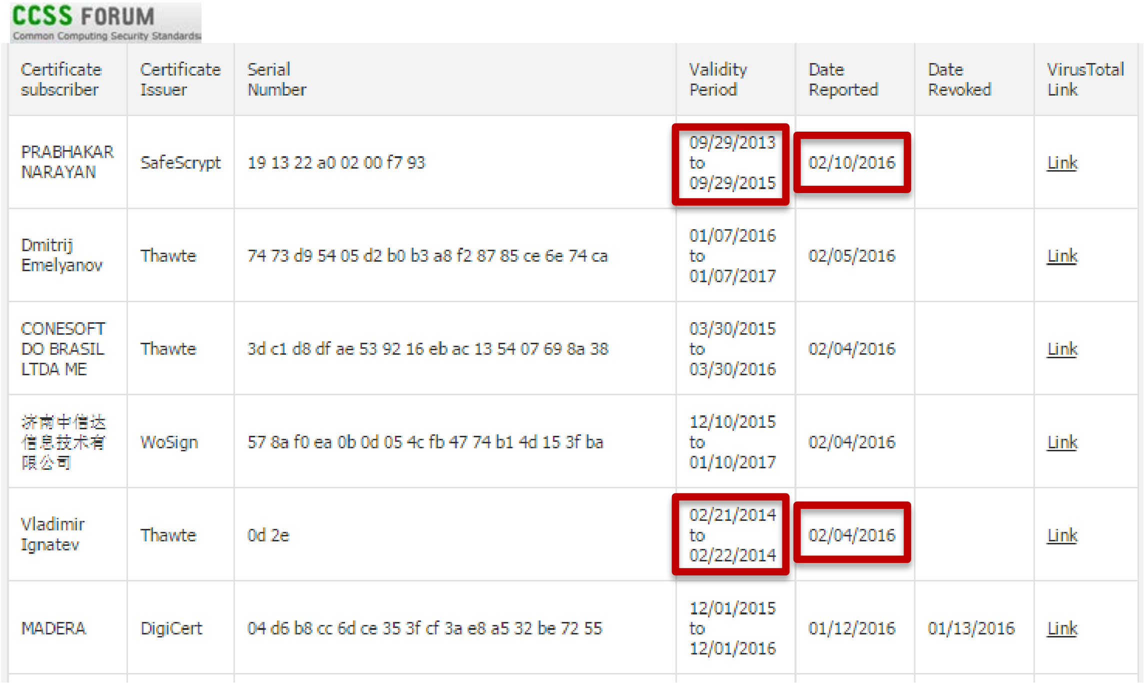
Signed-Malware Continues to Increase



The Ugly Truth – Revocation Doesn’t Work

- Oct 1, 2015 -> Sign malware with stolen code signing certificate with timestamp Oct 1, 2015
- Nov 1, 2015 -> Code signing certificate revoked
  - Malware can’t run on systems that check CRL
- Dec 31, 2015 -> Code signing certificate expires and is removed from CRL
- Jan 1, 2016 -> malware runs again as trusted on systems

Signed Malware



CCSS FORUM
Common Computing Security Standards

Certificate subscriber	Certificate Issuer	Serial Number	Validity Period	Date Reported	Date Revoked	VirusTotal Link
PRABHAKAR NARAYAN	SafeScrypt	19 13 22 a0 02 00 f7 93	09/29/2013 to 09/29/2015	02/10/2016		Link
Dmitrij Emelyanov	Thawte	74 73 d9 54 05 d2 b0 b3 a8 f2 87 85 ce 6e 74 ca	01/07/2016 to 01/07/2017	02/05/2016		Link
CONESOFT DO BRASIL LTDA ME	Thawte	3d c1 d8 df ae 53 92 16 eb ac 13 54 07 69 8a 38	03/30/2015 to 03/30/2016	02/04/2016		Link
济南中信达信息技术有限公司	WoSign	57 8a f0 ea 0b 0d 05 4c fb 47 74 b1 4d 15 3f ba	12/10/2015 to 01/10/2017	02/04/2016		Link
Vladimir Ignatev	Thawte	0d 2e	02/21/2014 to 02/22/2014	02/04/2016		Link
MADERA	DigiCert	04 d6 b8 cc 6d ce 35 3f cf 3a e8 a5 32 be 72 55	12/01/2015 to 12/01/2016	01/12/2016	01/13/2016	Link

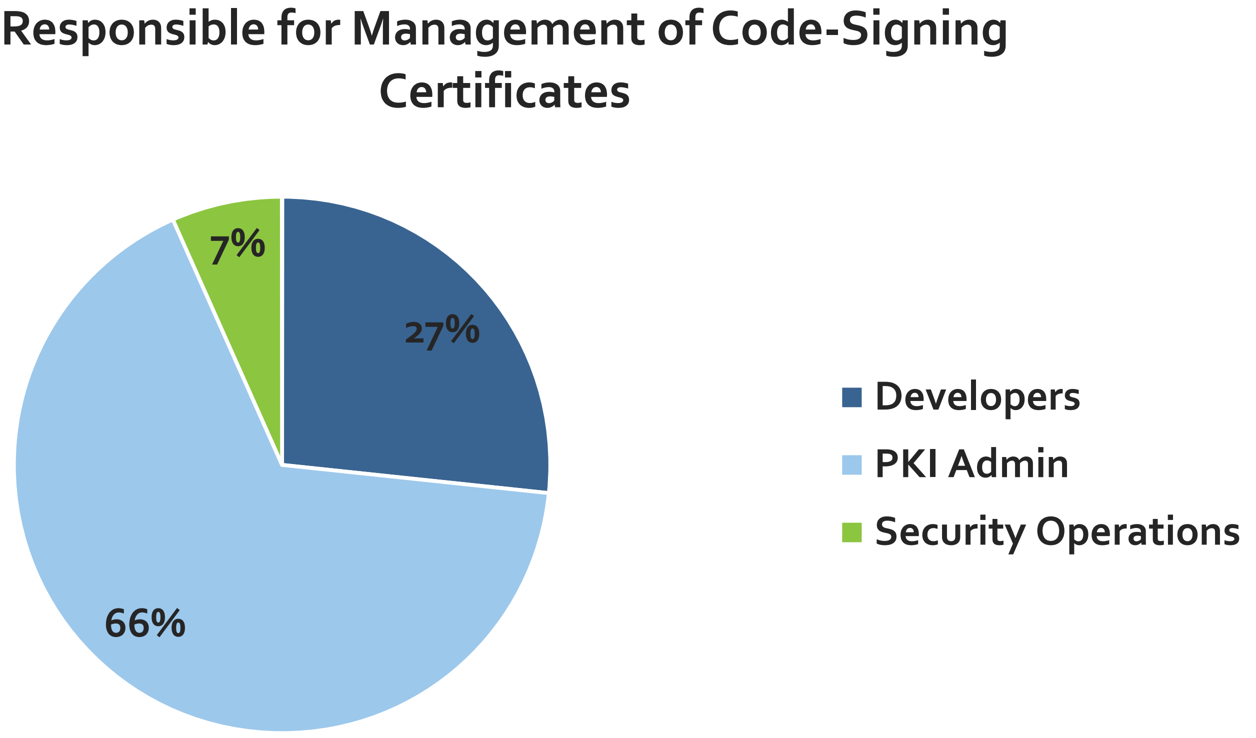
Note the expiration date of the certificates used to sign the malware and when it was discovered

The Carbon Problem

Bypassing Security Controls

Year	Organization	Attack	Source
2012	Adobe	Compromised code signing server used to sign malware	Compromised code signing server
2013	Bit9	Stolen code-signing certificate used to sign malware	Stolen from developer machine
2014	HP	Stolen code-signing certificate used to sign malware	Stolen from developer machine
2015	Dell	Sign fake certificates for MITM attacks or malicious code	eDellRoot self-signed CA installed on all new Dell machines*
2016	SBO Invest	multiple code signing certificates used to sign Spymel	Stolen code signing certificates

Who’s Responsible for Protecting the Keys?



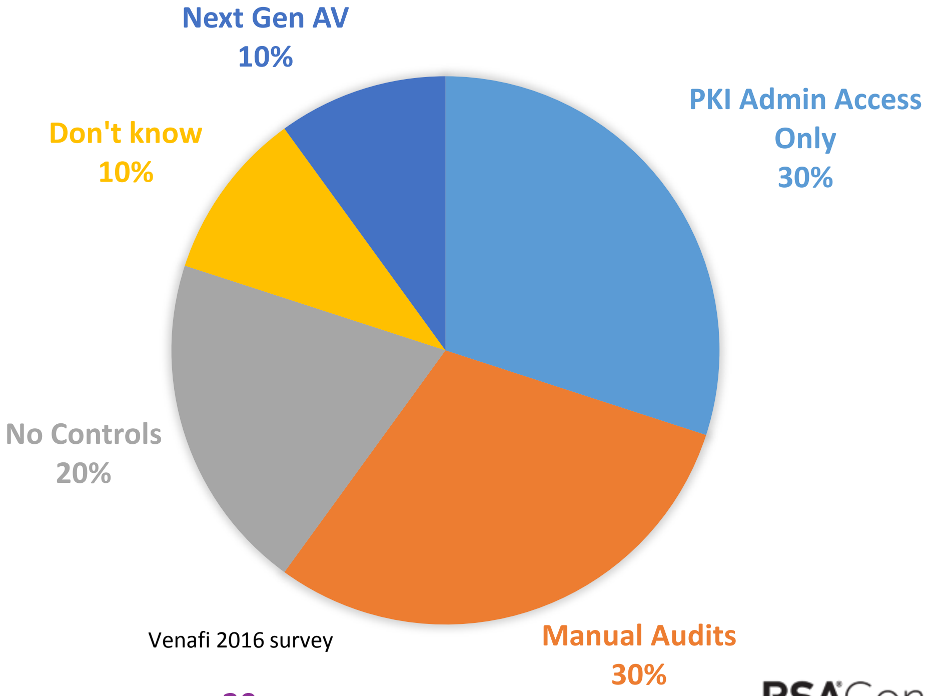
Venafi 2016 survey

Protecting Against a Compromise

At least

70% don’t have effective controls in place

CONTROLS IN PLACE TO ENSURE CODE-SIGNING PROGRAM IS NOT AT RISK OF A COMPROMISE



Venafi 2016 survey

The Problems with Closed Systems

- Not using signatures at all to validate updates (Automotive, Embedded Devices).
- Signing Keys/Certificates are blindly trusted and can't be revoked in case of CA/key compromise (IoT).
- Closed System CAs are not subjected to the usual public CAs security audits (WebTrust only has an audit criteria for EV Code Signing issuing CA).

How Do Attacks on Closed Systems Happen

- Exploiting the code signing process.
- Exploiting the update/upgrade process:
  - MITM attacks when updates are retrieved (either exploit TLS connection validation issues in existing client libraries)
  - Exploit signature validation vulnerabilities during manual update process
- Exploit another vulnerability in the firmware to get access to the device and then use the upgrade/update path to gain further access.

3 Suggested Steps To Mitigating Code Signing Abuse

Mitigating Code Signing Abuse – Step 1

- Find out what signed code you have
- Find out who is performing the code-signing in your organization
- Find out where code-signing certificates are stored and who has access to them

Mitigating Code Signing Abuse – Step 2

- Start publishing code-signing usage
- Require CAs to publish code signing certificate issuance

Mitigating Code Signing Abuse – Step 3

- Establish security controls to limit access to code signing certificates
- Identify any misuse or irregularities for code signing practices within your organization
- Validate:
  - ✓ What code is being signed
  - ✓ Who is signing it
  - ✓ Where it is being signed
  - ✓ When it was signed

Gartner

“Certificates can no longer be **blindly trusted.”**

Questions?