presented by

Building Secure Firmware with HardwareSecurityModules(HSM)

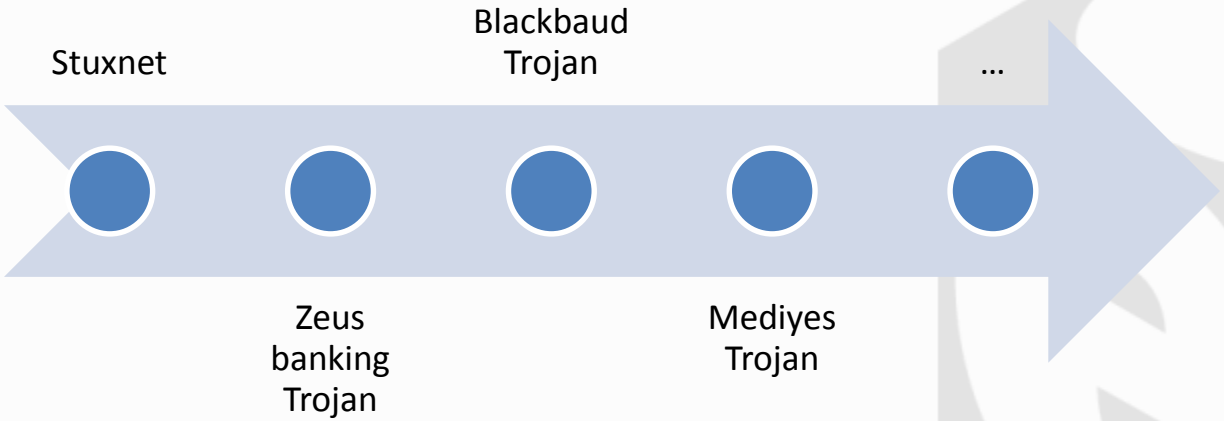
UEFI Summerfest – July 15-19, 2013

Presented by Vishal manan (Microsoft Inc.)

Agenda

- PKI for Secure Boot
- HSM overview
- KPI's for HSM
- Key Generation using HSM
- Good practices for HSM usage
- Questions

Stolen Private keys



Secure Boot relies on PKI

Key/db Name	Variable	Owner	Details
PKpub	PK	OEM	PK – 1 only. Must be RSA 2048 or stronger
Microsoft KEK CA	KEK	Microsoft	Allows updates to db and dbx.
Microsoft Windows Production CA	db	Microsoft	This CA in the Signature Database (db) allows Windows8/Windows Server 2012 to boot
Forbidden Signature Database	dbx	Microsoft	List of known bad Keys, CAs or images from Microsoft

+ Required for Secure Firmware Updates (not mandated by UEFI but by NIST 800-147)

Key/db Name	Owner	Details
Secure firmware update key	OEM	Recommendation is to have this key be different from PK. Must be RSA 2048 or stronger

+ any other (proprietary) keys

HSM Basics

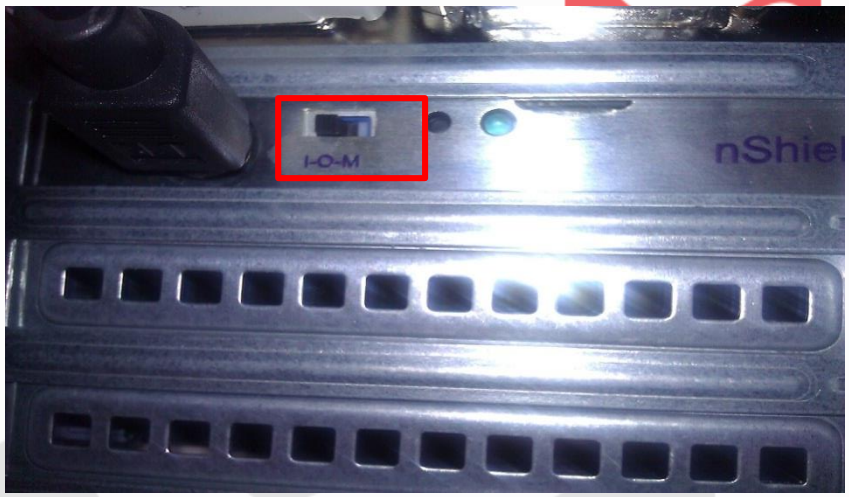
- Hardware device to generate and protect crypto keys
  - The private key never leaves the HSM
    - Either stored on the HSM
      or
    - Encrypted on the HSM
- Role based two factor authentication
- Supports M of N authentication
- Compliance with FIPS 140-2 level 2/3/4
  - Tamper evident or tamper resistant

- Better Performance of crypto operations(offloading servers) with onboard crypto-processors
- **In the world of security every bit counts!!**

HSM Device Overview

3 main components*

1. PCI/USB or Network card –
   - Has a switch to change mode
     - I – Init
     - O – Operation
     - M – Maintenance



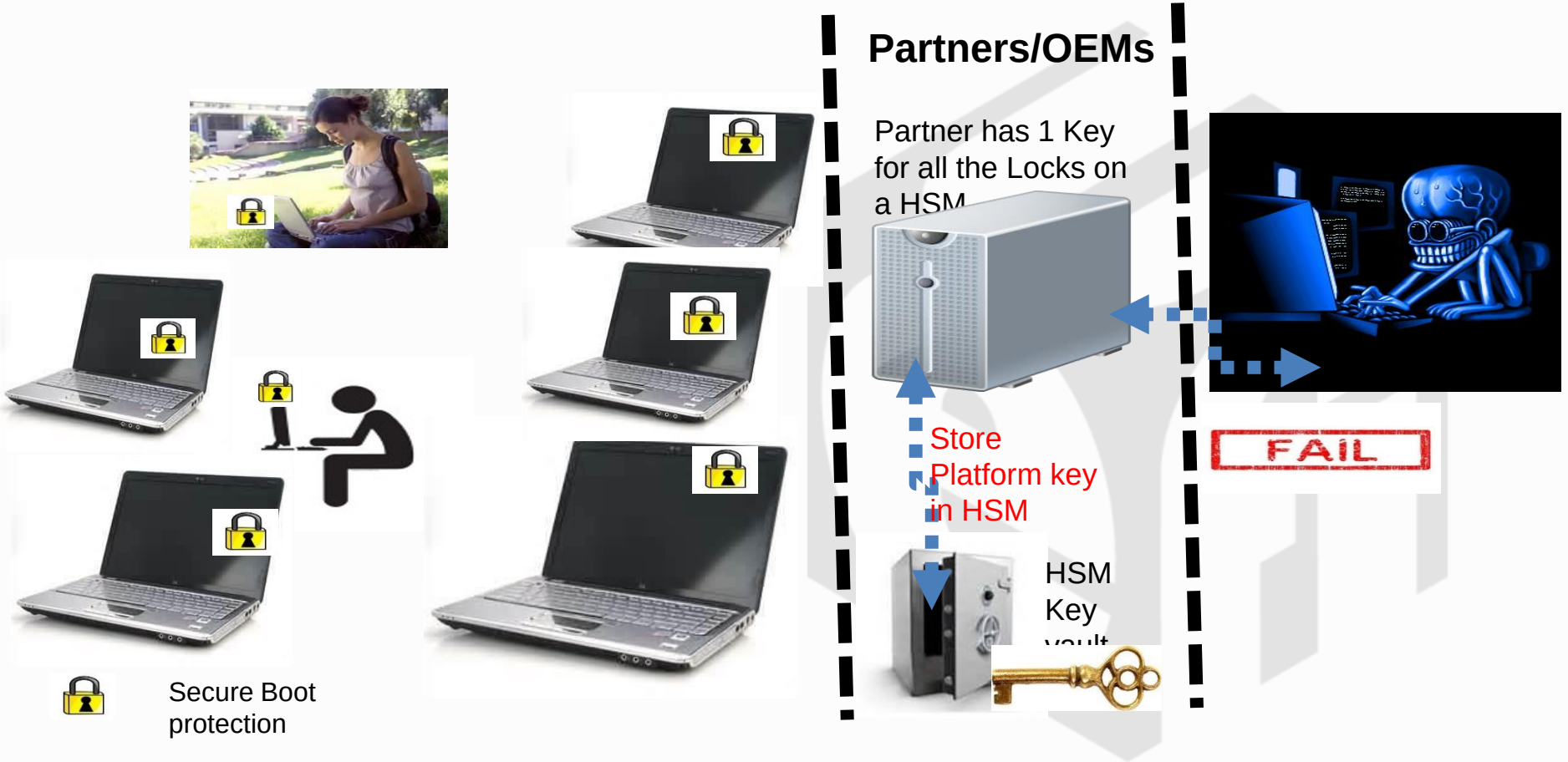
2. Smart Card reader/PED based

3. Smart Cards or USB authentication tokens

**We show the Thales HSM as an Example*

Secure Boot Without HSM – Bad News…

Secure Boot key management with HSM



Using HSM for Generating Certificates

- Leverage inbox certutil.exe
- Certificate attributes:
  - Key algorithm – RSA-2048
  - Hash algorithm – SHA-256
  - Self-signed certificate or derive it off an enterprise CA if you have one
  - Decide on the validity period for the certificate
  - Pick HSM CNG as the CryptographicServiceProvider
    - Need a CSP which can do Microsoft CNG to support SHA-256 hashing algorithm
- Back up the certificate

  As good practice please always backup the certificate you generated.

HSM usage KPI's

- Creates a Security framework and hence Lowers chances of Private Key leak
  - Lower support costs
- Can store keys on the HSM and back it up
  - HSM not as susceptible to data loss as a server
  - Network HSM can allow for High Availability (HA)
- Key deletion and changes require at least n of m people to be present
- Protects against industrial and political espionage
- Can be used for creating CA' s

HSM for key generation

Certificate creation using HSM

Create certificate

```
certreq.exe -new request.inf PK.cer
```

Sample request.inf file may look like:

```
[Version]
Signature= "$Windows NT$"
[NewRequest]
ValidityPeriod = Years
ValidityPeriodUnits = 6
Subject = "CN=Corporation TODO Platform Key,O=TODO Corporation,L=TODO_City,S=TODO_State,C=TODO_Country"
MachineKeySet = true
RequestType=Cert
Exportable =  FALSE
HashAlgorithm = SHA256
KeyAlgorithm = RSA
KeyLength = 2048
KeyContainer = "PKContainer"
ProviderName = "nCipher Security World Key Storage Provider"
KeyUsage = 0xf0
```

Key generation using HSM(contd.)

Validate the certificate

```
certutil -store -v my
"7569d364a2e77b814274c81ae6360ffe"//CERT. Serial #
```

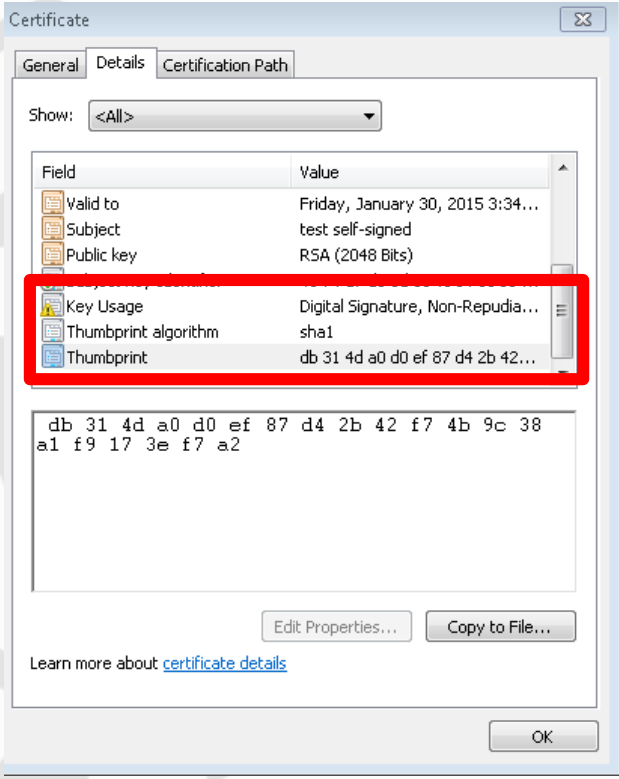
my
================ Certificate 16 ================
X509 Certificate:
Version: 3
Serial Number: 7569d364a2e77b814274c81ae6360ffe

Signing with HSM(contd.)

Available with the Windows SDK and used for signing binaries

Command

```
signtool.exe  sign /v /fd sha256 /sha1
"db314da0d0ef87d42b42f74b9c38a1f9173ef7
a2" /sm /p7  .\ /p7co
1.2.840.113549.1.7.1
/p7ce  DetachedSignedData
<binarytosign>.bin
```



Paramaters specific to generating certificates with HSM

`/sm` – use local certificate store

`/sha1` – Hash of the certificate

Managing keys – using HSM vendor tools

Key Operations

The buttons on this panel enable you to create new keys for a wide range of applications, to list details of existing keys, or to import an external application key.

You can discard an existing key by clicking the List Keys button below, highlighting the appropriate entry, and then clicking the Discard Key button.

Before you can use a module to generate keys, you must have either:

- initialized a security world using the module
- reprogrammed the module with an existing security world.

Use the options in the Module Operations panel to either initialize a security world or reprogram a module. Click the Modules menu button on the sidebar in order to go to the Module Operations panel.

Generate Key — create a new application key

List Keys — list all keys in the current security world

Import Key — import an application key from an outside source

Key Listing

Selecting a key from the list below displays that key's parameters.
You can then click the Remove Key button in order to remove the selected key from your security world, or you can make another selection.

Key List

Key Name	Application	Protection	NVRAM
PK	PKCS#11	Module Protected	No
pk_cert	PKCS#11	Module Protected	No
test_pk11	PKCS#11	Module Protected	No
vmanan	Unknown (`mscapi')	No Key	No

Good practices for HSM usage

- ❑ Read the Whitepaper on Security planning
  http://technet.microsoft.com/en-us/library/cc723503.aspx
  - ❑ Read the HSM vendors User manual
- ❑ Decide on Security roles

Operational Staff	Security Policies
Security Officers	Access Control Rules
Transaction Authorizer	Risk Control Strategies
Key Management	Operational Procedures
IT Administrator	7X24 Availability
Factory floor lead	Contingency plan
Outsourcing Agent (ISP)	Disaster Recovery

- ❑ Chose FIPS 140-2 level 3
- ❑ Pick n and m >1 for n of M authentication

Good practices for HSM usage (contd.)

- Have an AdministerCardSet for the HSM and an OperatorCardSet for Secure Boot
- Use an HSM CSP which supports SHA 256 and Microsoft CNG API such as "nCipher Security World Key Storage Provider“
- Generate Certificate for PK, Secure Firmware update key and optionally other components such as OEM KEK
  - Self Signed or derived from a CA
  - Uses RSA 2048 as encryption algorithm
  - SHA 256 as hash algorithm
  - Decide on validity period
  - Backup the certificate
  - Label the certificate with the model # of the machine
  - Generate new certificates at a regular cadence

Good practices for HSM usage (contd.)

- Setup and use HSM vendor GUI (KeySafe) utility for better key management
- Test generation of certificates in a production environment
- Test sign sample PK.bin/KEK.bin with the private key stored in the HSM and use the HSM as CSP
- Backup the HSM metadata(Security world/partitions…) on multiple sets of media periodically
- Try Restoring a deleted key using backup data
- Make sure Disaster recovery works

Thanks for attending the UEFI Summerfest 2013

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by