

サイバーエスピオナージとTPM

須崎有康 Kuniyasu Suzuki

独立行政法人産業技術総合研究所 (AIST)

セキュアシステム研究部門 (RISEC)



TCG日本支部(JRF)第6回公開セキュリティーワークショップ

3/December/2014

<http://www.trustedcomputinggroup.org/jp/jrfworkshop/workshop6>

Who am I?

独立行政法人 産業技術総合研究所
セキュアシステム研究部門 @茨城県つくば市
のコンピュータセキュリティ研究者



- 最近の仕事 <https://staff.aist.go.jp/k.suzaki/>
 - 仮想化を使ったセキュリティの強化や脆弱性の研究
 - 制御システム用ホワイトリスト制御技術
 - KNOPPIX日本語版のメンテナンス
 - KNOPPIX Trusted Computing Geeks も作りました



アウトライン

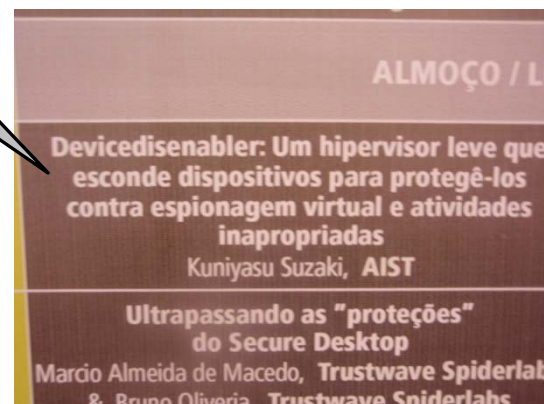
- モバイルガジェットの高性能デバイスを使ったサイバーエスピオナージ(電子的諜報活動)
- デバイスを認識させないハイバーバイザー DeviceDisEanbler
 - TPMによる暗号鍵管理
- DeviceDisEanblerを展開させるための条件

元になった発表

BlackHat Sao Paulo 2014

“DeviceDisEnabler: a lightweight hypervisor which hides devices to protect cyber espionage and tampering”

ポルトガル語のプログラム



モバイルガジェットにどれくらいのデバイスが入っているか知っていますか？

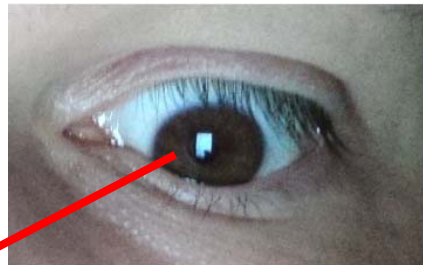
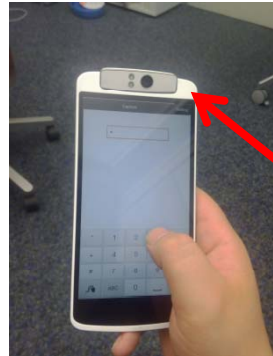
- マイクとスピーカ
- デジタルカメラ
- GPS
- ジャイロスコープ
- その他、多様なセンサー
- **これらのデバイスが入ったのはそれほど昔ではない。**
 - 2000年前後のPDA(携帯情報端末。例 Palm Pilot, Apple Newton)と呼ばれるものにはこのようなデバイスがないものが多かった。初期のiPadにもカメラが無かった。
- **現在のモバイルガジェットは従来のコンピュータと言うより、センサーデバイスの塊になっている。**

デバイスの性能を知っていますか？

- マイク、スピーカ
 - CD クオリティ (44.1kHz)以上
- デジタルカメラ
 - 100M pixel以上
- GPS
 - 10m以内の位置検出
- ジャイロスコープ
 - 20 Hz以上のサンプリング
- 高性能デバイスはサイバーエスピオナージ(諜報活動)の格好のターゲット。

Facial Reflection Keylogger

[T.Fiebig, WOOT'14]



このカメラが顔(目)の写真を撮ります。

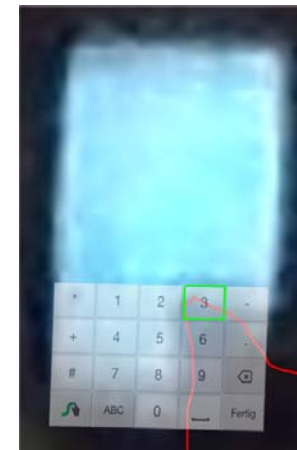
ズーム



親指検出



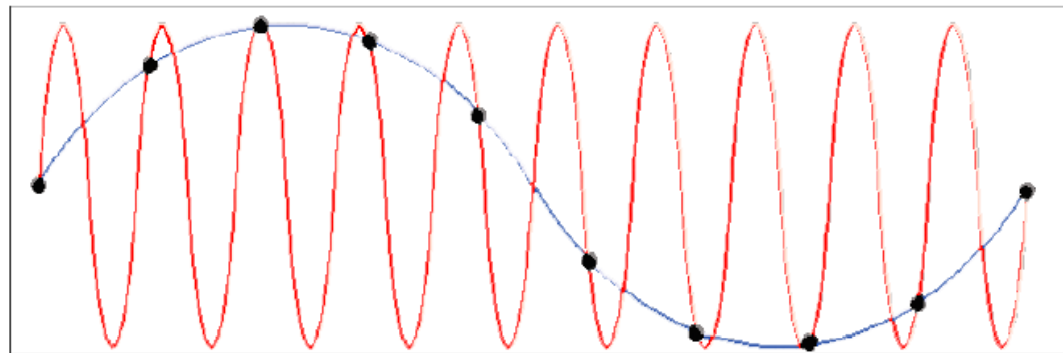
キーボードをマップ



T.fiebig, j.krissler and r.hanesch, "Security Impact of High Resolution Smartphone Cameras" woot 2014.
<https://www.usenix.org/conference/woot14/workshop-program/presentation/fiebig>

ジャイロスコープによる盗聴

- Gyrophone [USENIX Security 14, BlackHat Europe 14] はジャイロスコープで音声の解析ができることを示した。
 - 利点: マイクの使用には許可を取る必要があるが、ジャイロスコープは必要なし。
 - 問題点: ジャイロスコープのサンプリングは 20-200Hz で音声 (male 85 - 180 Hz, female 165 - 255 Hz) が取れない。
 - エイリアシングによって音声解析できることを示した。



モバイルガジェットの活用シナリオ

- モバイルガジェットは重要情報を扱う工場、会議、病院で広く使われるようになった。
- **管理者は仕事で使うモバイルガジェットで不要なデバイスを利用禁止にしたい。**



工場



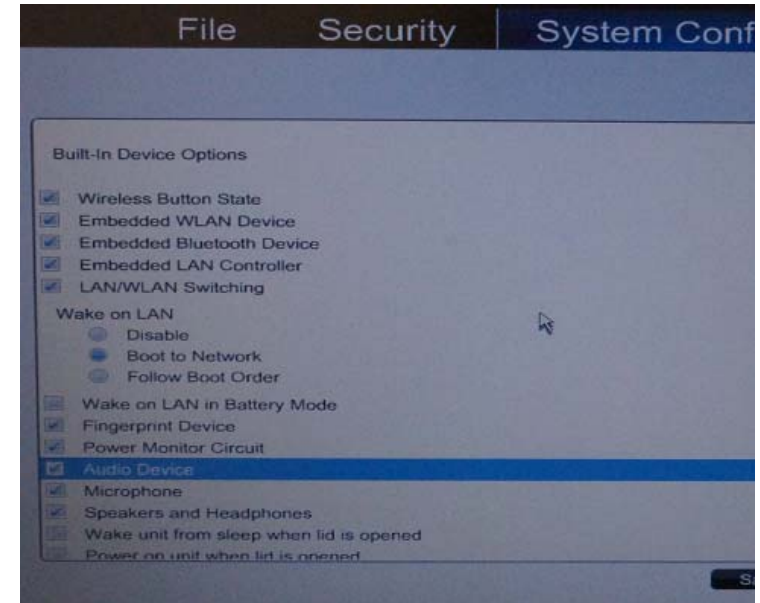
リモートミーティング

その他の脅威

- モバイルガジェットのデバイスは攻撃者ばかりでなく、ユーザ(社員)も使いたい！
- ユーザ(社員)が対応策を回避するかもしれない。
- 管理者は攻撃者ばかりでなく、ユーザも対象として対策技術を考えなくてはならない。

現在の対応策

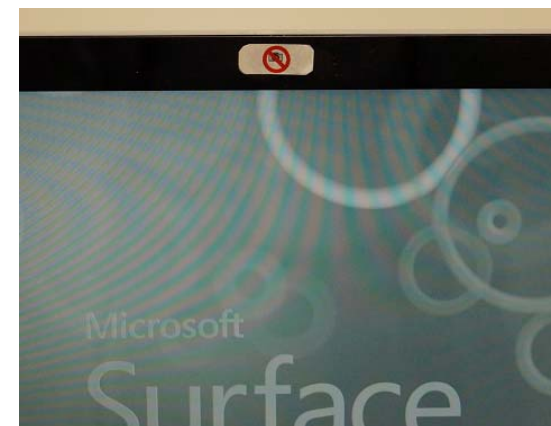
- BIOS/EFI によってデバイスを使用不可にする
 - 有効だが、全てのモバイルガジェットで有効なわけではない。



- セキュリティグッズ
 - プロテクションキャップ
 - セキュリティシール(カメラ用)



これらはユーザの良心に依存。



提案する対策方法

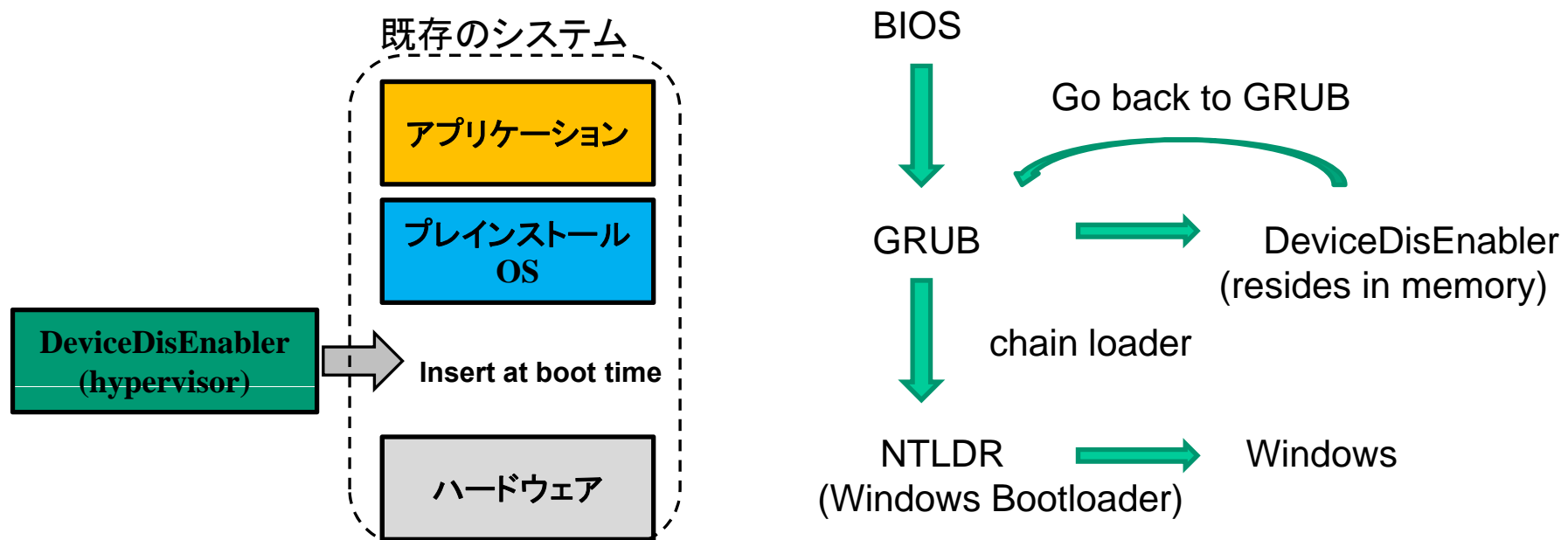
- “*DeviceDisEnabler (DDE)*”: サイバーエスピオナー
ジと改竄を防止する軽量ハイパーバイザー

特徴

1. 多くのモバイルガジェットに適用とするため、軽量で
既存OSに挿入可能なハイパーバイザー
2. OSからデバイスを隠蔽
3. 改竄防止
 - ハードディスクの一部をDDEが暗号化するので、
DDEなしではOSを立ち上げることが出来ない。
 - 暗号鍵はユーザから隠蔽されている。

挿入可能なハイパーバイザー

- 軽量のタイプI (ベアメタル) ハイパーバイザー
 - パラパススルーアーキテクチャ(BitVisor[VEE'09])
 - デバイスマodelが無く、ゲストOSがデバイスに直接アクセスできる
 - 小さなトラステッドコンピューティングベース(TCB)
 - HostOS 無しはTCBを小さくできる
- DDE はブートローダのチェーンロードを使ったOSの起動前に挿入される



(2) PCIデバイスの隠蔽

- モバイルガジェットには多くのPCIデバイスがある
- Tool: PCI-Z
 – <http://www.pci-z.com/>

(ThinkPad Helix)



PCI-Z 1.3 - PCI devices information utility

File Database Report Help

System information
 CPU name: Intel(R) Core(TM) i5-3427U CPU @ 1.80GHz, 4 logical CPUs
 Computer name: LENOVO-PC
 User name: knoppix
 Operating system: Microsoft Windows 8
 Available memory (MB): 3014

Type	Vendor	Device	Subsystem	PCI device
SMBus	Intel Corporation	7 Series/C210 Series Chipset Family SMBus Controller		VEN_8086&DEV_1E22&SUBSYS_220717AA
Serial controller	Intel Corporation	7 Series/C210 Series Chipset Family KT Controller		VEN_8086&DEV_1E3D&SUBSYS_220717AA
PCI bridge	Intel Corporation	7 Series/C210 Series Chipset Family PCI Express Root Port 2		VEN_8086&DEV_1E12&SUBSYS_220717AA
ISA bridge	Intel Corporation	QS77 Express Chipset LPC Controller		VEN_8086&DEV_1E56&SUBSYS_220717AA
USB controller	Intel Corporation	7 Series/C210 Series Chipset Family USB Enhanced Host Controller #1		VEN_8086&DEV_1E26&SUBSYS_220717AA
Host bridge	Intel Corporation	3rd Gen Core processor DRAM Controller		VEN_8086&DEV_0154&SUBSYS_220717AA
VGA compatible con...	Intel Corporation	3rd Gen Core processor Graphics Controller		VEN_8086&DEV_0166&SUBSYS_220717AA
Audio device	Intel Corporation	7 Series/C210 Series Chipset Family High Definition Audio Controller		VEN_8086&DEV_1E20&SUBSYS_220717AA
SATA controller	Intel Corporation	7 Series Chipset Family 6-port SATA Controller [AHCI mode]		VEN_8086&DEV_1E03&SUBSYS_220717AA
PCI bridge	Intel Corporation	7 Series/C210 Series Chipset Family PCI Express Root Port 1		VEN_8086&DEV_1E10&SUBSYS_220717AA
USB controller	Intel Corporation	7 Series/C210 Series Chipset Family USB xHCI Host Controller		VEN_8086&DEV_1E31&SUBSYS_220717AA
USB controller	Intel Corporation	7 Series/C210 Series Chipset Family USB Enhanced Host Controller #2		VEN_8086&DEV_1E2D&SUBSYS_220717AA
Communication cont...	Intel Corporation	7 Series/C210 Series Chipset Family MEI Controller #1		VEN_8086&DEV_1E3A&SUBSYS_220717AA
Network controller	Intel Corporation	Centrino Advanced-N 6205 [Taylor Peak]		VEN_8086&DEV_0085&SUBSYS_C2208086

Database: The PCI ID Repository | Version: 2014.11.15 | http://www.pci-id.org/ Right click on the list for options.

OSがPCI上のデバイスを認識する仕組み

- OS がPCIバス上のデバイスを認識するには、“**PCI configuration Register**”の情報を使う。
 - これにはベンダーID, デバイスID, デバイスクラス, メモリマップアドレスなどの情報含む。
 - ベンダーID はPCI-SIGによって定義されている

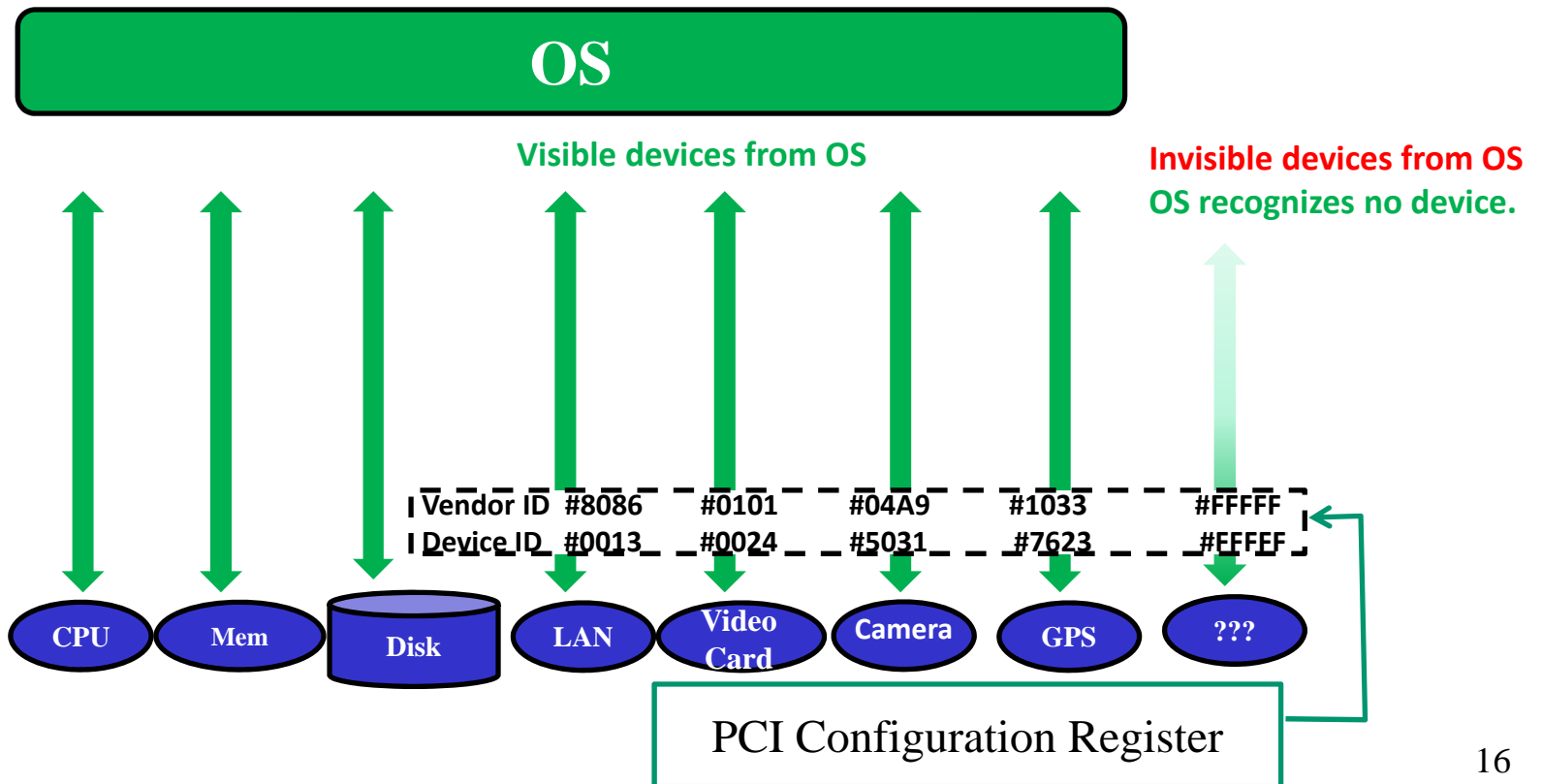
PCI Configuration Register

- I/O port: 0x0cfc

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
Device ID																Vendor ID												0x00					
Device Status																Device Control												0x04					
Class Code																				Revision ID								0x08					
Header Type																0x0c																	
Base Address 0																																0x10	
Base Address 1																																0x14	
Base Address 2																																0x18	
Base Address 3																																0x1c	
Base Address 4																																0x20	
Base Address 5																																0x24	
																																0x28	
Subsystem ID																Subsystem Vendor ID												0x2c					
																																0x30	
Reserved																0x34																	
Reserved																																0x38	
																Interrupt Pin				Interrupt Line				0x3c									
Undefined																																0x40	
																																~	
																																0xfc	

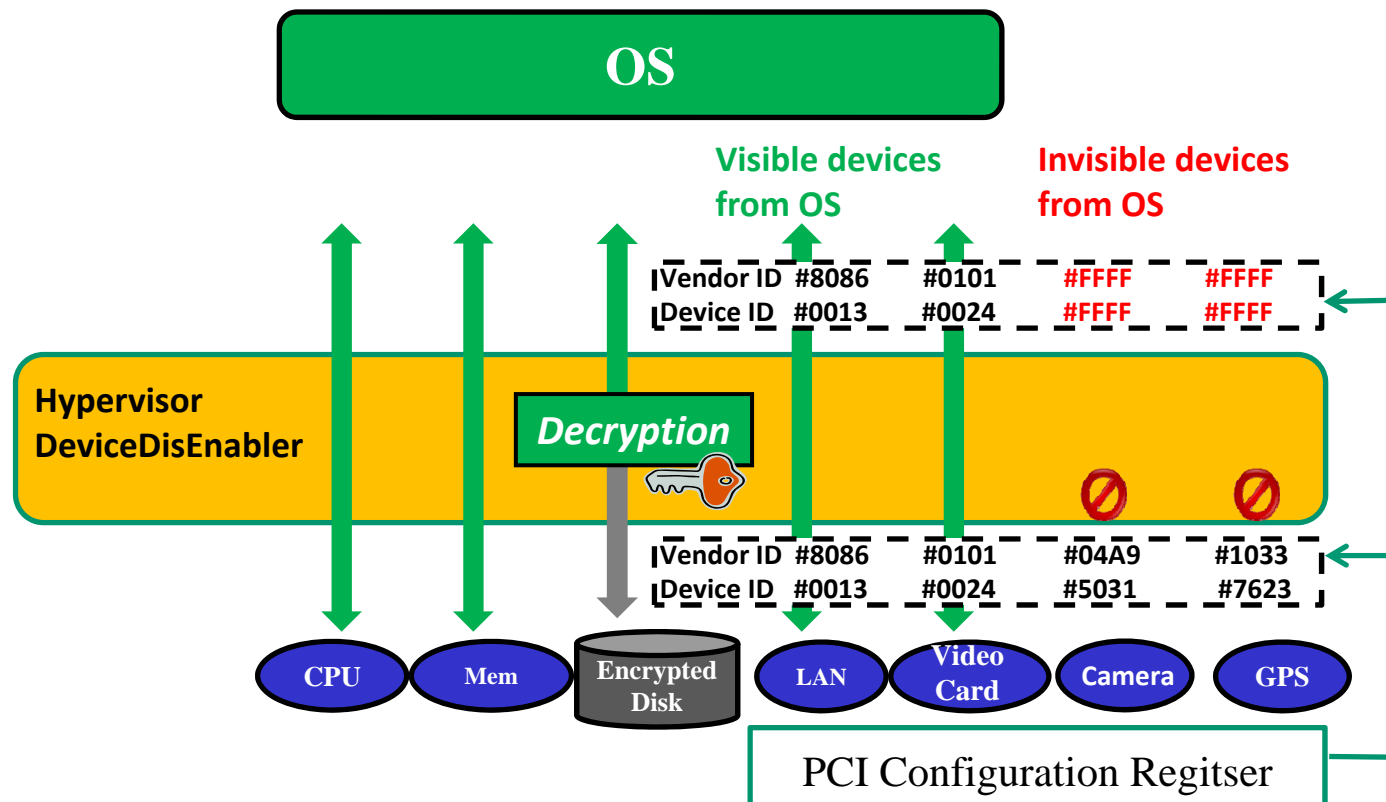
通常のOSによるデバイス認識

- PCIバス上のデバイスを知る為に、OSはI/Oポートをスキャンする。
- x86/AMD64アーキテクチャCPU では**I/O 命令 (i.e., IN and OUT)** を使ってI/Oポートにアクセスする。
 - VendorID, DeviceIDが”#FFFF”の場合はデバイスが無いことを示す。



DDEによるデバイス隠蔽

- I/O命令がOSから呼び出されるとIntel/AMDの仮想化アーキテクチャにより、ハイパーバイザー(DDE)に制御が移る。
- DDEはPCI configuration Registerの内容を検査してOSから隠すべきデバイスと認識した場合、VendorID とDeviceID に#FFFFを返す。
- OS はPCIバスにデバイスが無いものと認識し、そのデバイスは使われない。



DDEによるデバイス隠蔽

- DDEは2種類のタイプでデバイスを隠蔽できる
 - デバイス種類 (Vendor ID and Device ID)
 - 個々のデバイスでなく、ベンダーのこの製品という単位
 - カテゴリ毎 (PCI device class codeの定義による)

Vendor ID	Vendor name
0x05ac	Apple, Inc.
0x04B3	IBM
0x1010	Video Logic Ltd.
0x104D	Sony Corporation
0x1061	8x8 Inc.
0x106B	Apple Inc.
0x13B5	ARM Ltd
0x12E1	Nintendo Co. Ltd.
0x13B5	ARM Ltd
0x15AD	VMware Inc.
0x15C6	Technical University Of Budapest
0x8086	Intel Corporation
0x8087	Intel
0xA304	Sony
0xF5F5	F5 Networks Inc.

Class code	Class Name
0x00	Unclassified device
0x01	Mass storage controller
0x02	Network controller
0x03	Display controller
0x04	Multimedia controller
0x05	Memory controller
0x06	Bridge
0x07	Communication controller
0x08	Generic system peripheral
0x09	Input device controller
0x0a	Docking station
0x0b	Processor
0x0c	Serial bus controller
0x0d	Wireless controller
0x0e	Intelligent controller
0x0f	Satellite communications controller
0x10	Encryption controller
0x11	Signal processing controller
0x12	Processing accelerators
0x13	Non-Essential Instrumentation
0xff	Unassigned class

(3) 改竄防止

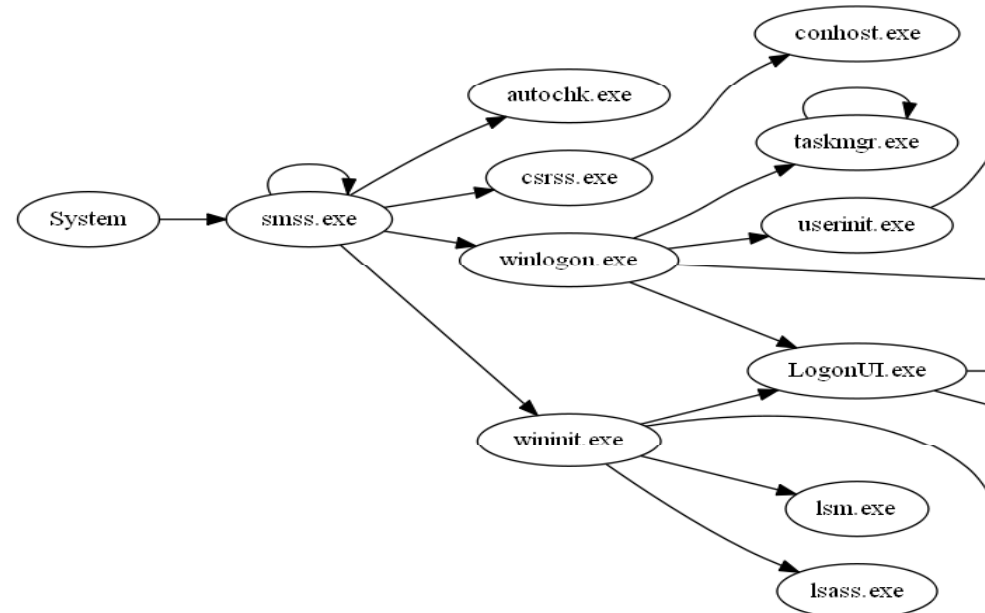
- 残念ながらユーザがハイパーバイザーを取り除いて、あるいは改竄して、デバイスを使う可能性を排除できない。
- DDEの対応策
 - DDEはディスクの一部を暗号化し、DDE無しではOSが起動できないようにしたい。
- 問題点
 - 残念ながらOS(Windows)の起動を止めるのは簡単ではない。

OSのブートを止める困難点

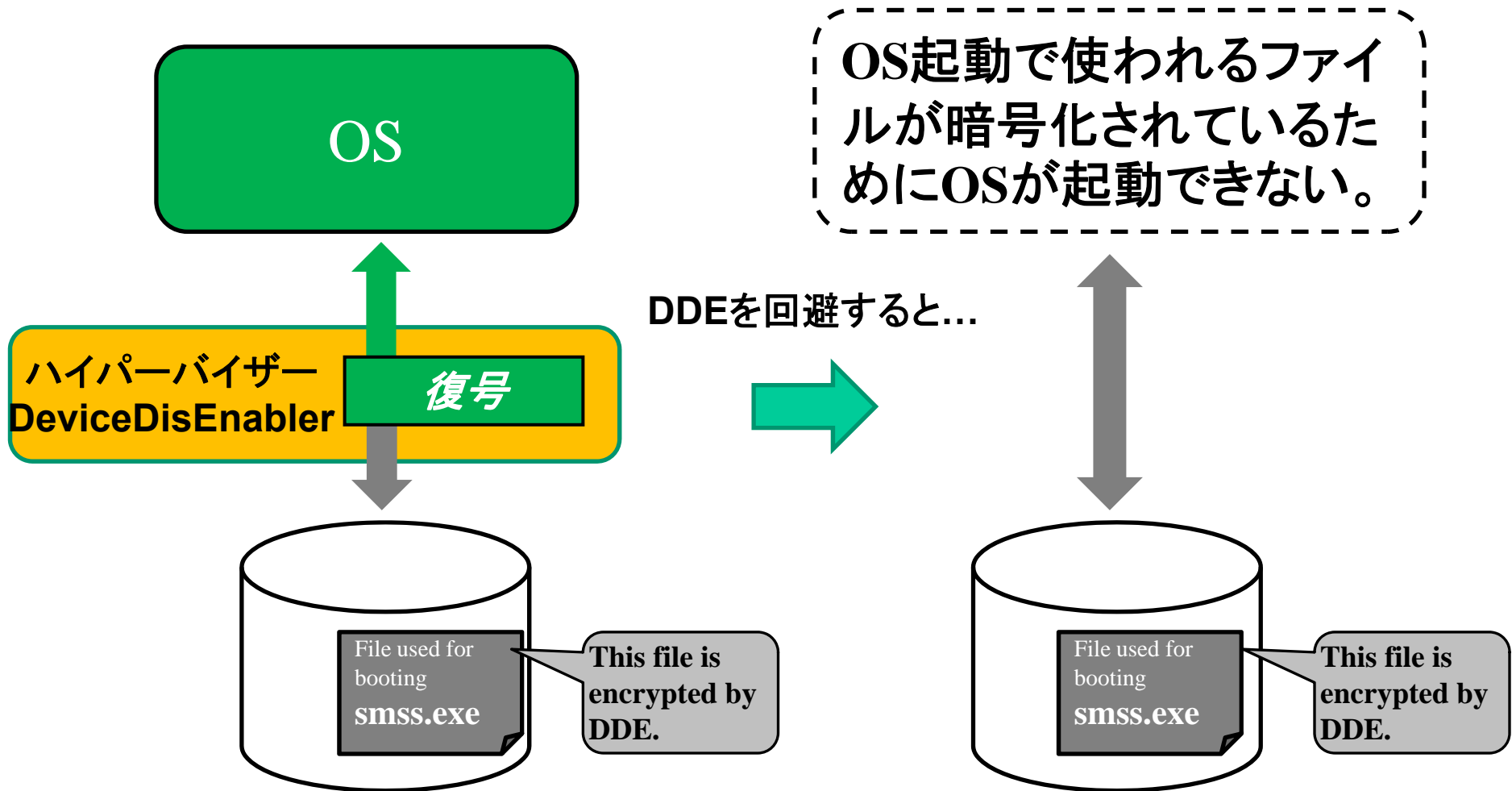
- BitVisor はハードディスクの領域(blocks)を暗号化する機能を持っている。
 - ディスクが盗まれた場合の機密保持には有効
- 残念ながら、BitVisorのディスク暗号化機能をそのままOS(Windows) のパーティションに適用できない。なぜなら**ブートシーケンス内でハイパーバイザーを経ないでアクセスしている**。
 - 推測: Kernelを立ち上げるブートシーケンスのどこかでBIOSを使ったディスクアクセスがあり、それをBitVisorが補足できない
 - このため、パーティション丸毎の暗号化ではOSを起動できない
 - (補足) LinuxならminirootとRootFSのパーティションを分けて、RootFSのみ暗号化することで解決できる。

Windowsの起動の止め方

- カーネルの起動を止めるのではなく、カーネルの起動後のユーザ空間の処理を止める。
- OSのユーザ空間のブートシーケンスを解析して、起動に必要なファイルのいずれかを暗号化する。
- 今回はsmss.exeを選択



DDEによるWindows起動の停止

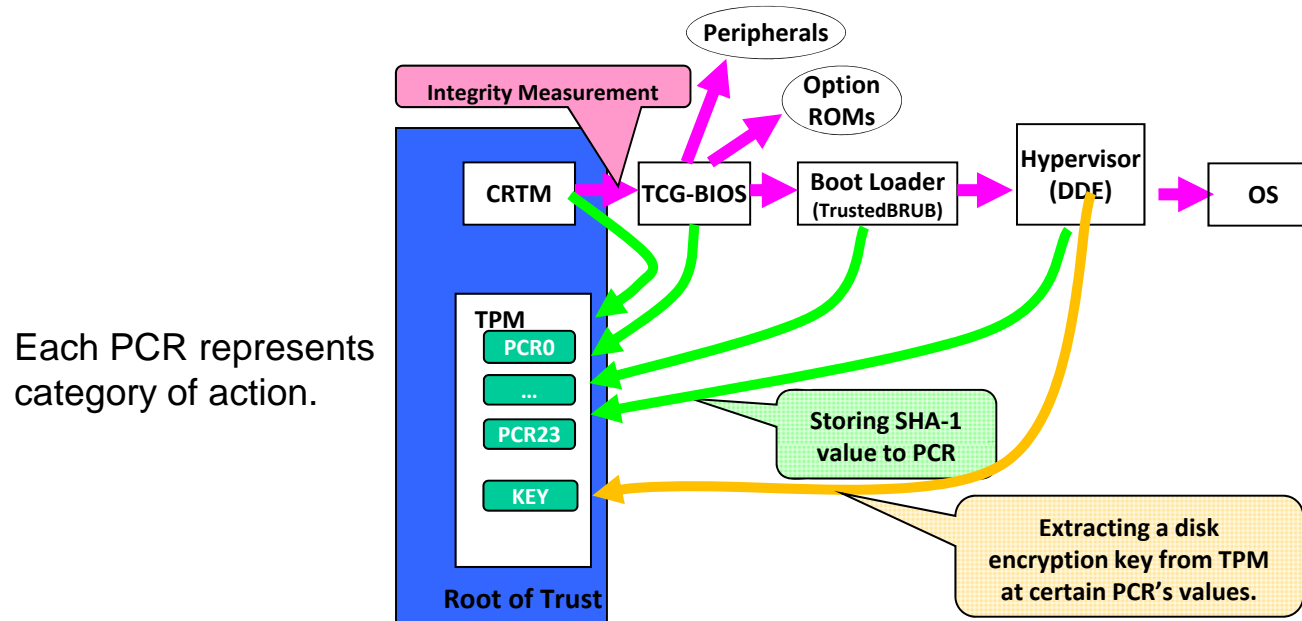


暗号鍵の隠蔽

- **DDEの暗号鍵はユーザに知られてはならない。**
- BitVisor では暗号鍵をただ単にバイナリに含ませるだけ。
 - これでは、攻撃者は簡単にバイナリを比較することで鍵を取り出すことができる。
- セキュリティチップTPM (Trusted Platform Module)に鍵を隠す方法を実装した。
 - ***“Trusted Boot”と“TPM non-volatile storage”の活用***

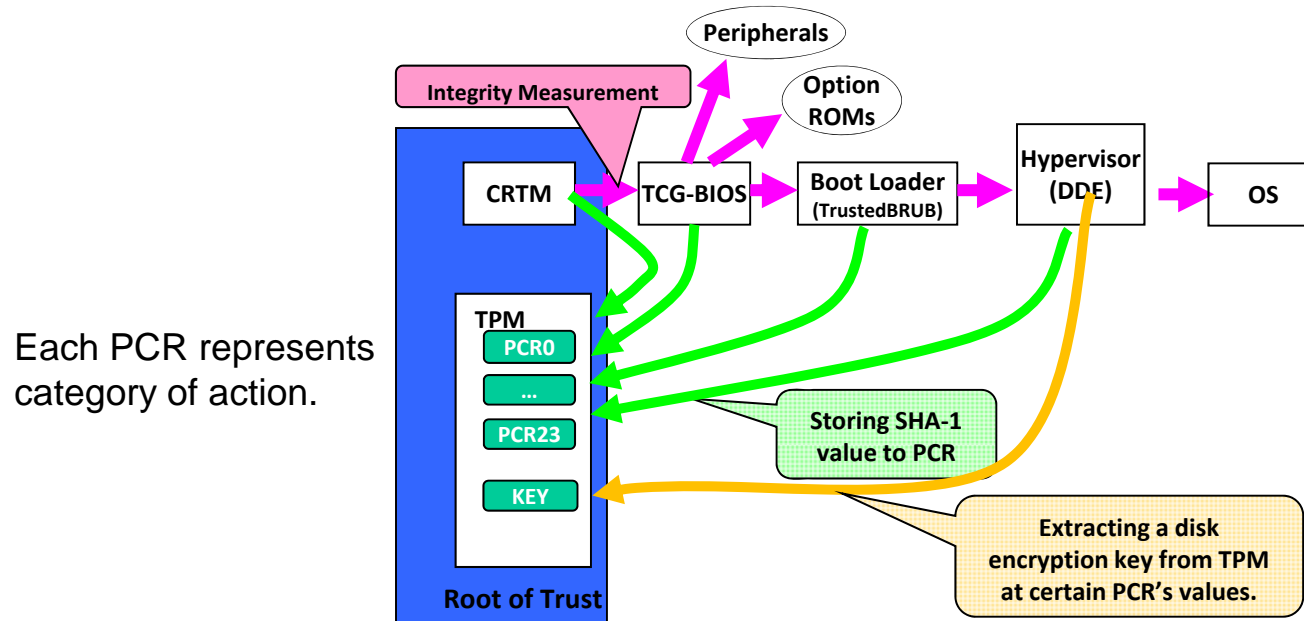
TPMに鍵を隠蔽する方法 (1/3)

- TPMを基軸とするTrusted Bootはブートシーケンスを計測し、その完全性を検証する仕組みを持つ(Chain of Trust)。
 - 各ブートシーケンスのSHA-1値は、“extend” 操作でTPMのPCR (Platform Configuration Register) に保存される。
 - $PCR = SHA-1(PCR + SHA-1(Component))$
 - PCRの値からどのようなブートシーケンスを行ったか検証可能。



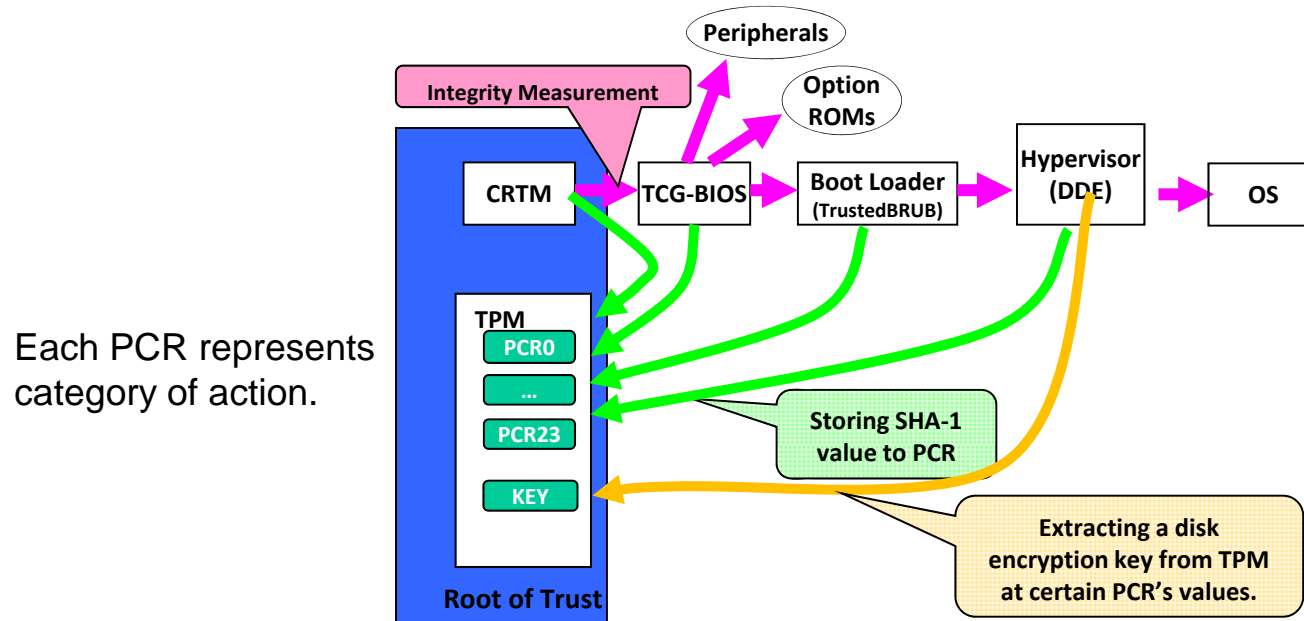
TPMに鍵を隠蔽する方法 (2/3)

- Chain of Trustを維持するためには各コンポーネントが次のコンポーネント計測する機能が必要。
 - BIOSはTCG-BIOS
 - ブートローダはTrusted GRUB



TPMに鍵を隠蔽する方法 (3/3)

- 暗号鍵はTPM内に保存することが可能。その鍵は特定のPCR値のみに取り出せるように設定できる。
 - PCR値が変化すれば(DDEが改竄されれば)、PCR値が異なり、暗号鍵を取り出すことが出来ない。
- これにより**ユーザはDDEを使わざるを得ない**。



Chain of Trust

- ThinkPad Helixでのブートシーケンス
 - 起動で使われるデバイスやソフトウェアは TPMに記録される
 - $PCR = SHA-1(PCR + SHA-1(Component))$

各PCR はそれぞれが
担当する計測値を保持

PCR	SHA1	Event
0	4b81c044c1472a34c73da87d7ad3a64ba62e9047	08 [S-CRTM Version]
6	fcad787f7771637d659638d92b5eee9385b3d7b9	05 [Wake Event 6]
0	8841e9e7d8eb4c753d2ef7dc9f89a07c756cb30b	07 [S-CRTM Contents]
0	3d9766e45814d6374d9a85aa519071dc82574017	01 [POST CODE]
1	b83f6c64a1727add477a94874f3f11f29d531c47	09 [CPU Microcode]
4	9069ca78e7450a285173431b3e52c5c25299e473	04 []
2	199804c152f10535cd88f8f5d607ae55e9e2f3ef	06 [Option ROM]
5	cd0fdb4531a6ec41be2753ba042637d6e5f7f256	80000007 []
0	afbf30b554a35d0ba6a469934d35cf9f58eec6af	80000009 []
1	8de522ea7b732f0bf261ed931245c5c7e75fedbb	80000009 []
0	9069ca78e7450a285173431b3e52c5c25299e473	04 []
1	9069ca78e7450a285173431b3e52c5c25299e473	04 []
2	9069ca78e7450a285173431b3e52c5c25299e473	04 []
3	9069ca78e7450a285173431b3e52c5c25299e473	04 []
5	9069ca78e7450a285173431b3e52c5c25299e473	04 []
6	9069ca78e7450a285173431b3e52c5c25299e473	04 []
7	9069ca78e7450a285173431b3e52c5c25299e473	04 []
1	1f3c97f0b6d45a46ec1aa91e5868322dea94d76c	80000002 []
4	c1e25c3f6b0dc78d57296aa2870ca6f782ccf80f	05 [Calling INT 19h]
4	d564bb707b030e193fdd3ddae8818703225c49c3	05 [Booting BCV Hard Disk]
4	f2e7a20ef1397308f937841b55040905ff7cabca	0d [IPL]
5	c358aaa78d400ad539f90d542e5519aa4e403714	0e [IPL Partition Data]
4	e479a239ff8d17b2391782a86e19ca873ec6536c	0d [IPL]

TPM non-volatile storage

- TPM は“***TPM non-volatile storage***”と呼ぶストレージシステムがあり、特定のPCR値の時にのみデータが取り出せる。
- DDEの暗号鍵をTPM non-volatile storageに保存して、DDEの改竄を防ぐ。
 - DDEが改竄されるとPCR値が異なり、鍵が取り出せない。
- Reference
 - TPM Main Part 3 Commands, Specification Version 1.2, Level 2 Revision 116, 1 March 2011

http://www.trustedcomputinggroup.org/files/static_page_files/72C33D71-1A4B-B294-D02C7DF86630BE7C/TPM_Main-Part_3_Commands_v1.2_rev116_01032011.pdf

TPM non-volatile storageのインターフェイス

- TPM non-volatile storageへは TCG-BIOSが提供する APIによってアクセスできる。

API of TCG BIOS	Description
TPM_NV_DefineSpace	<ul style="list-style-type: none"> •TPM non-volatile storageの領域を確保するAPI •領域へのアクセスは“index” 番号を通して行う •特定PCR値の時のみにアクセスできるように制限可能
TPM_NV_WriteValue	<ul style="list-style-type: none"> •TPM non-volatile storageにデータを書き込むAPI •登録したPCR値の時のみアクセスできる。
TPM_NV_ReadValue	<ul style="list-style-type: none"> •TPM non-volatile storageにデータを読み出すAPI •登録したPCR値の時のみアクセスできる。

TPM non-volatile storageの具体例

- Index番号によってアクセス可能
- この領域へはPCR[0-7,12-14] が特定値(ハッシュ値)の時のみアクセス可能。

On ThinkPad Helix

tpm_nvinfo

NVRAM index : 0x00010016 (65558)

PCR read selection:

PCRs : 0, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14

PCRs to verify

Localities : 0x7

Hash : bcea2524269cafd359d69caa850e209481feec4

Hash of values
of PCRs

PCR write selection:

PCRs : 0, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14

PCRs to verify

Localities : 0x7

Hash : bcea2524269cafd359d69caa850e209481feec4

Hash of values
of PCRs

Permissions : 0x00000000 ()

bReadSTClear : FALSE

bWriteSTClear : FALSE

bWriteDefine : FALSE

Size : 32 (0x20)

PCRの具体例

On ThinkPad Helix

Trusted GRUB はPCR[12-14]を利用

Original DDE

```

PCR-00: 27 CD 64 2F DA 95 EA 09 3B 8C AE BC 68 9F FA C7
2A 59 76 01
PCR-01: E2 60 C4 57 A9 DC 8B C1 3C 5D E8 23 9F 2B 6B 71
86 19 72 19
PCR-02: F2 E5 65 2A DC 7F 57 8A F0 89 9D F1 0F 6B AE A1
13 08 19 E2
PCR-03: B2 A8 3B 0E BF 2F 83 74 29 9A 5B 2B DF C3 1E A9
55 AD 72 36
PCR-04: AA C6 8F 43 8F 5C 23 4E BD 70 F7 46 7D 51 18 4E
BD A3 CA 55
PCR-05: 01 C2 F5 26 13 11 B9 6F 4B BF A4 39 14 AC CA 6B
CD A2 65 41
PCR-06: EE 1B 0F 99 7D 75 17 B2 86 BC 9D 73 A4 CF 74 2C
65 A7 69 BE
PCR-07: B2 A8 3B 0E BF 2F 83 74 29 9A 5B 2B DF C3 1E A9
55 AD 72 36
PCR-08: 93 41 C4 1A 6D EA 42 08 65 16 B8 4B AF AF 48 3C
CD 96 36 91
PCR-09: 1B 60 78 EA 42 8E FA 3A 2A D2 A9 7E 22 04 90 7C
1A E6 33 A9
PCR-10: 3D C7 DF C4 CB B0 EC D3 9F B2 75 14 4B 41 E0 42
52 AF C1 17
PCR-11: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
PCR-12: 98 CB C3 5A 43 22 54 CB CB DD E6 04 30 B1 89 D9
54 E4 E7 F8
PCR-13: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
PCR-14: EB 17 E0 8C 08 E0 1E D6 8B 86 62 14 62 E4 70 24

```

TPM non-volatile storageから鍵を取り出すにはPCR[0-7, 12-14]を使う。

PCR[0-7] はTrusted GRUB以前のブートシーケンスを検証するのに使う。

DDEが改変されるとPCR[12-14] が変わり、鍵が取り出せない。

鍵取り出しの失敗例

- DDE が改変されるとTPM non-volatile storageから暗号化鍵が取り出せない。

```
panic(CPU0): tpm_nv_acquirekey  
s:shell r:reboot ?
```


現状の動作環境

- DDE は下記の条件のノートPC、タブレットで動作可能
 - x86/ADM64 CPU
 - TPM 1.2
 - TCG BIOS (EFIは非対応)

 - PCI デバイスのみ制御可能
 - OS 非依存(動作確認はWindows 7,8,Linuxで行った)

対応モバイルガジェットの拡大計画

- 要点はTPMとTCG-BIOS
 - TPMの対応
 - atom CPUを使ったモバイルガジェットにはTPMが付いているものが少ない。
 - ARM CPUを使ったモバイルガジェットにはTPMが付いているものが少ない。
 - 例外: Samsung Chromebook2ではTPMが付いている。
 - TCG-BIOS
 - MS Surface はTPMが付いているが、起動がEFIのみなので未対応。

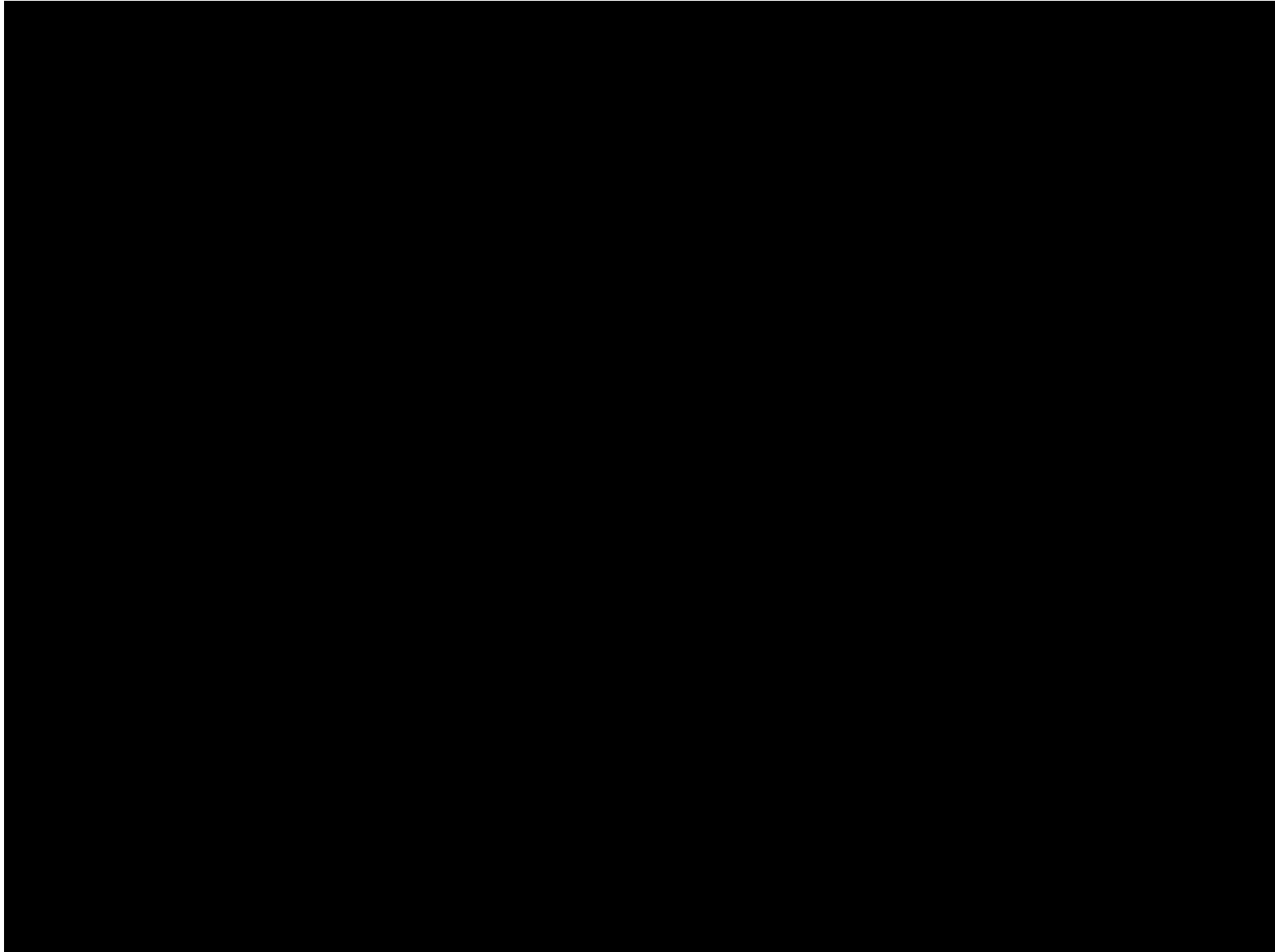
ビデオデモ

- 3つのブート
 - Windows8のスタンドアローンブート
 - smss.exe がDDEによって復号されないのでブートが失敗する。
 - Customized DDE
 - 暗号鍵が取得できないのでブートが失敗する。
 - DDE起動後のWindows8の起動
 - 正しくブート！

Just Fun!

Trusted GRUB has 3 boot options

- Windows 8
- Hacked DDE (Customized DeviceDisEnabler)
- DDE



まとめ

- モバイルガジェットの高解像度デバイスはサイバーエスピオナージ(諜報活動)で使われる危険性がある。
 - 管理者はこれらの不要なデバイスを職場で禁止したい。
- デバイスをOSから隠蔽する軽量ハイパーバイザー“DeviceDisEnabler”を提案した。
- DeviceDisEnabler はTPMを活用した改竄防止機能を含んでおり、ユーザからの回避を防ぐ。
- 今後の予定
 - EFI boot対応。(Microsoft Surfaceでの活用)
 - USBデバイス隠蔽
 - ARM対応

Special Thanks

- 八木都志樹 Toshiki Yagi, AIST
- 吉本道隆 Michitaka Yoshimoto, AIST
- 古原和邦 Kazukuni Kobara, AIST

- BitVisorの開発者の皆様
 - <http://www.bitvisor.org/>

これらのデバイスが何か問題でも？

- Yes!
- 高性能デバイスはサイバーエスピオナージ(諜報活動)の格好のターゲット。
 - 既に幾つかの事例や研究結果がある。

マイクを使った盗聴

- 報道によるとマルウェア “Bundestrojaner” (連邦トロジヤン)が空港で税関・出入国管理を通過する際にインストールされた。
 - 連邦トロジヤンはコードの中にあつた「C3PO-r2d2-POE」というストリングから「R2D2」も呼ばれる。
- 連邦トロジヤンはSkypeの通話を盗聴し、リモートのサイトにデータを転送していた。
- 連邦トロジヤンはChaos Computer Club (CCC) によって2011年に発見。
 - WikiLeaksによるとドイツの州政府からの発注。

GPSによる悪意あるトラッキング

- 日本の「カレログ」が有名だが、海外では“Cerberus” や “mSpy” が堂々と販売されている。
 - カレログはGPSコントロールマネージャとして販売。社会問題になり、サービスを停止。



Current Countermeasures

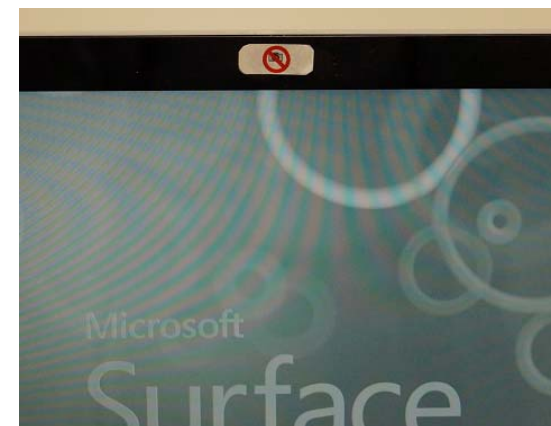
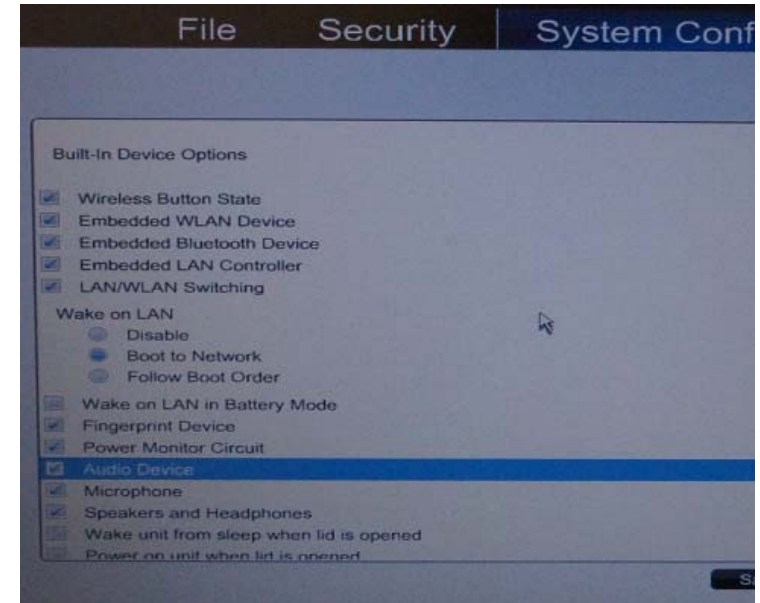
- BIOS/EFI によってデバイスを使用不可にする
 - 有効だが、全てのモバイルガジェットで有効なわけではない。
- Samsung KNOX はデバイスを使用不可に出来るが、Samsung's Androidのみ。
- セキュリティグッズ

プロテクションキャップ

セキュリティシール(カメラ用)



これらはユーザの良心に依存。



OSのブートを止める困難点

- BitVisor はハードディスクの領域(blocks)を暗号化する機能を持っている。
 - ディスクが盗まれた場合の機密保持には有効
- 残念ながら、BitVisorのディスク暗号化機能をそのままOS(Windows)のパーティションに適用できない。なぜならブートシーケンスでハイパーバイザーを経ないでアクセスしている。
 - 推測: Kernelを立ち上げるブートシーケンスのどこかでBIOSを使ったディスクアクセスがあり、それをBitVisorが補足できない
 - このため、パーティション丸毎の暗号化ではOSを起動できない
 - (補足) LinuxならminirootとRootFSのパーティションを分けて、RootFSのみ暗号化することで解決できる。

- ARM仮想化拡張機能
 - ARM Architecture Virtualization ExtensionおよびLarge Physical Address Extension (LPAAE)は、ARMアーキテクチャ準拠のプロセッサ向け仮想マシンハイパーバイザ
- ARMv7A virtualization extensions
- ARM Cortex A15 processor
 - ハイパーバイザー用のHYP Mode。