POC2015
November 5-6, 2015 Seoul, Korea

Cyber Security Required to Protect Human Life

FFRI Inc.
CEO Yuji Ukai
http://www.ffri.jp

Introduction of Our Research

Our Policy & Motivation

Policy of FFRI Basic Research Laboratory

Automotive Security
The number of the ECU is increasing by digitizing various functions of the vehicle. Hijacking vehicle or attacking ECU through automotive network is feared because CAN protocol used by automotive network doesn't consider the security.

iOS Security
iOS has been considered safe because app provide scheme is limited by App Store and app running on sandbox. But malware is inclreasing recent years by new diversified attack techniques.

IoT Security
IoT devices exposed to threat which did not need to be assumed by connecting devices which did not connect to the internet. Microsoft released "Windows 10 IoT Core" supported RPi2 by free as "Windows IoT series" along with emergence of "Windows 10".

Our Work (International Conferences)

- escar Asia 2014
  - Security for Embedded Systems and Solution of Fuzzing
- CODEBLUE 2014
  - A security assessment study and trial of TriCore-powered automotive ECUs
- CODEBLUE 2015
  - iOS malware trends and the malware detection with the dedicated gadgets
  - Threat Analysis of Windows 10 IoT Core and Recommended Security Measures

Our Work (Research Papers)

- May, 2015
  - Trend of Next Gen In-Vehicle Network Standard and Current State of Security
- Jun, 2015
  - Security of Windows 10 IoT Core
- Jul, 2015
  - A Survey of Threat in OS X and iOS
- Sep, 2015
  - Latest Security Reports of Automobile and Vulnerability Assessment by CVSS v3
- Oct, 2015
  - Overview of TPM2.0 and the usage example on IoT devices

Code Execution Theory on ECU (CODE BLUE 2014)

We conduct joint research with ETAS is a system provider in the automotive industry, the research results about the attack approach to TriCore of Infineon, which is often used in European cars of the ECU was presented at CODE BLUE 2014.

Possible Vulnerabilities in ECU Software

- Non-Memory Corruption Vulnerabilities
  - Access Control Issues
  - Encryption Strength Issues
  - Inappropriate Authentication
  - Conflictions
  - Certificate / Password Management Issues
  - etc.

- Memory Corruption Vulnerabilities
  - Buffer Overflow
  - Integer Overflow
  - Use After Free
  - Null Pointer Dereferences
  - Format String Bugs
  - etc.

Since it was difficult to obtain and analyze actual ECU Software, we hypothesized about the possibility of memory corruption vulnerabilities.

Consideration of Memory Corruption Vulnerabilities and Possible Attacks

- Buffer Overflow
  - Stack Overflow
  - Heap Overflow
- Integer Overflow
  - Hypothesized that integer overflows can cause of heap overflows
- Format String Bug
  - Possible to overwrite an arbitrary value in an arbitrary address, hypothesized that attacks are possible
- Use After Free
  - Implied attacks are possible because C++ code is executable with TriCore
- Null Pointer Dereference
  - Trap occurs by access to memory address zero, hypothesized attacks are possible

Possibility of Buffer Overflow Attacks

- Stack Overflows
  - For TriCore, unlike x86 and others, the return address is saved in the address register (A11) instead of the stack, therefore overwriting the return address using a stack overflow is not possible.
- Heap Overflows
  - Will examine TriCore's heap management in the near future

Considering Attack Possibilities Using TriCore's Control Mechanism

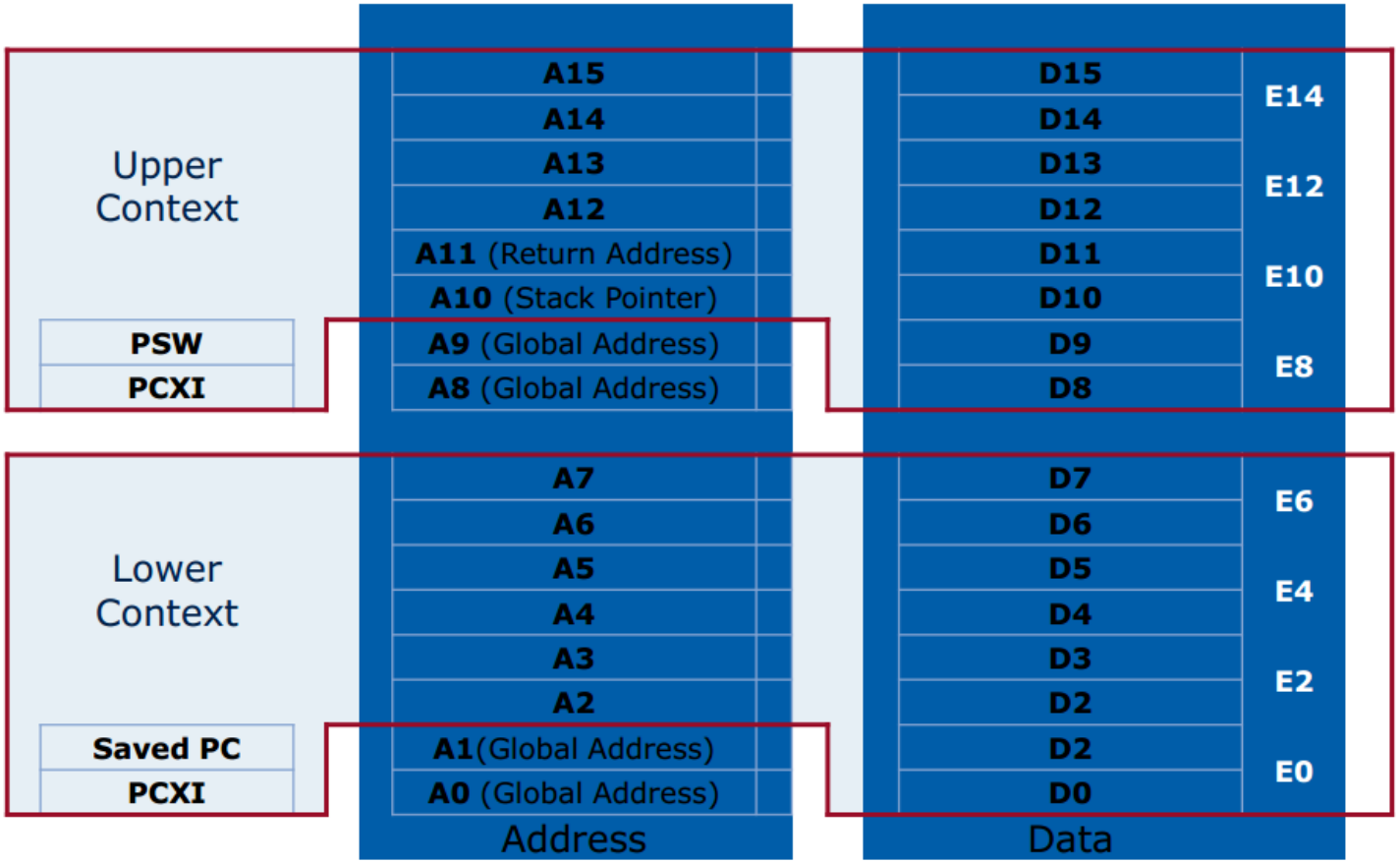
- Preconditions
  - **It is possible to overwrite data by using memory corruption vulnerabilities**
  - **Under the above condition, considered ways to execute arbitrary code**

- TriCore's Control Mechanism
  - Context Management Mechanism
  - Interrupt/Trap Mechanism

Overview of Context of TriCore

- About Context
  - The register value is CSA (Context Save Area)
    Saves and restores in TriCore's unique memory space
- Types of Context
  - 2 Types: Upper context and Lower context
  - Upper context
    - call command, interrupt, automatically saves when trapped
  - Lower context
    - Explicitly saved by using a dedicated command, used for passing parameters

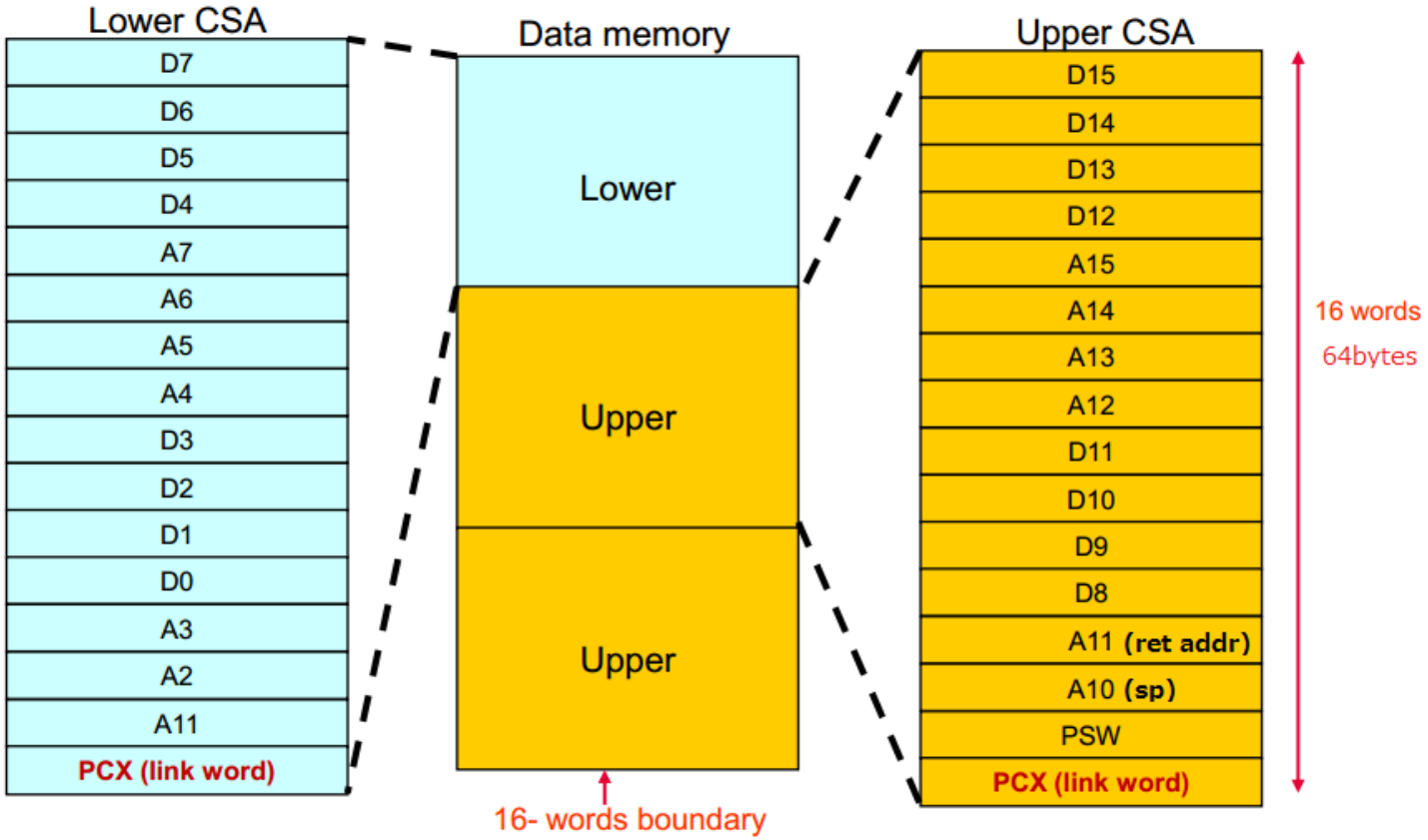
Registers Saved in the CSA



Reference: Tricore Architeture Overview
http://www.infineon-ecosystem.org/download/schedule.php?act=detail&item=44

CSA Configuration

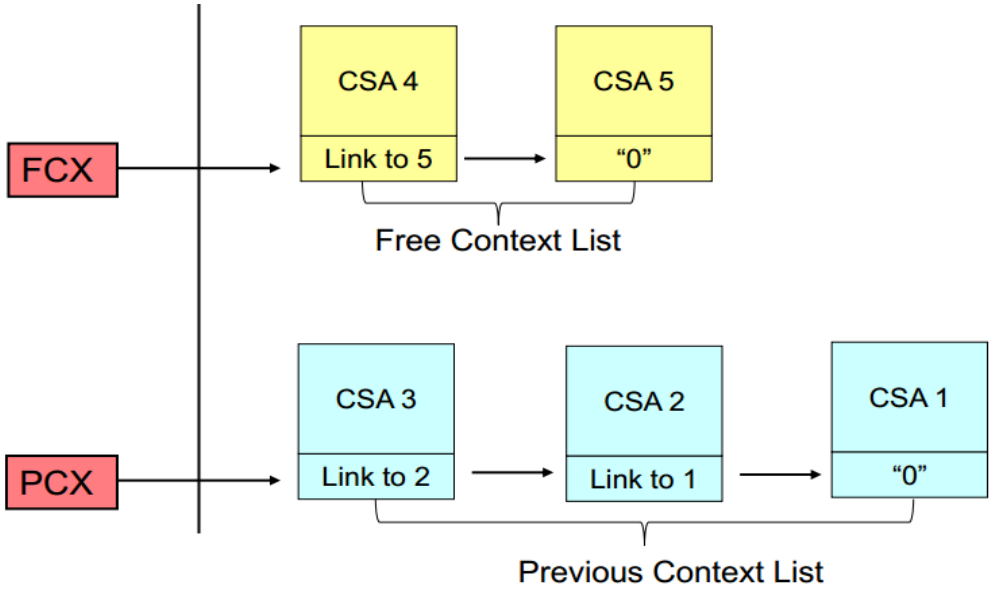
- Context Save areas can hold 1 upper or 1 lower context.
- CSA are aligned on a 16-word boundary.



Reference: Tricore Architeture Overview
http://www.infineon-ecosystem.org/download/schedule.php?act=detail&item=44

CSA Management

- CSA is Managed by Link Lists
  - Used CSA List (PCX) , Unused CSA List (FCX)
  - Pointer to the first element of each list is PCX, stored in FCX register
    - However, needs to be converted because it is not a raw address



Reference: Tricore Architeture Overview
http://www.infineon-ecosystem.org/download/schedule.php?act=detail&item=44

Code Execution Methods Using Context

- Method 1 : CSA Overwriting
  - By overwriting any return address saved in the CSA using a memory corruption vulnerability, it is possible to run code of an arbitrary address
- Method 2 : CSA Injection
  - By overwriting a Link word of the CSA using a memory corruption vulnerability, it is possible to restore crafted Upper context (including return address) and run arbitrary code.

CSA Overwrites on a Simulator

- Code on the right is the result of execution without augments
  func1
  func2
  func3
- Rewrite the return address (*ret) within func2 saved in the CSA to func3 address (0x80000360)
- When returned to func1, the A11 register value restores to func3 (0x80000360)
- Jump to func3 on func1 return

```
#include <stdio.h>

int func3()
{
    printf("func3\n");
    return 0;
}
int func2()
{
    unsigned int *ret;
    printf("func2\n");
    ret=0xD0004F4C;
    *ret=0x80000360;
    return 0;
}
int func1()
{
    printf("func1\n");
    func2();
    return 0;
}
int main( int argc, char** argv)
{
    func1();
    if (argc == 1)
    {
        func3();
    }
}
```

CSA Overwrites on a Simulator

- CSA overwrite using an evaluation board was possible in the same way as the simulator
  - There are memory protections to prevent CSA overwrites by default.
  - May be possible to exploit on actual ECU Software

Wrap up

- Considered attack methods on TriCore-powered ECU software in which memory corruption vulnerabilities exist.
- If memory corruption vulnerabilities exist, it may be possible to execute arbitrary code.
  - By altering the CSA, is is possible execute arbitrary code.
- This research is a result of a study of logical attack methods and a demo conducted on a vulnerable software sample. This study does NOT indicate anything about existing threats on actual ECU software.

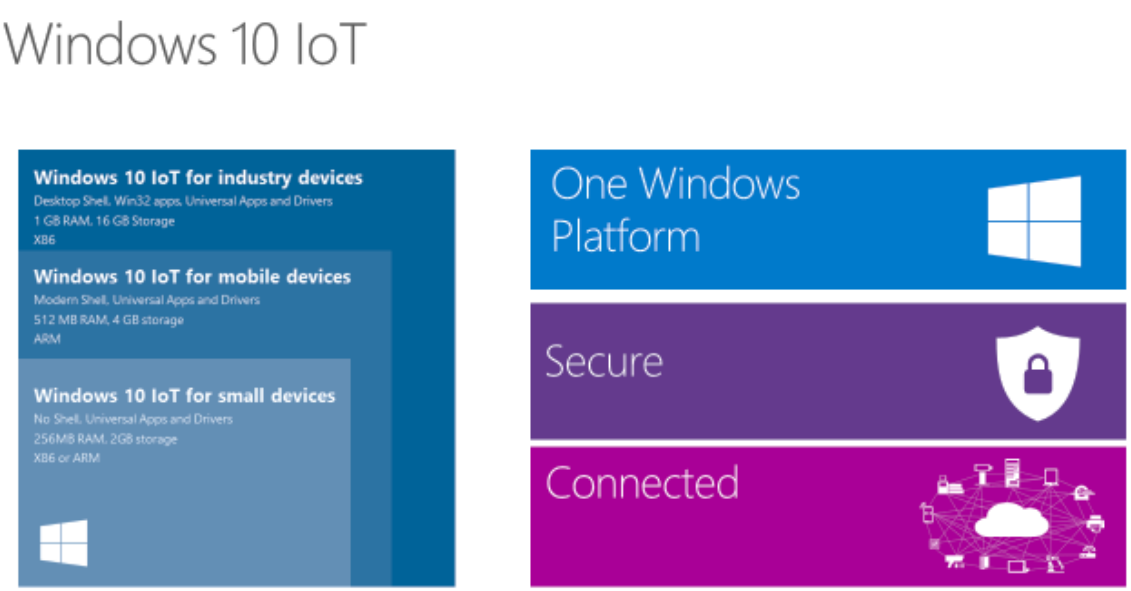
Threat analysis of Windows IoT (CODE BLUE 2015)

As Windows IoT series, newly released Windows 10 IoT Core might also user other than the manufacture to use the Windows 10 IoT Core for that support the single-board computer, such as Raspberry Pi 2.

Therefore we presented at CODE BLUE 2015 the security risks at the time of use and threat analysis of Windows 10 IoT Core.

Changed from “Embedded” to “IoT”

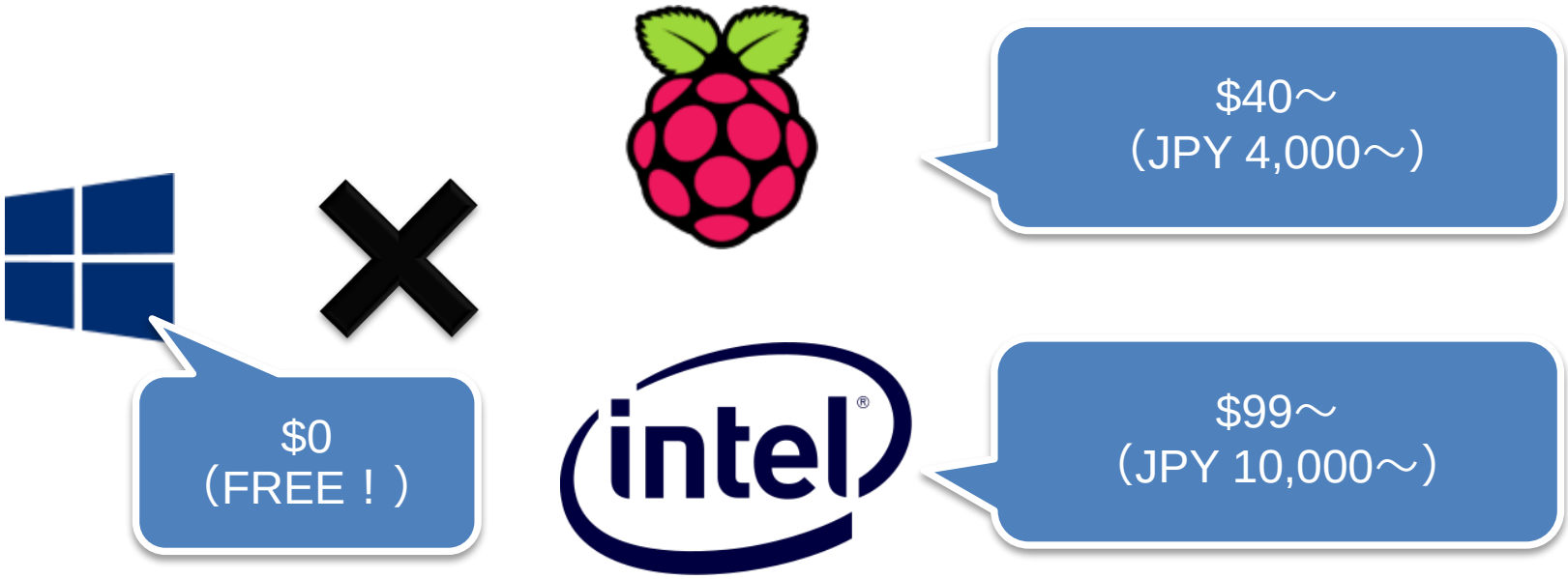
- Windows Embedded series, it has been redesigned in Windows IoT series in conjuction with the launch of Windows 10.
- Windows 10 IoT Core of minimum configuration in Windows IoT series is intended for small devices such as sensors.



http://az648995.vo.msecnd.net/win/2015/03/IoT-1.png

Windows on Single-board Computers

- Windows supports Raspberry Pi 2 and MinnowBoard MAX (Intel) popular worldwide.
- Windows 10 IoT Core that can be used on those single-board computer is provided free.



The difference of IoT Core and Windows 10

Windows 10 IoT Core

Supports

- DEP
- ASLR
- Control Flow Guard
- Windows Firewall

Windows 10 IoT Core

Doesn't support

- Windows Update
- Windows Defender
- User Account Control

Survey of Network Services

- First of all, it was investigated for possibility of attacks from remote.
- Using the port scan results of investigating the TCP and UDP port, it was found that multiple ports are opened.
- Among the open port, this time focused on the FTP and remote debugging service that seems to be prone to attack.

Survey of Network Services（cont.）

- By default, the command line of the process that is using the port that has been open are as follows.

Port	Nmap	Command line
21.tcp	ftp	ftpd.exe
22.tcp	ssh	C:¥windows¥System32¥svchost.exe -k SshSvcGroup
135.tcp	msrcp	C:¥windows¥system32¥svchost.exe -k RPCSS
445.tcp	microsoft-ds?	System
4020.tcp	trap?	C:¥RDBG¥msvsmon.exe /CHILDSERVER 188 "+:4020" {5D8A1EE3-3C96-4562-AD8A-8E4740A26577} 0x3 148 140 13c 144 /silent- /servicemode-
5985.tcp	wsman?	System
8080.tcp	http-proxy	System
9955.tcp 9955.udp	unknown	C:¥windows¥system32¥svchost.exe -k LocalService
47001.tcp	unknown	System

Survey of Network Services（cont.）

- This time the examined services are used for the following purposes.

Port No.	Nmap	Command Line
21.tcp	ftp	ftpd.exe
22.tcp	ssh	
135.tcp	msrcp	
445.tcp	microsoft-ds?	System
4020.tcp	trap?	C:¥RDBG¥msvsmon.exe /CHILDSERVER 188 "+:4020" {5D8A1EE3-3C96-4562-AD8A-8E4740A26577} 0x3 148 140 13c 144 /silent- /servicemode-
5985.tcp	wsman?	
8080.tcp	http-proxy	
9955.tcp 9955.udp	unknown	C:¥Windows¥System32¥svchost.exe -k LocalService
47001.tcp	unknown	System

In order to edit the startup file.
(from commentary of Microsoft official site)

In order to perform remote debugging from Visual Studio 2015.

••• FTP is unnecessary authentication ?

- From the execution of port scan results, it can be seen that the FTP service can be anonymous login.

- Since the banner output is also different from the FTP service that other Windows provides, we've examined the binary.

```
Scanned at 2015-09-26 00:14:16 ???? (?W???) for 83s
PORT      STATE    SERVICE       REASON          VERSION
21/tcp    open     ftp           syn-ack ttl 128
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| d--------- 1 user group 0 Jul 10 13:13 CrashDump
| d--------- 1 user group 0 Jul 10 13:13 Data
| d--------- 1 user group 0 Jul 10 13:13 EFI
```

FTP authentication is not required specifications

- As a result of investigation, there was originally no authentication logic to ftp.exe !

```
SetUser
PUSH.W          {R11,LR}
MOV             R11, SP
LDR             R1, =a331User
POP.W           {R11,LR}
B.W             PostReply
; End of function SetUser
```

```
SetPassword
PUSH.W          {R11,LR}
MOV             R11, SP
LDR             R1, =a230UserLoggedI ; "230 User logged in.\r\n"
POP.W           {R11,LR}
B.W             PostReply
; End of function SetPassword
```

Summary of FTP service

- FTP service of Windows 10 IoT Core does not originally have the authentication feature.
- FTP service is set to start by default always.
- FTP service is “C:¥” is set to the default root directory.
  - It is possible to override some of the important files, such as startup files.

Threat to Confidentiality

- Sniffing/Password cracking（or Steal）
  - Access to Web UI is likely to be stolen the password when the communication because it uses “HTTP communication + Basic authentication” by default has been sniffing.
  - There is a possibility that the password cracking is attempted by the attacker to attack as well services such as HTTP and SSH for home routers that are problem for some time.
- Unauthorized access/Spoofing
  - Since there is no such as account setup wizard during setup, built-in account is likely to be operational the default password.

Threat to Integrity

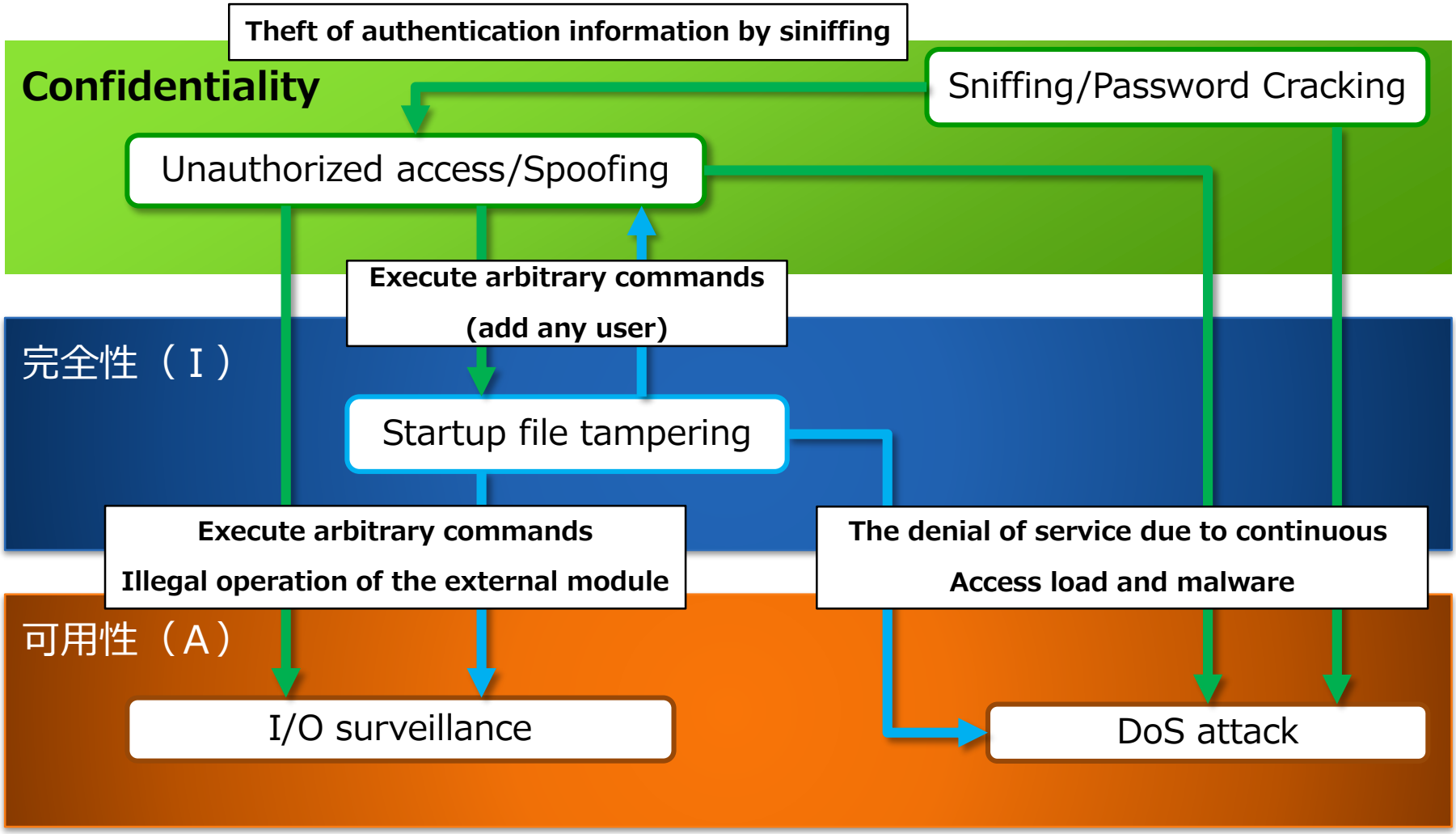
- Startup file tampering
  - Possible to modify the setup files by exploiting the FTP service of the default settings.
  - Since the startup file is run as a batch file, attacker can execute arbitrary commands. (such as user added)

Threat to Availability

- DoS Attack
  - Password cracking attempts and continuous REST API call might elicit a denial of service as a result.

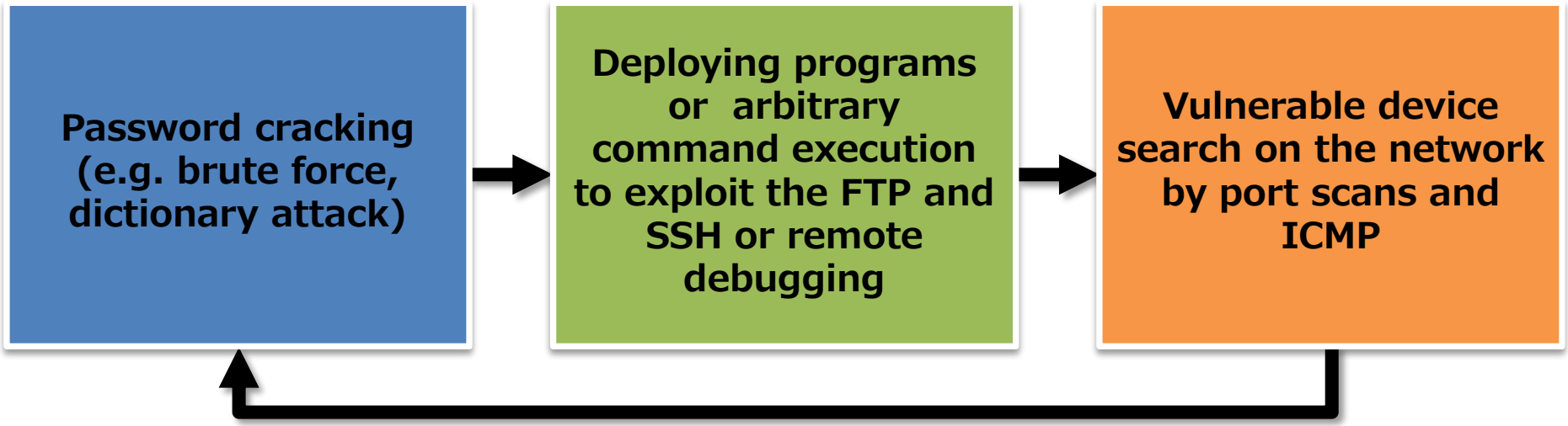
- I/O surveillance
  - Unauthorized access and spoofing might be allowed as a result of the illegal operation of cameras and sensors that are connected to the device.

Relationship diagram of threat



These Threats are likely to be exploted by malware

- Threats in the system configuration at the time of default is almost the same as the home router.
  - May be operated by default password.
  - Web UI access without encryption.
  - No authentication of remote services.
- Therefore, there is a high possibility of attack by malware worm type of repeating invasion and infection.



Wrap up

- Windows 10 IoT Core concept for security is different from the Windows desktop version.
- It might be left behind vulnerable devices because there is no Windows Update feature.
- If Microsoft want to get feedback from a lot of user by the support of single-board computer and provided free, a minimum of security is to be ensured.

The Cutting Edge of Attack Technique to iOS (CODE BLUE 2015)

We are conducting research focused on attacks on iOS that many of attack cases in recent years has been reported.

Embedded Framework is a framework which can be officially supported by the dynamic loading from iOS8, we found that it is possible to new attacks by combining the Method Swizzling.

We have presented the measures and how it works in CODE BLUE 2015.

Crumbling Myths of Security

- **Common Apps are only provided by App Store. And there is severe review.**
  - **Cydia**
  - **iOS Developer Enterprise Program**

- **All apps work on sandbox**
  - **XARA（Cross-App Resource Access）**
  - **Quicksand（By Managed App Configuration）**

Embedded Framework + Method Swizzling = MITA on iOS

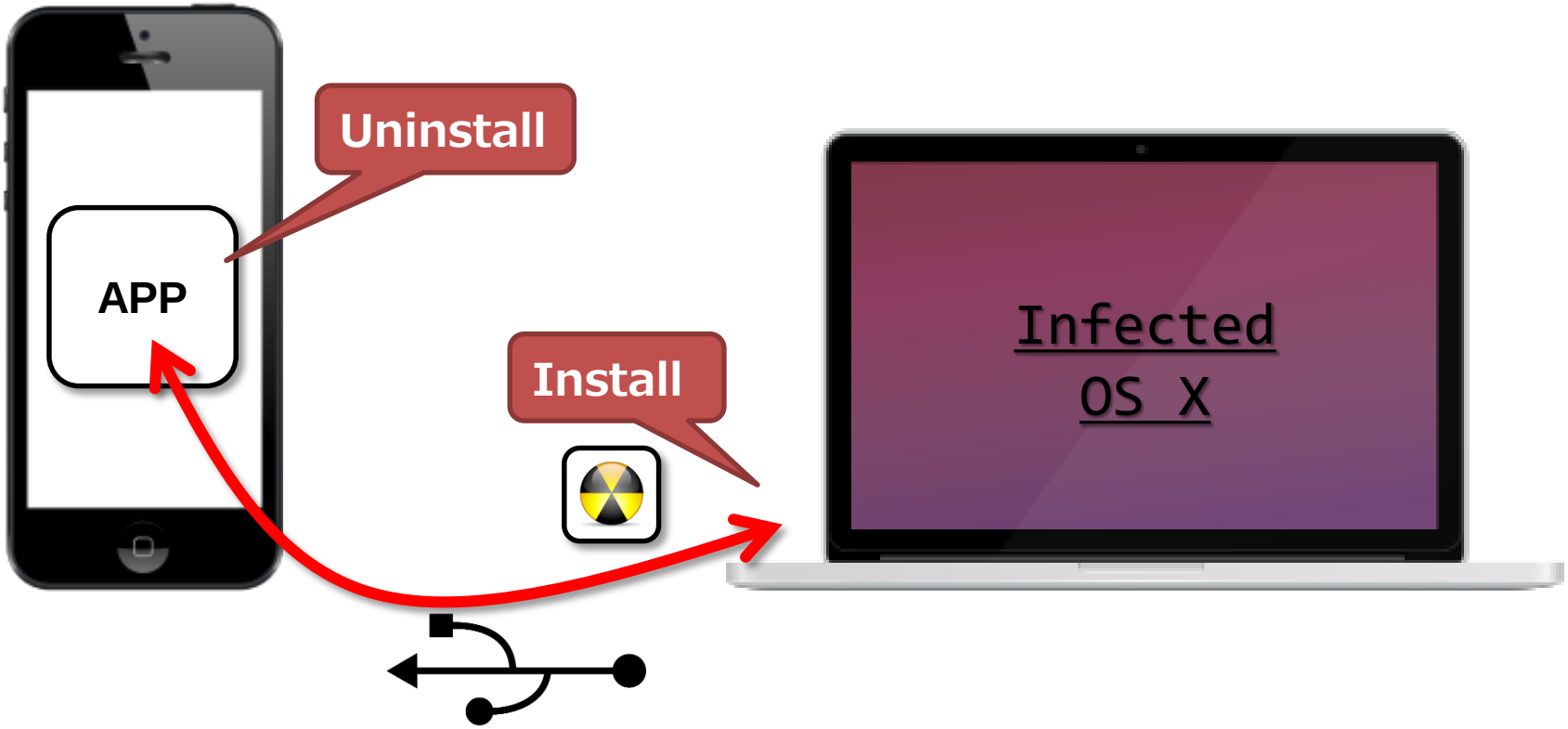
Embedded Framework

```
# Hijack function
Fake transfer function (Account number, Amount of money, Token)
{
    Transfer function ("000-0001440", 1440, token ) ;
}
```

Target App

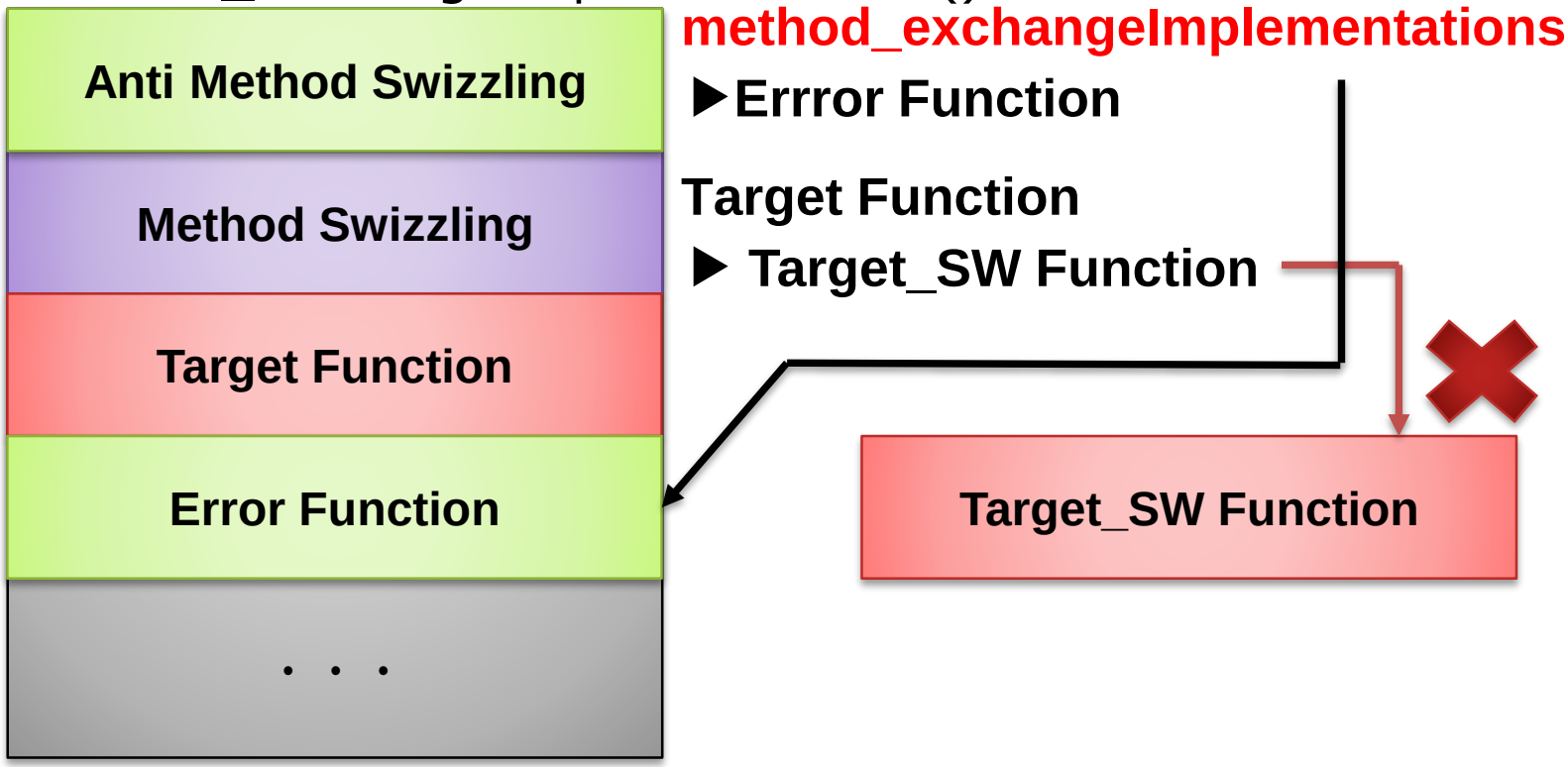
```
# Function declaration
Transfer function
(Account number, Amount of money, Token)

# Exchange functions
Transfer function
=> Fake transfer function

# Calling Function
Transfer function
("000-0000123", 1000,
                    token) ;
```

Rewriting to patched app

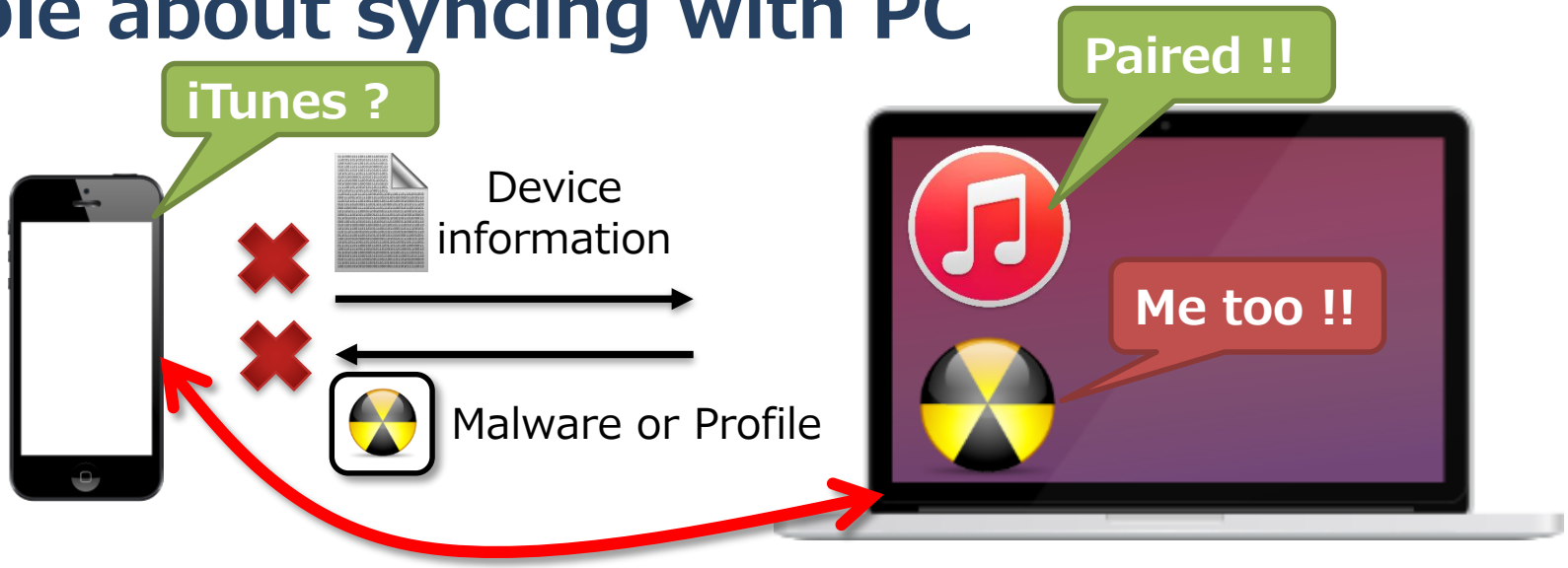


Anti Method Swizzling

- You can prevent Method Swizzling by encrypting source code and do Method Swizzling “method_exchangeImplementations()“.



Proposal for specifications improvement to Apple about syncing with PC



- Alert when third party software requests control.
- Alert when install or sync app that hasn't not App Store's signature.
- Alert when regular app is uninstalled.

Wrap up

- Myth has been broken, because malware use “iOS Developer Enterprise Program (iDEP)” or sync feature and other.
- Threat of MITA on iOS to exploit the attack technique by Embedded Framework and Swizlling Method is becoming a reality.
- We suggest to Apple about sepcifications improvement proposal about sync to PC.

Conclusion

- Areas where we are research might did not need to consider the security threat until a few years ago, the situation has changed.

- And these areas have a relevance to each other, and ultimately threaten even the human life not only property.