Black Hat USA 2015 Survey Report

FFRI,Inc.
http://www.ffri.jp

Ver 2.00.01

About Black Hat USA

- The world's largest security conference in Las Vegas every summer
  - Briefings of cutting-edge security research
    - Threat demo, exploit technique, defense technology
    - They have breakthrough or advantage
    - Slides and papers are public on the Web
  - Yuji Ukai, CEO of FFRI,Inc. is a member of the review boards.
- Also, there are exhibitions of cyber security companies and hacker's original tools
  - Annual festival for cyber security worker
  - Participants are increasing by spotlight of cyber security
  - BSidesLV, DEFCON, USENIX Security were held in around the same time
- In this report, we introduce our focused briefings of Black Hat USA and DEFCON

Our Featured Research

- Vehicle
  - Remote Exploitation of an Unaltered Passenger Vehicle
    - Charlie Miller & Chris Valasek
  - How To Hack a Tesla Model S (DEFCON)
    - Marc Rogers & Kevin Mahaffey
  - Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars (DEFCON)
    - Samy Kamkar
- IoT
  - When IoT Attacks: Hacking a Linux-Powered Rifle
    - Runa A. Sandvik & Michael Auger
  - ZigBee Exploited the Good, the Bad, and the Ugly
    - Tobias Zillner & Sebastian Strobl

Our Featured Research

- Mobile
  - Attacking your “Trusted Core” Exploiting TrustZone on Android
    - Di Shen
  - TrustKit: Code Injection on iOS 8 for the Greater Good
    - Alban Diquet & Eric Castro & Angela On-kit Chow
- Malware, Exploit
  - ROPInjector: Using Return-Oriented Programming for Polymorphism and AV Evasion
    - Giorgos Poulios & Christoforos Ntantogian & Christos Xenakis
  - Exploiting the DRAM rowhammer bug to gain kernel privileges
    - Mark Seaborn & Halvar Flake
  - WSUSPect – Compromising the Windows Enterprise via Windows Update
    - Paul Stone & Alex Chapman

Our Featured Research

- Malware, Exploit
  - Server-Side Template Injection: RCE for the Modern Web App
    - James Kettle

- Reverse Engineering
  - Using Static Binary Analysis To Find Vulnerabilities And Backdoors in Firmware
    - Christopher Kruegel & Yan Shoshitaishvili

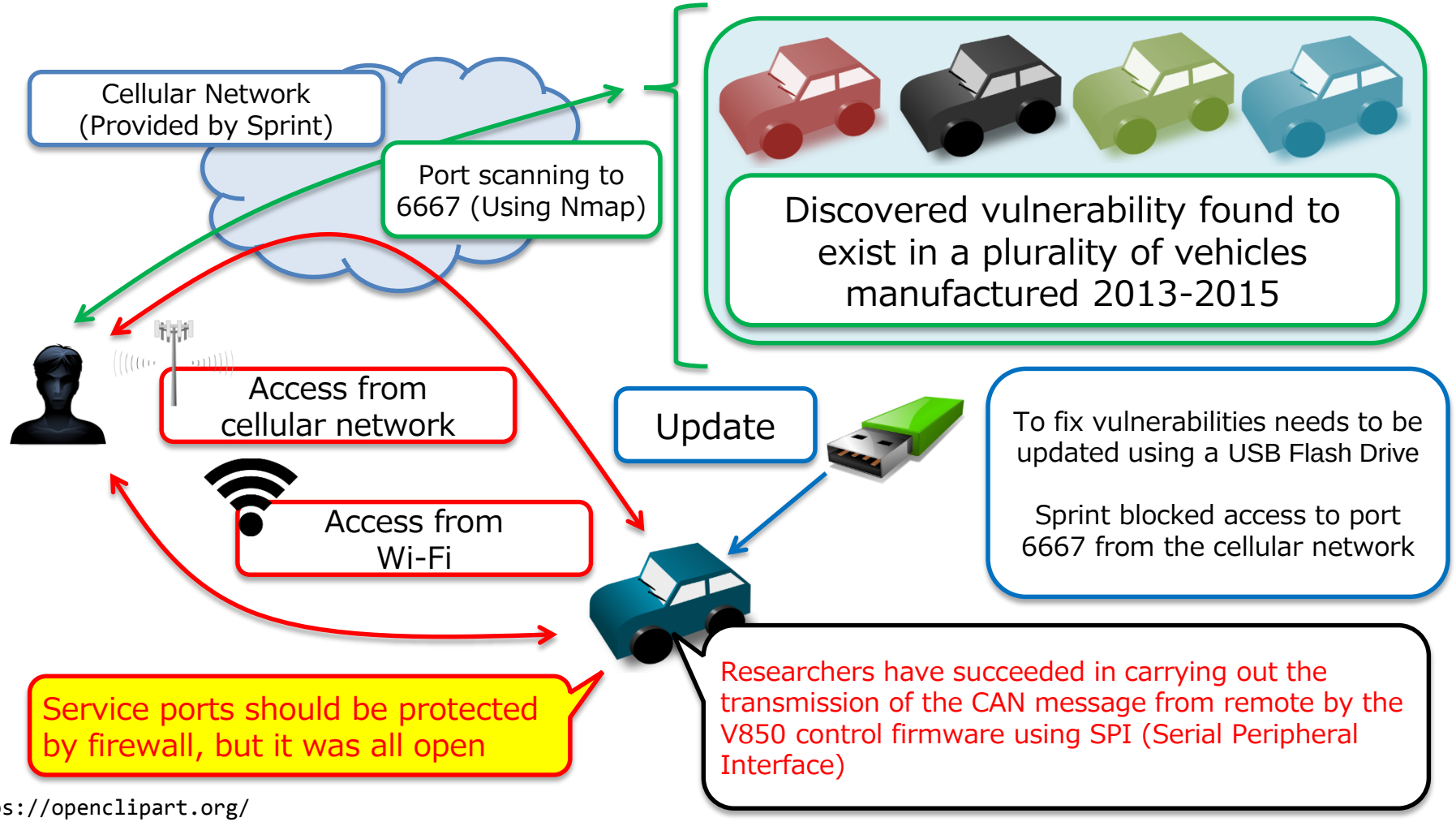
Remote Exploitation of an Unaltered Passenger Vehicle

- Demonstration of Chrysler's Jeep hacking by remote exploit
  - Researchers said “Jeep is Most Hackable” in Black Hat USA 2014

- The attack surface is U-Connect which is in-vehicle infotainment system via Wi-Fi or cellular network
  - The port 6667/tcp is open for D-Bus service, and anonymous user can access it
  - Researchers used Python and DFeet to analyze D-Bus service.
    (DFeet is a tool for debugging D-Bus)
  - As a result of scanning the network, it was found that there is the vulnerability in 2013-2015 models

- Researchers exploited head unit via D-bus at first
  Then they modified firmware to control car using update function
  - Head unit and microcontroller are connected by Serial-Peripheral Interface
  - The update function did not validate a firmware

Remote Exploitation of an Unaltered Passenger Vehicle



https://openclipart.org/

Remote Exploitation of an Unaltered Passenger Vehicle

- **Comments of FFRI researcher**
- Not implementation of firewall is fatal
  - Anyone can do port scanning. Therefore, open ports and services will always be target of attacks
  - This problem will be always pointed out by security experts
  - Lack of authentication for D-Bus service is also problem

- In-vehicle infotainment systems should implement a mechanism of automatic updates by OTA (On-The-Air)
  - Jeep firmware update possible only from USB flash drive

- In-vehicle infotainment systems should implement secure boot and secure update
  - TPM or TrustZone are available

How To Hack a Tesla Model S

- Demonstration of exploit via LAN (Ethernet) port on the Tesla Model S
  - Engine start from a laptop PC which is connected to vehicle
  - Malware which can stop engine remotely was created, and infected to vehicle
  - In addition, it can control power window, control suspension and stop power supply

- The vehicle changed to fail-safe mode when they attacked
  - Gear will be shifted in neutral if engine is stopped by attack
  - Measures have been considered against abnormal control instructions

- Tesla carried out firmware update by OTA to fix vulnerability

How To Hack a Tesla Model S

- **Comments of FFRI researcher**
- Diagnostic port of Tesla Model S from previously has been analyzed, and the results had been shared in forum by owners

- Infotainment system of Tesla Model S is Linux base
  - It is easy to develop malware for general OS

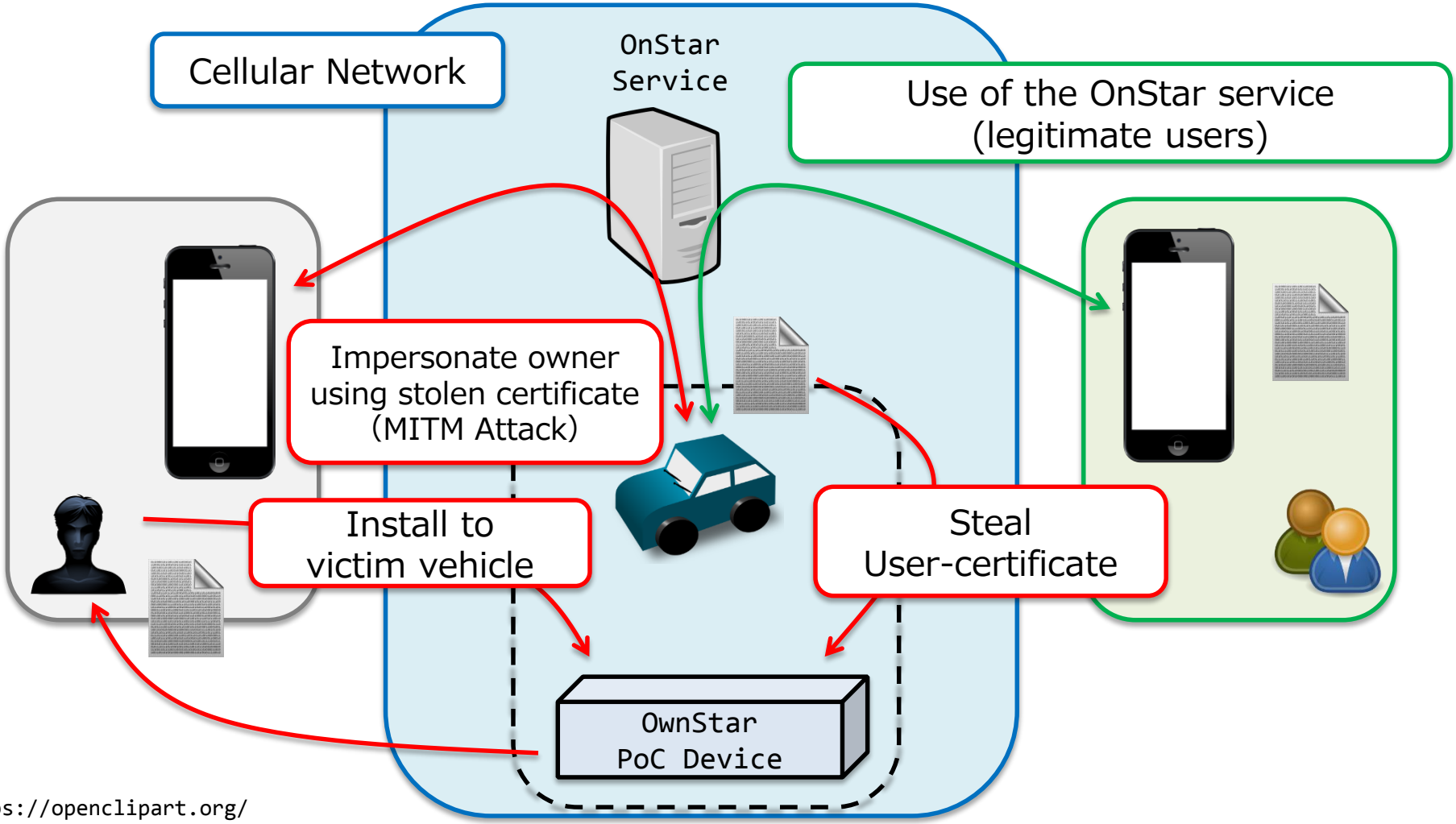
- Linux-based OS will be widely used for in-vehicle infotainment in the future
  - Security measures are required because Linux is an OS exposed to frequent attacks

- Fail-safe mode for abnormal situation
  - Really scary attack is disabled or avoidance of fail-safe mode

Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars

- Demonstration of MITM attack in telematics service which provided by GM
- An attacker steal users certificate by exploiting vulnerability of RemoteLink app
  - Raspberry Pi-based devices have been used in the PoC
- As a result, attacker is possible to perform all the operations that user can operate in the app from remote
- A vulnerability was discovered in iOS app for the telematics service
  - GM said Android/Windows version have no problem

Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars



https://openclipart.org/

Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars

- **Comments of FFRI researcher**
- When using app to control the vehicle, we need to consider the risk of MITM

- Mobile app is easy to reverse engineering compared with embedded software
  - Price, Availability, CPU Architecture

- Such a threat has become easier to achieve than ever by development of single-board computer and SDR technology
  - Also affected drone and various IoT devices

- Therefore, defense of vehicle itself and comprehensive security is necessary to for services in the future

When IoT Attacks: Hacking a Linux-Powered Rifle

- Analysis of TP750
  - Analyzing next-generation rifle equipped with Linux based support system works on ARM
  - The Rifle has AP that is possible to access from smartphone
  - They found multiple vulnerabilities by reverse engineering them.
    - For example “Fixed WPA2 key”, and “password(4-digit PIN) for advanced mode is possible to break by brute force attack"
  - Manufacturer said part of them was fixed
- **Comments of FFRI researcher**
  - The Rifle is not used by a lot of people
  - But it's dangerous to left these vulnerabilities
    Because thing connected to the internet might used with other device
  - We recommend design that premise to be attacked

ZigBee Exploited the Good, the Bad, and the Ugly

- About IoT
  - In 2022, 500 smart devices are into the home. And we are not expected these devices connected by wired
  - But wireless LAN is not practical. So, there is ZigBee
- Security of ZigBee
  - ZigBee is expected for IoT but it has vulnerabilities. For example default trust center link key is fixed
  - There is need to focus to the security modeled on history of TCP/IP

- **Comments of FFRI researcher**
  - If connecting devices by traditional way for concept of IoT is not efficient. Expecting other way like ZigBee is natural thing
  - We recommend to focus to high layer security. For example encrypting packet because key is fixed

Attacking your “Trusted Core” Exploiting TrustZone on Android

- An example of TrustZone exploit
  - Target smartphone is Huawei Ascend Mate 7 (SoC: Hisilicon Kirin 925)
- The exploit caused by Huawei's original TEE software implementation
  - Some vulnerabilities were found in both Normal World and Secure World
- Strategy of TrustZone exploit
  - Rooting Android and Disabling SE for Android in the Normal World
  - Then sending and executing shellcode in the Secure World
- Demonstration: Bypass of security mechanism and theft of fingerprint data

- **Comments of FFRI researcher**
  - Approach is royal road, but he was analyzing patiently reverse engineering and black box architecture
  - Unique implementation software tend to have vulnerabilities
  - In particular, handling of memory address in driver should be careful

TrustKit: Code Injection on iOS 8 for the Greater Good

- Deregulation of iOS app development (Embedded Frameworks)
  - If you want to publish any app, you must all codes are static and linked to binary. But it has been relaxed since iOS 8
  - Now , we can embed third party framework and they can load dynamically
- Hooking function by adding framework on non-jailbreak
  - This way is possible to hook functions and it's not need modifying app code by loading third party framework
  - It means maybe able to hook SSL
  - Speakers published this way named "TrustKit"

- **Comments of FFRI researcher**
  - Vulnerabilities created by new feature or changing specification through OS updating often found in such as iOS and OSX
  - Works on non-jailbreak means big impact because that it is possible to create malicious app like sniffing SSL packet

ROPInjector: Using Return-Oriented Programming for Polymorphism and AV Evasion

- Injecting ROP shellcode into harmless executable file for AntiVirus Evasion.
  - Low suspiciousness, Generally code injection, Polymorphism
- Four challenges
  - AntiVirus evasion, No damage to original PE,
    No writable section, Implementation as general tool
- Injection procedure (7 steps)
  - See presentation slides
- Evaluation conditions
  - Implementation: Native Win32 C
  - Original harmless file: 9 PE files(32bit)
  - Shellcode: Metasploit Reverse TCP and meterpreter reverse TCP
  - AntiVirus vendor: 57 in VirusTotal
- Results
  - Almost 100% AntiVirus Evasion

ROPInjector: Using Return-Oriented Programming for Polymorphism and AV Evasion

- Current signature-based detection methods are no longer effective
  - They shown that by using ROP we can reduce the footprint to benign stack modifying instructions
- Behavioral analysis is tough to perform exhaustively
  - They shown how to easily bypass it by running right before process exit
- “Default distrust all” policy
  - Checksums and certificates is the poor user’s last line of defense at the moment

- **Comments of FFRI researcher**
  - Pattern matching based AntiVirus would be powerless
  - This evaluation result has very impact
  - We are interesting to quality as a tool
  - We pray that cyber criminals do not use this method

Exploiting the DRAM rowhammer bug to gain kernel privileges

- Rowhammer causes bit flips in adjacent rows
- How would one exploit a truly random bit flip in physical memory?
  - Identify data structure which makes privilege escalation by random bit flip
  - Fill as much memory as possible with this data structure
  - Wait for the bit flip to occur
- Types of memory error: random (e.g. cosmic ray) vs. repeatable
  - Rowhammer is indictable by software, and often repeatable
  - Repeatable bit flips gives more control
- How to row hammer on x86
  - Requirement #1: Bypass the cache → x86 CLFLUSH instruction
  - Requirement #2: Search for bad rows
  - Requirement #3: Pick >=2 addresses

Exploiting the DRAM rowhammer bug to gain kernel privileges

- Experiment Results: rowhammer-test
  - Allocates 1GB, looks for bit flips in this
  - Risky: Could corrupt other processes or the kernel
  - Bit flips occurred on some laptop (Model 2010-2011)
- Two exploits:
  - Systems rely on memory staying constant!
  - Native Client (NaCl) sandbox in Chrome
    - Bit flip in validated-to-be-safe code
  - Linux kernel privilege escalation
    - Bit flip in page table entries (PTEs)
    - Spray physical memory with page tables
    - Privilege escalation in 7 easy steps
      - In practice, there are many complications.

Exploiting the DRAM rowhammer bug to gain kernel privileges

- As software-level sandboxes get better, attackers will likely target more esoteric bugs, such as hardware bugs
- Rowhammer: not just a reliability problem
- Hard to verify that hardware meets spec
  - Vendors should adopt security mindset
  - Vendors should be more transparent

- **Comments of FFRI researcher**
  - Their exploit approach is very novelty
  - Two exploit example is shown, but it seems difficult for the general attacker to practical use this technique.
  - There is a possibility to be considered a similar approach in the Windows and OS X

WSUSPect – Compromising the Windows Enterprise via Windows Update

- One approach for an intruder to privilege escalation in WSUS environment
- WSUS Security
  - SSL not enabled by default
  - WSUS uses SSL for metadata only, not for update files.
  - All updates must be signed by Microsoft
- WSUS Attacks
  - If SSL not used we could MITM update traffic
  - Updates are signed so cannot be modified
- Windows Update respects user proxy settings
  - Modify proxy settings for MIMT
  - Injecting a fake update
  - Running PsExec via “Command Line Installation”
    - download and run any Microsoft-signed exe
    - With arbitrary command-line arguments

WSUSPect – Compromising the Windows Enterprise via Windows Update

- Attack Scenario 1
  - Client PC configured to use WSUS over HTTP
  - Malicious low-priv user
  - PsExec is likely to be detected, but it can be substituted by BgInfo
    - bginfo ¥¥attacker¥share¥config.bgi /nolicprompt /timer:0
- Attack Scenario 2
  - Attacker has access to corporate subnet, no domain creds
  - Attacker can perform ARP spoofing / WPAD injection
- **Comments of FFRI researcher**
  - It is surprising that they found the hole of Windows Update and WSUS that trusted by many people
  - Potentially abusable Microsoft-signed file might also exist in addition to BgInfo
  - I want to know how to exploit to force installed vulnerable drivers

Server-Side Template Injection: RCE for the Modern Web App

- Vulnerability of template engine
  - Template engine that to generate dynamically web page has RCE vulnerability
- How it works
  - RCE works if accept user argument directly because template syntax works there
  - For example embedded php code
- Affected template engine
  - FreeMarker, Velocity, Smarty, Twig, Jade, etc.
- **Comments of FFRI researcher**
  - Direct assign user argument to template engine from framework is dangerous
  - Developer needs secure coding because a lot of major template engine vulnerable
  - And we found same vulnerability on Python based web framework named "bottle"

Using Static Binary Analysis To Find Vulnerabilities And Backdoors in Firmware

- Presentation of binary analysis method of IoT devices firmware
- Binary analysis framework called “angr”
  - http://angr.io/
- Researchers described “Symbolic Execution Engine” for discovering a vulnerability related to authentication bypass
  - It tracks conditional branch, and estimates value that may be set to variable when it reaches requested path
- **Comments of FFRI researcher**
  - Vulnerability and backdoor inspection of firmware is an important issue
  - This presentation shows an approach that may be solution of challenge
  - However, example of vulnerability and backdoor found by this approach is not shown

Conclusions

- Vehicle and IoT security research are increasing
  - Non-IT expertise and breakable target object are required for these security research
  - Attacking cost is high compared to generic computer
  - There is a risk that damage to human body by attack
  - However, defense is the same level as the information device
    - The following measures are not enough
      - Threat analysis in design
      - Pentest before product release
      - Update framework for vulnerability fix

- It has been shown limit of pattern matching in multiple anti-malware research

- Research of iOS and Android are increasing
  - Threat will become reality in the future

References

- When IoT Attacks: Hacking a Linux-Powered Rifle
  - https://www.blackhat.com/docs/us-15/materials/us-15-Sandvik-When-IoT-Attacks-Hacking-A-Linux-Powered-Rifle.pdf
- ZigBee Exploited the Good, the Bad, and the Ugly
  - https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly.pdf
- Attacking your “Trusted Core” Exploiting TrustZone on Android
  - https://www.blackhat.com/docs/us-15/materials/us-15-Shen-Attacking-Your-Trusted-Core-Exploiting-Trustzone-On-Android.pdf
- TrustKit: Code Injection on iOS 8 for the Greater Good
  - https://www.blackhat.com/docs/us-15/materials/us-15-Diquet-TrustKit-Code-Injection-On-iOS-8-For-The-Greater-Good.pdf
- ROPInjector: Using Return-Oriented Programming for Polymorphism and AV Evasion
  - https://www.blackhat.com/docs/us-15/materials/us-15-Xenakis-ROPInjector-Using-Return-Oriented-Programming-For-Polymorphism-And-Antivirus-Evasion.pdf
- Exploiting the DRAM rowhammer bug to gain kernel privileges
  - https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf
- WSUSPect – Compromising the Windows Enterprise via Windows Update
  - https://www.blackhat.com/docs/us-15/materials/us-15-Stone-WSUSpect-Compromising-Windows-Enterprise-Via-Windows-Update.pdf
- Server-Side Template Injection: RCE for the Modern Web App
  - https://www.blackhat.com/docs/us-15/materials/us-15-Kettle-Server-Side-Template-Injection-RCE-For-The-Modern-Web-App-wp.pdf
- Using Static Binary Analysis To Find Vulnerabilities And Backdoors in Firmware
  - https://www.blackhat.com/docs/us-15/materials/us-15-Kruegel-Using-Static-Binary-Analysis-To-Find-Vulnerabilities-And-Backdoors-In-Firmware.pdf

Contact Information

E-Mail : research—feedback@ffri.jp
Twitter : @FFRI_Research