EE515
Security of Emerging Systems

Yongdae Kim

KAIST

Admin

- Find your group members and discuss about projects

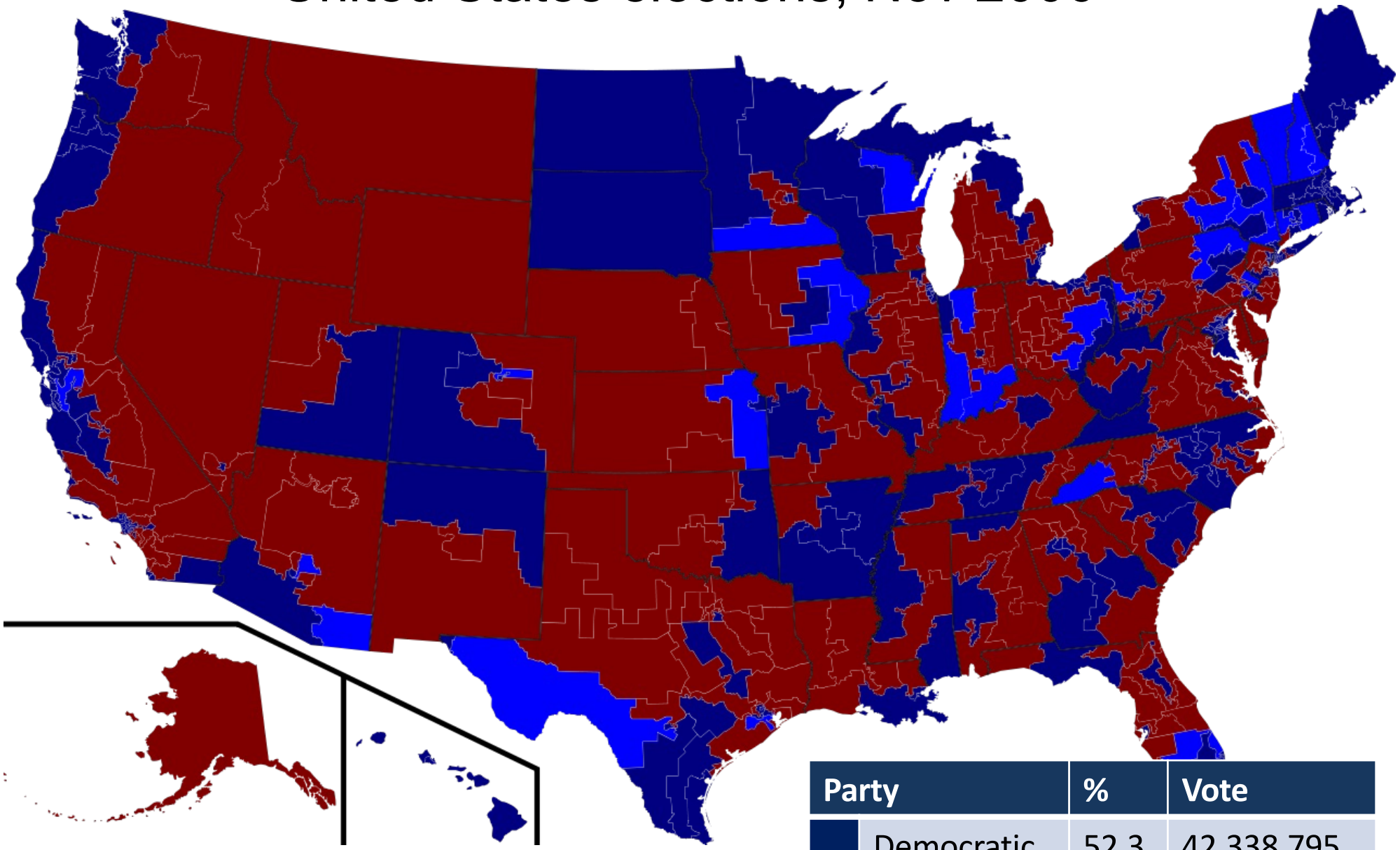
Security Analysis of the Diebold AccuVote-TS Voting Machine

EVT '07
A. Feldman, J. Halderman, and E. Felten

Presenter
Jinseob Jeong

This file is originally written by Dawon Park and Donhwan Kwon,
Revised by Jinseob Jeong

United States elections, Nov 2006



Party	%	Vote
Democratic	52.3	42,338,795
Republican	44.3	35,857,334

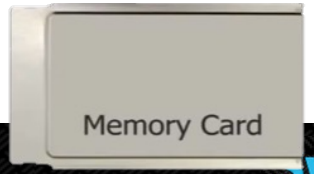
Voting

Paper-based Voting

Electronic Voting

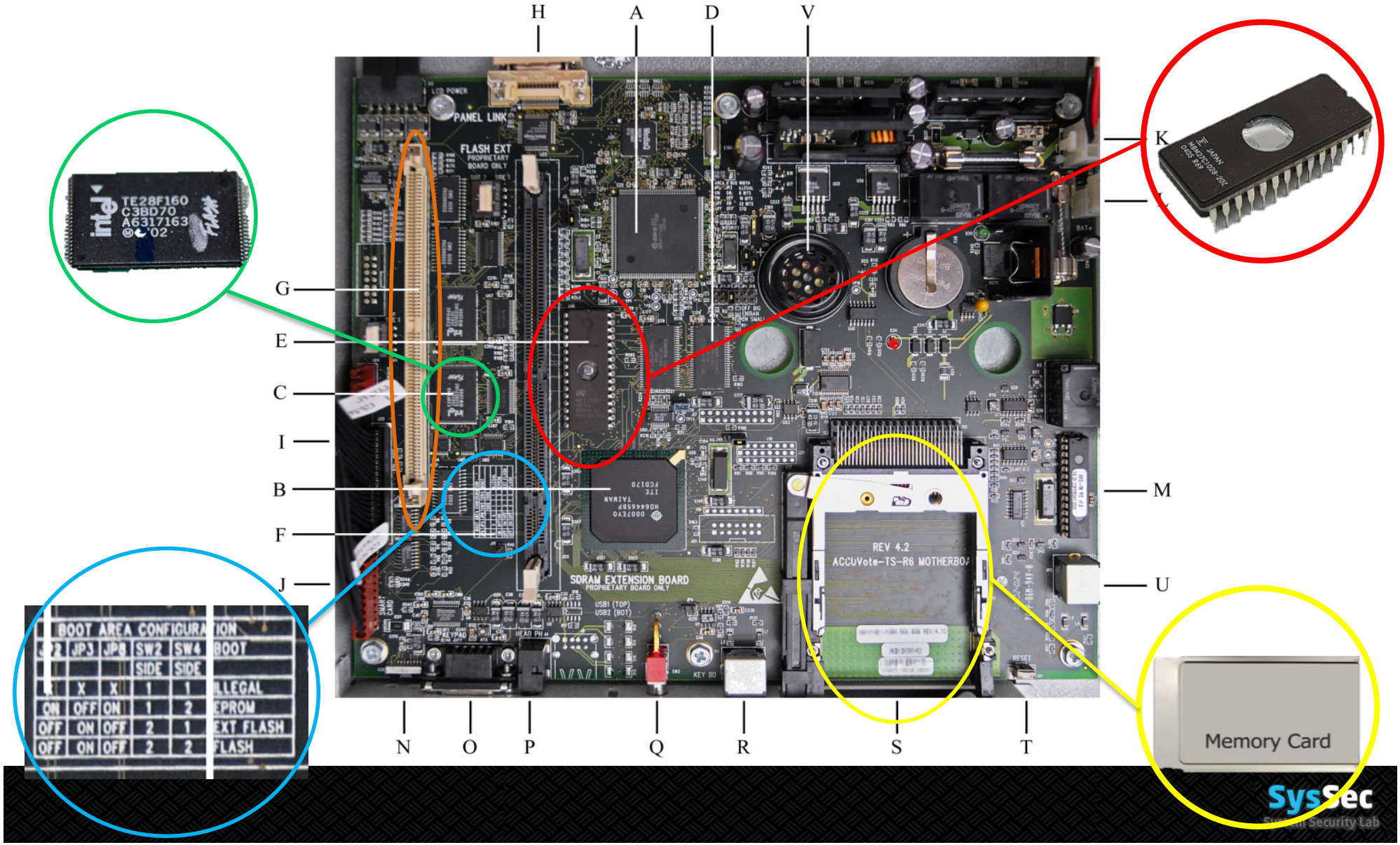
AccuVote-TS Voting Machine

Software	Software
Windows CE	
Direct Recording Electronic (DRE)	



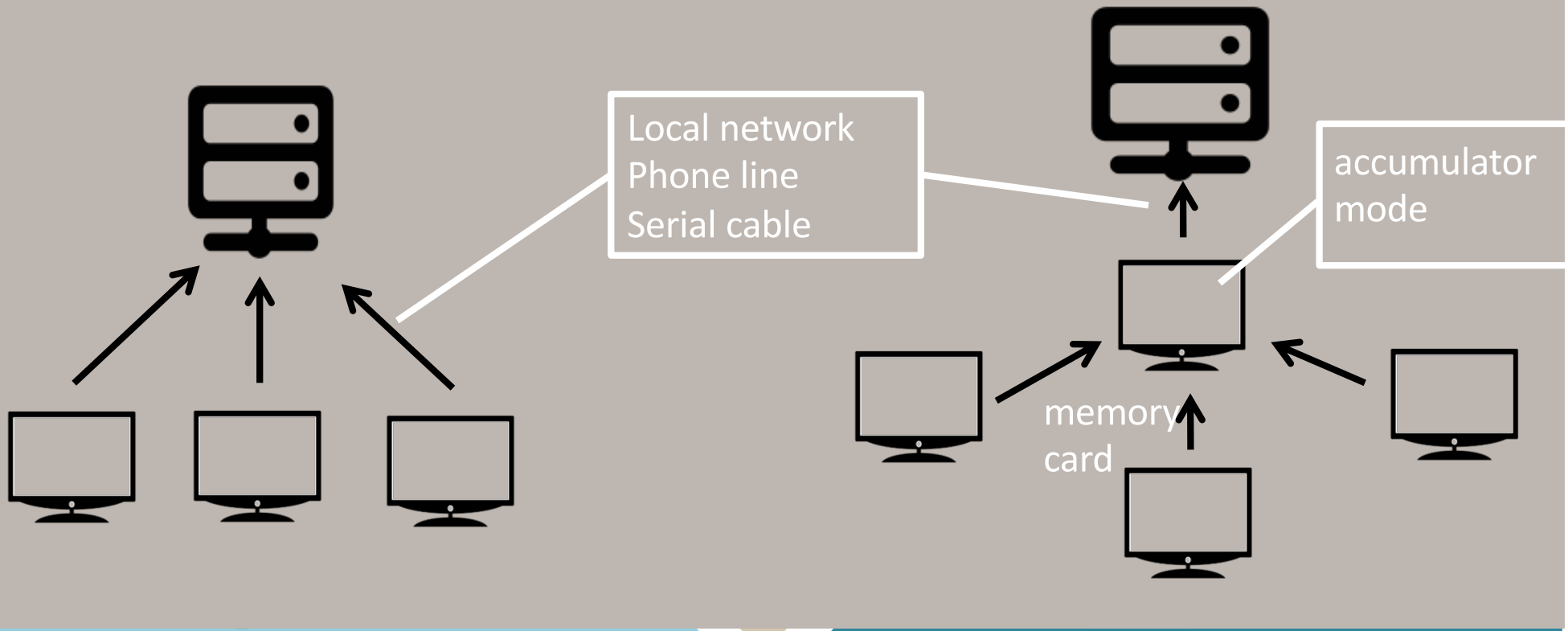
SysSec
System Security Lab

AccuVote-TS Voting Machine



SysSec
System Security Lab

AccuVote-TS Voting Machine



- Voter access card
  (valid -> invalid)

- **On-board Flash memory,**
  Flash memory card

- Local network
- **Accumulator mode**

Attacker's Goal

Attacker's Goal

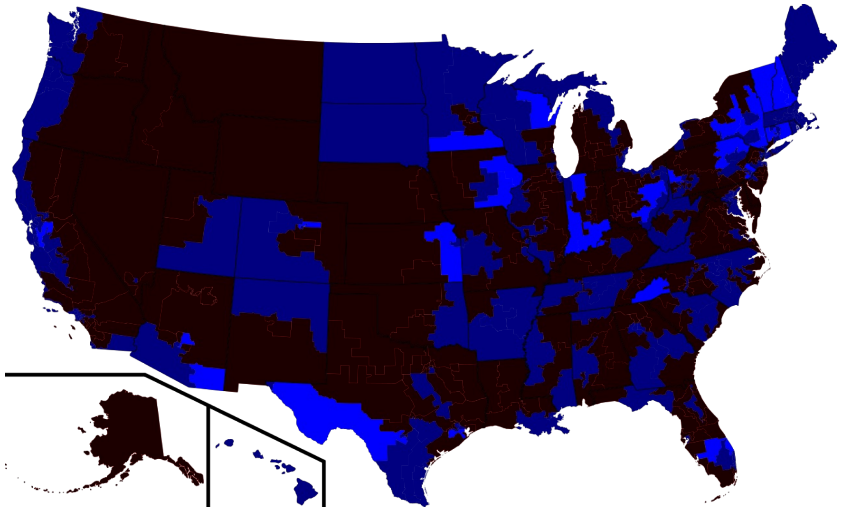
Vote Stealing

Party	%	Vote
Democratic	52.3	42,338,795
Republican	44.3	35,857,334

5% (4,048,777)

Party	%	Vote
Republican	49.3	39,906,111
Democratic	47.3	38,290,018

Denial of Service



Vulnerability

Direct Installation

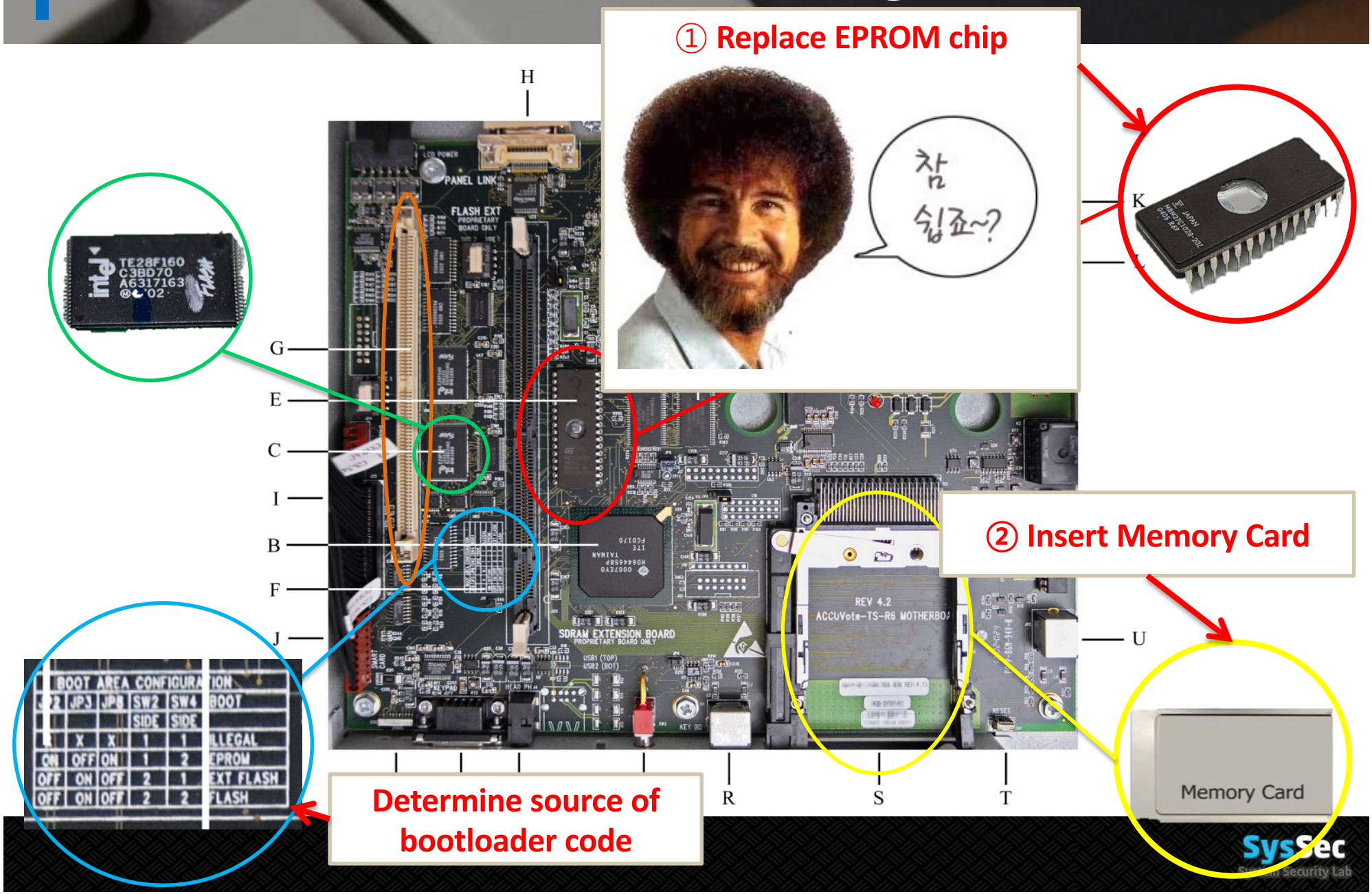
- Easy to physically access to the motherboard
  - EPROM chip, removable memory card, power button
- Source of bootloader code is changeable
  - EPROM chip / On-board flash memory / Memory card
- Not verify authenticity of files
  - fboot.nb0, nk.bin, EraseFFX.bsq, explorer.glb, .ins file

Spreading Virus

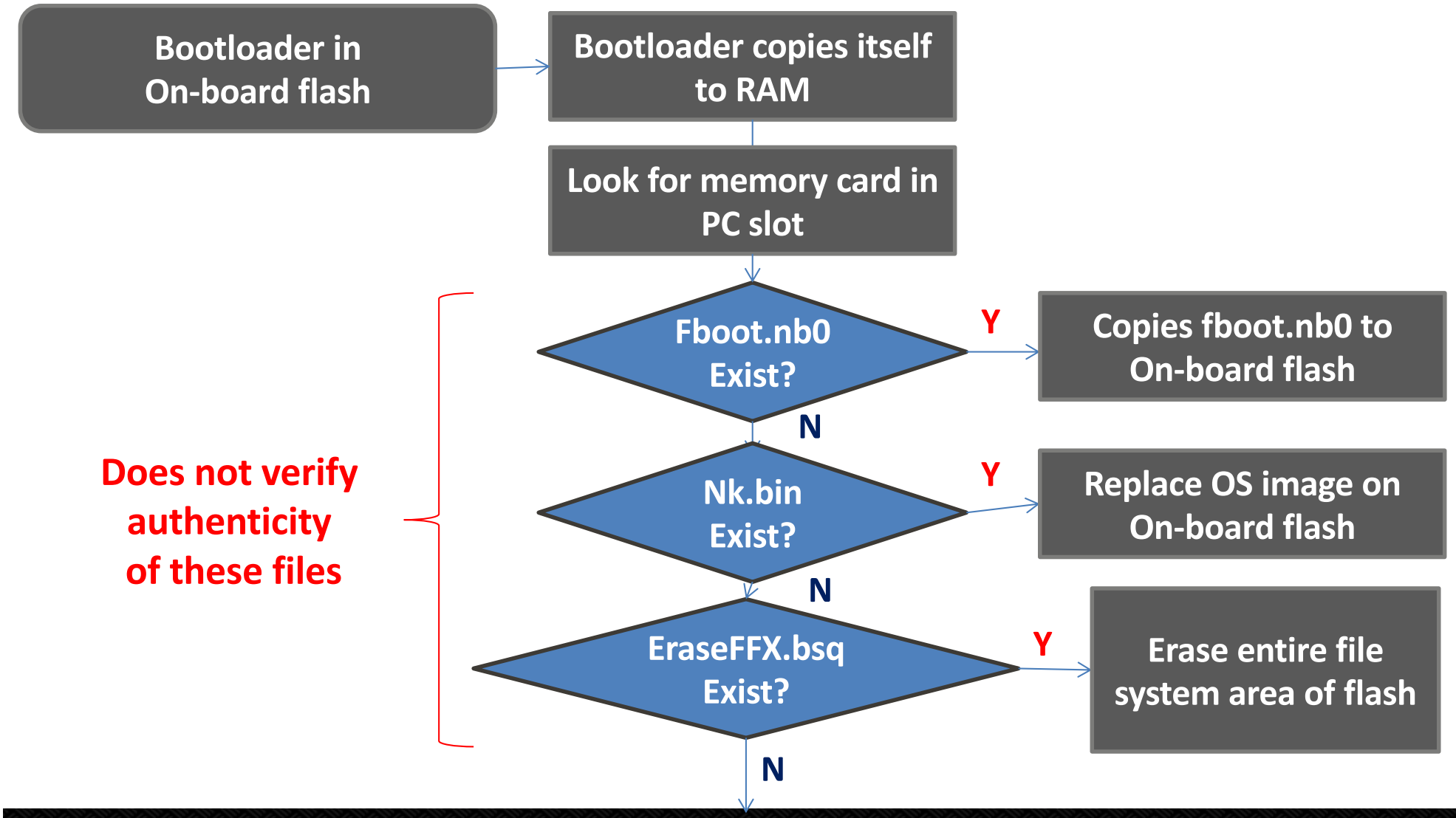
- Removable memory card can spread out virus

Attack Scenario – installing malware

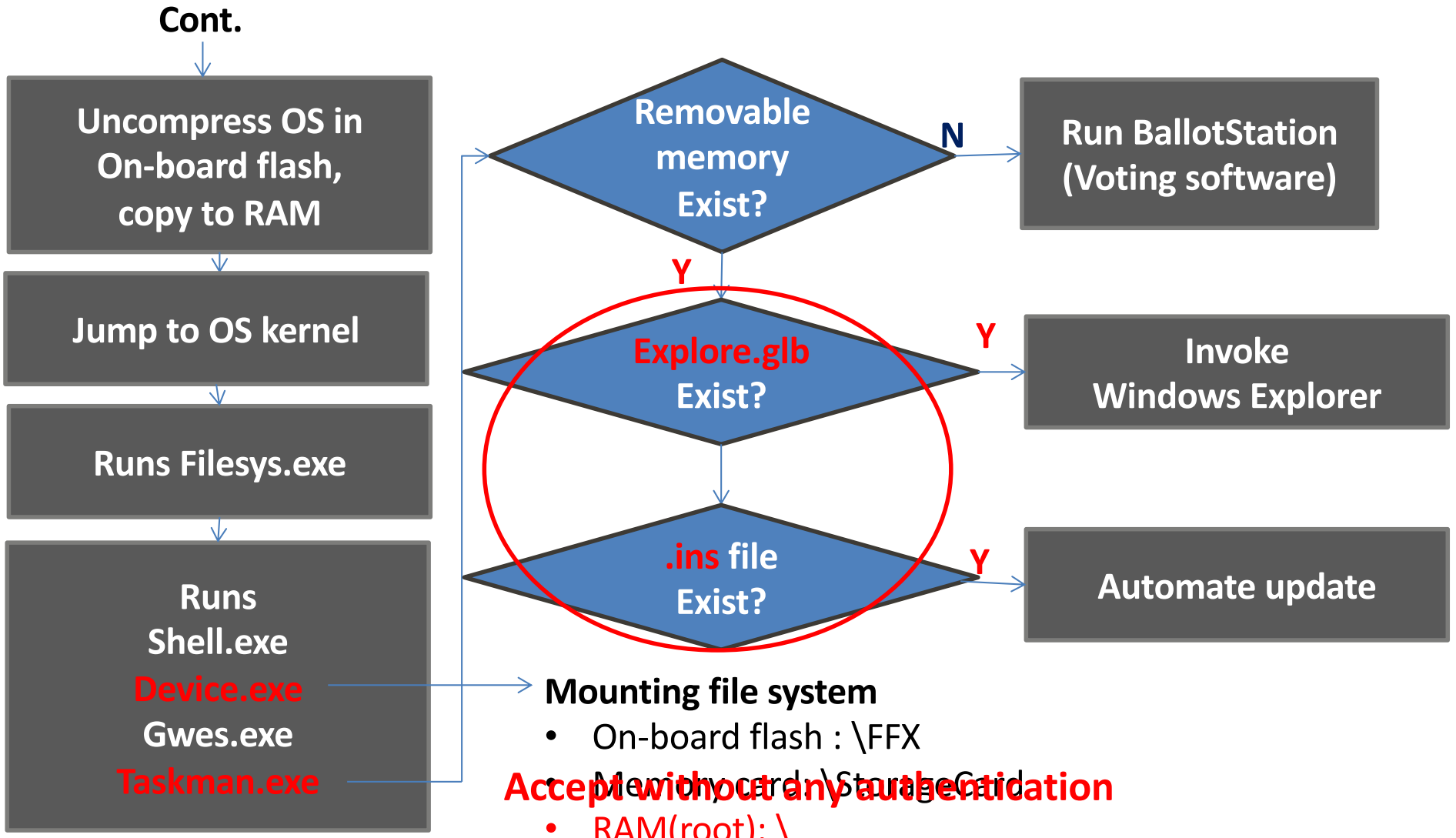
Attack Scenario – installing malware



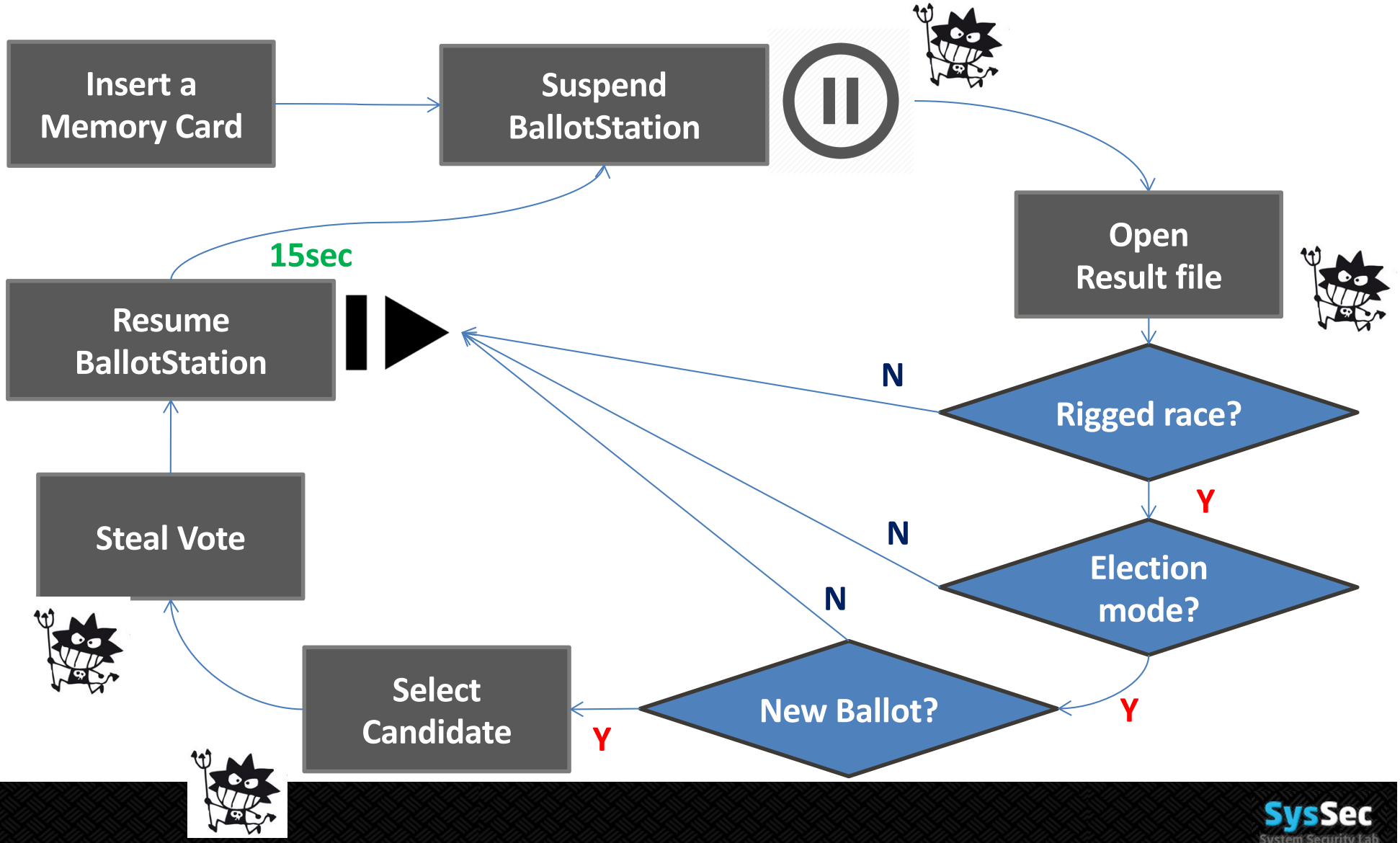
Attack Scenario – installing malware



Attack Scenario – installing malware



Attack Scenario – stealing vote



Mitigation

- S/W & H/W modification
  - Code signing & signature verification
  - Person confirm for software updates
  - Not use rewritable storage -> tamper-proof logs, records
- Physical access control : broken seal cause DoS
- Parallel testing : simulation pattern, secret knock
- Effective certification system : Strong Certifica
- Software independent design : printout paper

Conclusion

- H/W & S/W encompassing study of a widely used DRE
- Demonstration of vote-stealing and virus spreading
- Warning for large scale fraud
- Proving H/W architecture limitation of the target

Limitation & Future work

- General attack idea -> Attack through network
- Malicious action of voters : copy card or re-enable invalid card
- Physical access is not so easy during voting

Another Story – Diebold

`02

`03

`03

Diebold Election Systems to Become Premier Election Solutions

Increased Operational Independence, Concentrated Focus on Elections Systems

Industry Will Strengthen Premier's Competitive Advantage

Aug 16, 2007, 01:00 ET from Premier Election Solutions, Inc.

Diebold CEO resigns after reports of fraud litigation, internal woes

John Byrne

Harri Hursti
- **Hardware & compiled boot-loader**
- Problems with software update

Feldman, Halderman, Felten
- Reverse engineer hardware & software
- Confirmed earlier studies by **demo**

Another Story – Diebold

`07

David Wagner, California
- **TTBR(Top to Bottom Review)**
- "Deep architectural flaws"
- Buffer overflow, weak cryptography

`07

Ohio
- Project EVEREST
- "Yet more vulnerabilities"

`09

Election Systems & Software
Take over

`10

Dominion Voting System
Take over

Electronic voting in Korea

Secure?

> **OOO 당, 왜그러나 또 '선거 조작?'… '1번이 OOO 선장' 괴문자 파문**
>
> K-보팅 주소도 그대로 노출됐다. 비밀 보장을 위해 각 유권자에게 알파벳 6자리로 된 고유번호와 보안코드가 제공됐음에도 특정인의 비밀코드가 고스란히 노출돼 클릭하면 자동 연결된다.

> **'나가수' 뽑은 선관위 전자투표 보안기술 엉터리**

Thanks

Comprehensive Experimental Analyses of Automotive Attack Surfaces

2018.9.27
Hyunki kim

- **Hyunki Kim** S. Checkoway, D. McCoy Roesner, and T. Kohno, "Comprehensi
- **R1 Shuxuan Zhou** Kyong-Tak Cho ar Vulnerable", CCS'16
- **R2 Byungkyu Lee** M. Contag and G. "How They Did It: An Analysis of Emis

Authers: **Stephen Checkoway**, Damon McCoy, Brian Kantor,
Danny Anderson, Hovav Shacham, Stefan Savage, (UCSD)
Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno (UW)

KAIST

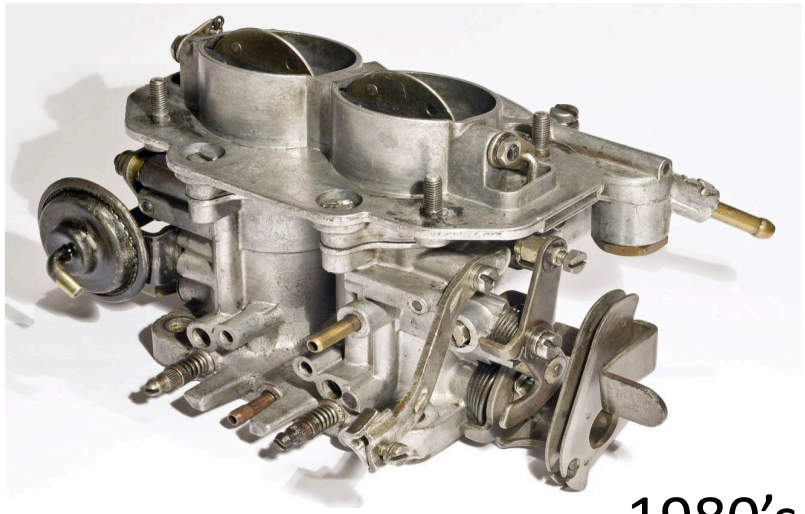
Written by Sanha Park

Intro

Intro

- Jeep Cherokee hacked in 2015

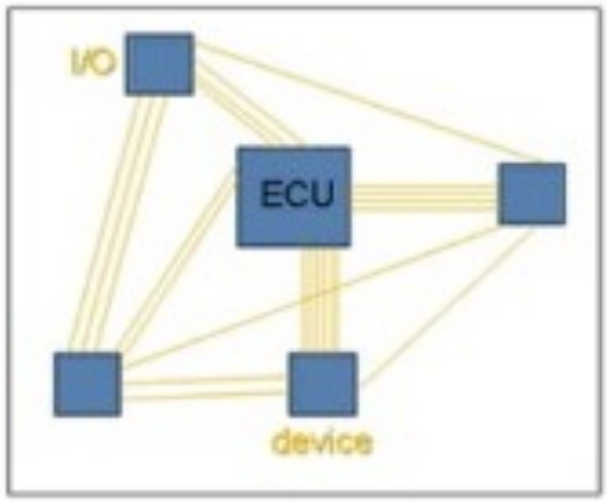
Why can we attack?



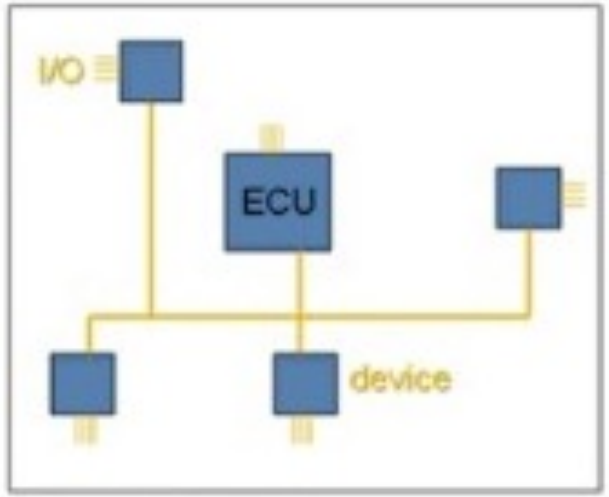
1980's

Today

Without CAN

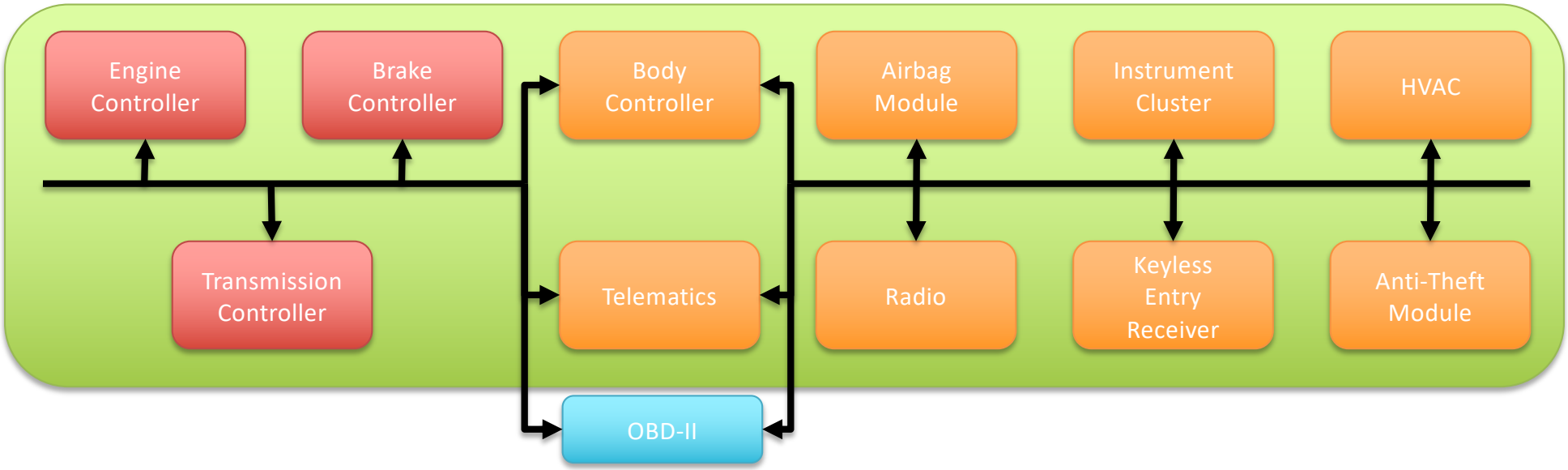


With CAN



Why can we attack?

Cars' system



- ECU(Electronic Control Unit) :
  - Ubiquitous computer controller
- ECU interconnection driven by safety, efficiency, and capability requirements
- But, also has some fatal shortcomings

Oakland 2010, they showed…

- Safety-critical systems can be compromised
  - Selectively enable/disable brakes
  - Stop engine
  - Control lights
- Owning one ECU = total compromise
- ECUs can be reprogrammed (while driving!)

- Limit: Need physical access

[Oakland'10] koscher et al. Experimental Security Analysis of a Modern Automobile.

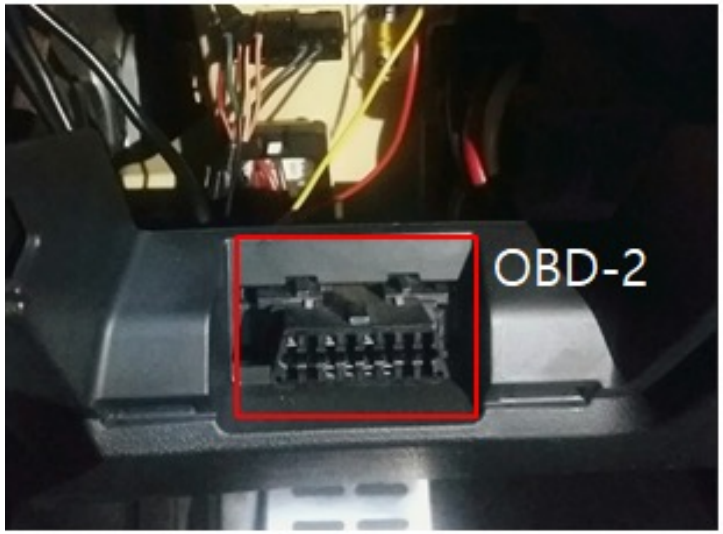
Threat model

- Technical (theoretical) Capabilities
  - Capabilities in analyzing the system
  - Focuses on making technical capabilities realistic

- Operational (real-time) capabilities
  - Show how malicious payload is delivered
  - Attack vector
    - » Indirect physical access
    - » short-range wireless access
    - » long-range wireless access

Indirect physical

- Definition:
  - Attacks over physical interfaces
  - Constrained: Adversary may not **directly** access the physical interfaces herself
- OBD(stands for On Board Diagnostic)



Indirect physical

- Definition:
  - Attacks over physical interfaces
  - Constrained: Adversary may not **directly** access the physical interfaces herself
- Extends attack surface to the device

Short-range wireless

- Definition: Attacks via short-range wireless communication (meters range or less)

Bluetooth

TPMS

Remote key

Immobilizer

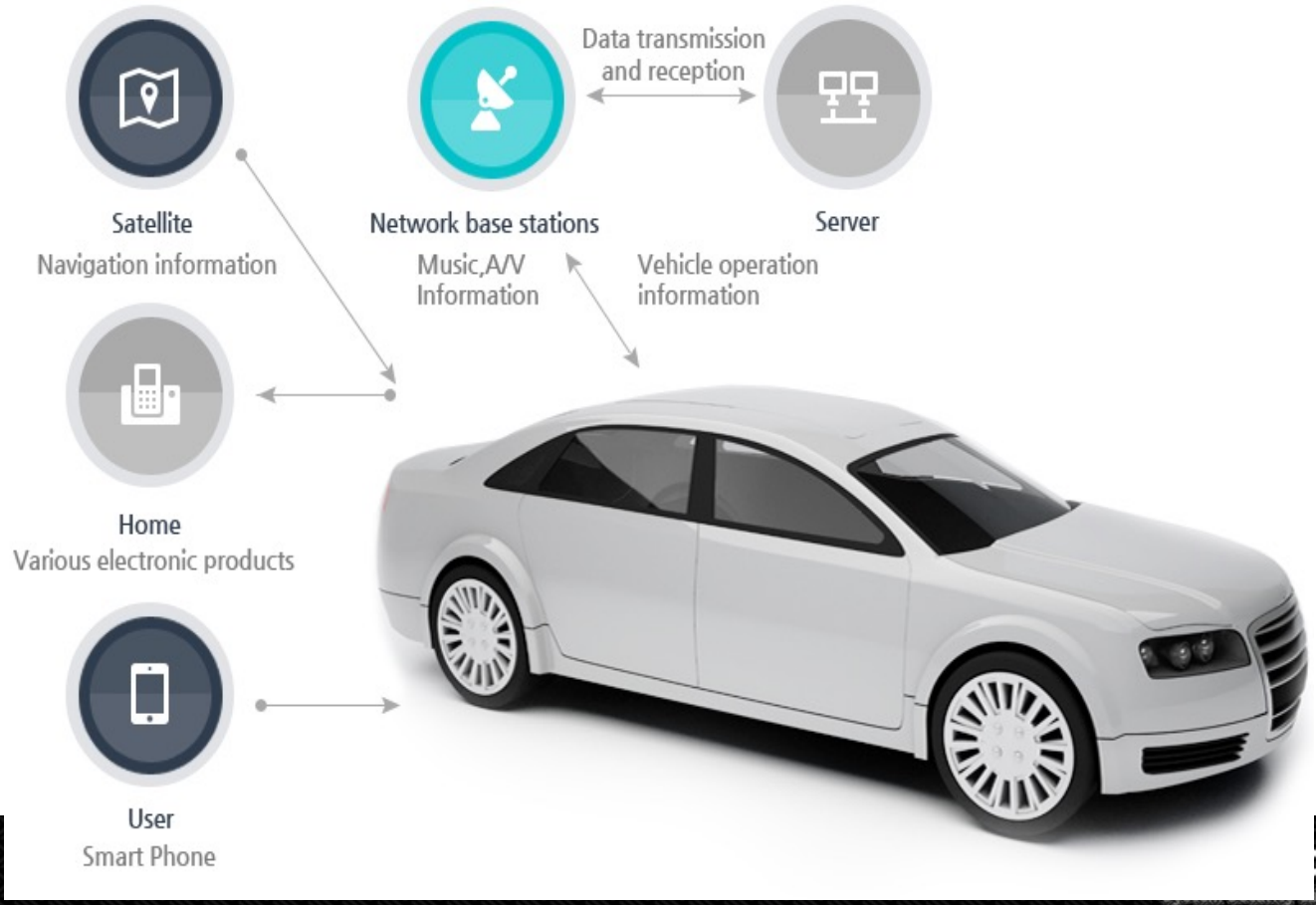
Long-range wireless

- Definition: Attacks via long-rage wireless communication (miles, global-scale)
- Broadcast channel
  - Satellite Radio, GPS, RDS

Satellite Radio

Long-range wireless

- Definition: Attacks via long-rage wireless communication (miles, global-scale)
- Addressable channel
  - Telematics



Attack surfaces explored in depth

- Components we compromised
  - **Indirect physical: Media player, OBDII**
  - **Short-range wireless: Bluetooth**
  - **Long-rage wireless: Cellular**

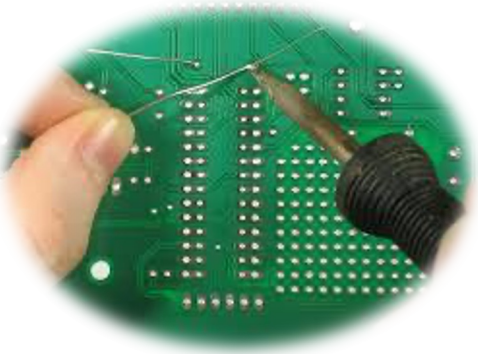
- Every attack vector leads to complete car compromise

Premise

- No direct physical access
- Already know how to deal with CAN signal
- Recent made sedan, 2 same model

Overall methodology

- Extract device's firmware
  - Read memory out over the CAN bus (CarShark)
  - Desolder flash memory chips in ECUs



- Reverse engineering firmware
  - IDA Pro
  - Custom tools

- Identify and test vulnerable code paths

Indirect physical: Media player attack

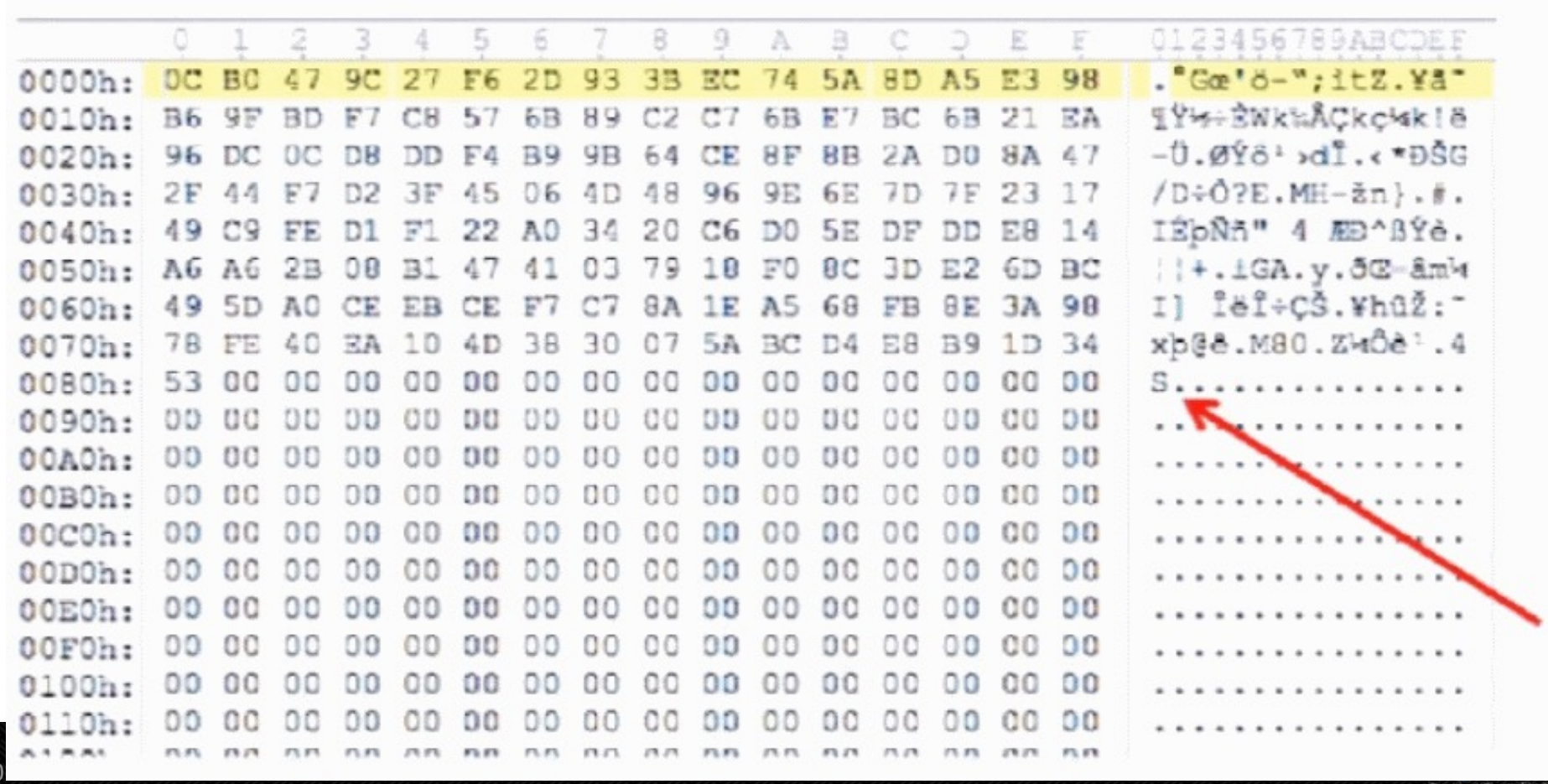
- Code for ISO-9660 leads to
  - Vulnerable : in a module that uploads firmware.

./usr/share/scripts/update/installer/system_module_check.lua

```
 91    local fname= string.format("%s/swdl.iso", os.getenv("USB_STICK")
or "/fs/usb0")
 92    local FLAGPOS=128
 93
 94    local f = io.open(fname, "rb")
 95    if f then
 96       local r, e = f:seek("set", FLAGPOS)
 97       if r and (r == FLAGPOS) then
 98          local x = f:read(1)
 99          if x then
100             if x == "S" then
101                print("system_module_check: skip ISO integrity
check")
```

Indirect physical: Media player attack

- Code for ISO-9660 leads to
  - Vulnerable : in a module that uploads firmware

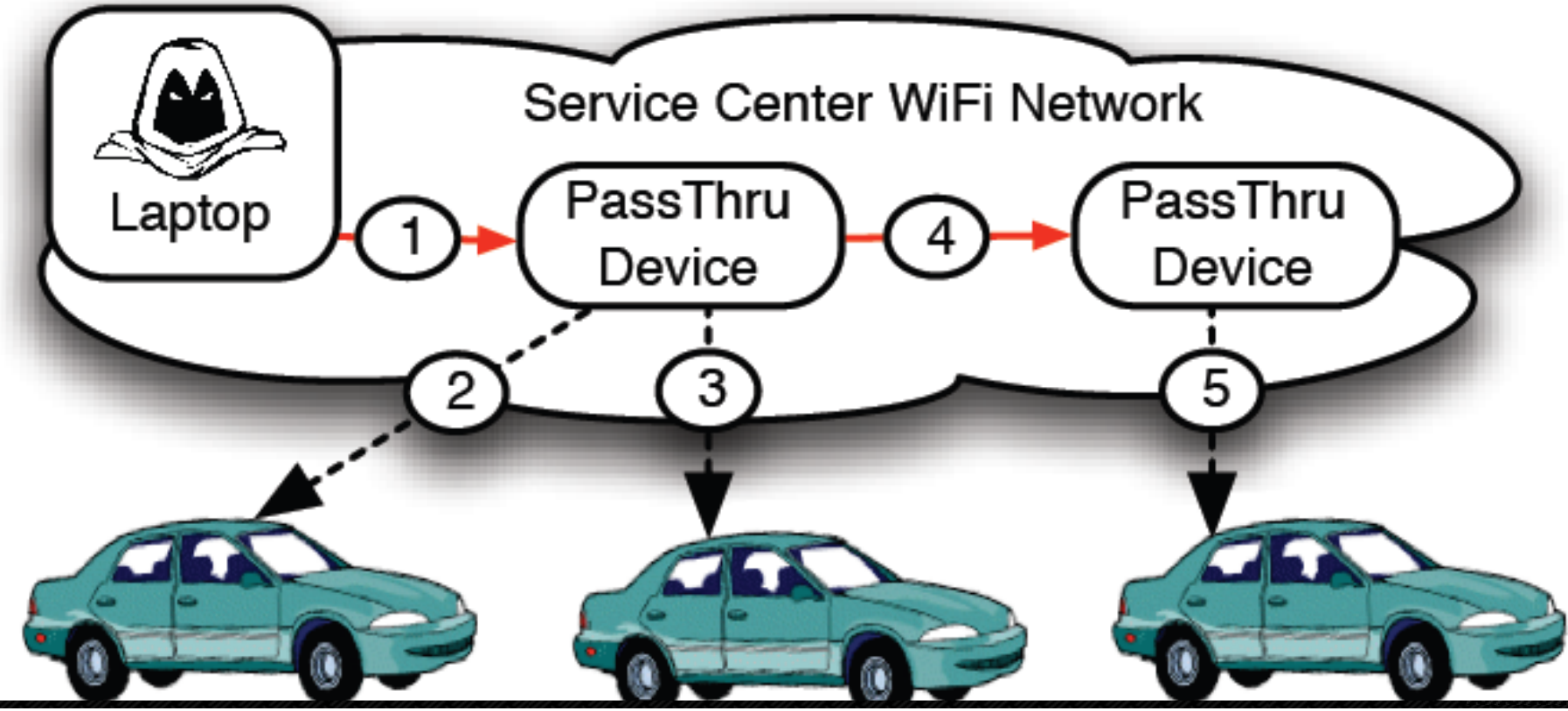


Short-range wireless: OBDII

- PassThru device has no authentication method

1. Connect to same WiFi with device to get to CAN bus

2. Implant malicious code inside the device



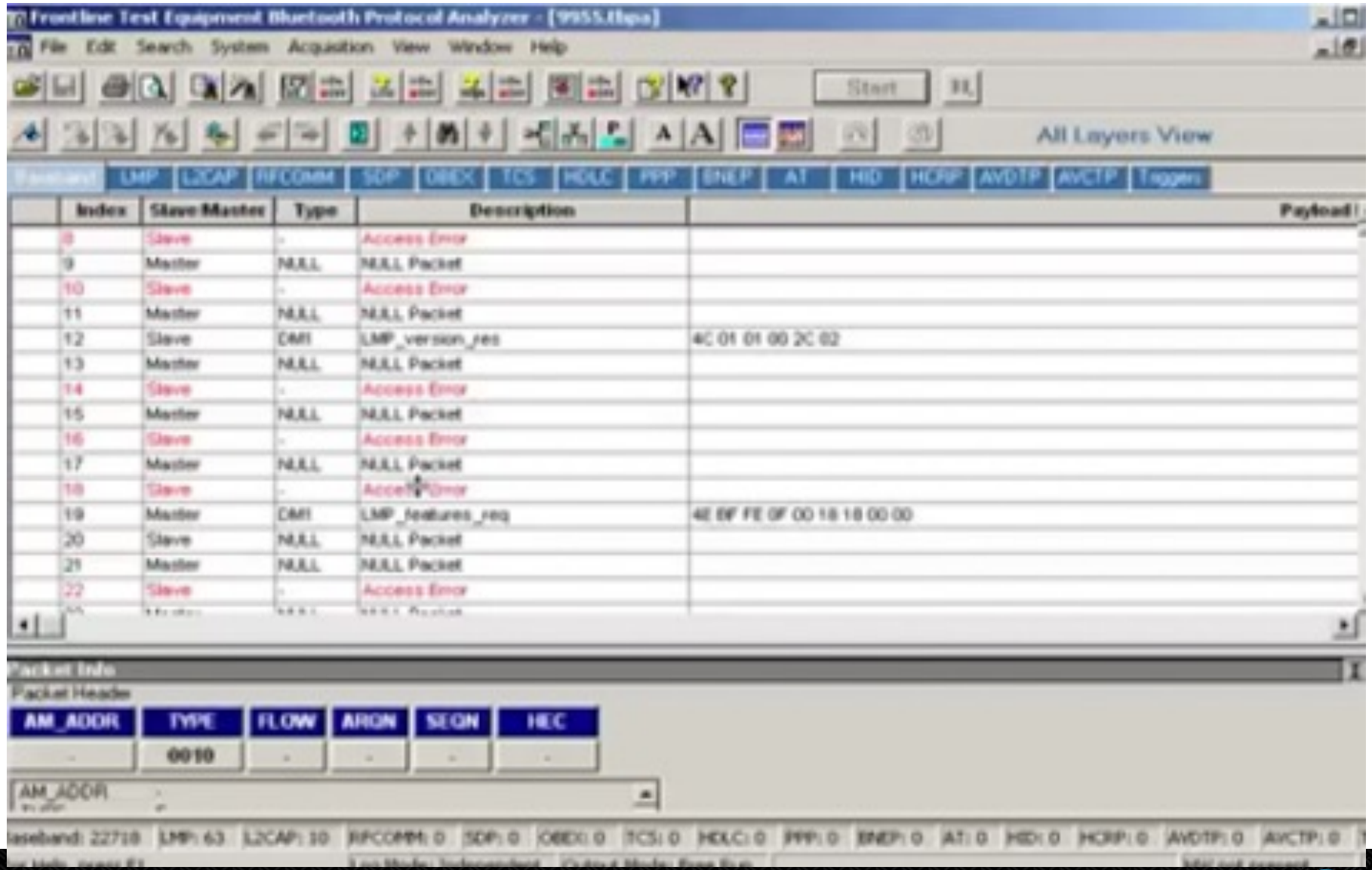
Short-range wireless: Bluetooth attack

- Custom-built code contains vulnerability
  - Strcpy() bug → execute arbitrary code (Bufferoverflow)

1. Using owner's smartphone as stepping-stone
   - Trojan Horse application
   - Check whether other party is telematics unit
     → if so it sends our attack payload

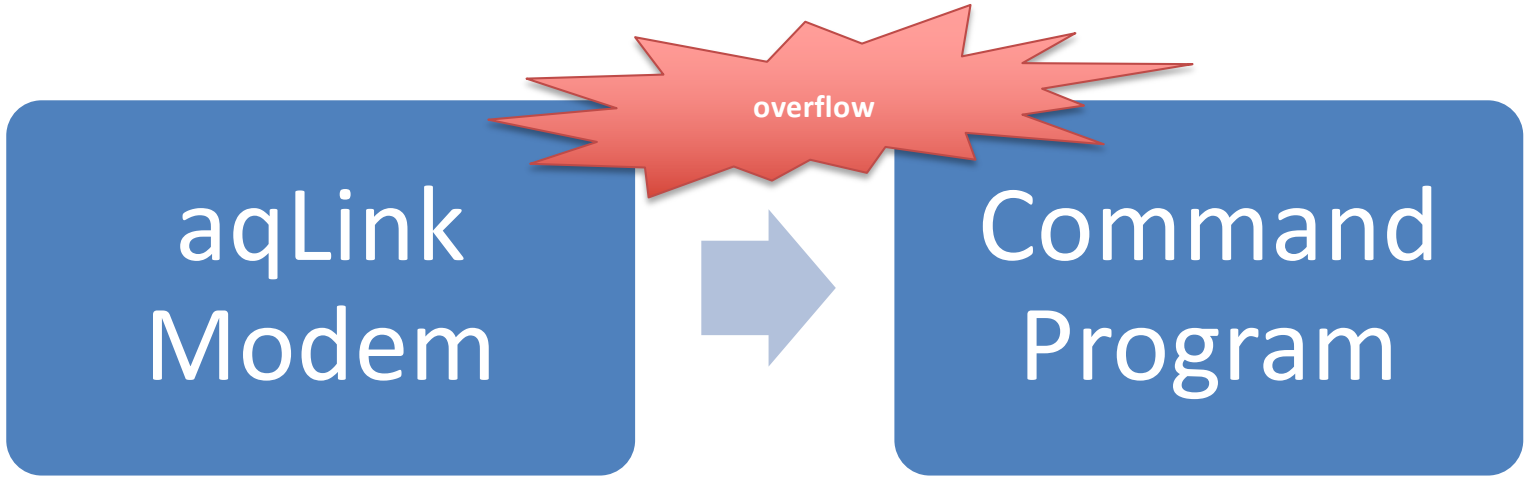
2. Can directly pair with Bluetooth undetectably
   - USRP software radio
   - MAC address ; 2ways to get
   - Brute force PIN ;10hrs per car

Short-range wireless: Bluetooth attack



Long-range wireless: Cellular attack

1. Attack @ Lowest level of protocol stack



Use 1024bytes packet size

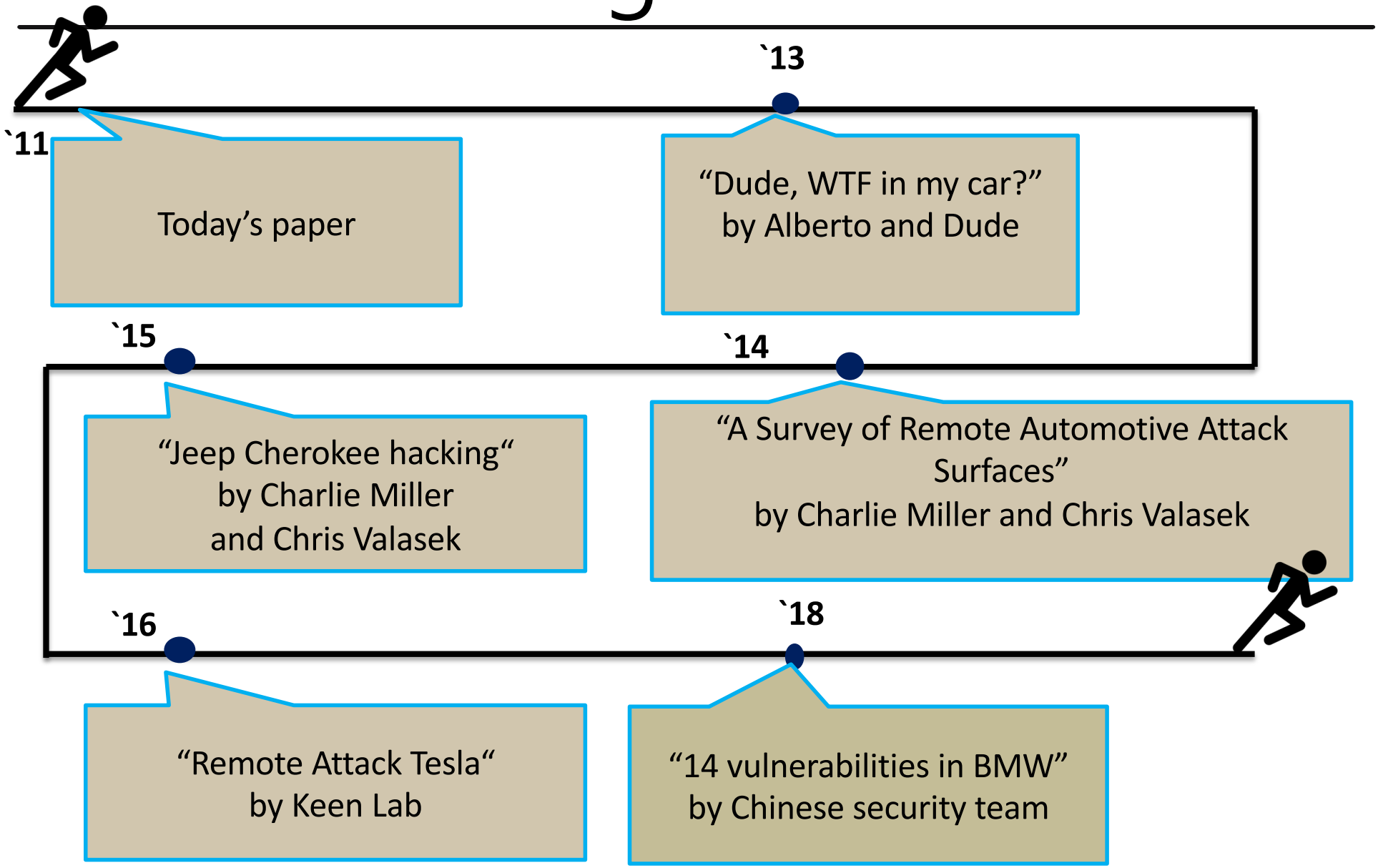
Maximum 100bytes packet

Car theft

1. Compromise car
2. Get Car's INFO (GPS…)
3. Unlock doors
4. Start engine
5. Bypass anti-theft

Where to go from here?

Where to go from here?



Where to go from here?

- Stakeholders responding today:
  - SAE, USCAR, US DOT

- Recommendation : lessons from the PC world
  - Avoid unsafe function
  - Remove unnecessary binaries e.g.) ftp/telnet/vi
  - ASLR (Address Space Layout Randomization)
  - Stack cookies
  - Limited inbound calls

Where to go from here?

Achieve excellence in automotive software security

Penetration testing	Replicate the steps a threat agent takes to find vulnerabilities, and receive clear guidance on how to eliminate them in your server-side applications and APIs.
Dynamic application security testing (DAST)	Identify security vulnerabilities while web applications are running, without the need for source code.
Mobile application security testing (MAST)	Find vulnerabilities regardless of where they exist, including in client-side code, server-side code, third-party libraries, and underlying mobile platforms.
Embedded application security testing (EAST)	Verify the functional and security performance of embedded systems, and identify vulnerabilities in the embedded software stack.
Software composition analysis (SCA)	Detect third-party open source components in source code and binaries. Track and remediate vulnerabilities during development and in containers in production. Identify third-party licenses, and set policies to avoid noncompliance.
Tools	Synopsys provides industry-leading tools for software composition analysis, static code analysis, fuzz testing and protocol testing, and interactive security testing.
Architecture and design	Security testing and threat modeling help you find architectural, design, and system defects and flaws.
Cloud security	Run applications securely in the Cloud.
Agile and CI/CD	Build security into modern agile SDLCs.
Training	Synopsys creates security training courses delivered as instructor-led, eLearning, and virtual classes.
Build Security In programs	Synopsys offers the BSIMM, the Maturity Action Plan, security metrics, and software security initiative programs.

Where to go from here?

- Future work
  - Developing new protocol alternative to CAN bus
  - Research how to encrypt CAN message
  - CAN monitoring system to catch external attack

Summary

- Current autos have broad (and increasing) external attack surface
- They demonstrated real attacks that compromised safety-critical systems
- **Industry and government are responsible**


Q&A

Good Questions from Students

- The authors suggest removing extraneous binaries. Would this approach be any effective in practice, given that the attackers already have arbitrary code execution primitives?
- Why is this paper famous? Where is the evaluation part?
- Mitigations can be bypassed?
- Could one get access to a smartphone's sensitive information through its Bluetooth connection with a compromised car?
- Can't we use security solutions for PCs or smartphones?

Best questions from students

- Seunghyun Lee: How relevant are the attacks presented in the paper as of today, where latest vehicles employ “CAN gateway” that regulates data flow between the CAN bus and infotainment systems
- Hyeon Heo: Can these attack be automated using static analysis or taint analysis?
- Dongok Kim: Cars have their own OS, FSD makes implementation complex. Memory safety issues and logic vulnerabilities are reduced. Considering these, status of quo for automobile security? How can we trade off security vs. cost?
- Balentin Guittard: what about a software whose goal would be to limit and control the access of an ECU to the CAN / other ECUs?

Questions that may not be answered

- Do modern vehicles still vulnerable to these attacks?
- What is the most dangerous path to exploiting automobile security?
- Were there any real world attack cases?

Tesla and GPS Spoofing

GPS Spoofing Effect on
Tesla Autopilot Cruise Speed

Tesla Blinding AEB

Tesla Model S
Camera Blinding Effect on AEB
Demo

DoS Using Fake Base Station

Denial of Service attack using
FAKE base station

CAN Protocol Analysis