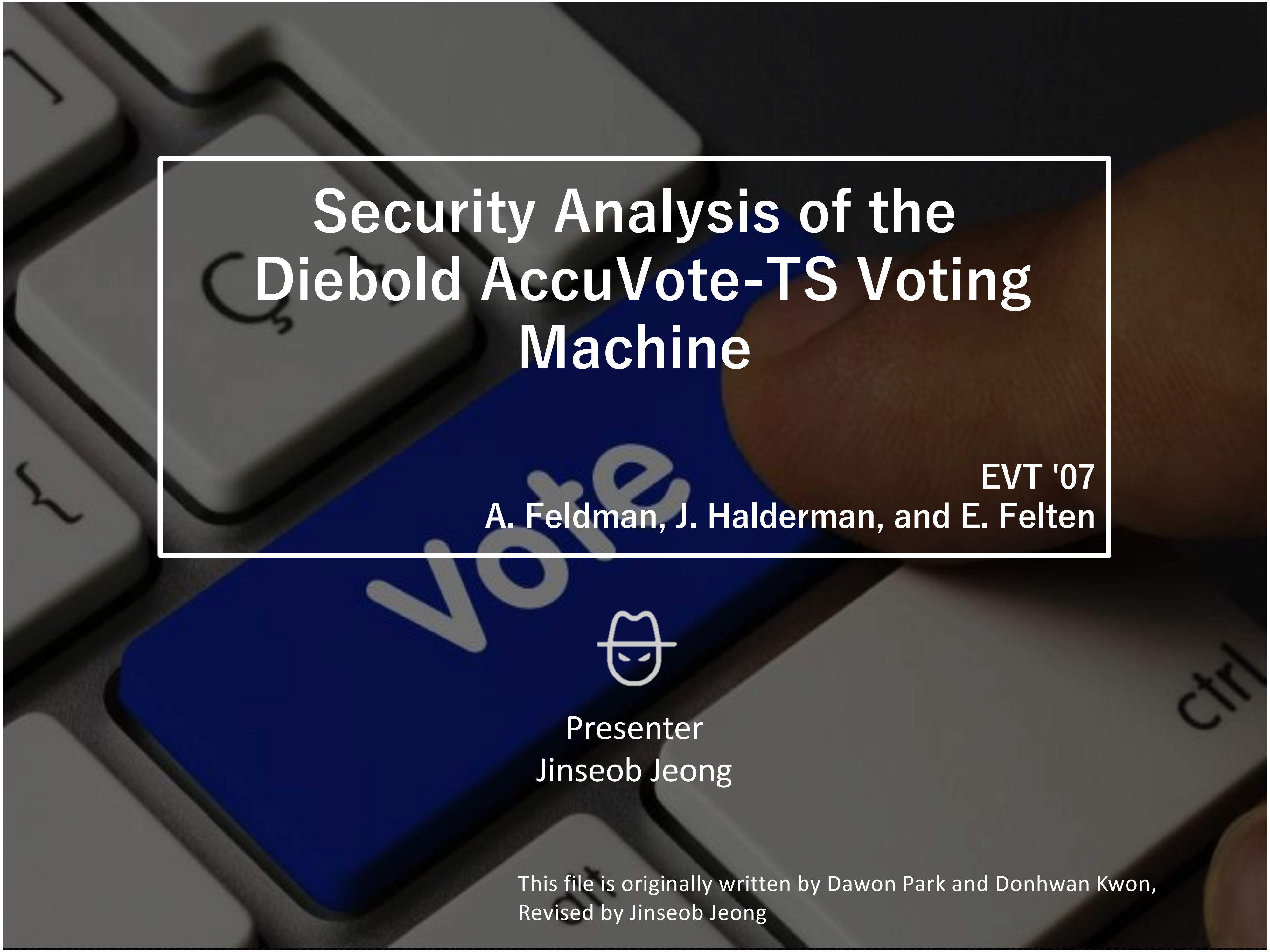
EE515
Security of Emerging Systems

Yongdae Kim

KAIST

Admin

- Find your group members and discuss about projects



Security Analysis of the Diebold AccuVote-TS Voting Machine

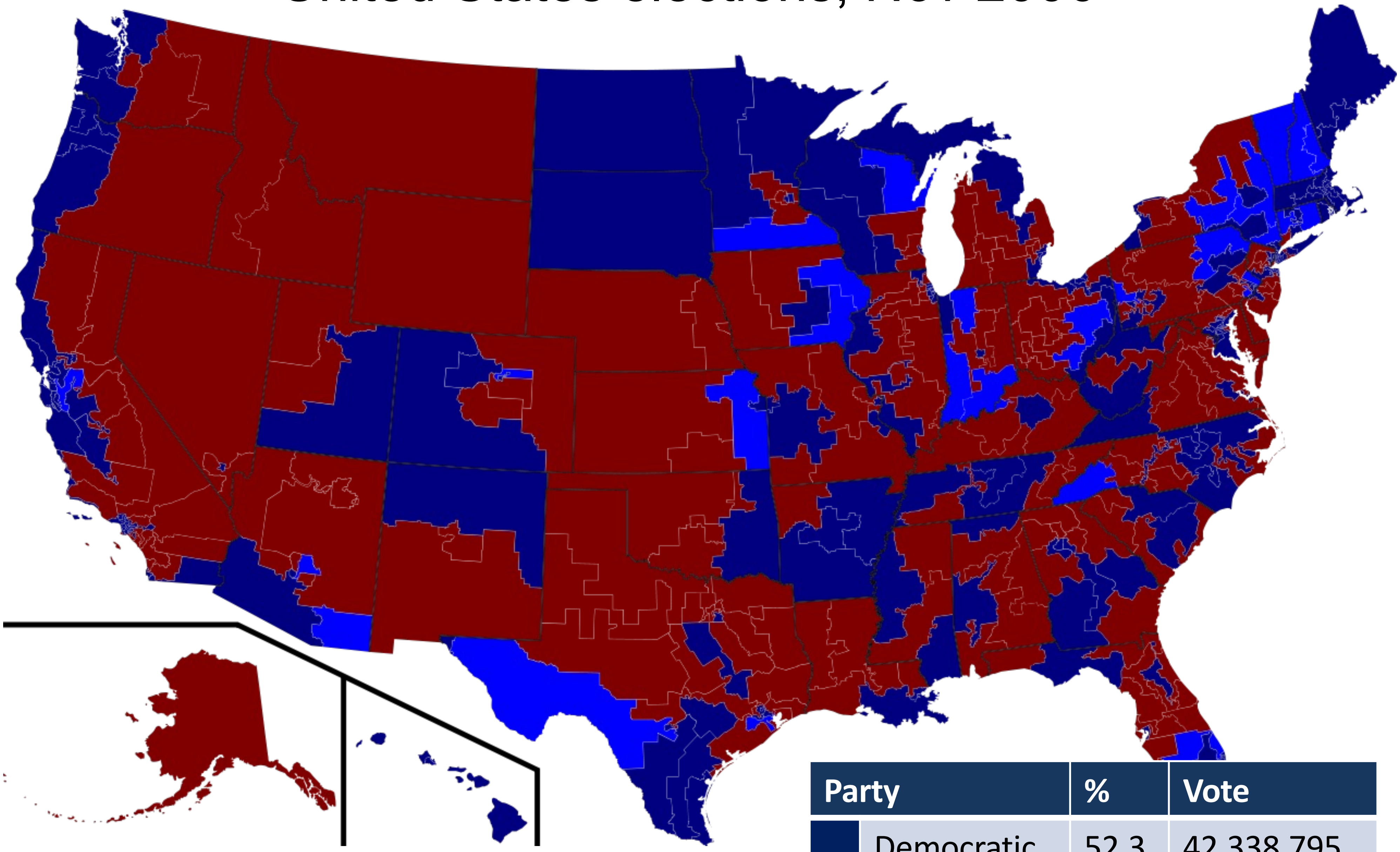
EVT '07

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



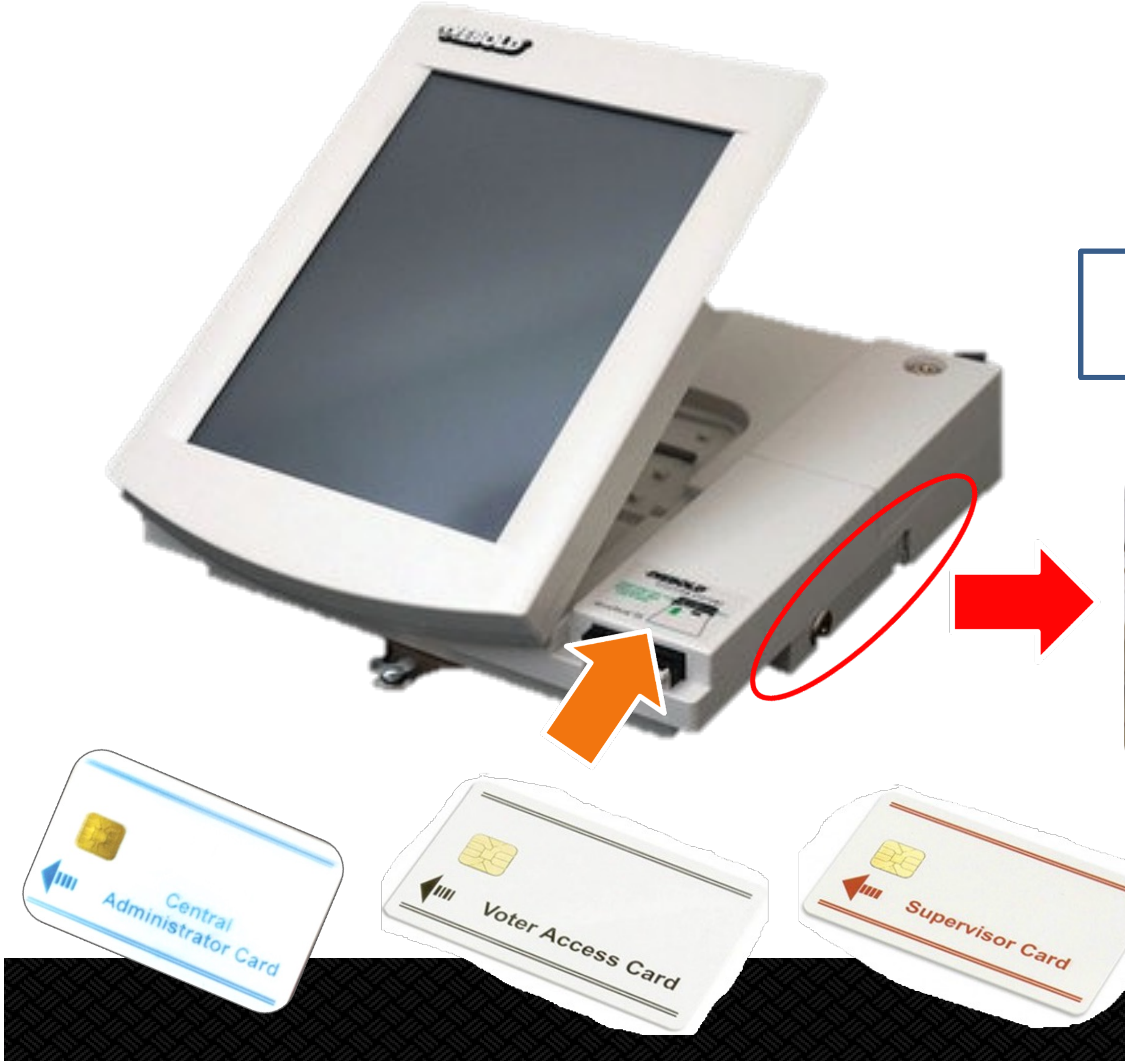
	Party	%	Vote
🟦	Democratic	52.3	42,338,795
🟥	Republican	44.3	35,857,334

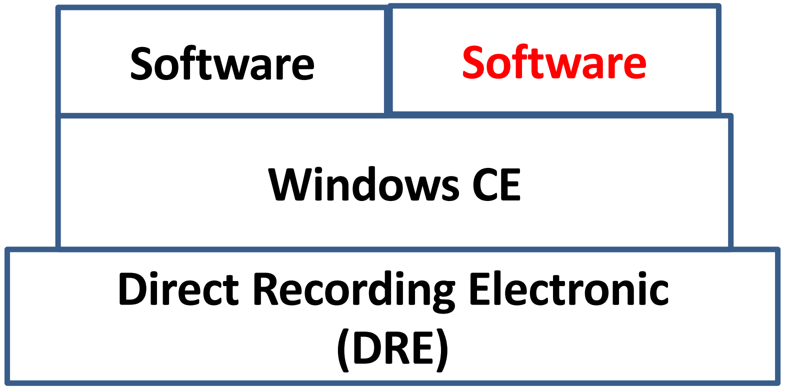
Voting

Paper-based Voting

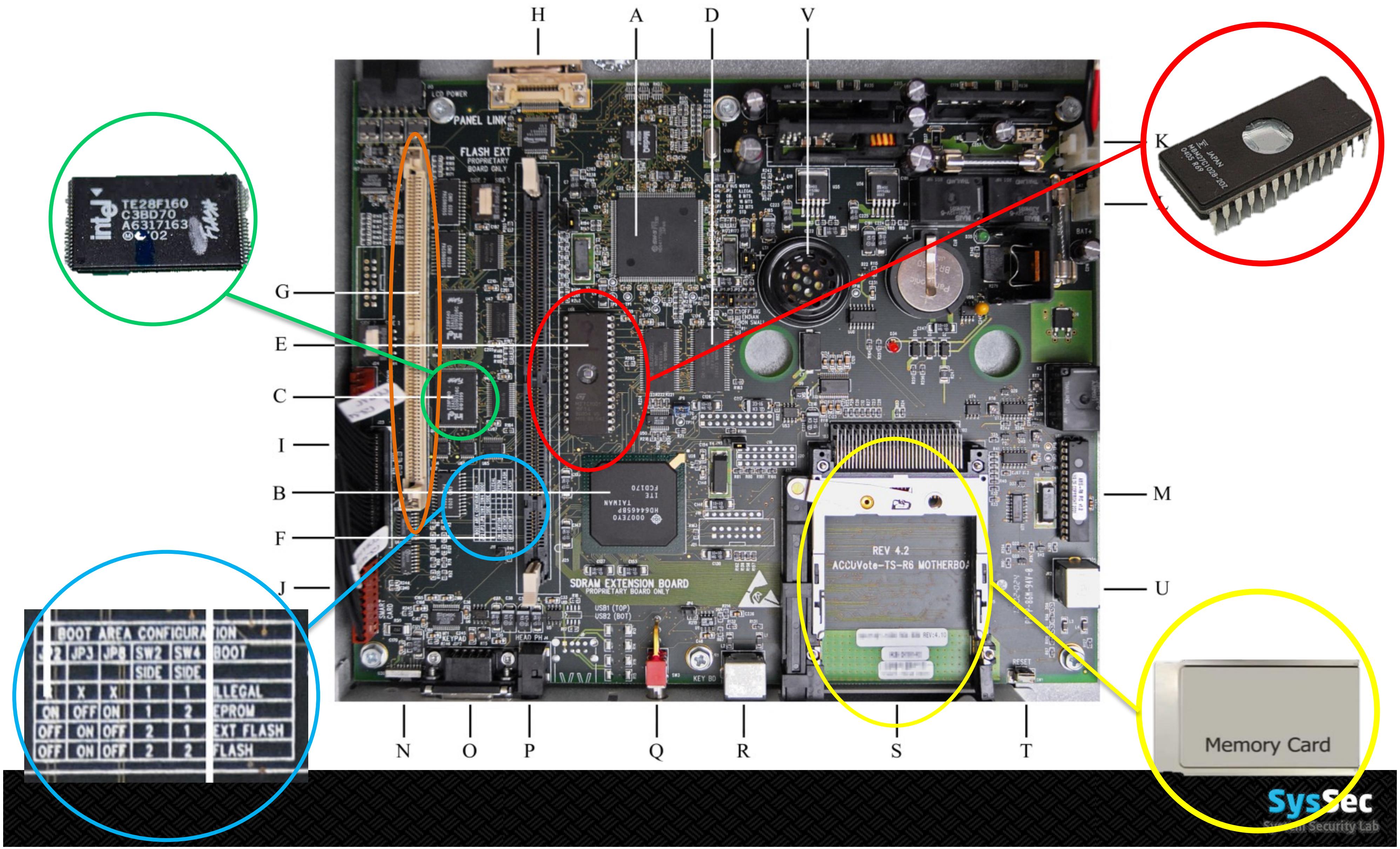
Electronic Voting

AccuVote-TS Voting Machine



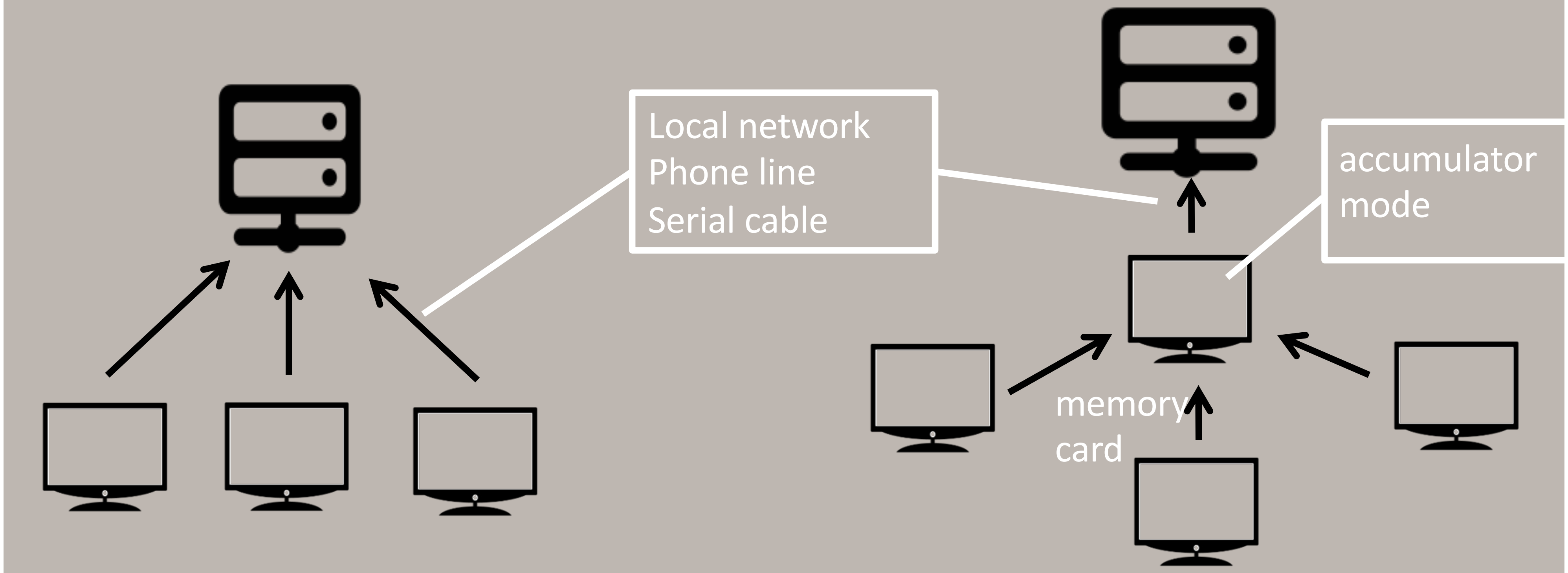


AccuVote-TS Voting Machine



BOOT AREA CONFIGURATION					
JP2	JP3	JP8	SW2	SW4	BOOT
			SIDE	SIDE	
X	X	X	1	1	ILLEGAL
ON	OFF	ON	1	2	EPROM
OFF	ON	OFF	2	1	EXT FLASH
OFF	ON	OFF	2	2	FLASH

AccuVote-TS Voting Machine



Local network
Phone line
Serial cable

accumulator mode

memory card

- Voter access card (valid -> invalid)

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

- Local network
- **Accumulator mode**

Attacker’s Goal

Attacker's Goal

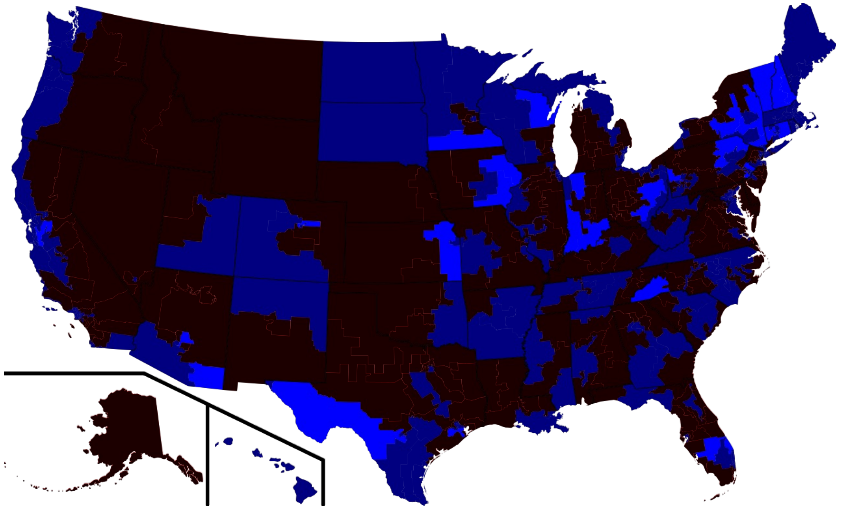
Vote Stealing

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

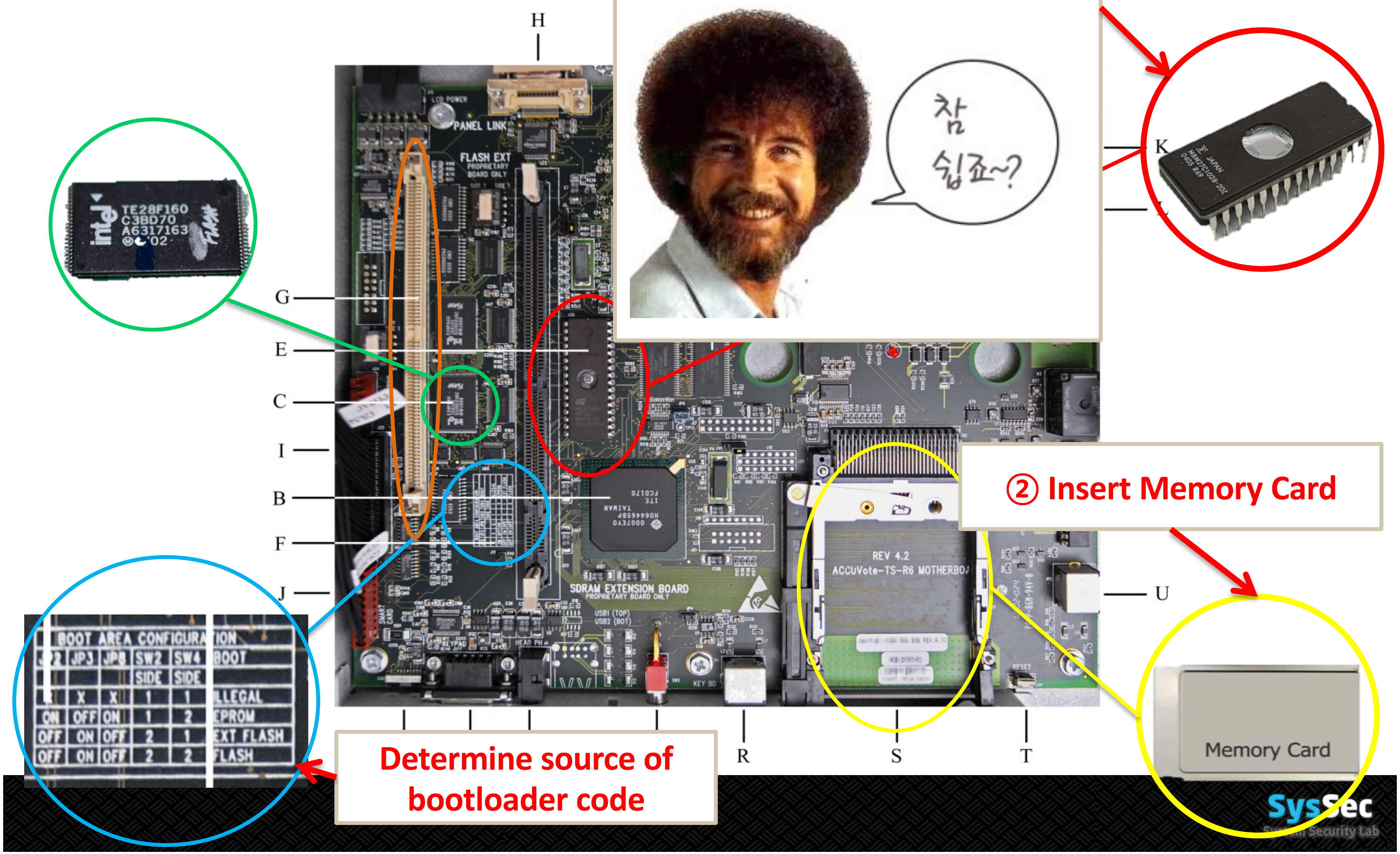
- Easy to physically access to the motherboard
  - EPROM chip, removable memory card, power button
- Source of bootloader code is changeable
  - EPROM chip / On-board flash memory / Memory card
- Not verify authenticity of files
  - fboot.nb0, nk.bin, EraseFFX.bsq, explorer.glb, .ins file

Spreading Virus

- Removable memory card can spread out virus
  - Used for multiple machine, rewritable memory

Attack Scenario – installing malware

Attack Scenario – installing malware



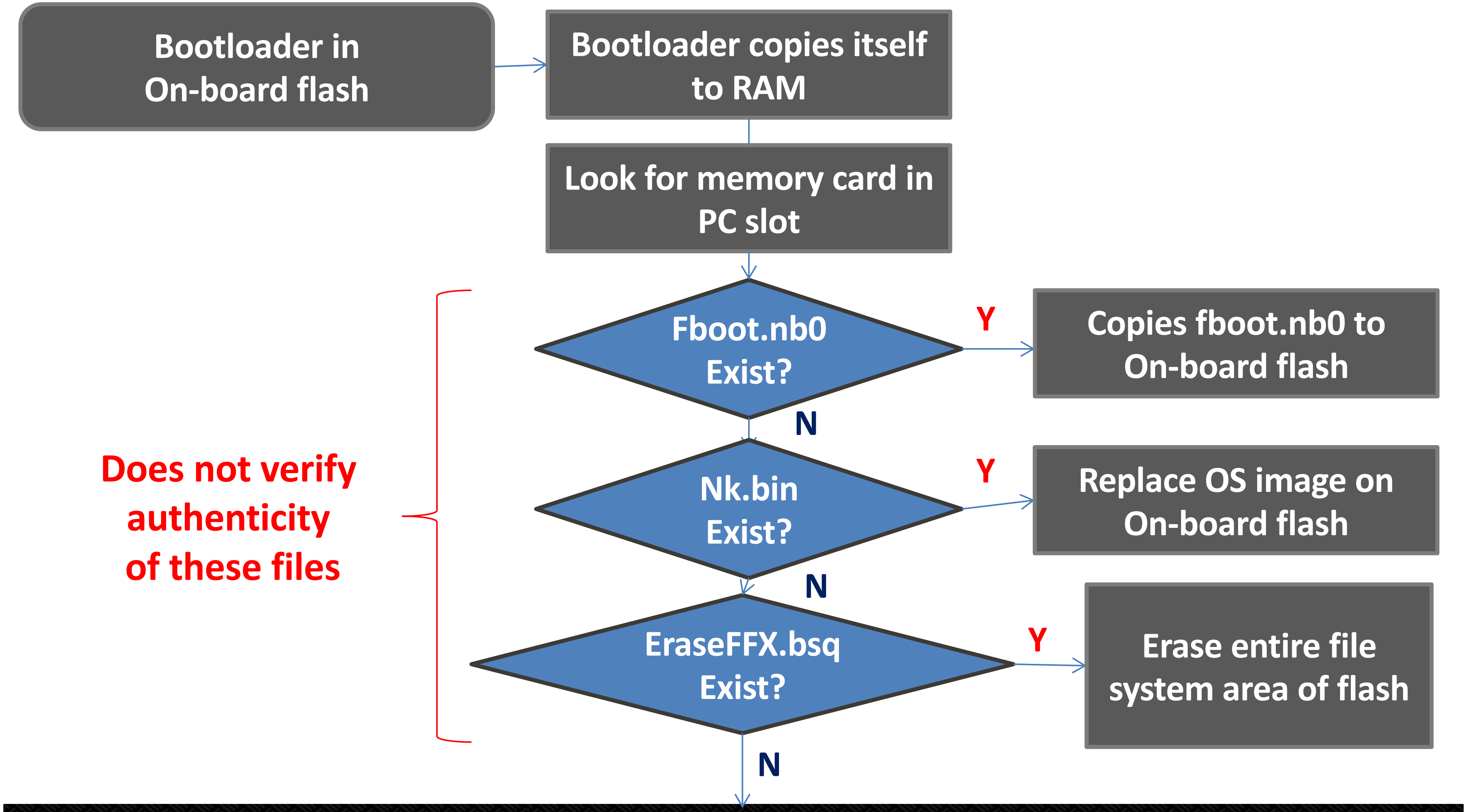
① Replace EPROM chip

② Insert Memory Card

Determine source of bootloader code

BOOT AREA CONFIGURATION					
JP2	JP3	JP8	SW2	SW4	BOOT
			SIDE	SIDE	
[illegible]	X	X	1	1	ILLEGAL
ON	OFF	ON	1	2	EPROM
OFF	ON	OFF	2	1	EXT FLASH
OFF	ON	OFF	2	2	FLASH

Attack Scenario – installing malware



Bootloader in On-board flash

Bootloader copies itself to RAM

Look for memory card in PC slot

Fboot.nb0 Exist?

Y

Copies fboot.nb0 to On-board flash

N

Nk.bin Exist?

Y

Replace OS image on On-board flash

N

EraseFFX.bsq Exist?

Y

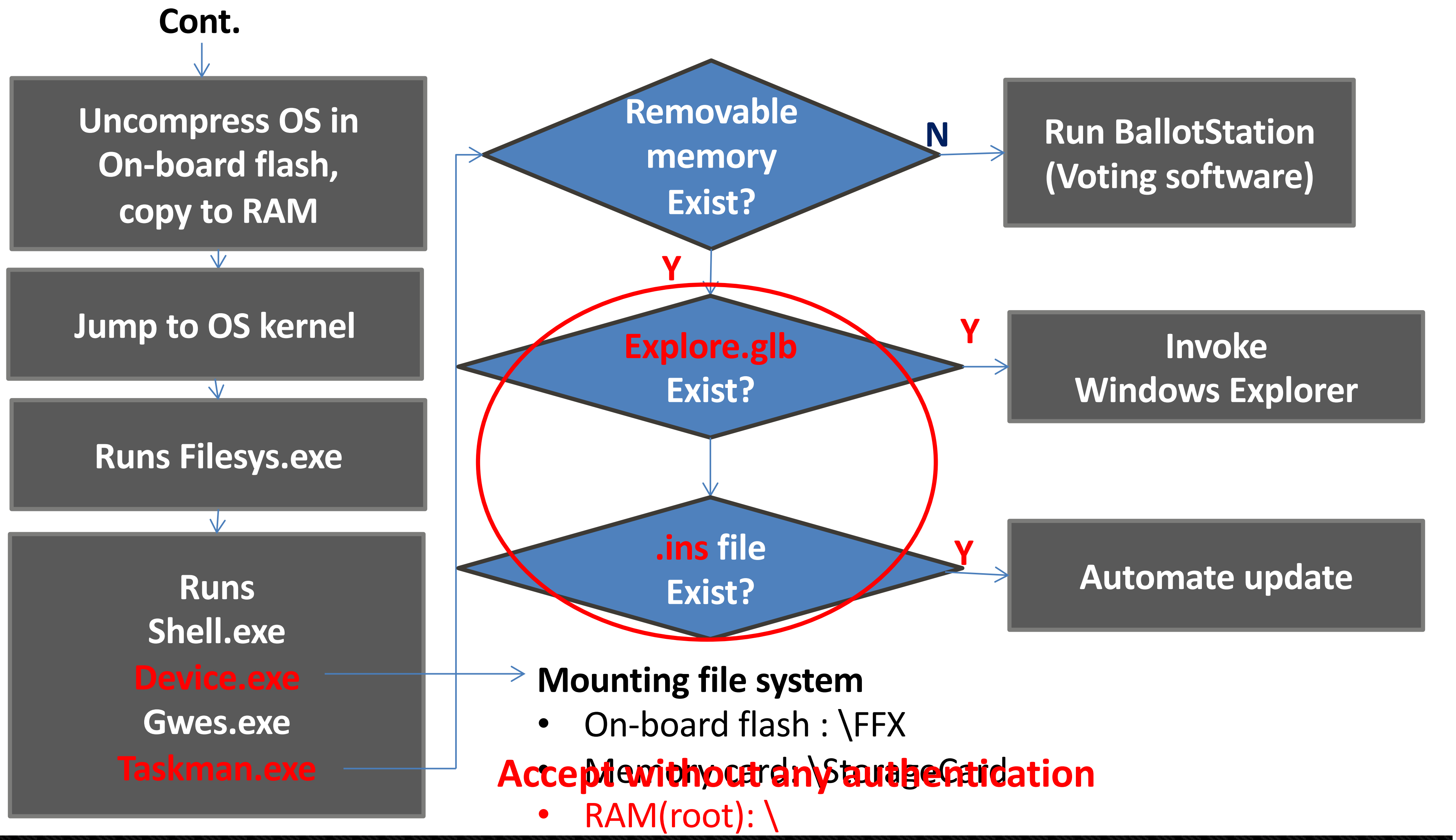
Erase entire file system area of flash

N

Cont.

Does not verify authenticity of these files

Attack Scenario – installing malware



Cont.

Uncompress OS in On-board flash, copy to RAM

Jump to OS kernel

Runs Filesys.exe

Runs
Shell.exe
Device.exe
Gwes.exe
Taskman.exe

Removable memory Exist?

N

Run BallotStation (Voting software)

Y

Explore.glb Exist?

Y

Invoke Windows Explorer

.ins file Exist?

Y

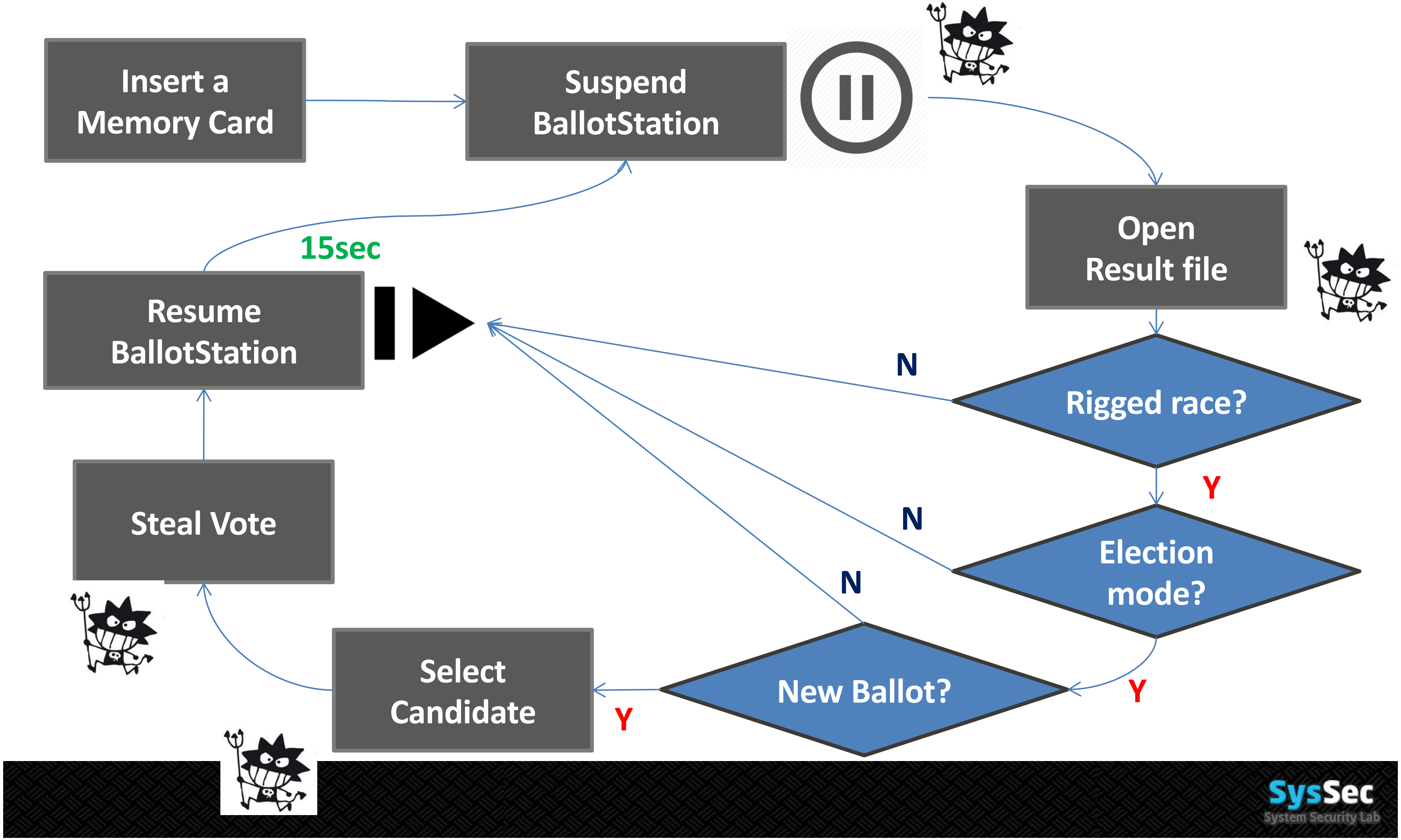
Automate update

Mounting file system

- On-board flash : \FFX
- Memory card: \StorageCard
- RAM(root): \

Accept without any authentication

Attack Scenario – stealing vote



Insert a Memory Card

Suspend BallotStation

Open Result file

Rigged race?

Election mode?

New Ballot?

Select Candidate

Steal Vote

Resume BallotStation

15sec

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

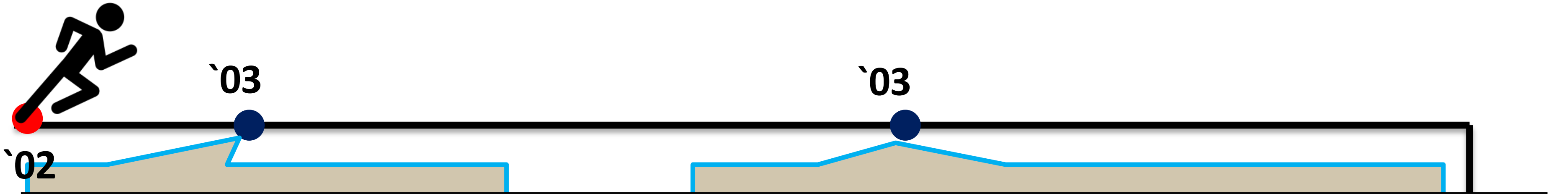
Conclusion

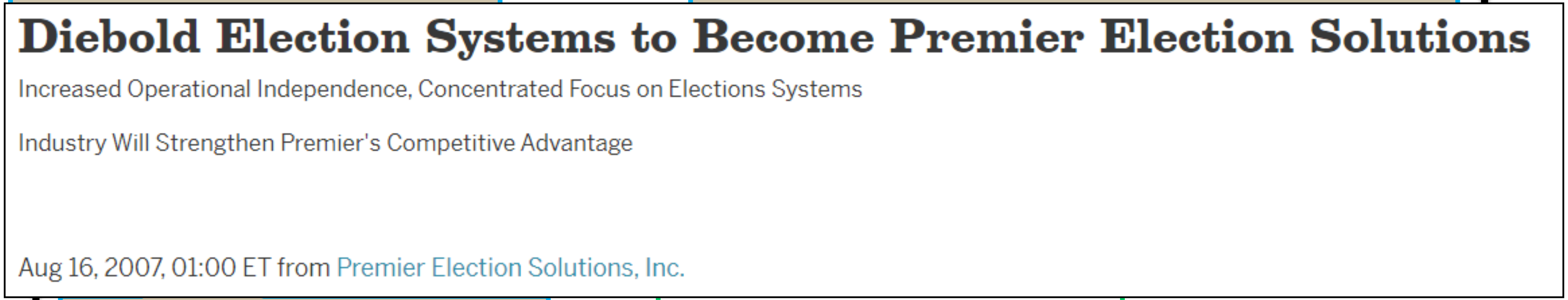
- H/W & S/W encompassing study of a widely used DRE
- Demonstration of vote-stealing and virus spreading
- Warning for large scale fraud
- Proving H/W architecture limitation of the target

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



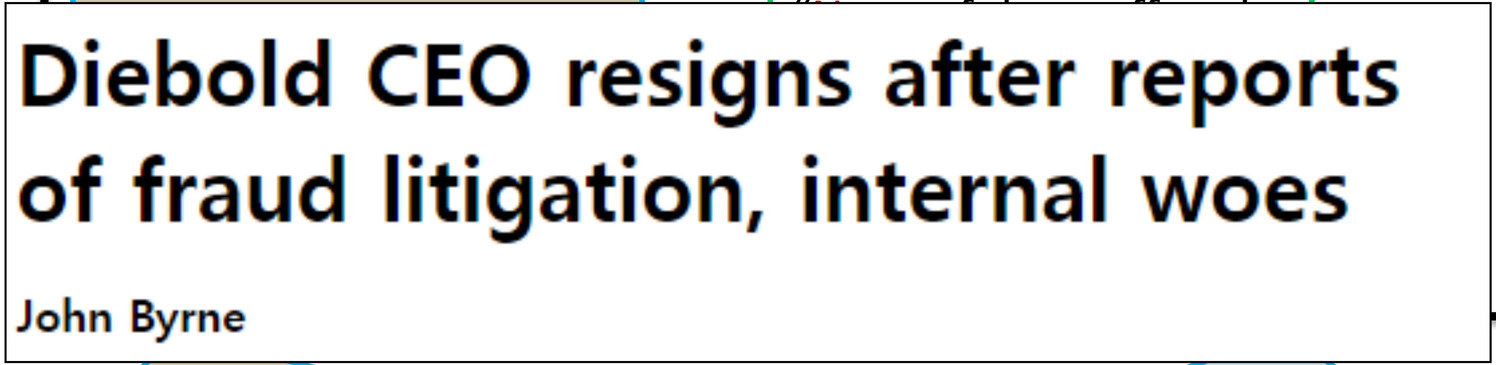


Diebold Election Systems to Become Premier Election Solutions

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

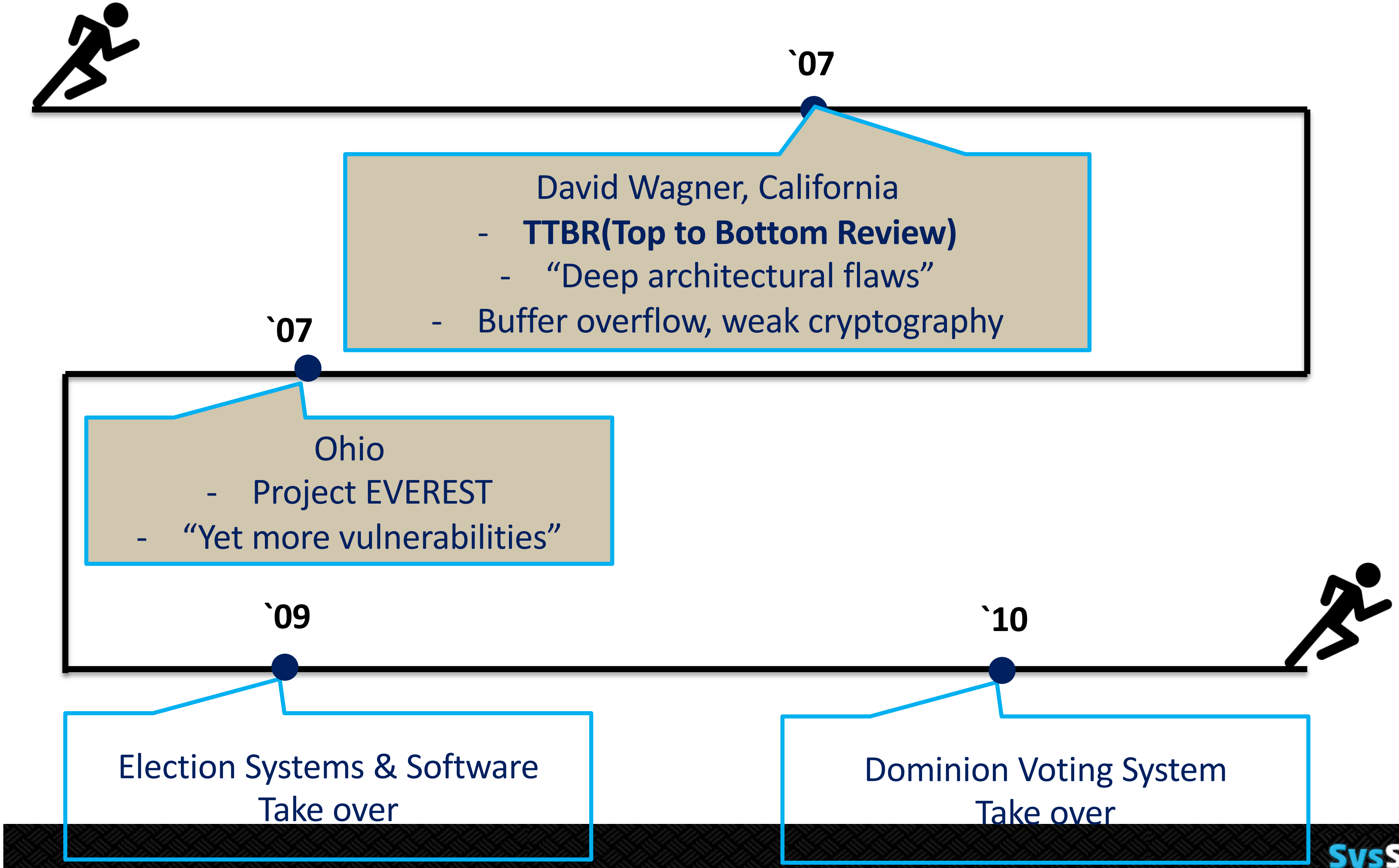
Harri Hursti

- **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

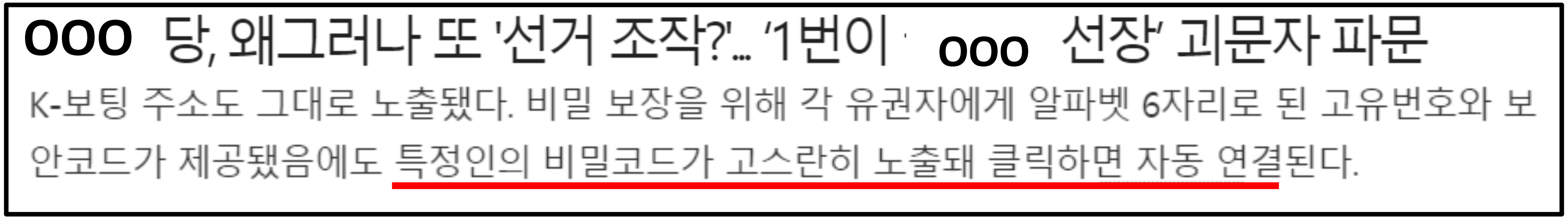
Election Systems & Software
Take over

`10

Dominion Voting System
Take over

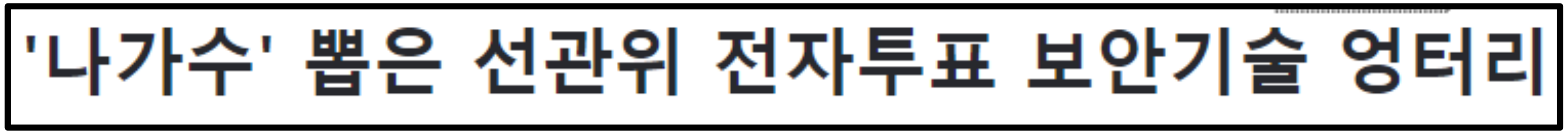
Electronic voting in Korea

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.



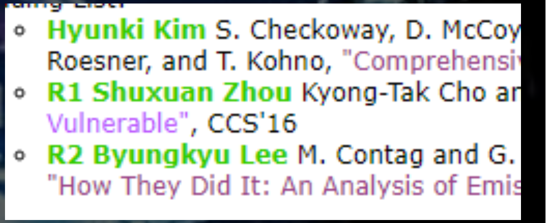
'나가수' 뽑은 선관위 전자투표 보안기술 엉터리

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)

Written by Sanha Park

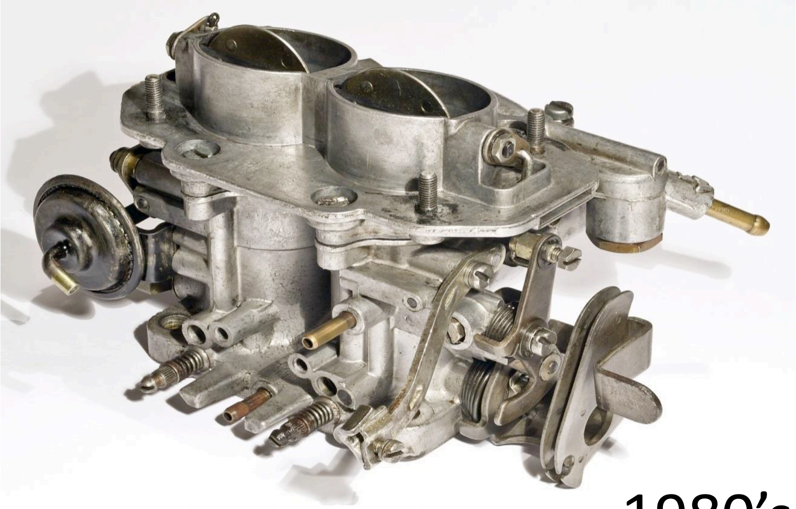
Intro

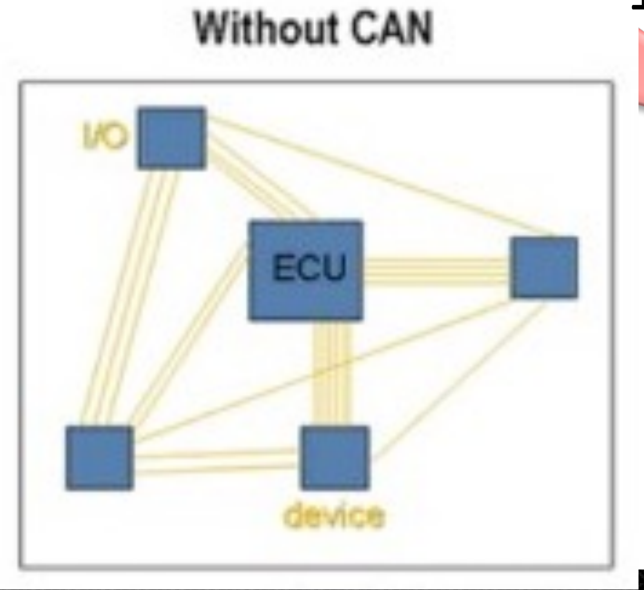
Intro

- ❑ Jeep Cherokee hacked in 2015

HK4 W7M
LOU FUSZ
CHARLIE MILLER
SECURITY ENGINEER, TWITTER
CHRIS VALASEK
DIRECTOR OF VEHICLE SAFETY RESEARCH, IOACTIVE

Why can we attack?

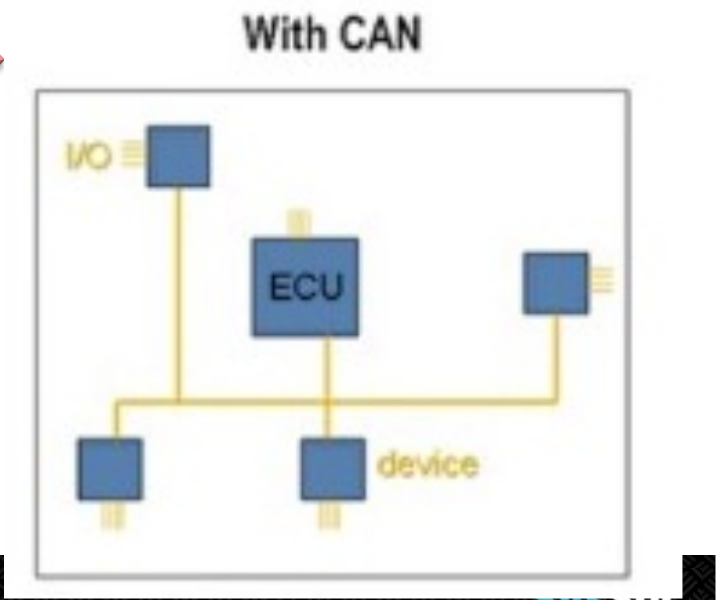




Without CAN

1980's

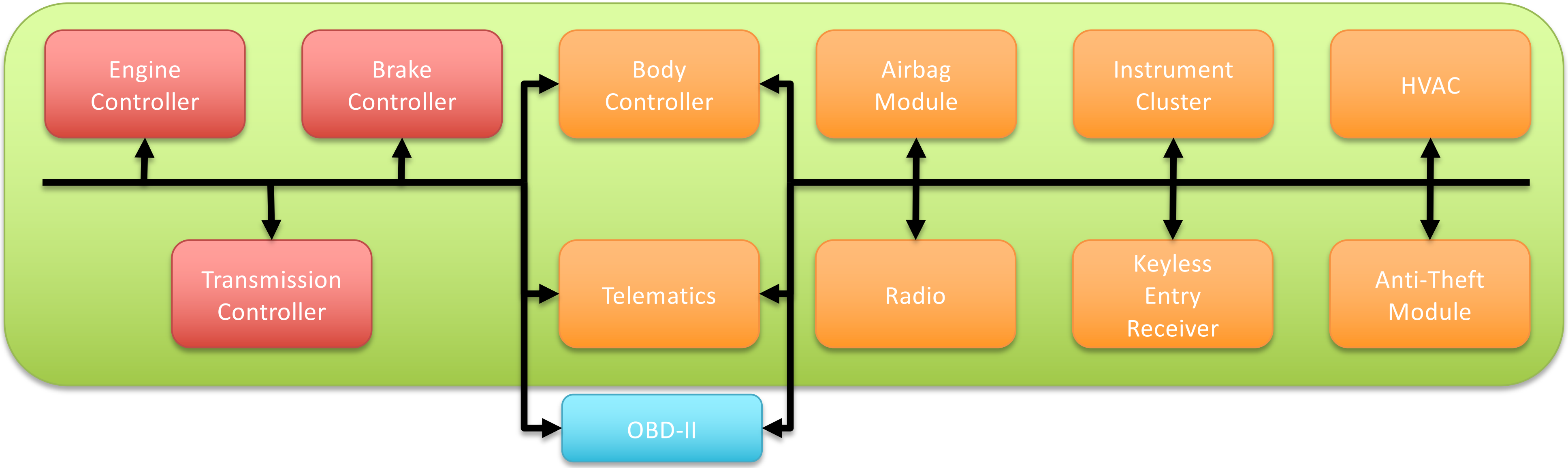
Today



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.

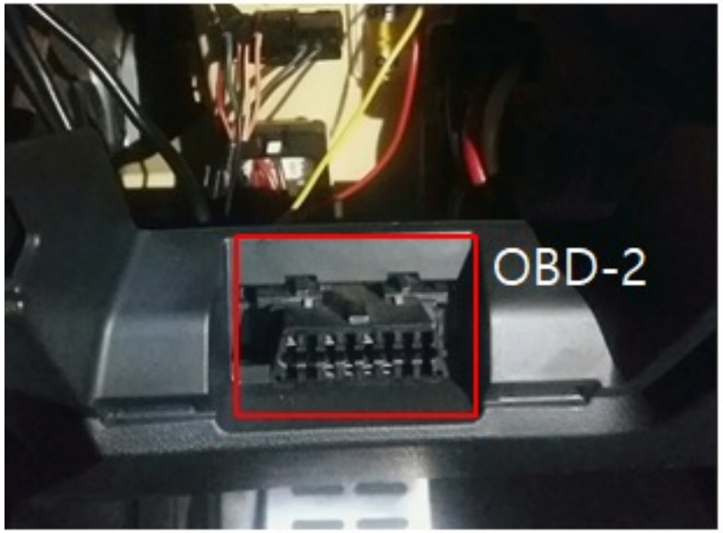
Threat model

- ❑ 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)



Port

Scanner

PassThru

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

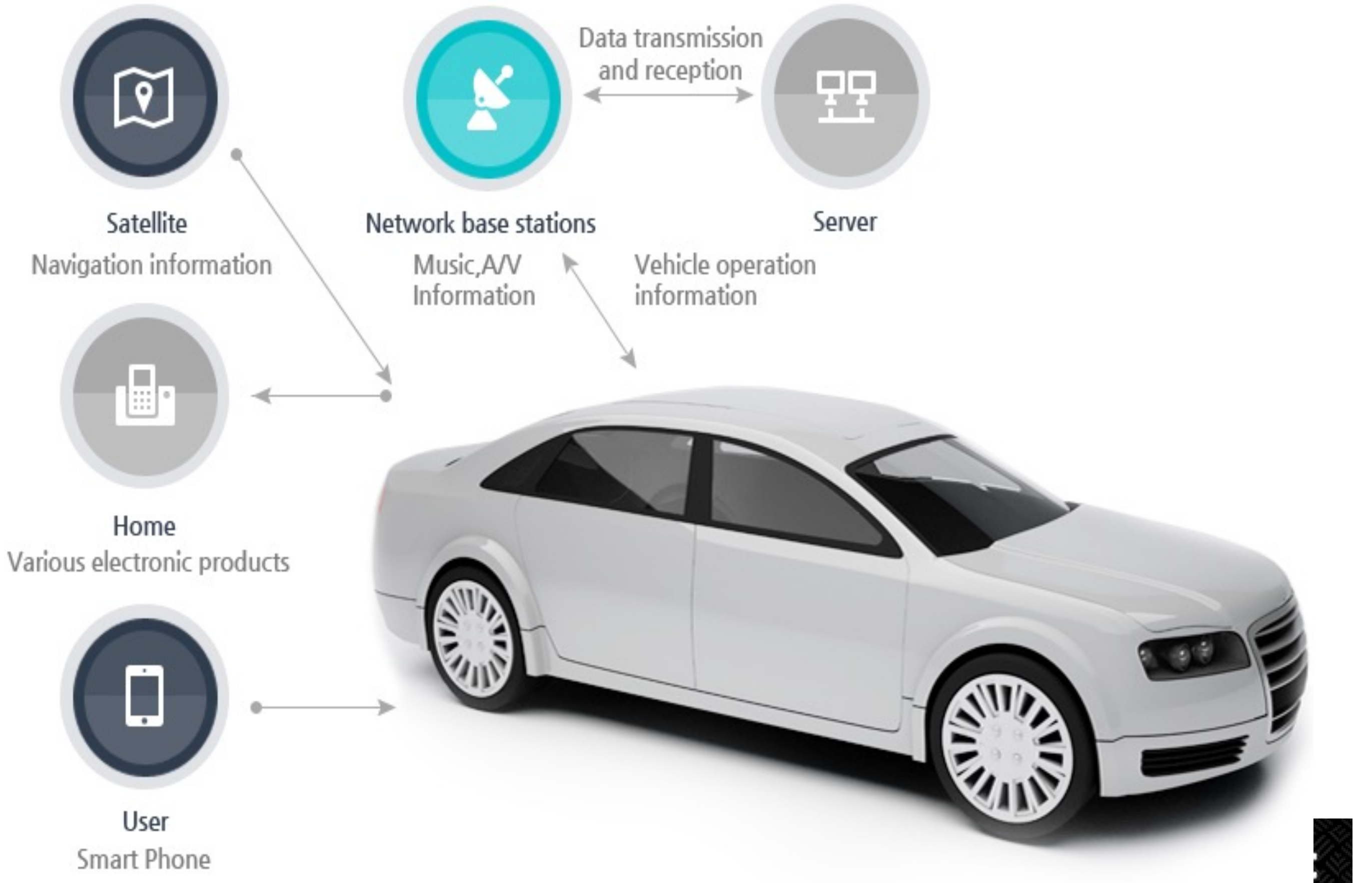
- ❑ Definition: Attacks via long-rage wireless communication (miles, global-scale)
- ❑ Broadcast channel
  - ▹ Satellite Radio, GPS, RDS

Artist:
Title:
HD-2: -
HD-3: -

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**

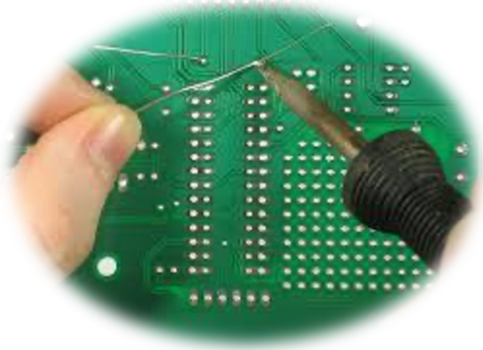
- Every attack vector leads to complete car compromise

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

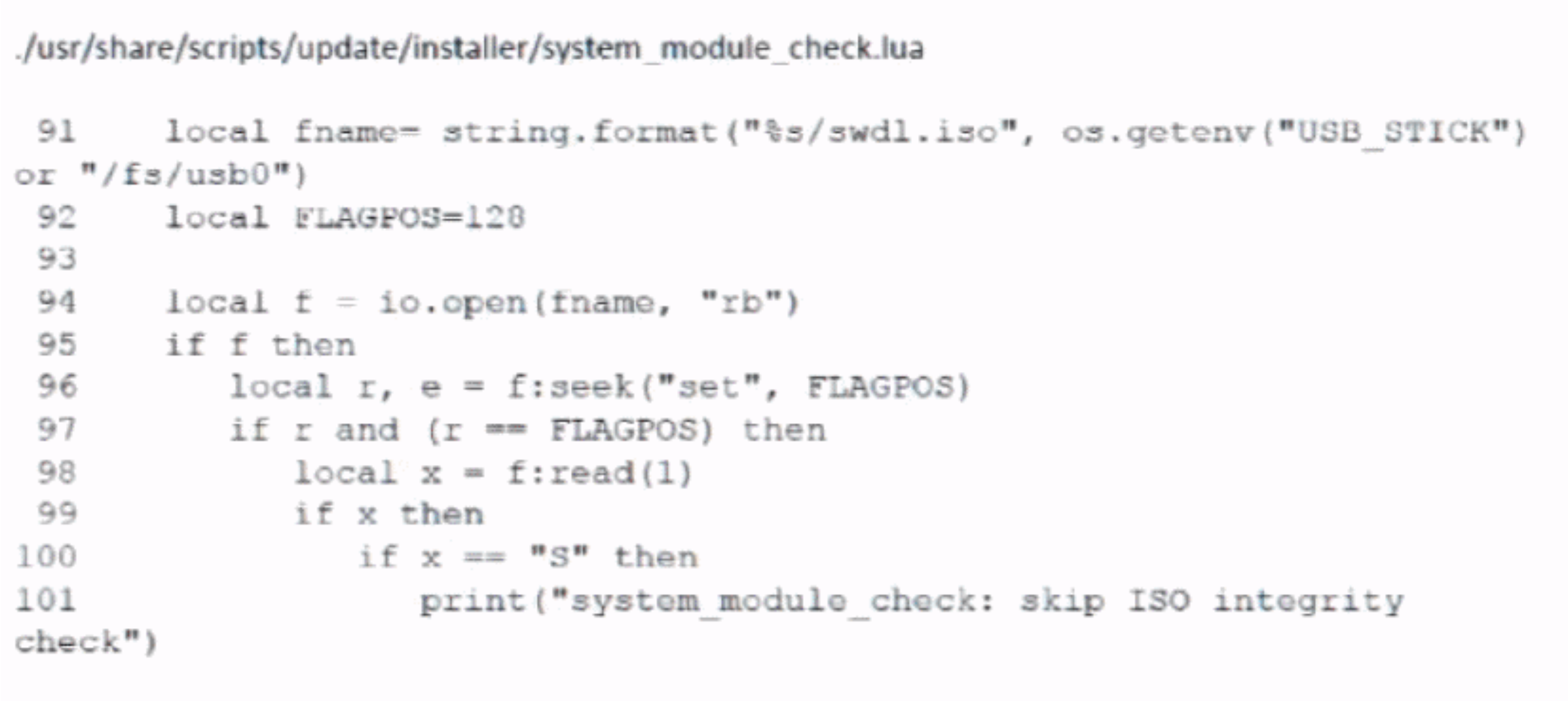


- Reverse engineering firmware
  - IDA Pro
  - Custom tools

- Identify and test vulnerable code paths

Indirect physical: Media player attack

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



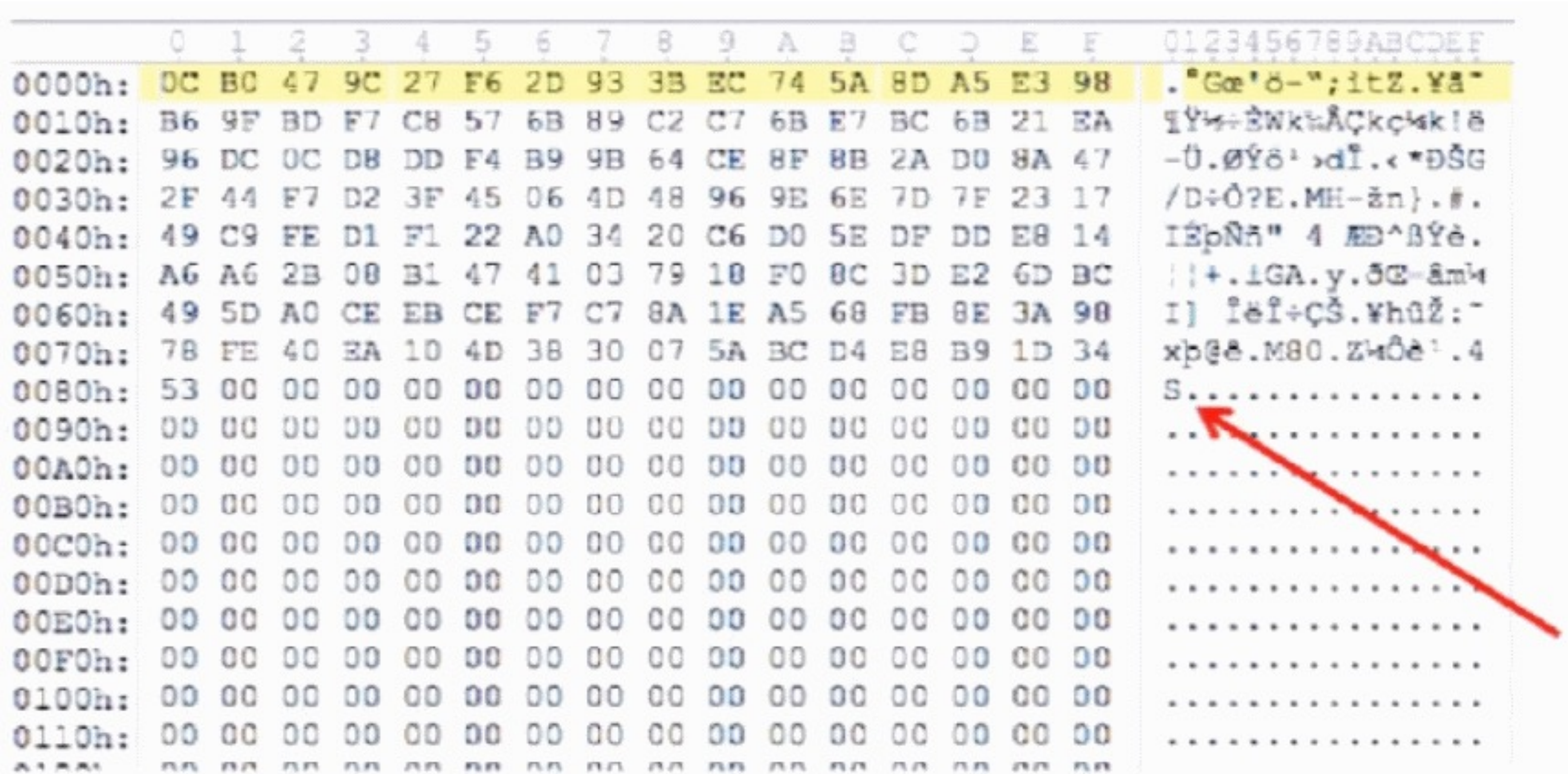
./usr/share/scripts/update/installer/system_module_check.lua

```
local fname= string.format("%s/swdl.iso", os.getenv("USB_STICK") or "/fs/usb0")
local FLAGPOS=128

local f = io.open(fname, "rb")
if f then
   local r, e = f:seek("set", FLAGPOS)
   if r and (r == FLAGPOS) then
      local x = f:read(1)
      if x then
         if x == "S" then
            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



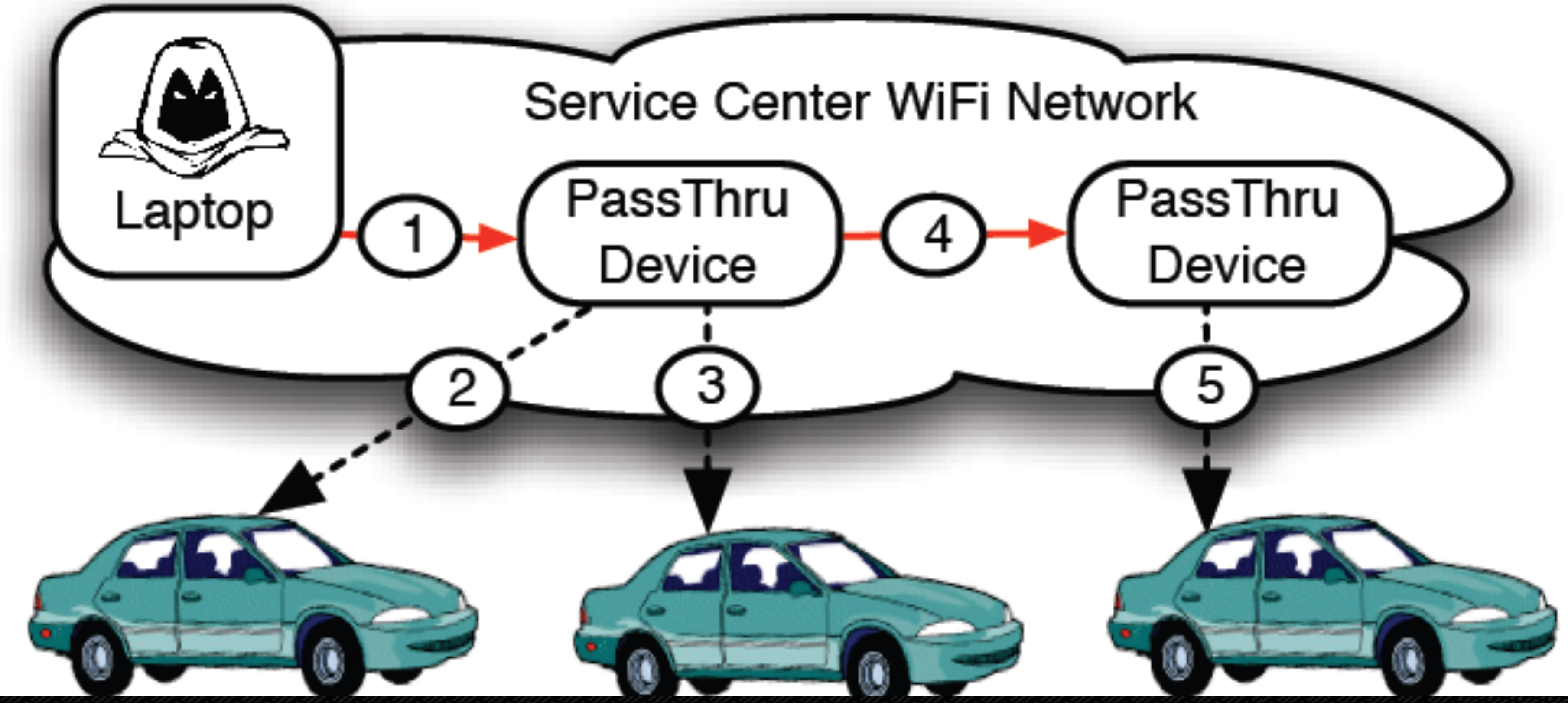
```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0123456789ABCDEF
0000h: DC BC 47 9C 27 F6 2D 93 3B EC 74 5A 8D A5 E3 98   .¼Gœ'ö-";ìtZ.¥ã˜
0010h: B6 9F BD F7 C8 57 6B 89 C2 C7 6B E7 BC 6B 21 EA   ¶Ÿ½÷ÈWk‰ÂÇkç¼k!ê
0020h: 96 DC 0C D8 DD F4 B9 9B 64 CE 8F 8B 2A D0 8A 47   –Ü.ØÝô¹›dÎ.‹*ÐŠG
0030h: 2F 44 F7 D2 3F 45 06 4D 48 96 9E 6E 7D 7F 23 17   /D÷Ò?E.MH–žn}.#.
0040h: 49 C9 FE D1 F1 22 A0 34 20 C6 D0 5E DF DD E8 14   IÉþÑñ" 4 ÆÐ^ßÝè.
0050h: A6 A6 2B 08 B1 47 41 03 79 10 F0 8C 3D E2 6D BC   ¦¦+.±GA.y.ðŒ=âm¼
0060h: 49 5D A0 CE EB CE F7 C7 8A 1E A5 68 FB 8E 3A 98   I] ÎëÎ÷ÇŠ.¥hûŽ:˜
0070h: 78 FE 40 EA 10 4D 38 30 07 5A BC D4 E8 B9 1D 34   xþ@ê.M80.Z¼Ôè¹.4
0080h: 53 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   S...............
0090h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00A0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00B0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00C0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00D0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00E0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00F0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0100h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0110h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

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



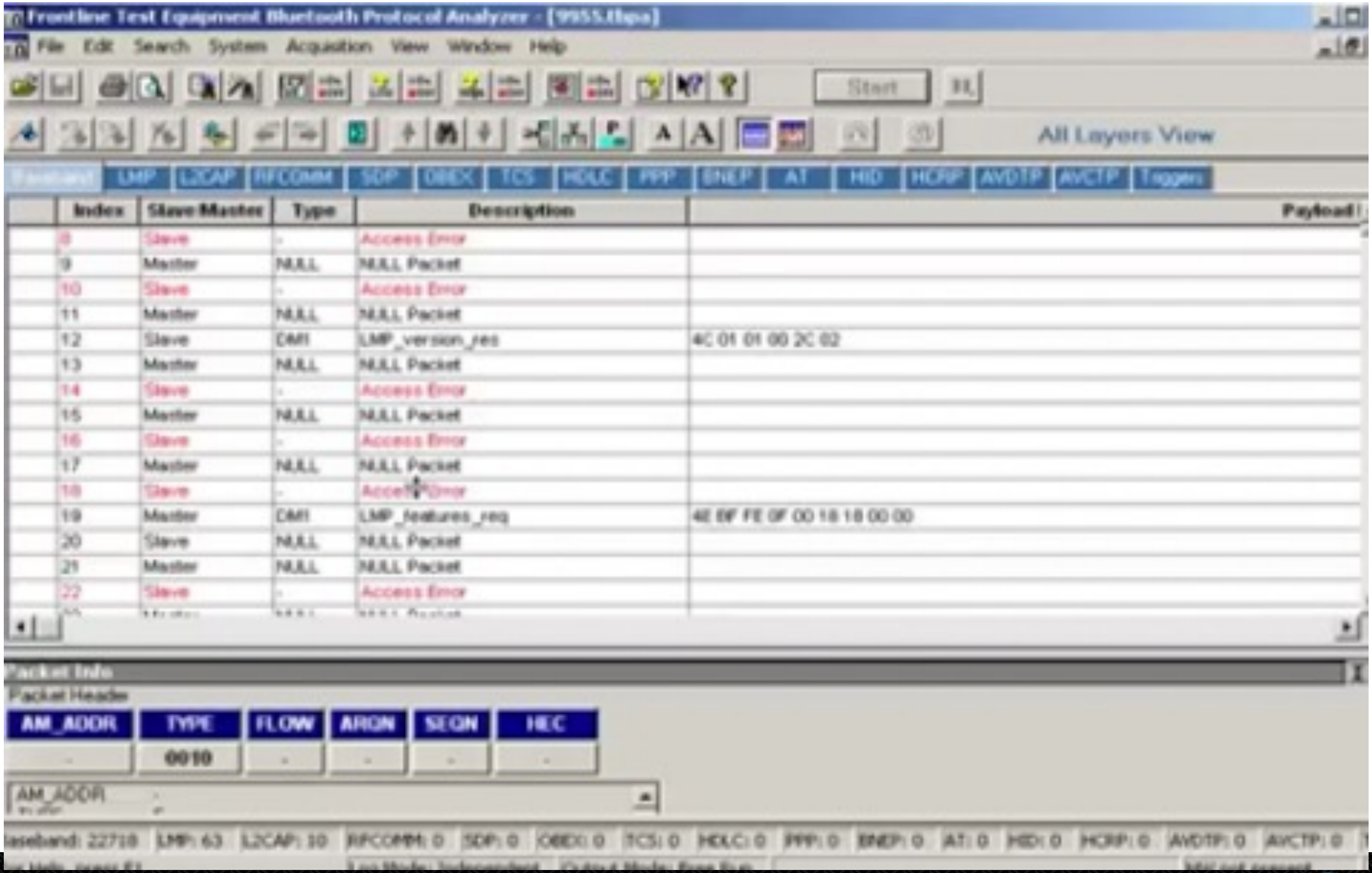
Short-range wireless: Bluetooth attack

- ❑ 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



	Index	Slave/Master	Type	Description	Payload
	8	Slave	-	Access Error	
	9	Master	NULL	NULL Packet	
	10	Slave	-	Access Error	
	11	Master	NULL	NULL Packet	
	12	Slave	DM1	LMP_version_res	4C 01 01 00 2C 02
	13	Master	NULL	NULL Packet	
	14	Slave	-	Access Error	
	15	Master	NULL	NULL Packet	
	16	Slave	-	Access Error	
	17	Master	NULL	NULL Packet	
	18	Slave	-	Access Error	
	19	Master	DM1	LMP_features_req	4E 8F FE 0F 00 18 18 00 00
	20	Slave	NULL	NULL Packet	
	21	Master	NULL	NULL Packet	
	22	Slave	-	Access Error	

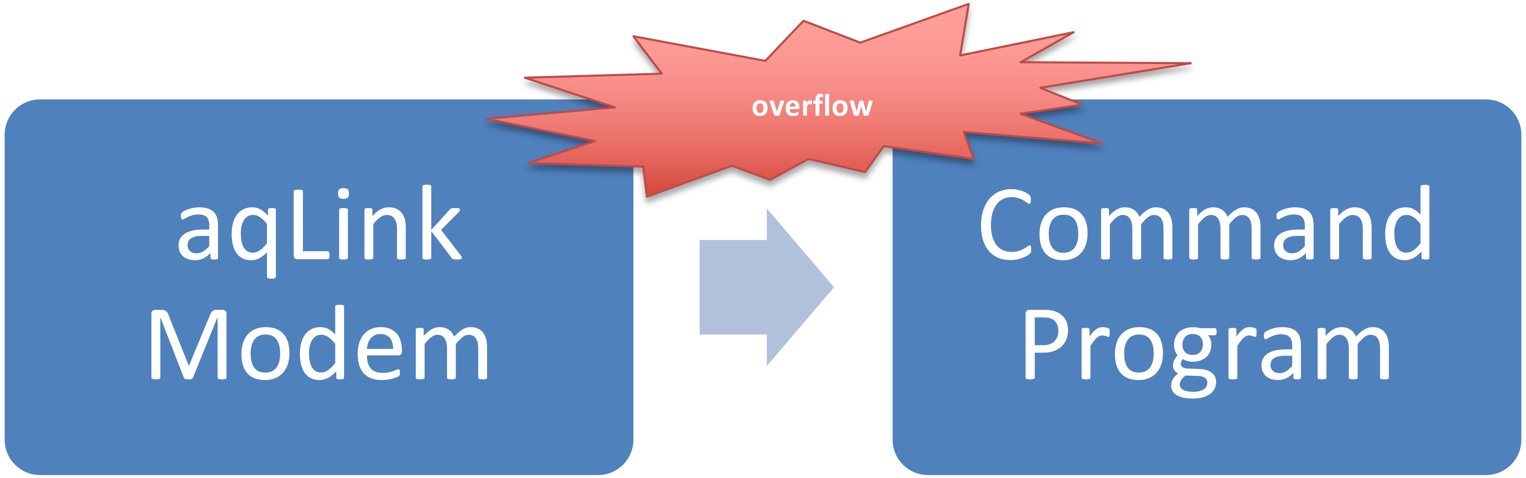
AM_ADDR	TYPE	FLOW	ARQN	SEQN	HEC
-	0010	-	-	-	-

AM_ADDR -

aseband: 22718 LMP: 63 L2CAP: 10 RFCOMM: 0 SDP: 0 OBEX: 0 TCS: 0 HDLC: 0 PPP: 0 BNEP: 0 AT: 0 HID: 0 HCRP: 0 AVDTP: 0 AVCTP: 0

Long-range wireless: Cellular attack

1. Attack @ Lowest level of protocol stack



Use 1024bytes packet size

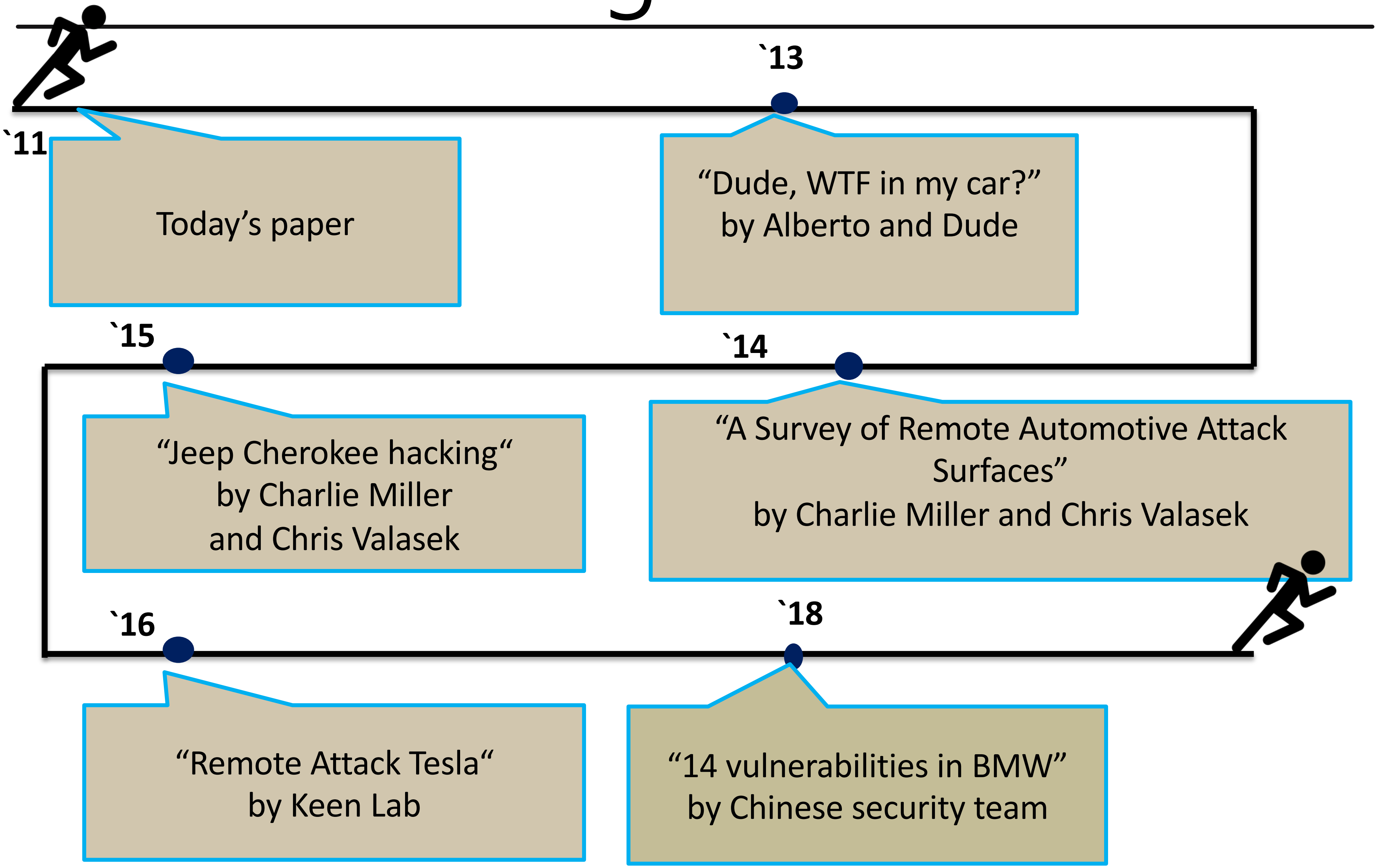
Maximum 100bytes packet

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?



`11

Today’s paper

`13

“Dude, WTF in my car?”
by Alberto and Dude

`14

“A Survey of Remote Automotive Attack Surfaces”
by Charlie Miller and Chris Valasek

`15

“Jeep Cherokee hacking“
by Charlie Miller
and Chris Valasek

`16

“Remote Attack Tesla“
by Keen Lab

`18

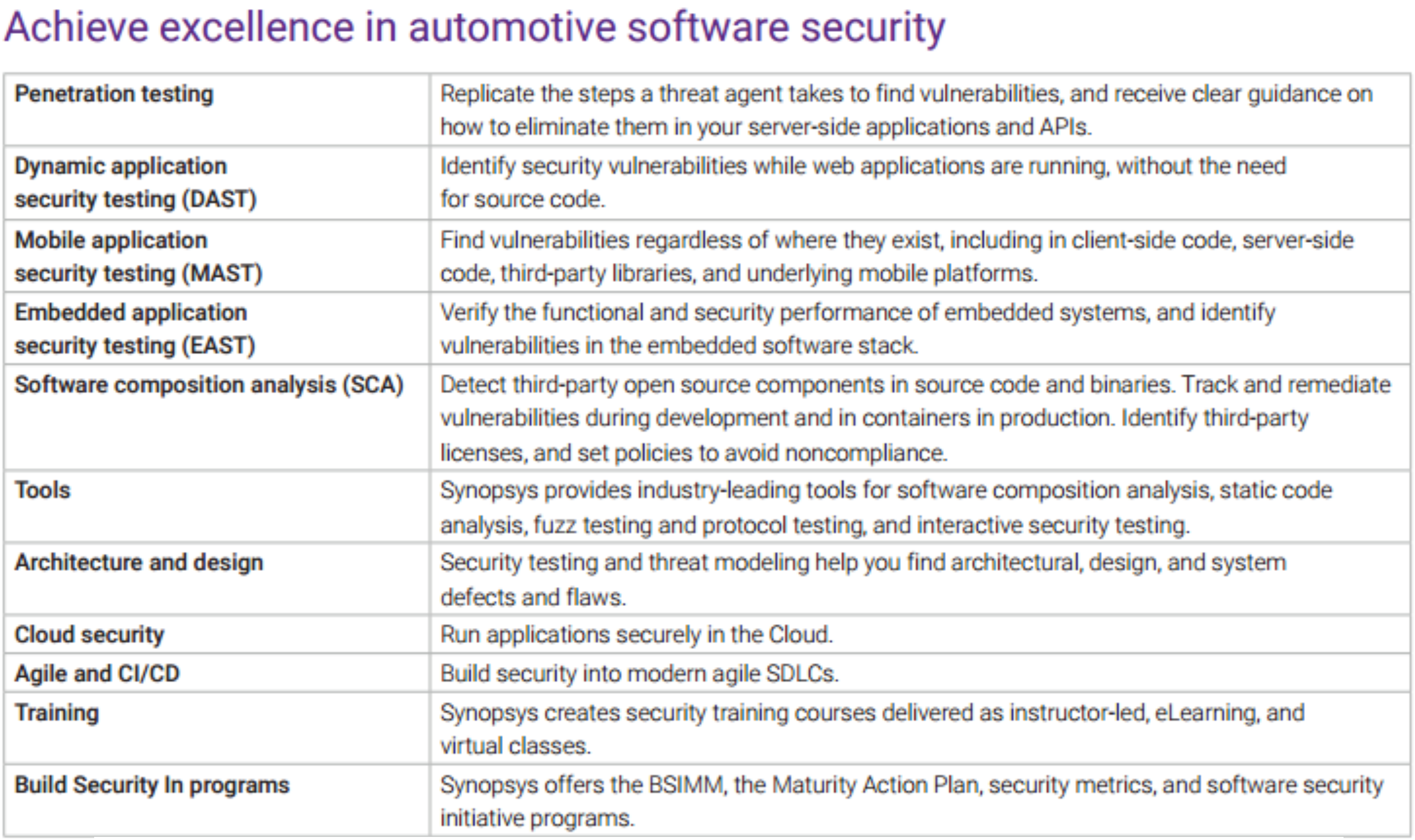
“14 vulnerabilities in BMW”
by Chinese security team

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.

SAE INTERNATIONAL ISO MISRA AUTOMOTIVE GRADE LINUX UL DEKRA

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**

SAE®

USCAR

DEPARTMENT OF TRANSPORTATION
UNITED STATES OF AMERICA

Q&A

Good Questions from Students

- ❑ The vendors separate CAN bus components considering security these days?
- ❑ Could immobiliser be also exploited through a vulnerability?
- ❑ Can’t we use security solutions for PCs or smartphones?
- ❑ The authors suggest quite simple defense mechanisms that are easy to implement, particularly when taken outside the context of cars. Do you think that the authors have proposed effective solutions (the solutions look simple)
- ❑ Based on this research, are there any security requirements or standards that can be applied in the design phase of modern vehicles?
- ❑ What forensic techniques could be used to trace back incidents of compromise?
- ❑ What is the key distinguishment between "Direct physical" and "Indirect physical”?
- ❑ How possible is it for manufacturers to remotely patch vulnerabilities in automobile electronics?
- ❑ Given the growing number of connected cars, could large-scale attacks on vehicle fleets become a reality soon?

Best questions from students

- ❑ Munim: Given the increasing complexity of automotive systems, how can manufacturers ensure that outsourced components are secure, especially when they lack access to the source code? Can binary code analysis be a feasible and scalable solution?
- ❑ Donghyun: Given the widespread use of 3rd-party components in automotive manufacturing, how feasible is it for car manufacturers to implement the proposed defenses uniformly across all models and production lines? Specifically, what challenges do you anticipate in coordinating security updates across such a diverse ecosystem of suppliers and manufacturers?
- ❑ Boris: Since a modern car is basically multiple computers connected together through a network, would it make sense to implement a firewall in order to detected unusual traffic?

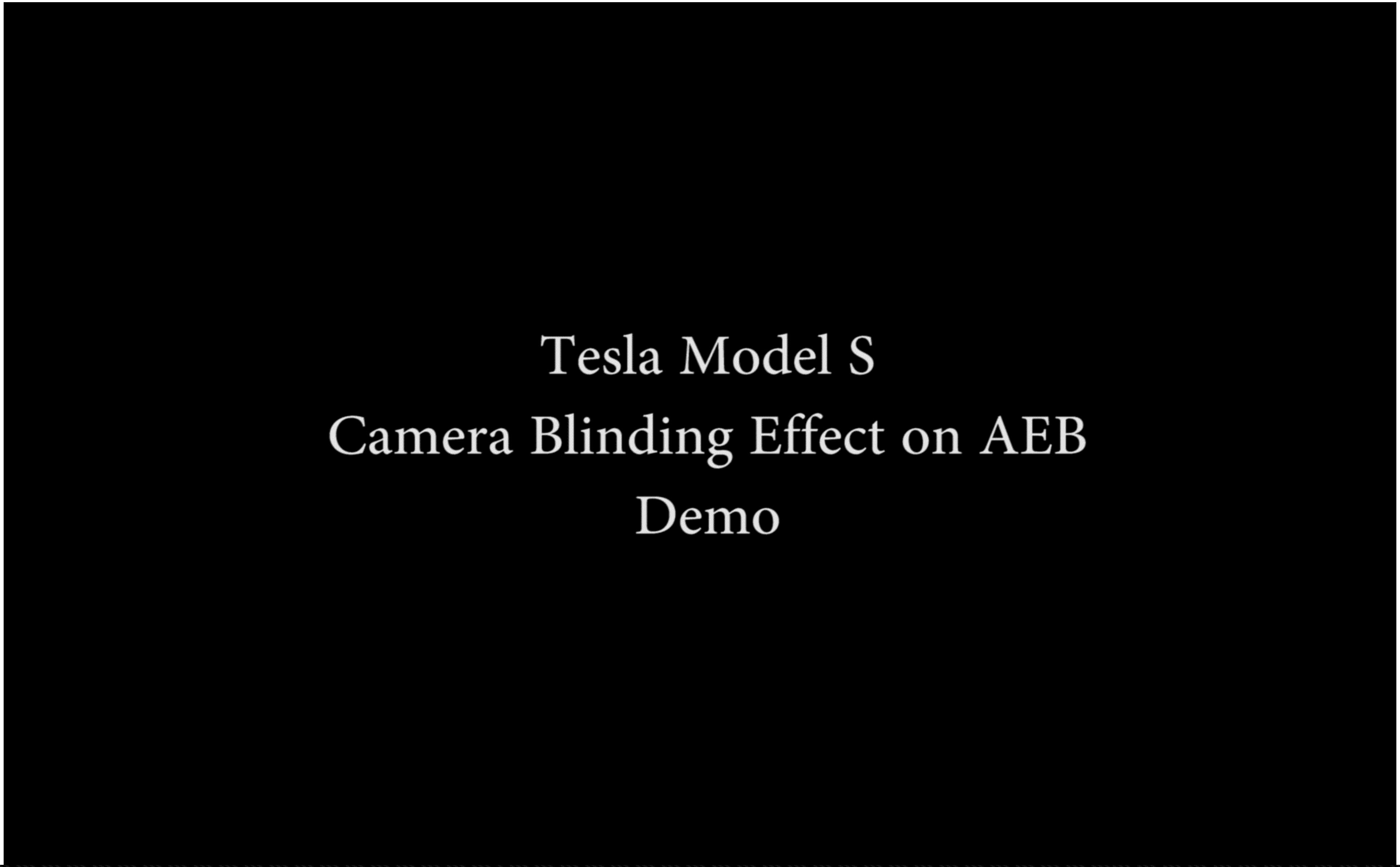
Questions that may not be answered

- ❑ Grammar/spell check using GPT/Claude
- ❑ For real-world testing of such vulnerabilities, do you believe it’s ethical to conduct experiments on actual cars?
- ❑ Is there any current research or paper in this area?
- ❑ Do modern vehicles still vulnerable to these attacks?
- ❑ Is Tesla’s self-driving vulnerable?
- ❑ Were there any real world attack cases?
- ❑ Are major automobile companies in Korea also conducting analyses to detect the vulnerabilities?
- ❑ Is it more effective to focus on detecting anomalies in ECU communications using AI?

Tesla and GPS Spoofing

GPS Spoofing Effect on
Tesla Autopilot Cruise Speed

Tesla Blinding AEB

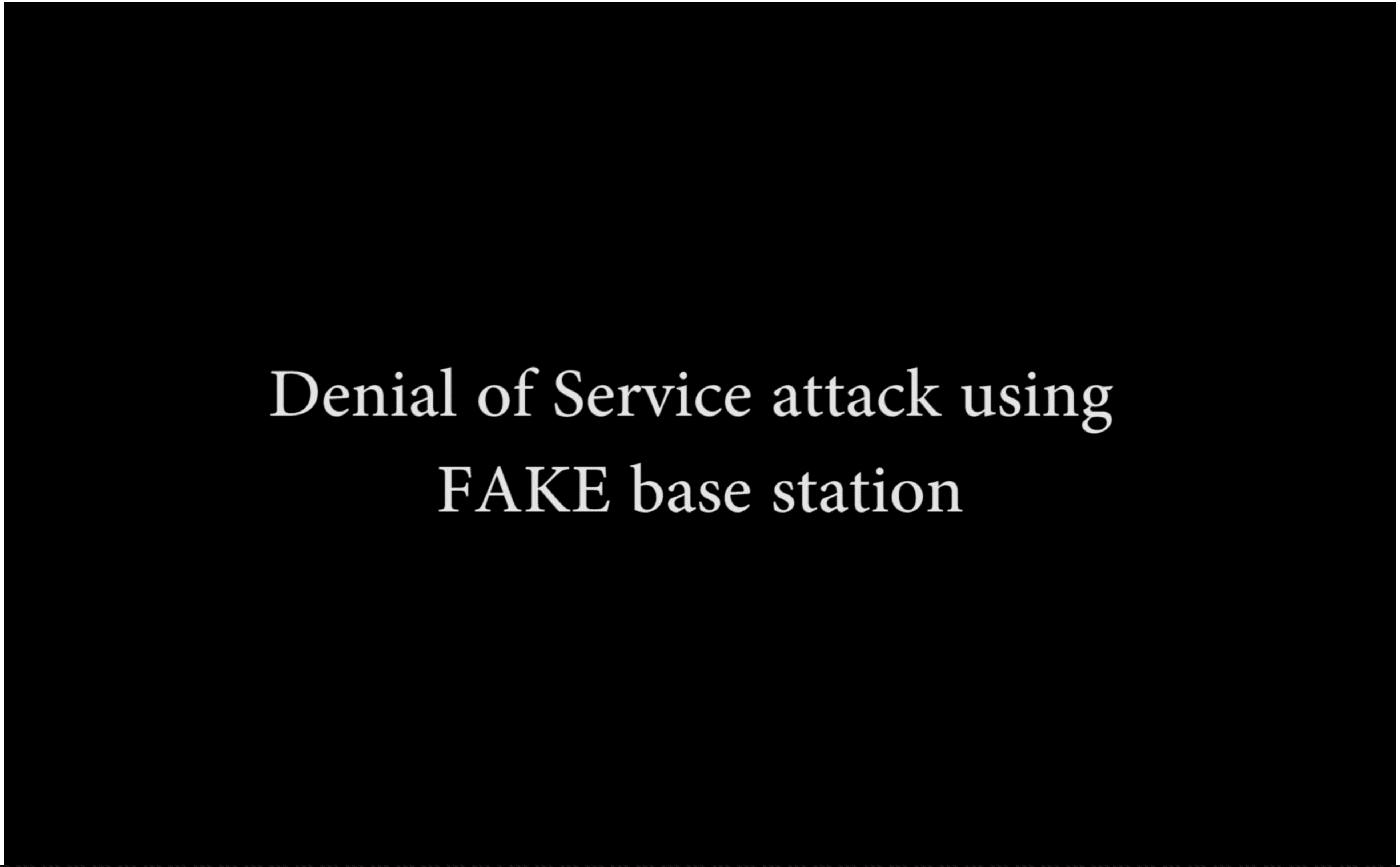


Tesla Model S

Camera Blinding Effect on AEB

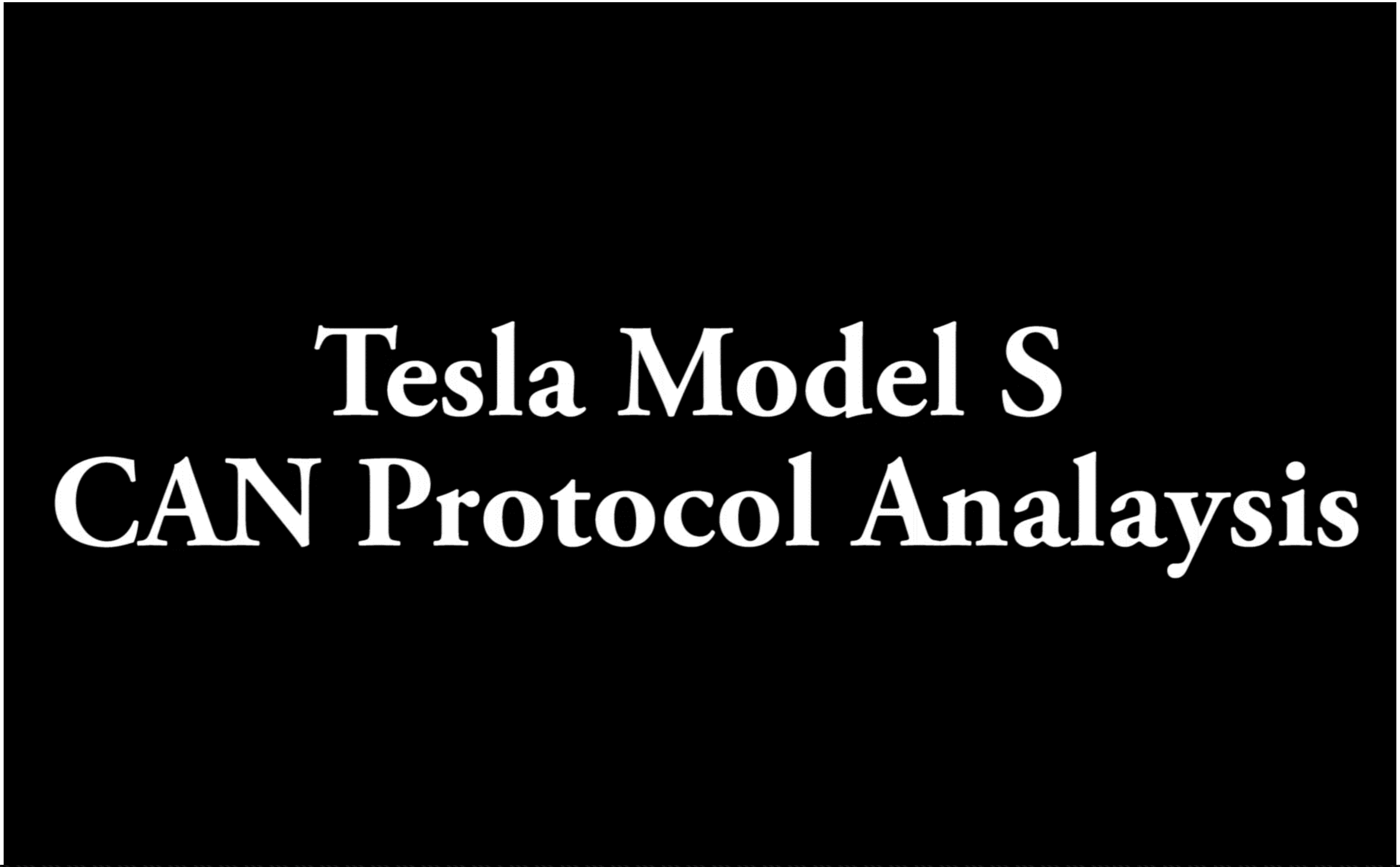
Demo

DoS Using Fake Base Station



Denial of Service attack using
FAKE base station

CAN Protocol Analysis



Tesla Model S
CAN Protocol Analaysis