Announcements

Today: *Last lecture*, special topic on smart transportation security

- **Attention:** It’s within the scope of final exam

Final exam: **12/12, 1:30-3:30 PM**

- Should be in this class room (HSLH 100A)
- Bring *your photo ID* with you
DNS: Domain Name Service

DNS maps symbolic names to numeric IP addresses (for example, \texttt{www.uci.edu} $\leftrightarrow$ 128.195.188.233)
Cached Lookup Example

Client

ftp.ics.uci.edu

Local DNS recursive resolver

ftp.ics.uci.edu
ftp=128.195.15.5

root & edu DNS server

uci.edu DNS server

ics.uci.edu DNS server
DNS “Authentication”

Request contains random 16-bit transaction id $\rightarrow$ TXID

Response accepted if TXID is the same
Stays in cache for a long time (TTL)

Client

$\text{www.ics.uci.edu}$

$\text{Local DNS recursive resolver}$

$\text{www.ics.uci.edu}$

$\text{NS uci.edu}$

$\text{root & edu DNS server}$

$\text{ics.uci.edu DNS server}$
DNS Spoofing / DNS Cache Poisoning

6.6.6.6

Trick client into looking up www.foo.com (how?)

Guess TXID, www.foo.com is at 6.6.6.6
Another guess, www.foo.com is at 6.6.6.6
Another guess, www.foo.com is at 6.6.6.6

Several opportunities to win the race
If attacker loses, has to wait until TTL expires
... but can try again with host1.foo.com, host2.foo.com, etc.
... but what’s the point of hijacking host2.foo.com?
DNS Spoofing / DNS Cache Poisoning

6.6.6.6

Trick client into looking up <random>.foo.com

Guessed TXID, very long TTL
I don’t know where <random>.foo.com is
Ask the authoritative server at www.foo.com
It lives at 6.6.6.6

www.foo.com

<random>.foo.com

Local resolver

ns.foo.com
DNS server

If attacker wins, future DNS requests for www.foo.com will go to 6.6.6.6
The cache is now poisoned... for a very long time!
No need to win future races!
DNSSEC

• Goals: authentication and integrity of DNS requests and responses

• PK-DNSSEC (public key)
  – DNS server signs its data (can be done in advance)
  – How do other servers learn the public key?

MORE INFO: http://www.dnssec.net/presentations
Lecture 17
CS 134

Smart Transportation Security

Qi Alfred Chen
Department of Computer Science
Recent interest: Autonomy software security in smart transportation

Connected Vehicle (CV)  Autonomous Vehicle (AV)

[Logos of U.S. Department of Transportation, Toyota, Ford, Qualcomm, Lyft, Uber, Tesla, Aurora, GM, Waymo, Apollo]
Recent interest: Autonomy software

Disengagements per 1000 miles

[Graph showing disengagements per 1000 miles for various companies, including Waymo, GM Cruise, Zoox, Nuro, and others.]

[Log scale on the y-axis, ranging from 0.001 to 10000.]
BotRide Service Zone
We have 19 Pick up/Drop off points
Recent interest: Autonomy software security in smart transportation

Connected Vehicle (CV)  Autonomous Vehicle (AV)
Recent interest: Autonomy software security in smart transportation

Connected Vehicle (CV)  Autonomous Vehicle (AV)
Recent interest: Autonomy software security in smart transportation

**Connected Vehicle (CV)**

[ISOC NDSS’18]  
*First software security analysis* of a CV-based transportation system

**Autonomous Vehicle (AV)**

[ACM CCS’19]  
*First software security analysis* of LiDAR-based AV perception

*Autonomy software*
Recent interest: Autonomy software security in smart transportation

Connected Vehicle (CV)

[ISOC NDSS’18]
*First software security analysis* of a CV-based transportation system

Autonomous Vehicle (AV)

[ACM CCS’19]
*First software security analysis* of LiDAR-based AV perception
Background: Connected Vehicle technology

- Wirelessly connect vehicles & infrastructure to dramatically improve **mobility & safety**
- Will **soon** transform transportation systems today
  - 2016.9, USDOT launched **CV Pilot Program**
First security analysis of CV-based transp.

- **Target**: Intelligent Traffic Signal System (I-SIG)
  - Use real-time CV data for intelligent signal control
  - USDOT sponsored design & impl.
  - Fully implemented & tested in Anthem, AZ, & Palo Alto, CA
    - ~30% reduction in total vehicle delay
  - Under deployment in NYC and Tampa, FL

CV = Connected Vehicle  OBU = On-Board Unit  RSU = Road-Side Unit
Threat model

• Malicious vehicle owners deliberately control the OBU to send spoofed data
  – OBU is compromised physically\(^1\), wirelessly\(^2\), or by malware\(^3\)
• Can only spoof data, e.g., location & speed
  – Can’t spoof identity due to USDOT’s vehicle certificate system

1 Koscher et al. @IEEE S&P’10 2 Checkoway et al. @Usenix Security’11 3 Mazloom et al. @Usenix WOOT’16
Attack goals

Traffic congestion

*Increase total delay of vehicles in the intersection*

Personal gain

*Minimize attacker’s travel time (at the cost of others’)*
Attack goals

**This work**

**Traffic congestion**

*Increase total delay of vehicles in the intersection*

**Personal gain**

*Minimize attacker’s travel time (at the cost of others’)*
Analysis approach overview

Source code

Traffic snapshots from simulator

Analysis of Attack input data flow

Data spoofing strategies

Dynamic analysis

Spoofing option enum

Increased delay calc

Spoofing w/ high delay inc

Exploit construction

Congestion creation exploit

Congestion creation vuln.
2 distinct types of algorithm-level vulnerabilities:
One single attack vehicle can greatly manipulate traffic control!
COP (Controlled Optimization of Phases)

**Input:** All vehicles’ location & speed

**Dynamic programming**

**Output:** Signal plan (green light length & order) with *lowest total delay*

1: 5 sec → 2: 3 sec → 1: 7 sec (total delay: 15 sec)

Delay = 0

Delay = 0

Delay = 15
COP (Controlled Optimization of Phases)

Data from one single vehicle: Very hard to affect signal plan

- Commonly, 1 vehicle vs > 25 vehicles’ delay in 5 conflicting lanes
  - Can’t change even 1 sec

Delay = 15

+3\times n
Vuln #1: Last vehicle advantage

- **Attack**: Spoof to arrive as late as possible to increase the delay of queuing vehicles in other lanes
Cause: Effectiveness & timeliness trade-off

- COP on RSU = \(4\text{-}5\) sec \(\iff\) decision time \(< 3\) sec
- To meet timeliness requirement, **customize COP to limit the # of servings per lane** \(\implies\) **Sub-optimal COP**
  - By default, only serve each lane **once**

---

RSU = Road-Side Unit
Vuln #2: Curse of transition period

- I-SIG has 2 operation modes based on PR:
  - PR ≥ 95%, full deployment: Directly run COP
  - PR < 95%, transition: COP becomes ineffective, use an unequipped vehicle estimation algorithm as pre-processing step

PR = Penetration Rate
Unequipped vehicle estimation algorithm

Yes (full deployment period)

No (transition period)

PR ≥ 95%

Unequipped vehicle estimation

COP algorithm

Free flow region

Slow-down region

Queuing region

Vulnerable

PR = Penetration Rate
Vulnerable queue estimation

- Data from **one single attack vehicle** can add **30-50** “ghost” vehicles to COP input
- Dramatically increase length of (wasted) green light
Attack video demo

• Demo time!
  – https://www.youtube.com/watch?v=3iV1sAxPuLo
Recent interest: Autonomy software security in smart transportation

**Connected Vehicle (CV)**

[ISOC NDSS’18]

*First software security analysis* of a CV-based transportation system

**Autonomous Vehicle (AV)**

[ACM CCS’19]

*First software security analysis* of LiDAR-based AV perception
Recent interest: Autonomy software security in smart transportation

Connected Vehicle (CV)

[IntOC NDSS'18]
First software security analysis of a CV-based transportation system

Autonomous Vehicle (AV)

[ACM CCS'19]
First software security analysis of LiDAR-based AV perception
Background: Autonomous Vehicle technology

- Equip vehicles with various types of sensors to enable self driving
Goal: First security analysis of AV autonomy software

• New attack surface: Sensors
  – Key input channel for critical control decisions
  – Public channel shared with potential adversaries
    • Fundamentally unavoidable attack surface
Background: AV autonomy software & possible sensor attacks

- Camera/LiDAR/RADAR:
  - **Spoofing attack**: inject spoofed obstacles -> emergency brake, rear-end collision etc.
Background: AV autonomy software & possible sensor attacks

- **Camera/LiDAR/RADAR:**
  - **DoS attack**: prevent victim from performing object detection -> collide into a front vehicle
Background: AV autonomy software & possible sensor attacks

- GPS:
  - **Spoofing attack**: Make victim deviate from the lane -> crash into cars in the wrong way or adjacent lanes
Background: AV autonomy software & possible sensor attacks

- **GPS:**
  - **DoS attack:** Victim unable to localize itself -> deviate from lane -> crash to cars in wrong way or adj. lanes
Goal: First security analysis of AV autonomy software

• New attack surface: Sensors
  – Key input channel for critical control decisions
  – Public channel shared with potential adversaries
    • *Fundamentally unavoidable attack surface!*

• LiDAR
Background: LiDAR basics
Background: LiDAR attacks

- Known attack: LiDAR spoofing
  - Shoot laser to LiDAR to inject points

How to use this to attack AV software control logic?

\(^1\) Shin et al.@CHES’17
First security analysis of LiDAR-based perception in AV

- **Target:** Baidu Apollo AV software system
  - Production-grade system, drive some buses in China already
  - Open sourced (“Android in AV ecosystem”)
  - Partner with 100+ car companies, including BMW, Ford, etc.

- **Attack:** LiDAR spoofing attack from road-side laser shooting devices to create fake objects
  - Trigger undesired control operations, e.g., emergency brake
LiDAR input workflow in Apollo

Point cloud data → ROI filter → Data aggregation → Deep learning model → Objectness
LiDAR input workflow with attack

Point cloud data

ROI filter

Data aggregation

Deep learning model

Objectness

Spoofed data points from LiDAR spoofing
LiDAR input workflow with attack

Point cloud data → ROI filter → Data aggregation → Deep learning model → Objectness

Data trace of LiDAR spoofing → Attack data synthesis

Attack parameters:
- Rotation
- Scale
- Height
Analysis approach

Point cloud data → ROI filter → Data aggregation → Deep learning model → Objectness

Attack data synthesis

Input: Math function

Gradient descent

Data trace of LiDAR spoofing

- Rotation
- Scale
- Height

Increase

Change
Analysis approach

Point cloud data → ROI filter → Data aggregation → Deep learning model → Objectness

Math function for pre-processing steps → Attack data synthesis

Attack parameters:
- Rotation
- Scale
- Height

Data trace of LiDAR spoofing

Gradient descent

Input: Math function
Analysis results

- Successfully find attack input that can inject fake object!
Security implication: Emergency brake attack

- Cause AV to decrease speed from $43 \text{km/h}$ to $0 \text{km/h}$ within 1 sec!
Security implication: Car “freezing” attack

• “Freeze” an AV at an intersection forever!
Conclusion

- Initiated **the first research efforts** to perform security analysis of autonomy software in CV/AV systems
- Discovered **new attacks**, analyzed **root causes**, and demonstrated **security & safety implications**
- **Only the beginning** of CV/AV software security research
  - Initiated the **ACM AutoSec workshop** to build community
  - Interested in joining? **Fill this form:**
    https://forms.gle/S7QzGkVMTcLzFvcT8

**Contact:**
Qi Alfred Chen  
Computer Science, UC Irvine  
Email: alfchen@uci.edu  
Homepage: https://www.ics.uci.edu/~alfchen/