IoT Security issues related to the future Networked Car

Koji Nakao
Distinguished Researcher,
Network Security Research Institute, NICT
(Yokohama National University with Prof. Yoshioka)
Contents

1) IoT problems – in relation to the networked car
   • Observing current IoT Attacks
   • Analyzing IoT Attacks
   • Understanding Infected IoT devices

2) Key findings and Conclusion
Recently, “scanning to Port 23 (telenet)” is getting larger!!

- Capturing packets through dark-net in real time basis.
- Color indicates the protocol types.

- UDP
- TCP SYN
- TCP SYN/ACK
- TCP Other
- ICMP
Telnet (23) attacks on Darknet have rocketed

Darknet Size -> 270,000 IP Addresses (2015/May)
Attacking hosts are IoT devices

150,000 attacking IPs

361 models observed in 4 months

Devices are inferred from their web interface and telnet banners.
In the case of Connected Car, More Attack Surfaces can be recognized and many IoT devices will be located in the car!

Why IoT devices?

• 24/7 online
• No AV
• Weak/Default login passwords
• with global IP address and open to Internet
We would like to know..

- What kind of malware?
- How many different kinds?

- What IoT devices are targeted?

- What the attackers do after compromising these devices?

We propose the first honeypot for IoT.
Our Challenges

**Honeypot**

- Emulating diverse IoT devices
- Handling to capture malware of different CPU architectures

**Sandbox: IoTBOX**

- Handle to run malware of different CPU architectures

IoT devices listening on Telnet

IoT malware of different CPU Architecture

- ARM
- MIPS
- SUPERH
- X86
- PPC
Emulating different devices (IoTPOT)

- **3-way handshake**
- **Welcome message & Login prompt**
- **Authentication**
  - root
  - 12345
- **Command Interaction**
  - `cat /bin/sh`
  - corresponding responses

**Device Profile**

**Banner Interaction**
- Do Echo, Do NAWS, Will Echo

**Authentication**
- **Different Banner Interactions**
- **Different User ID/Pass**
- **Different Interactions**

**NAWS (Negotiate About Window Size)**

- Scanning Internet on port 23 to get different banners
- Obtain weak/default ID/Pass by web search
- Always accept/reject incoming challenges
- Accept after several challenges

- Learn from actual devices
- System with general configuration for embedded devices
  (E.g., OpenWRT or Debian based embedded OS)
IoTPOT results

- During 122 days of operations [April 01 to July 31 - 2015]

900,394 Malware Download Attempts

Malware of 11 different CPU architectures

93% of downloaded binaries are new to Virus Total (2015/09)
Analyzing attacks

• Intrusion
  • Pattern of User ID/Password challenges

• Infection
  • Telnet Command Sequences from Attacker

• Monetization
  • Behaviors of second stage malware (i.e. binaries and shell scripts)
**Example 1: DDoS (DNS Water Torture attacks)**

- **Cache DNS server at ISP**
  - 9a3jk.cc.zmr666.com?
  - elirjk.cc.zmr666.com?
  - pujare.cc.zmr666.com?
  - oi4an.cc.zmr666.com?

- **Authoritative DNS for “zmr666.com”**
  - 9a3jk.cc.zmr666.com?
  - elirjk.cc.zmr666.com?
  - pujare.cc.zmr666.com?
  - oi4an.cc.zmr666.com?

- **Infected devices**

- **No resource**

- **Delayed reply**
Example 2: Click fraud

Infected devices imitates user clicks to advertising web sites
Example 3: Stealing credential from PPV (Pay Per View)

Particular set top boxes are being targeted (such as dreambox)
For Understanding Infected IoT devices, looking back on devices visiting IoTPOT

More than 60 different types (361 models) of devices visit IoTPOT

- We scan back on port 23/TCP and 80/TCP
- More than 60 type of devices visit us
Web interfaces of devices attacking us
AS with more than 1,000 infected Devices

- China
- Turkey
- Russia
- Korea
- India
- Ukraine
- Spain
- Colombia
- Germany
- Britain
- Libya
- Argentina
- Malaysia
- Mexico
- Taiwan
- Vietnam
- Hong Kong
- Brasil
- USA
- India
- Korea
- Russia
- France
- Thailand
- Phillipine
- Israel
- Italy
- Canada
- UK
- Libya
- Ukraine
- Spain
Categorizing device types

- **Surveillance Group**
  - IP Camera
  - DVR (Digital Video Recorder)

- **Networking Related Devices**
  - Router
  - Gateway
  - Modem
  - Bridge
  - Security Appliance

- **Telephone System**
  - VoIP Gateway
  - IP Phone
  - GSM Router
  - Analog Phone Adapter

- **Infrastructure**
  - Parking Management System
  - LED display control system

- **Industrial Control System**
  - Solid State Recorder
  - Internet Communication Module
  - Data Acquisition Server
  - BACnet I/O Module

- **Personal**
  - Web Camera
  - Personal Video Recorder
  - Home Automation Gateway

- **Broadcasting Facility**
  - Digital Video Broadcaster
  - Digital Video Scaler
  - Video Encoder/Decoder
  - Set Top Box

- **Other**
  - Heat Pump
  - Fire Alarm System
  - Disk Recording System
  - Optical Imaging Facility
  - Fingerprint Scanner

Devices are inferred from their web interface and telnet banners.
Attacks observed in IoTPOT from the following data source last year (2015).

Time Stamp visiting IoTPOT: 2015-03-09 and 2015-03-14
Country (IP) from Italy
HTTP Title: Web2Park -Amministrazione

Web2Park®
I believe this Web2Park has already been patched and no more scan attacks were observed in our IoT POT since last year.
Smart+Connected City Infrastructure Management: IoT Use Cases

**Smart+Connected City Parking**
- Give citizens live parking availability information to reduce circling and congestion

**Smart+Connected City Traffic**
- Monitor and manage traffic incidents to reduce congestion and improve livability

**Smart+Connected City Safety & Security**
- Automatically detect security incidents, shorten response time, and analyze data to reduce crime

**Smart+Connected City Location Services**
- Provide view of people flow data to aid planning and leverage location data for contextual content and advertising

**Smart+Connected City Lighting**
- Manage street lighting to reduce energy and maintenance costs

Presentation in ITU-T by Mr. Mikhail Kader
Smart+Connected City Parking: How It Works

Solution Components
1. Sensors on parking spots
2. New generation of parking meters
3. Video camera with analytics

Data Flow
1. Sensors detect parking events
2. Correlation of sensor and meter events to generate meter violations
3. Cameras detect no-parking and loading zone violation events
1 Streetline sensor gateway
2 Cisco IP Camera
3 Cisco Wireless Mesh Network for connectivity
4 Streetline parking data and analytics application
5 Video analytics for violation detection
6 Streetline citizen application to find real-time parking availability and payments
7 Streetline enforcement application for parking enforcement officer
Key findings and Conclusion

• **Malware**
  - At least 6 DDoS malware families target IoT devices via Telnet
  - Malware samples of 11 different CPU architectures are captured
  - 93% of samples are new to Virus Total
  - One family has quickly evolved to target more devices with as many as 9 different CPU architectures

• **Targets**
  - More than 60 types (361 models) of IoT devices are infected

• **Monetization**
  - 11 types of DDoS attacks
  - Scans (TCP/23,80,8080,5916 and UDP/ 123,3143)
  - Fake web hosting
  - Click fraud attacks
  - Stealing credential of PPV

In the connected car environment, **Malware Infection** to the Car Components (IoT devices) should be carefully considered!!

<Key Security Controls>
1. Threat observation and analysis
2. Malware/intrusion detection
3. Software Remote Update (ITU-T)
4. Data Confidentiality
   – Light-weight crypto
5. Appropriate Authentication and Access control
6. Incident handling and Information (threat) sharing

IoT devices

The Networked Car environments
Introduction of draft Rec. X.itssec-1
“Scope”

Functionality of Head Unit
- Status check of ECUs
- Log collection
- In-car diagnosis function

Functionality of Server
- Stored Data Definition
  - Auth info
  - Log Audit
- With considerations of privacy concerns

Communication protocol
- Between Car and Manufacturer / Garage
- Encryption
- Authentication

Diagnosis of on-board devices
- Status check of ECUs
- Log collection
- Verification of update module

Update Server and Log Repository at Car Manufacturer / Garage center
1. ECU generates a digest of its SW with its own secret key and sends it to the head unit

2. Head unit verifies the digest

3. Head unit merges collected digest and resigns it with its own secret key, then sends it to the manufacturer center

4. The center determines an update program, signs it with own secret key, sends it to the head unit.

5. Head unit verifies the update program and transmits it to ECU

6. ECU applies the update program by itself

7. Again, ECU generates a digest, sign it and sends it to the head unit for verification

8. Head unit verifies the digest

9. Head unit resigns the digest with own secret key, and sends it to the center

10. The center determines whether SW update process is successful or not. After the process has done successfully, the center stores the update log into own DB.
Thank you for listening Q&A

Contact: Koji Nakao (ko-nakao@nict.go.jp)