IoT Security in Action!

Julien Vermillard, Sierra Wireless
@vrmvrm - jvermillard@sierrawireless.com

EclipseCON EU 2016
Introduction
Managing connected devices

Why it is simple to exploit non-secured systems

How simple to have the minimum security
Network security
Deep dive: demo setup

- IoT Device (MangOH)
- Internet Gateway (Linux PC)
- Attacker
- Local network
- Internet
Man-in-the-Middle?

IoT Device (MangOH)

Internet Gateway (Linux PC)

Attacker PC
Linux
Ettercap

Local network
Before attack

IoT Device (RaspberryPI)

Attacker PC
Linux
Ettercap

Internet Gateway (Linux PC)

Traffic to Internet

Local network

Internet
ARP poisoning

I’m the gateway!

Route everything to the gateway

Attacker PC
Linux
Ettercap

Internet Gateway
(Linux PC)

Traffic to Internet

Local network

IoT Device
(MangOH)

Internet
DNS spoofing

DNS query

`iot.eclipse.org`

IoT Device (MangOH)

LWM2M connection

Fake DNS response

Attacker PC
- Linux
- Ettercap

Internet Gateway (Linux PC)

Local network

Internet
With TLS/DTLS?

- DNS query: `iot.eclipse.org`
- IoT Device (Raspberry Pi)
- DTLS handshake failure
- Fake DNS response
- Attack PC: Linux, Ettercap
- Internet Gateway (Linux PC)
- Local network
- Internet
Security with gateways

- Sensor network (ex: Zigbee)
- Gateway: collect data and push to cloud
- Low or no security
- Secure transport

- Public network
- Cable, 4G, etc.
Security with gateways

- Sensor network (ex: Zigbee)
- Local wireless sniffing
- Attack gateway
get access to all the network

cable, 4G, etc.

public network
End-to-end security

Low power nodes: security starts here

Sensor network (ex: Thread)

Router

See only encrypted communication
Not your Achilles’ heel
Other benefits of IP to the edge device

Simplicity: only IP networks

Topology flexibility compared to gateway

Scaling IP routing is something well known
Can we trust wireless network?

Wifi password?
GPRS encryption?
3G/4G femtocell?
Zigbee?
Bluetooth?

Not talking of plain text wireless network :)

Not talking of plain text wireless network :)
Example: GPRS

Example: 3G/4G femtocell

Figure 22: Sprint Airave 2.0

After running the exploit our Airave devices could be accessed via Telnet, essentially giving us a Busybox [32] shell on the device. We assumed that this would provide us the tools required to communicate with the Jeep over the cellular network.

Much to our delight, we were able to ping the Jeep and communicate via D-Bus over the cellular network! This meant that we could possibly broaden the range of our attack and use the same exploit that was being used to leverage remote commands via Wi-Fi without any alterations and against default vehicles (i.e. not just ones that had Wi-Fi enabled).

Generally speaking this was a huge win, but we realized that the range was still quite limited and were hoping for more, and more we shall have...
Key management

You will have a fleet of device
They needs secrets (key, password, etc..)
Unique across devices
You need to be able to change those secrets
You will probably don’t trust your factory
Lightweight M2M Bootstrap

Flash bootstrap credentials
Lightweight M2M Bootstrap

I only have bootstrap credentials or I can’t reach final server
Lightweight M2M Bootstrap

Give me key and my server(s)

Bootstrap Server
Lightweight M2M Bootstrap

Bootstrap Server

New keys and server(s) URLs and ACL
Lightweight M2M Bootstrap

Bootstrap Server

Registration

Home Automation Server

Device Manag. Server
Secret key rotation using bootstrap?

Renew or upgrade your secret:

1 - Device authenticate with the bootstrap server
2 - Bootstrap server rewrite the bootstrap secret

Next bootstrap the device use the new bootstrap secrets
Public Key Infrastructure?
How to verify a certificate

1. Identity to verify
   - Identity
   - Expiration ~3 y
   - Public Key
   - Issuer Identity
   - Issuer Signature

2. Intermediate
   - Identity
   - Expiration ~5 y
   - Public Key
   - Issuer Identity
   - Issuer Signature

3. Root trust
   - Root CA Identity
   - Root Public Key
   - Root Signature

Steps:
- Find Identity
- Verify Identity
- Find Intermediate
- Verify Intermediate
- Find Root trust
- Verify Root trust
Enrollment with PKI

Generate

Private  Public
Enrollment with PKI

Certificate Request

Private  Public

Certification Authority
Enrollment with PKI

Generate

Private
Public

Certificate

Certification Authority

Sign using CA private key a X.509 certificate

CA
Private
Enrollment with PKI

Generate

Private

Public

Sign using certificate for authentication

Service

CA Public
Still not IoT friendly

A lot of enterprise protocols:

**IKE**: Internet Key Exchange [RFC2409]

**CMP**: Certificate Management Protocol [RFC4210]

**SCEP**: Simple Certificate Enrollment Protocol [draft-gutmann-scep-02]

**EST**: Enrollment Over Secure Transport [RFC7030]

**IEEE 802.1AR**: Secure Device Identity [802.1AR]

But still nothing ready to use for constrained networks & devices
Firmware download
Firmware download

HTTP GET

IoT Device (RaspberryPI)

Internet Gateway (Linux PC)

Local network

Send firmware with backdoor

Attacker PC
Linux
Ettercap
CMS (Cryptographic Message Syntax)

See RFC5652 (replaces PKCS #7)

Used to digitally sign, digest, authenticate, or encrypt arbitrary message content.

Supported by OpenSSL (CLI: openssl cms)
Secure boot

Hardware (ROM) enforces booting only correctly signed code

Often based on ECDSA signature

Hardware based $\Rightarrow$ no algorithm agility
Open-source solutions are there

**Eclipse IoT:**
- Leshan, Wakaama, TinyDTLS, Scandium, Paho,
  - Mosquitto, Hono

**OpenSSL, Mbed TLS**

**CFSSL**

**GnuPG**

**U-Boot**
Thanks!

Twitter: @vrmvrm
Mail: jvermillard@sierrawireless.com
Credits

Tom Medley - The Noun Project
Guilhem - The Noun Project
Giuditta Valentina Gentile - The Noun Project
Sergey Krivoy - The Noun Project
Jon Anderson - The Noun Project
Ryan Beck - The Noun Project
Edward Boatman - The Noun Project