Securing the Internet of Things

Paul Fremantle
CTO, WSO2 (paul@wso2.com)
PhD researcher, Portsmouth University (paul.fremantle@port.ac.uk) @pzfreo

Paul Madsen*
Technical Architect, PingIdentity (pmadsen@pingidentity.com) @paulmadsen

*Paul M helped me with the initial content, but I take responsibility for anything you don’t like in this slide deck.
ongoing experiment with 'connected toilet' will unfortunately prevent me from presenting on OAuth for IoT at @eclipsecon with @pzfreo
About me

• CTO and Co-Founder WSO2
  – Open Source Middleware platform

• Part-time PhD looking at security

• Working in Apache for 14 years

• Working with Cloud, SOA, APIs, MQTT, IoT
Firstly, does it matter?
WEAR THIS BIOSENSOR SO MANAGEMENT CAN MONITOR YOUR HEALTH DURING THE DAY.

WOW. I DIDN'T KNOW YOU CARED SO MUCH ABOUT MY HEALTH.

OH, I DO.

EMPLOYEE 479 DOESN'T HAVE SHALLOW BREATHING. YOU CAN GIVE THAT ONE SOME MORE WORK.
A fridge full of spam: Hacked domestic appliances send a torrent of junk email

Monday 20 Jan 2014 10:24 pm
Search Queries

- inurl:/view.shtml
- inurl:ViewerFrame?Mode=
- inurl:ViewerFrame?Mode=Refresh
- inurl:view/index.shtml
- inurl:view/view.shtml
- liveapplet
- intitle:"live view" intitle:axis
- intitle:liveapplet
- allintitle:”Network Camera NetworkCamera”
- intitle:axis intitle:"video server”
- intitle:liveapplet inurl:LvAppl
- intitle:"EvoCam” inurl:"webcam.html”
- intitle:"Live NetSnap Cam-Server feed”
- intitle:”Live View / - AXIS 206M”
- intitle:”Live View / - AXIS 206W”
- intitle:”Live View / - AXIS 210”
- inurl:indexFrame.shtml Axis
- intitle:start inurl:cgistart
- intitle:"WJ-NT104 Main Page”
- intitle:snc-z20 inurl:home/
- intitle:snc-cs3 inurl:home/
- intitle:snc-rz30 inurl:home/
When 'Smart Homes' Get Hacked: I Haunted A Complete Stranger's House Via The Internet

“I can see all of the devices in your home and I think I can control them,” I said to Thomas Hatley, a complete stranger in Oregon who I had rudely awoken with an early phone call on a Thursday morning.

He and his wife were still in bed. Expressing surprise, he asked me to try to turn the master bedroom lights on and off. Sitting in my living room in San Francisco, I flipped the light switch with a click, and resisted the Poltergeist-like temptation to turn the television on as well.

“They just came on and now they’re off,” he said. “I’ll be darned.”

So what is different about IoT?

• The longevity of the device
  – Updates are harder (or impossible)
• The size of the device
  – Capabilities are limited – especially around crypto
• The fact there is a device
  – Usually no UI for entering userids and passwords
• The data
  – Often highly personal
• The mindset
  – Appliance manufacturers don’t think like security experts
  – Embedded systems are often developed by grabbing existing chips, designs, etc
Physical Hacks

Karsten Nohl and Henryk Plotz. MIFARE, Little Security, Despite Obscurity
Or try this at home?
http://freo.me/1g15BiG

Atmel AVR High Voltage Fuse Reset on a Breadboard

Submitted by pomprocker on December 19, 2008 - 2:48am.

Here is what happens when you don't plan well...A big hairy mess.

Parts:

1 - Breadboard
1 - Set of breadboard jumper wires
2 - ATmegas (one good one, and the one you're locked out of)
2 - Regulated power sources, 12vdc and 5vdc
1 - LED
1 - 2N3903 or 2N3904 transistor (available at Radio Shack)
1 - Tactile/Momentary Button (Omron B3F-1000 is a popular one, can be stuck into a breadboard)
20 - 1K Ohm Resistors, 1/4 watt is fine.
Technical Report

Number 630

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Semi-invasive attacks –
A new approach to
hardware security analysis

Sergei P. Skorobogatov

http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-630.html
Hardware recommendations

• Don’t rely on obscurity
Hardware recommendations

• Don’t rely on obscurity
• Don’t rely on obscurity
• Don’t rely on obscurity
• Don’t rely on obscurity
• Don’t rely on obscurity
• Don’t rely on obscurity
Hardware Recommendation #2

• Unlocking a single device should risk only that device’s data
The Network
Crypto on small devices

- Practical Considerations and Implementation Experiences in Securing Smart Object Networks

<table>
<thead>
<tr>
<th>Key length (bits)</th>
<th>Execution time (ms); key in SRAM</th>
<th>Memory footprint (bytes); key in SRAM</th>
<th>Execution time (ms); key in ROM</th>
<th>Memory footprint (bytes); key in ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>66</td>
<td>40</td>
<td>70</td>
<td>32</td>
</tr>
<tr>
<td>128</td>
<td>124</td>
<td>80</td>
<td>459</td>
<td>64</td>
</tr>
<tr>
<td>512</td>
<td>25,089</td>
<td>320</td>
<td>27,348</td>
<td>256</td>
</tr>
<tr>
<td>1,024</td>
<td>199,666</td>
<td>640</td>
<td>218,367</td>
<td>512</td>
</tr>
<tr>
<td>2,048</td>
<td>1,587,559</td>
<td>1,280</td>
<td>1,740,267</td>
<td>1,024</td>
</tr>
</tbody>
</table>

RSA private key operation performance
# ROM requirements

<table>
<thead>
<tr>
<th>Library</th>
<th>ROM Footprint (Kilobytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvrCryptolib</td>
<td>3.6</td>
</tr>
<tr>
<td>Wiselib</td>
<td>16</td>
</tr>
<tr>
<td>TinyECC</td>
<td>18</td>
</tr>
<tr>
<td>Relic-toolkit</td>
<td>29</td>
</tr>
</tbody>
</table>

Summary of library ROM needs
ECC is possible (and about fast enough)

<table>
<thead>
<tr>
<th>Curve parameters</th>
<th>Execution time (ms)</th>
<th>Memory Footprint (bytes)</th>
<th>Comparable RSA key length</th>
</tr>
</thead>
<tbody>
<tr>
<td>128r1</td>
<td>1,858</td>
<td>776</td>
<td>704</td>
</tr>
<tr>
<td>128r2</td>
<td>2,002</td>
<td>776</td>
<td>704</td>
</tr>
<tr>
<td>160k1</td>
<td>2,228</td>
<td>892</td>
<td>1,024</td>
</tr>
<tr>
<td>160r1</td>
<td>2,250</td>
<td>892</td>
<td>1,024</td>
</tr>
<tr>
<td>160r2</td>
<td>2,467</td>
<td>892</td>
<td>1,024</td>
</tr>
<tr>
<td>192k1</td>
<td>3,425</td>
<td>1,008</td>
<td>1,536</td>
</tr>
<tr>
<td>192r1</td>
<td>3,578</td>
<td>1,008</td>
<td>1,536</td>
</tr>
</tbody>
</table>

ECDSA signature performance with TinyECC
I didn't have any accurate numbers so I just made up this one.

Studies have shown that accurate numbers aren't any more useful than the ones you make up.

How many studies showed that?

Eighty-seven.
<table>
<thead>
<tr>
<th>System type</th>
<th>Such as</th>
<th>Will it work?</th>
<th>The issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low end embedded</td>
<td>Atmel 8-bit AVR (most Arduino), TI MSP-430</td>
<td>No</td>
<td>SRAM</td>
</tr>
<tr>
<td>Mid-high end</td>
<td>Anything ARM based (e.g. STM</td>
<td>With some effort</td>
<td>Library, key and cipher suite</td>
</tr>
<tr>
<td>embedded</td>
<td>Discovery, TI Stellaris) inc. Arduino Due</td>
<td></td>
<td>suite wrangling</td>
</tr>
<tr>
<td>Linux OS</td>
<td>Raspberry Pi, BeagleBone, Arduino Yún</td>
<td>Yes</td>
<td>-</td>
</tr>
</tbody>
</table>

Won’t ARM just solve this problem?
Cost matters

8 bits
$5 retail
$1 or less to embed

32 bits
$25 retail
$?? to embed
Another option?

ATECC108
Atmel CryptoAuthentication

SUMMARY DATASHEET

Features

- Secure authentication and product validation device
- High-Speed Public Key Algorithm (PKI) Crypto Engine
  - FIPS186-3 Elliptic Curve Digital Signature Algorithm (ECDSA)
- NIST Standard P256, B283, and K283 Elliptic Curve support
- Superior SHA-256 Hash Algorithm: HMAC option
- Integrated capability for both Host and Client operations
- Best in class 256/283-bit key length, storage for up to 16 keys
- Guaranteed unique 72-bit serial number
- Internal high-quality FIPS Random Number Generator (RNG)
- 8.5Kb EEPROM memory for keys, certificates, and data
- 512 One Time Programmable (OTP) bits for fixed information or
SIMON and SPECK: New NSA Encryption Algorithms

The NSA has published some new symmetric algorithms:

**Abstract:** In this paper we propose two families of block ciphers, SIMON and SPECK, each of which comes in a variety of widths and key sizes. While many lightweight block ciphers exist, most were designed to perform well on a single platform and were not meant to provide high performance across a range of devices. The aim of SIMON and SPECK is to fill the need for secure, flexible, and analyzable lightweight block ciphers. Each offers excellent performance on hardware and software platforms, is flexible enough to admit a variety of implementations on a given platform, and is amenable to analysis using existing techniques. Both perform exceptionally well across the full spectrum of lightweight applications, but SIMON is tuned for optimal performance in hardware, and SPECK for optimal performance in software.

It's always fascinating to study NSA-designed ciphers. I was particularly interested in the algorithms' similarity to Threefish, and how they improved on what we did. I was most impressed with their key schedule. I am always impressed with how the NSA does key schedules. And I enjoyed the discussion of requirements. Missing, of course, is any cryptanalytic analysis.

I don't know anything about the context of this paper. Why was the work done, and why is it being made public? I'm curious.

[https://www.schneier.com/blog/archives/2013/07/simon_and_speck.html](https://www.schneier.com/blog/archives/2013/07/simon_and_speck.html)
Datagram Transport Layer Security (DTLS)

• UDP based equivalent to TLS
• https://tools.ietf.org/html/rfc4347

<table>
<thead>
<tr>
<th></th>
<th>ROM</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>State Machine</td>
<td>8.15</td>
<td>1.9</td>
</tr>
<tr>
<td>Cryptography</td>
<td>3.3</td>
<td>1.5</td>
</tr>
<tr>
<td>Key Management</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>DTLS Record Layer</td>
<td>3.7</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>16.15</strong></td>
<td><strong>3.9</strong></td>
</tr>
</tbody>
</table>

Table 1: Memory Requirements in KB
Key distribution
exemplar
/ɪɡˈzɛmplər/  🎧

noun

1. a person or thing serving as a typical example or appropriate model.
   "the place is an exemplar of multicultural Britain"

   synonyms: epitome, perfect example, shining example, model, paragon, ideal,
   type, exemplification, definitive example, textbook example,
   embodiment, essence, quintessence; More
CoAP

• Constrained Application Protocol
  – REST-like model built on UDP
  – Californium project coming soon to Eclipse IoT

• No authentication or authorization
  – Relies on DLTS or data in the body
MQTT

Publisher

Subscriber

Messenger
Free texting from Facebook
MQTT

• Very lightweight messaging protocol
  – Designed for 8-bit controllers, SCADA, etc
  – Low power, low bandwidth
  – Binary header of 2 bytes
  – Lots of implementations
    • Mosquitto, Paho, RSMB and Moquette from Eclipse
  – Clients:
    • Arduino, Perl, Python, PHP, C, Java, JS/Node.js, .Net, etc
• Plus an even lighter-weight version for Zigbee
  – MQTT-SN (Sensor Network)
MQTT

- Relies on TLS for confidentiality
- Username/Password field
Passwords

• Passwords suck for humans
• They suck even more for devices
How Tokens Work

User \rightarrow ResourceServer: login

ResourceServer \rightarrow AuthServer: token

AuthServer \rightarrow ResourceServer: validate token

ResourceServer \rightarrow User: ok

User \rightarrow ResourceServer: do something

User never passes their userid/password to the resource server

[Alternatively validate via signatures]
Why OAuth2?

• Widely implemented
• Pretty good
  – Of course there is never 100% agreement
  – Or certainty with security protocols
• Not just HTTP:
  – OAuth2 used with SSL
An OAuth 2 access token is like a hotel-room key card.

It gives access, all by itself without further checking, to a particular resource (in this case, room 238 at the Omni Interlocken in Denver.) Check.

It’s issued to a particular person, who has to be authenticated first (like by showing my driver’s license at the check-in.) Check.

Nothing on the outside tells you who it’s been issued to or what it’s for. Check.

It’s not obscured or encrypted, so you have to take good care of it (if a bad guy got it and knew what it was for, he could get into my hotel room and rob me blind.) Check.

You can give it to someone else and have them access the resource for you (like giving a colleague the card and asking them to go up to your room and get the VGA dongle that you stupidly left on the desk.) Check.

If you lose it, you can go back to the issuer and get another one which is functionally identical (somehow it wasn’t there when you got back from the bar, but the front desk can get you another, assuming you have your wallet and ID.) Check.

It expires after a while. (I gave it back to the front desk when I left because I knew it wouldn’t be useful any more.) Check.
Authorize The Visitor Widget to use your account?

This application **will be able to:**
- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

This application **will not be able to:**
- Access your direct messages.
- See your Twitter password.

You can revoke access to any application at any time from the Applications tab of your Settings page.

By authorizing an application you continue to operate under Twitter’s Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.
Why FIAM for IoT?

• Can enable a meaningful consent mechanism for sharing of device data
• Giving a device a token to use on API calls better than giving it a password
  – Revokable
  – Granular
• May be relevant for both
  – Device to cloud
  – Cloud to app
Two aspects using OAuth with IoT

- On the device
  - Tokens are good
  - Limiting the access of the device
- On the cloud
  - Putting users in control of their data
  - Just good current practice
- Demo with MQTT
  - But not just for MQTT
  - Also for the cloud, CoAP, and other protocols too
Demo components

- **CreateToken.py**
- **Mosquito** (Open Source MQTT Broker)
  - Acting as “Resource Server”
- **Mosquito_py_auth**
  - `mqtt-oauth2.py`
- **Refresher.py**
- **IdP**
  - WSO2 Identity Server
- **ESB Introspection API**

Flow:
- 1. IdP
- 2. Arduino
- 3. Mosquito
- 4. Refresher.py
- 5. Mosquito_py_auth
- 6. ESB Introspection API
Lessons learnt

• MQTT and MPU / I2C code is 97% of Duemilanove
  – Adding the final logic to do OAuth2 flow pushed it to 99%
  – No TLS in this demo is a big issue
• Different OAuth2 implementations behave differently (e.g. changing the refresh token every time you refresh)
• Need to be able to update the scope of token if this will work for long term embedded devices
  – Easy fix
• MQTT should have a well defined model for sending a message to just one client (securely)
What I haven’t covered enough of
Summary

• Think about security with your next device
• We as a community need to make sure that the next generation of IoT devices are secure
• We need to create exemplars
  – Shields
  – Libraries
  – Server software
  – Standards