Iot CyberSecurity

Vulnerabilità, attacchi e contromisure nel mondo IoT

Giuseppe Augiero

Agenda

- Introduction to the iot ecosystem.
- Threats, Vulnerability and Risks.
- Attacks and attack tools.
  - Device
  - Web.
  - Mobile.
  - Protocols.
- Countermeasures.

Disclaimer

- The informations on these slides should only be used for educational purposes.
- Any other use of the informations contained in these slides is prohibited.
- The author assumes no responsibility.

Introduction

Vulnerabilità, attacchi e contromisure nel mondo IoT

Iot

- "*The Internet of Things envisions a self-configuring, adaptive, complex network that interconnects things to the internet through the use of standard communication protocols. The interconnected things have physical or virtual representation in the digital world, sensing/actuation capability, a programmability feature, and are uniquely identifiable. The representation contains information including the thing's identity, status, location, or any other business, social or privately relevant information. The things offer services, with or without human intervention, through the exploitation of unique identification, data capture and communication, and actuation capability. The service is exploited through the use of intelligent interfaces and is made available anywhere, anytime, and for anything taking security into consideration*."

Iot - Security

Iot: Internet of Things

The **S** of Iot means **Security**

A complex ecosystem

- As security practitioners, we must be able to understand the value of these services and ensure that they are kept available and secure.
- Diversity and architectural heterogeneity.
- Sensor systems and internet connection.

Cps vs Iot

- I Cyber-Physical Systems (CPS) possiamo definirli come quei sistemi chiusi che con le loro architetture, gli attuatori, i misuratori e quanto altro gestiscono “industrialmente” un processo.

- It is a subset of the IOT.

- The substantial difference with the world of IOT is that cps are closed systems that have no connection to the Internet.

- The Iot, by definition, requires an Internet connection in order to provide its service.

Iot classification

Smart living environment for aging well	IoT systems support quality of life improvements while reducing care costs for the ageing population. These systems demonstrate the value of pervasive instrumentation and the impact that the IoT can make on an individual level.
Smart farming and food security	IoT systems enable precision farming and introduce new methods to assure food security and food safety. New autonomous technologies reduce workloads and increase quality.
Wearables	IoT systems become integrated into the fabric of our daily lives through integration with wearables, such as clothing, watches, and body-mounted devices.
Smart cities	IoT systems enable smart services for citizens, including transport, energy, health care, lighting, water, and waste. Populations will come to rely on these services, as on any other utility, as generations age.
Smart mobility	IoT systems transform the way we move, through the efficient management of traffic, automated transportation systems (for example, tolling), usage-based insurance, and connected and autonomous vehicles.
Smart water management	IoT systems enable more efficient water management capabilities while keeping our water supply safe and available.
Smart manufacturing	IoT systems such as industrial robotics and connected factories increase productivity and quality at manufacturing plants.
Smart energy	IoT systems support energy optimization across asset portfolios, including renewable plants, grid substations, control rooms, demand response applications, and **Electronic Vehicle (EV)** charging.
Smart buildings and architectures	IoT systems transform building management with a focus on occupant quality of life, through enhancements to lighting, comfort, temperature, air quality, water, nourishment, fitness, and energy use.

Iot ecosystem

A heterogeneous world !!!

Seven Layer - Iot ecosystem

- Physical devices and controllers.
- Connectivity.
- Edge computing.
- Data accumulation.
- Data abstraction.
- Application.
- Collaboration and processing.

Physical Device

- IoT devices often use a real-time operating system (RTOS) for process and memory management, as well as utility services that support messaging and other communications.

- The selection of each RTOS is based on the necessary performance, safety and functional requirements of the product.

Operating System

TinyOS	Optimized for low-power embedded systems. A framework that incorporates components that support development of an application-specific operating system. Written in NesC, which supports event-driven concurrency. Refer to `http://www.ann.ece.ufl.edu/courses/ee16935_10spr/papers/tinyos.pdf`.
Contiki	Supports IP, UDP, TCP, and HTTP, as well as 6loWPAN and CoAP. Designed for operation in low-power systems. Supports link layer encryption for 802.15.4 communications.
Windows 10 IoT	Supports bitlocker encryption and secure boot. Includes DeviceGuard and CredentialGuard features. Supports updates through **Windows Server Update Service** (**WSUS**).
QNX (Neutrino)	Operating System often used in vehicle infotainment systems. Includes security features such as sandboxing and fine-grained access controls.
Ubuntu Core	A read-only root file system, security sandbox for applications and separate (independent) update of applications from the OS. Allows categorization of applications as trusted or untrusted and supports **Unified Extensible Firmware Interface** (**UEFI**) secure boot. Learn more at `https://developer.ubuntu.com/en/snappy/guides/security-whitepaper`.
OpenWRT	A popular open source OS used often in wireless routers.
GreenHills IntegrityOS	A higher-assurance operating system.

Mantis	Embedded operating systems for wireless sensor platforms. Includes a kernel, scheduler, and networking stack. Supports remote update and remote login. Incorporates a sleep mode for power savings. Refer to: Sha, Carlson, et al. *Mantis OS: An Embedded* Multithreaded *Operating System for Wireless Micro Sensor Platforms. ACM Digital Library.*
Nano-RK	Tailored for surveillance and environmental monitoring applications. Supports energy-efficient mode of operation and preemptive multitasking. Runs on 2 KB RAM and 18 KB ROM.
Lite-OS	Supports a wirelessly accessible shell and a remote debugging system. Runs on 10 KB.
FreeRTOS	A general purpose RTOS. Supports add-on TCP networking and secure communications (TLS). Implementers can use cryptographic libraries such as WolfSSL with FreeRTOS.
SapphireOS	Supports mesh networking and device discovery. Includes Python tools and a RESTful API server.
BrilloOS	Runs on 32 to 64 MB RAM and optimized for consumer/home-based IoT devices.
uCLinux	Embedded Linux supports a variety of user applications, libraries, and tools. Learn more about uCLinux at `http://www.uclinux.org/pub/uClinux/FAQ.shtml`.
ARM Mbed OS	Incorporates a supervisory kernel (uVisor) that supports creation of isolated security domains on ARM Cortex M3, M4, and M7 MCUs with a **Memory Protection Unit** (**MPU**). Refer to `https://www.mbed.com/en/technologies/security/uvisor/`.
RIOT OS	Runs on 8-, 16-, and 32-bit platforms. Includes TCP/IP stack and supports 6LoWPAN, IPv6, UDP, and CoAP. Supports multithreading and requires 1.5 KB RAM and 5KB ROM.
VxWorks	Here are the two versions (VxWorks and VxWorks+). Includes optional add-on security profile with secure partitioning, secure boot, secure runtime, loader, and advanced user management. Supports encrypted containers and secure networking.
LynxOS	Supports TCP/IP, IPv6, and cellular communications. Supports 802.11 WiFi, ZigBee, and Bluetooth. Includes encryption support, access controls, and auditing and account management features.
Zephyr	Open source designed for resource-constrained systems. Project included a heavy focus on secure development practices. Implements nano-kernel and micro-kernel and supports Bluetooth, Bluetooth-LE, and 802.15.4 6LoWPAN.

Memories

- Configuration and storing of the secure parameters is a very critical point.
- Configuration settings that are applied to an operating system are often lost to the power cycle without battery-powered RAM or other persistent memory. In many cases, a configuration file is kept in persistent memory to provide the various networks and other settings necessary to allow the device to perform its functions and communicate with the outside world.
- It is very important to manage the root password, the other account passwords and the cryptographic keys stored on the devices when the device is turned off and on again.
- Each of these problems has one or more security implications.

Connectivity: Transport Protocol

- Iot use of **Tcp** and **Udp**.
- While tcp is used for Rest or for MQTT, the real protagonist is udp.
- **DTLS** (udp) is used to guarantee privacy and security.

Connectivity: Network Protocol

- **IPv4** and **IPv6** both play a role at various points within many IoT systems.
- IPv6 over Low Power Wireless Personal Area Networks (**6LoWPAN**) support the use of IPv6 in the network-constrained environments that many IoT devices operate within.
- **6LoWan** has been designed to support wireless internet connectivity at lower data rates for devices with very limited form factor.

Com. Protocol

Communication Protocol	Description
GPRS	All data and signals are encrypted using the **GPRS Encryption Algorithm (GEA)**. SIM cards are used to store identities and keys.
GSM	A **Time Division Multiple Access (TDMA)**-based cellular technology. SIM cards are used to store identities and keys.
UMTS	Signaling and user data are encrypted using a 128-bit key and the KASUMI algorithm.
CDMA	Code Division Multiple Access cellular technology. No SIM cards are used.
Long Range Wide Area Network (LoRaWAN)	Supports data rates between 0.3 Kbps and 50 Kbps. LoRAWAN networks use three keys: A unique network key, unique application key for end-to-end security, and device-specific key.
802.11	Wi-Fi. Standard wireless technologies used in many environments.
6LoWPAN	Low-power wireless **Personal Area Network (PAN)** designed to support automatic device network joining using a LoWPAN bootstrapping server to provision bootstrap information to 6LoPAN devices. A 6LoWPAN network includes an authentication server supporting mechanisms such as **Extensible Authentication Protocol (EAP)**. Bootstrap server can also be configured with a device blacklist.
ZigBee	ZigBee uses 802.15.4 for the physical and **Medium Access Control (MAC)** layers. ZigBee networks can be configured in star, tree, and mesh topologies. ZigBee security services provide key establishment, key transport, frame protection, and device management.
Thread	Thread uses 802.15.4 for the physical and MAC layers. Supports connection of up to 250 devices within a network. Uses AES encryption. Uses a **Password Authenticated Key Exchange (PAKE)**. New nodes join a network using a commissioner and DTLS to create a pairwise key that can be used to encrypt network parameters.
SigFox	Operates in the **Ultra Narrow Band (UNB)** in the 915 MHz (US) and 868 MHz (Europe) ranges. Devices sign messages with private keys. There is a limit of 140 messages per day per device, and SigFox supports anti-replay protections.
Near Field Communications (NFC)	Provide limited security protections. Often used in connection with another protocol. Short range support.
Wave 1609	Prevalent in CV communications. Relies heavily on IEEE 1609.2 certificates that support attribute tagging.

Threats, Vulnerability and Risks

Vulnerabilità, attacchi e contromisure nel mondo IoT

Essential security components

- **Confidentiality:** keep sensitive information secret and protected from disclosure.
- **Integrity:** ensuring that information is not changed, accidentally or intentionally, without being detected.
- **Authentication:** ensure that the source of the data comes from a known identity or endpoint.
- **Non-repudiation:** ensuring that an individual or a system cannot later deny having performed an action.
- **Availability:** ensure that information and capabilities are available when needed.

Threats

- IoT threats include all threats related to the management and manipulation of information to management, application, sensor and control data sent to and from IoT devices.

- IoT devices are subject to the same physical security, hardware, software quality, environment, supply chain and many other threats common to security sectors.

- In addition, IOT devices are subject to threats of physical reliability and resilience or to the degradation of the computing platform.

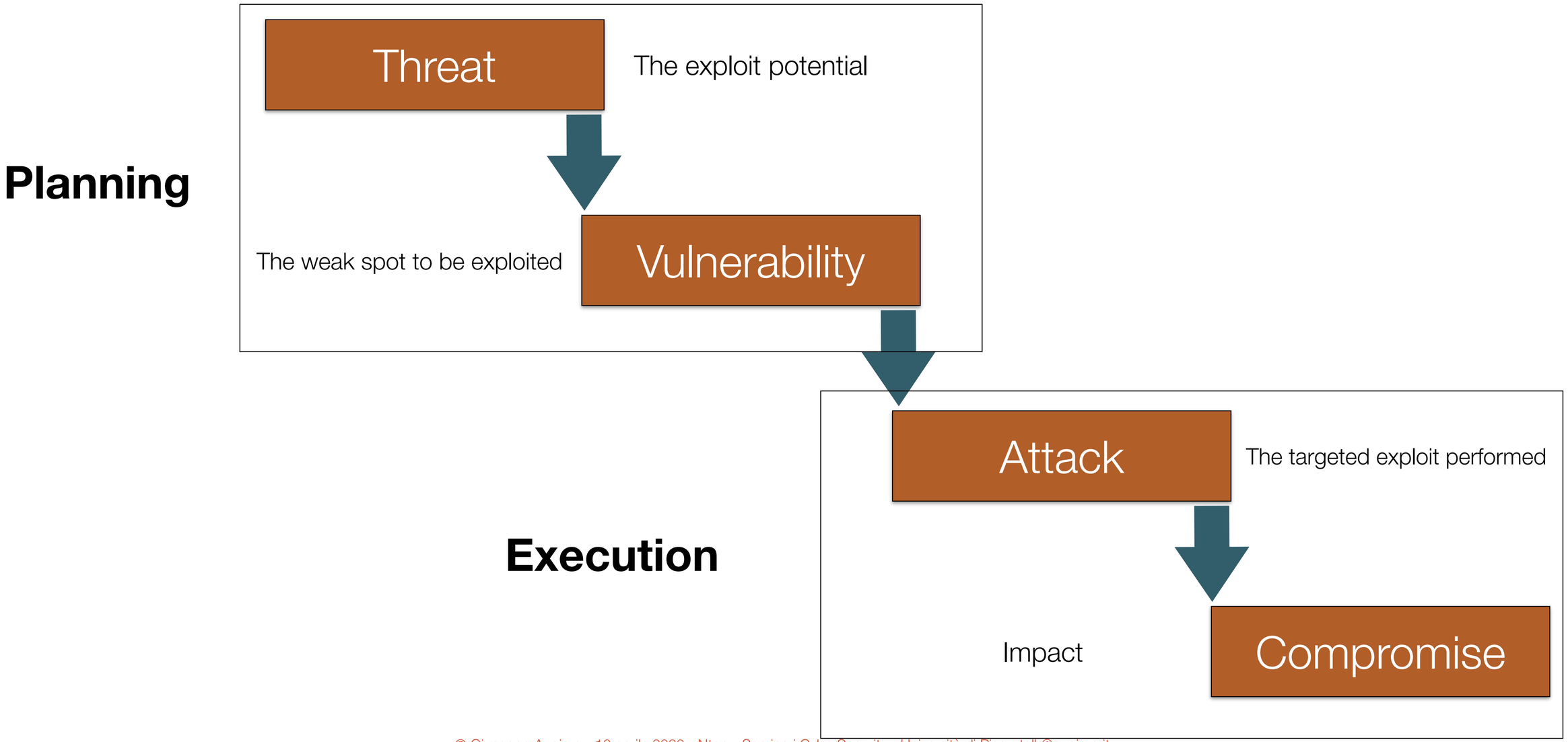
Vulnerability

- **Vulnerability** is the term we use to identify a weakness in the design, integration or operation of a system or device.

- Vulnerabilities are always present and countless new ones are discovered every day. Many online databases and web portals now provide us with automatic updates on newly discovered vulnerabilities.

- **Vulnerabilities are shortcomings.**

Threat actor



Risks vs Vulnerabilities

- **Risk exposure** is the measure of potential future **loss** resulting from a specific activity or event.

- It is different from **vulnerability,** because it depends on the probability of a particular event, attack or condition and has a strong connection with the motivations of an attacker.

- Vulnerability does not directly invoke impact or probability, but it is innate weakness itself. It can be easy or difficult to exploit or cause a small or large loss when it is exploited.

- Not all vulnerabilities will be known in advance. We call them **zero-day.**

Risk management

- Risk can be managed through threat modeling, which allows you to ascertain the following:
  - Impact and overall cost of a compromise.
  - How valuable the target can be for attackers.
  - Predicted skills and motivations of the attackers (based on the threat model).
  - Preliminary knowledge of a system or device vulnerabilities (for example, those identified in public notices, discovered during threat modeling and penetration tests).

Residual risk

- Since mitigation of security controls is never perfect, we still have a small residual amount of risk, generally called “**residual risk**”.

- **Residual risk** is often accepted as it is or offset by the application of other risk compensation mechanisms, such as insurance.

Types of attacks on IOT devices

- There are many types of attacks but the most significant for the IOT world are:
  - Wired and wireless scanning and mapping attacks.
  - Protocol attacks.
  - Eavesdropping attacks (loss of confidentiality).
  - Cryptographic algorithm and key management attacks.
  - Spoofing and masquerading (authentication attacks).
  - Operating system and application integrity attacks.
  - Denial of service and jamming.
  - Physical security attacks (for example, tampering and interface exposures).
  - Access control attacks (privilege escalation).

Ransomware

- We can translate the concept of Ransomware into the IOT world.
- Imagine that ransom attacks are being carried out on physical infrastructure or medical equipment.
- Examples:
  - A pacemaker that gives short non-lethal shocks.
  - A car that can't get out because the garage doors don't open.
  - Front door that opens when we are on vacation.

Attack campaigns

- For how many attacks we can define or imagine, in reality the number of types of attacks is always greater and often the attacks are customized for the target to be hit.

- **Generally only one type of attack is ever used**.

- An attack is the set of a campaign of sub-attacks grouped in sequence or other activities, each carefully chosen by a variety of intelligence methods (human social engineering, profiling, scanning, Internet research, and familiarity with the system).

- Each activity designed to achieve its immediate goal has a certain level of difficulty, cost and probability of success.

Attack surface

- Attack surface refer to the many ways in which a device can be compromised via an input source.
- This input source can take place via hardware, via software or wirelessly.
- In general, the greater the attack surface contained in a device, the greater the probability of a compromise.
- **Attack surfaces are entry points into the IoT device**.
- Each attack surface discovered will have an associated risk, probability, and impact.
- Attack surfaces are threats that have the potential to adversely affect a device to perform unwanted actions.

Threats modeling

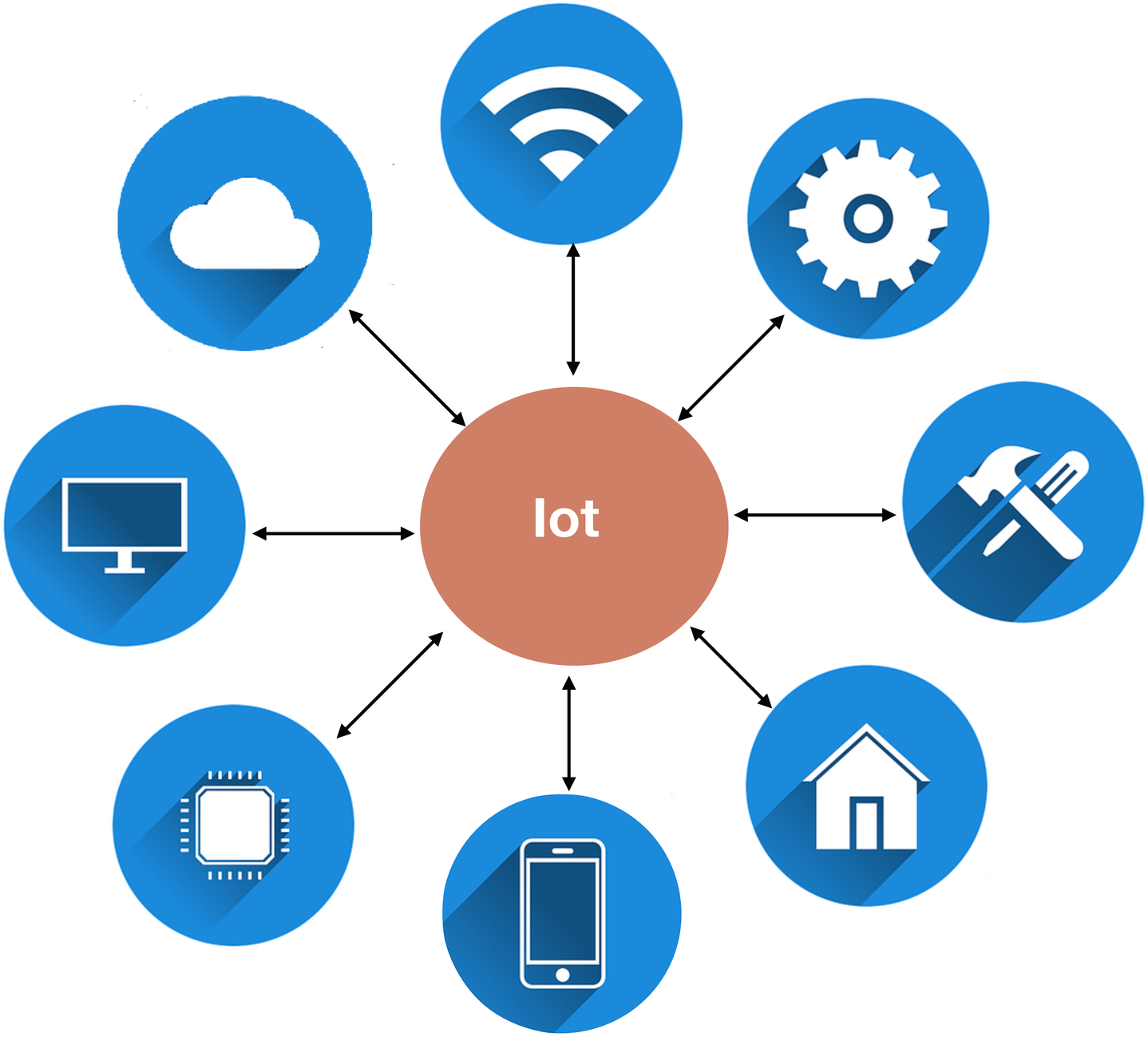
- To find out each attack surface, theoretical use cases must be considered before the test takes place or before the software is written.

- This exercise is known as **threat modeling**.

- There are different threat classification systems, depending on the sector; however, the most common are **DREAD** and the Common Vulnerability Scoring System (**CVSS**).

DREAD

- A threat classification system must be able to quantify the following risk variables:
  - **Potential:** how great is the damage if exploited?
  - **Reproducibility:** how easy is it to reproduce the attack?
  - **Ease of attack:** how easy is it to exploit the attack?
  - **Users affected:** how many users are exposed to risk?
  - **Detectability:** How easy is it to find vulnerability?



Attacks and attack tools

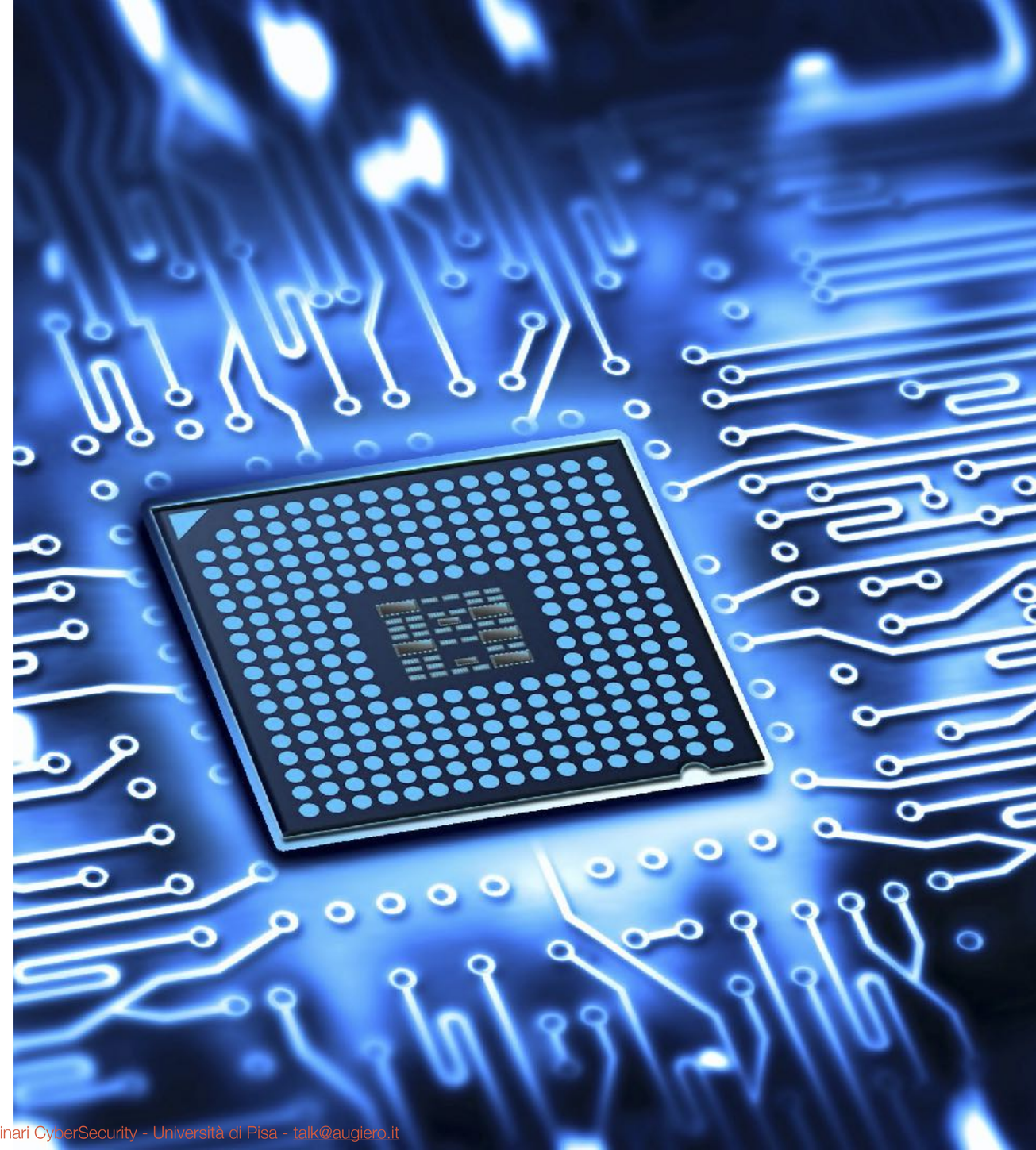
Vulnerabilità, attacchi e contromisure nel mondo IoT

How can I attack a system?

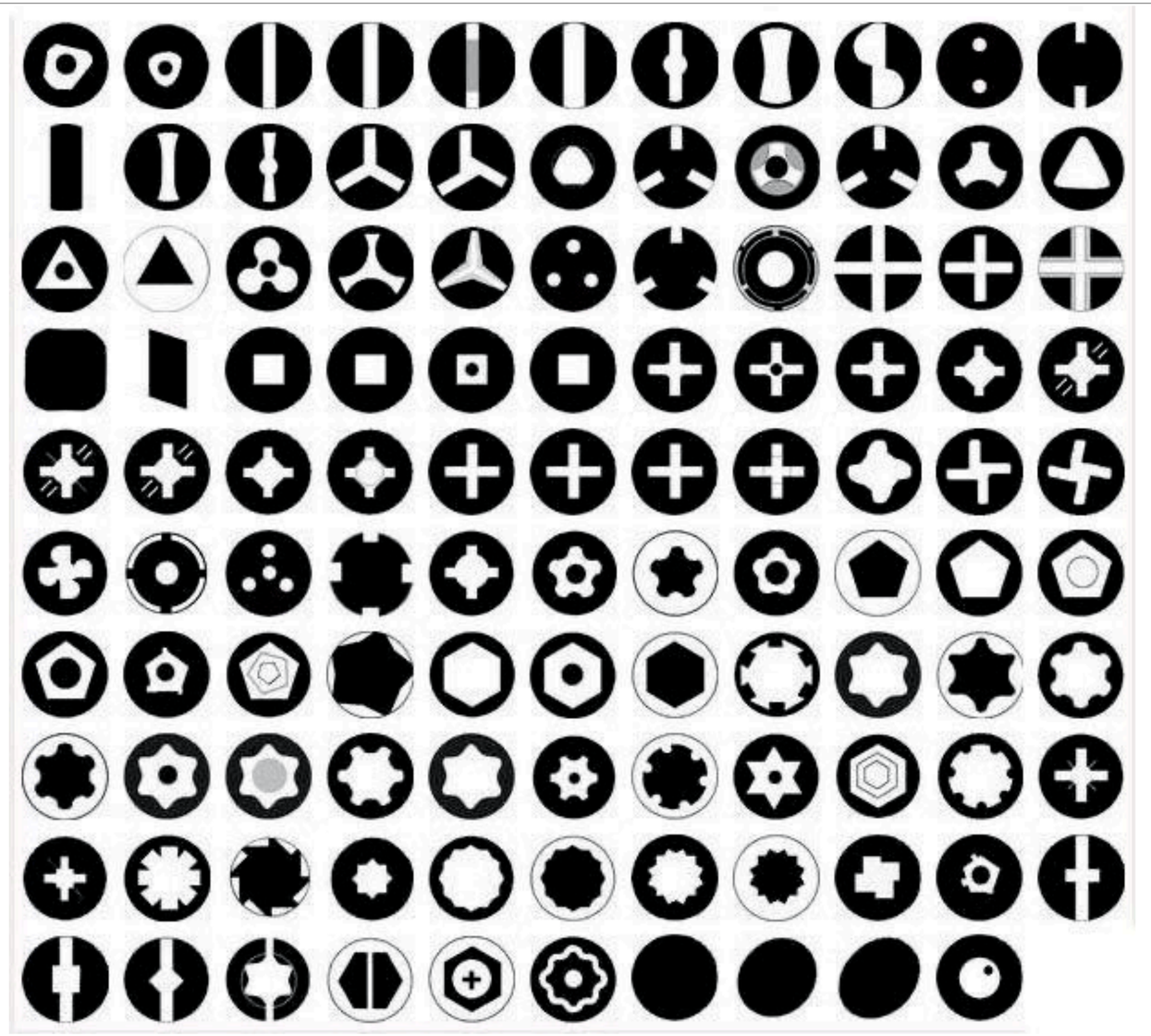
Just Thinking Outside The Box

Device

Vulnerabilità, attacchi e contromisure nel mondo IoT



Screws



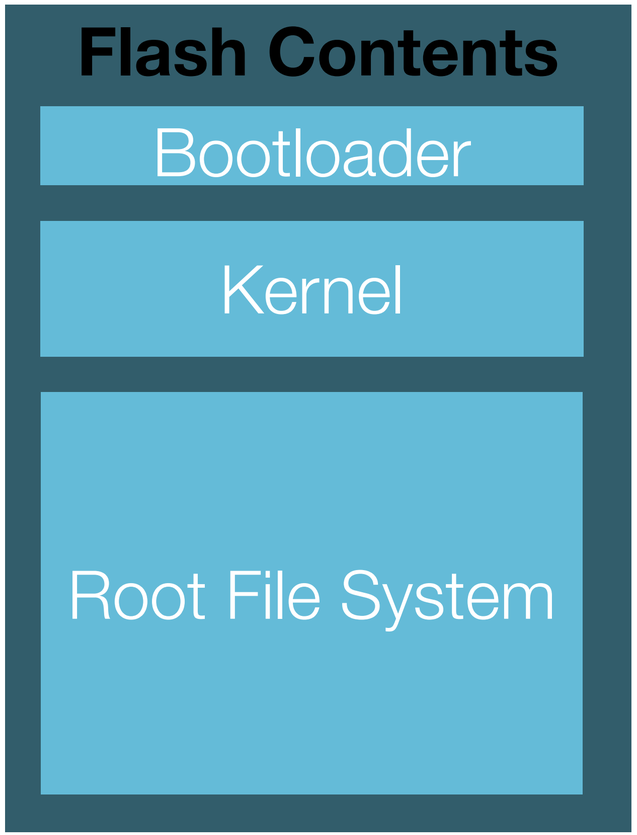
Firmware

- Firmware is the control center of IoT devices, which is why we may want to start analyzing its contents before other pieces of device components.

- Depending on the production sector of the IoT device, obtaining a firmware image and disassembling its contents can be trivial.

Struttura

- Generally the firmware is composed as follows:



Informations offered by the firmware

- By analyzing the firmware we can obtain:
  - Passwords.
  - API tokens.
  - API endpoints (URLs).
  - Vulnerable services.
  - Backdoor accounts.
  - Configuration files.
  - Source code.
  - Private keys.
  - How data is stored.

How to get the firmware

- Here are the possible cases of how to obtain the firmware to be analyzed:
  - Download the firmware directly from the vendor website.

Proxying

- Proxying network traffic while the device updates itself:
  - Use Linux Distribution (for example Kali Linux)
  - Enable IP forwarding:
    - *echo 1 > /proc/sys/net/ipv4/ip_forward*
  - Configure iptables to redirect traffic from destination port 80 to port 8080, which is what SSLstrip listens on:
    - *iptables -t nat -p tcp -A PREROUTING --dport 80 -j REDIRECT --to- port 8080*
  - Start SSLstrip:
    - *ssltrip -a -l 8080 -w logfile*
    - Attenzione all'HSTS
  - Start Ettercap:
    - *ettercap -T (o -G)*
  - Start Wireshark
    - *wireshark*

Dump Firmware

- Iot devices can have a Jtag, Serial, Spi service port.
  - In some cases it is necessary to weld the service door feet.
  - Once ready, just use the command: *flashrom* (with a little option) to dump the firmware.

Google Dorking

- If you can't download the firmware image, we can always use google dorking.
- We can search the google hacking database:
  - https://www.exploit-db.com/google-hacking-database

Analyze the firmware (1/2)

- The firmware contains the **bootloader**, the **kernel** and the **root file system**.
- We are interested in being able to analyze, in the first instance, the filesystem so that we can access the configuration files, keys and more of the device.
- **The firmware is a binary container and we do not know, at the start, the start offset of the file system and its size**.

Analyze the firmware (2/2)

- We can use hexdump and grep to search for the magic number of the file system (for example shsq for squashfs filesystem).
  - *Hexdump -C firmware.bin | grep -i 'hsqs')*
  - Once the offset is found, we can use dd to extract the file system:
  - *dd if=firmware.bin of=filesystem bs=1 skip=[offset (decimal)]*
- At this point we can examine the filesystem through the command:
  - *Unsquashfs*

Binwalk

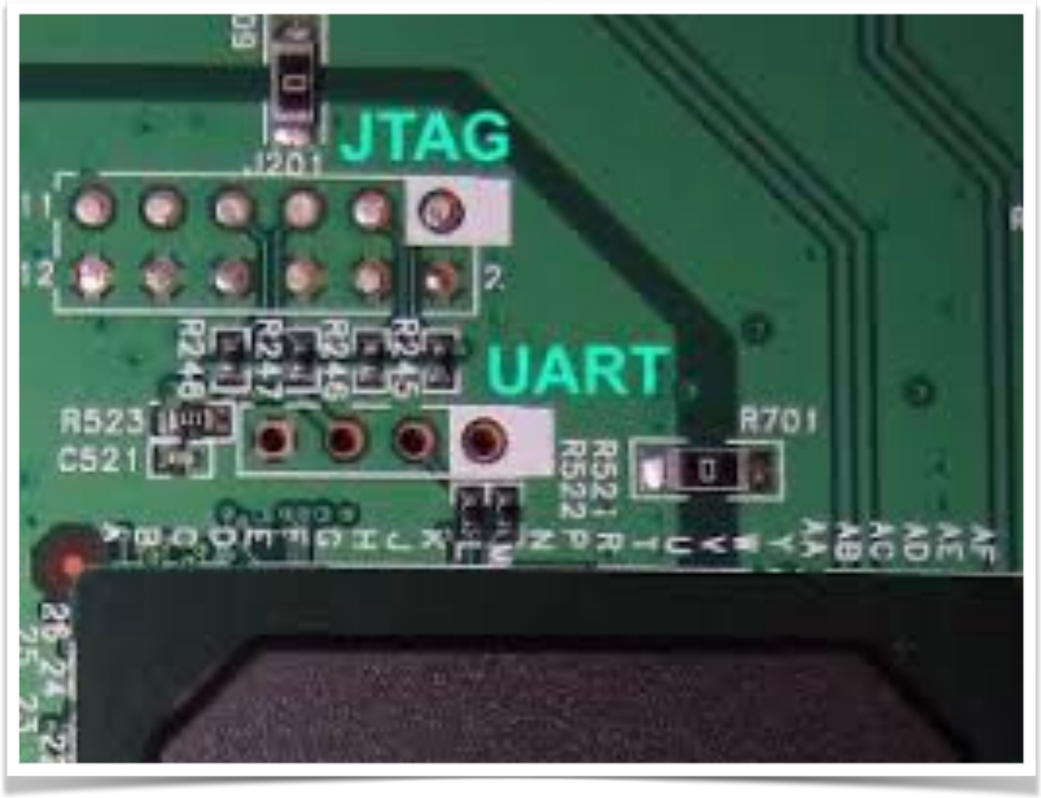
- We can automate the operations we performed using the command:
  - *binwalk -e firmware.bin*

Analyze the filesystem

- To analyze the filesystem we can use two approaches:
  - **Manual.** Searching for configuration files and passwords.
  - **Automatic.** Helping us with tools used for forensic analysis.

- We can perform the static analysis of the firmware through:
  - *firmwalker*

Uart



- Recognize baud rate:
  - https://github.com/devttys0/baudrate/blob/master/baudrate.py.

Glitch

- What do we do on the serial we have no console?
- **Glitch** is a way of causing failures in the system you are working on.
- Example (NAND glitching):
  - To access the bootloader, one of the I/O pins of our device's NAND flash is short-circuited on a GND pin. Note that this short circuit must be done at the moment the bootloader has started and the kernel is about to boot.
  - Once the boot procedure has been interrupted, it will be possible to start the system in single mode, thus bypassing the authentication system.
- However, you can also use power and voltage glitching techniques to perform things such as bypassing crypto and more.

Web

Vulnerabilità, attacchi e contromisure nel mondo IoT

Bruce Schneier

- **Cryptography is harder than it looks:**
  - Primarily because it looks like math…
- **Complexity is the worst enemy of security:**
  - The more complex a system is, the more lines of code, interactions with other systems, configuration options, and vulnerabilities there are.

Web App Security Testing

- The right methodology must be defined to carry out the correct search for software vulnerabilities.

- The identical techniques used for a penetration test can help us analyze an IOT product.

Owasp methodologies

- Introduction and objectives.
- Information gathering.
- Configuration and deployment management testing.
- Identity management testing.
- Authentication testing.
- Authorization testing.
- Session management testing.
- Input validation testing.
- Error handling.
- Cryptography.
- Business logic testing.
- Client-side testing.

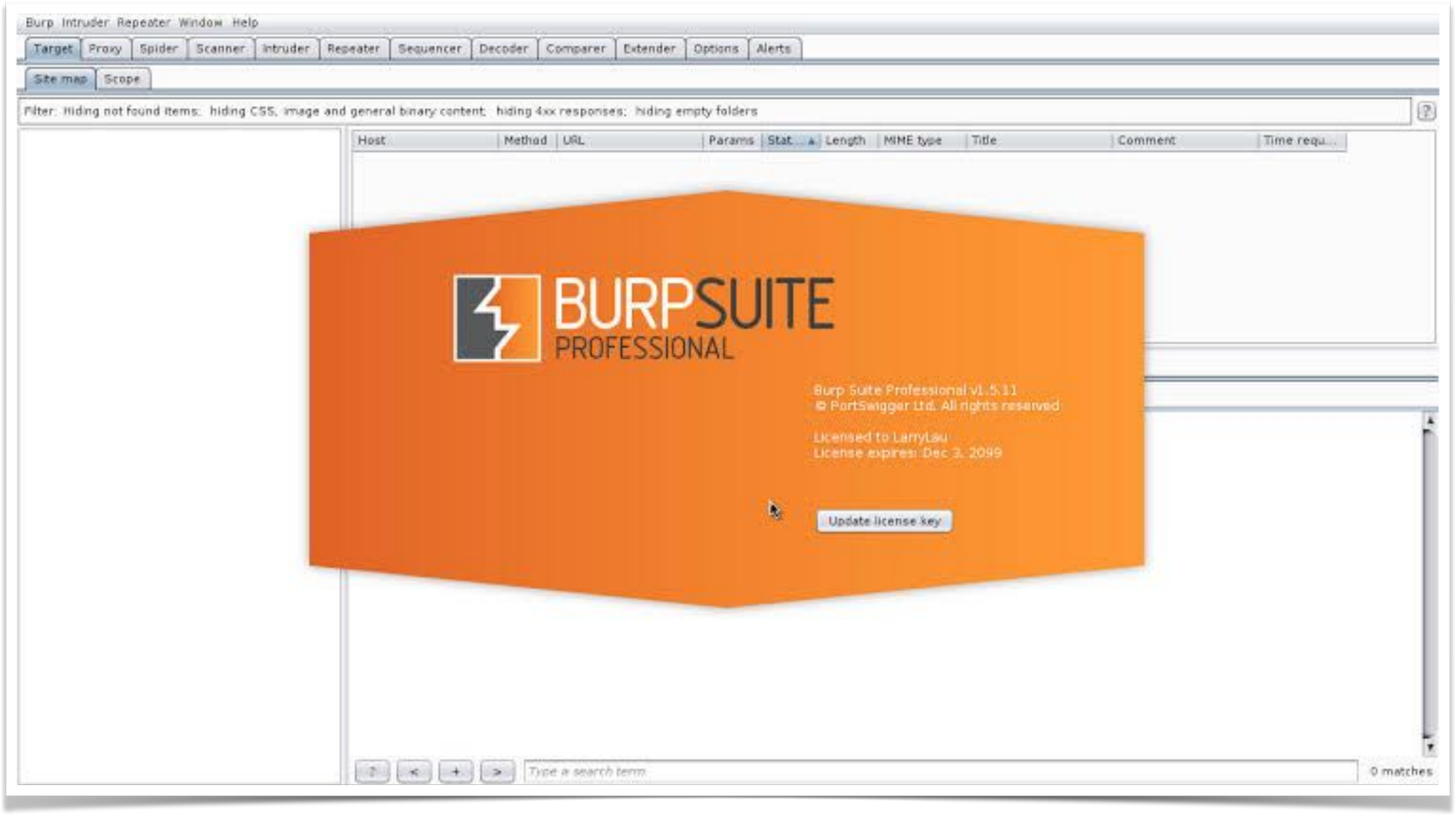
First Step

- The tools that we can use are varied and not all necessary.
- The first step is to choose the browser to use:
  - A good choice is to use Firefox as it offers countless add-ons.
  - The other browsers are not to be excluded. It may be helpful to use Internet Explorer for sites with ActiveX components.

Browser Plugin

- Some (basic) components to install are:
  - Wappalyzer.
  - FoxyProxy.
  - Cookie Manager.

Burp Suite


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Site map Scope
Filter: hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
Host Method URL Params Stat Length MIME type Title Comment Time requ...
BURPSUITE
PROFESSIONAL
Burp Suite Professional v1.5.11
© PortSwigger Ltd. All rights reserved
Licensed to LarryLau
License expires: Dec 3, 2099
Update license key
Type a search term
0 matches

Burp Suite - Live Demo

- Brute force Basic Authentication
  - http://pentesteracademylab.appspot.com/lab/webapp/basicauth
    - (Challenge 3)
  - CA: http://burp/cert

Burp Scanner

This listing contains the definitions of all issues that can be detected by Burp Scanner.

Name	Typical severity	Type index
OS command injection	High	0x00100100
SQL injection	High	0x00100200
SQL injection (second order)	High	0x00100210
ASP.NET tracing enabled	High	0x00100280
File path traversal	High	0x00100300
XML external entity injection	High	0x00100400
LDAP injection	High	0x00100500
XPath injection	High	0x00100600
XML injection	Medium	0x00100700
ASP.NET debugging enabled	Medium	0x00100800
HTTP PUT method is enabled	High	0x00100900
Out-of-band resource load (HTTP)	High	0x00100a00
File path manipulation	High	0x00100b00
PHP code injection	High	0x00100c00
Server-side JavaScript code injection	High	0x00100d00
Perl code injection	High	0x00100e00
Ruby code injection	High	0x00100f00
Python code injection	High	0x00100f10
Expression Language injection	High	0x00100f20
Unidentified code injection	High	0x00101000
Server-side template injection	High	0x00101080
SSI injection	High	0x00101100
Cross-site scripting (stored)	High	0x00200100
HTTP request smuggling	High	0x00200140
Web cache poisoning	High	0x00200180
HTTP response header injection	High	0x00200200
Cross-site scripting (reflected)	High	0x00200300
Client-side template injection	High	0x00200308
Cross-site scripting (DOM-based)	High	0x00200310
Cross-site scripting (reflected DOM-based)	High	0x00200311
Cross-site scripting (stored DOM-based)	High	0x00200312
JavaScript injection (DOM-based)	High	0x00200320
JavaScript injection (reflected DOM-based)	High	0x00200321
JavaScript injection (stored DOM-based)	High	0x00200322
Path-relative style sheet import	Information	0x00200328
Client-side SQL injection (DOM-based)	High	0x00200330
Client-side SQL injection (reflected DOM-based)	High	0x00200331
Client-side SQL injection (stored DOM-based)	High	0x00200332
WebSocket URL poisoning (DOM-based)	High	0x00200340
WebSocket URL poisoning (reflected DOM-based)	High	0x00200341
WebSocket URL poisoning (stored DOM-based)	High	0x00200342
Local file path manipulation (DOM-based)	High	0x00200350
Local file path manipulation (reflected DOM-based)	High	0x00200351
Local file path manipulation (stored DOM-based)	High	0x00200352
Client-side XPath injection (DOM-based)	Low	0x00200360
Client-side XPath injection (reflected DOM-based)	Low	0x00200361
Client-side XPath injection (stored DOM-based)	Low	0x00200362
Client-side JSON injection (DOM-based)	Low	0x00200370
Client-side JSON injection (reflected DOM-based)	Low	0x00200371
Client-side JSON injection (stored DOM-based)	Low	0x00200372
Flash cross-domain policy	High	0x00200400
Silverlight cross-domain policy	High	0x00200500
Cross-origin resource sharing	Information	0x00200600

Cross-site scripting (stored DOM-based)

Description

Stored DOM-based vulnerabilities arise when user input is stored and later embedded into a response within a part of the DOM that is then processed in an unsafe way by a client-side script. An attacker can leverage the data storage to control a part of the response (for example, a JavaScript string) that can be used to trigger the DOM-based vulnerability.

DOM-based cross-site scripting arises when a script writes controllable data into the HTML document in an unsafe way. An attacker may be able to use the vulnerability to construct a URL that, if visited by another application user, will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session with the application.

The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

Users can be induced to visit the attacker's crafted URL in various ways, similar to the usual attack delivery vectors for reflected cross-site scripting vulnerabilities.

Burp Suite automatically identifies this issue using static code analysis, which may lead to false positives that are not actually exploitable. The relevant code and execution paths should be reviewed to determine whether this vulnerability is indeed present, or whether mitigations are in place that would prevent exploitation.

Remediation

The most effective way to avoid DOM-based cross-site scripting vulnerabilities is not to dynamically write data from any untrusted source into the HTML document. If the desired functionality of the application means that this behavior is unavoidable, then defenses must be implemented within the client-side code to prevent malicious data from introducing script code into the document. In many cases, the relevant data can be validated on a whitelist basis, to allow only content that is known to be safe. In other cases, it will be necessary to sanitize or encode the data. This can be a complex task, and depending on the context that the data is to be inserted may need to involve a combination of JavaScript escaping, HTML encoding, and URL encoding, in the appropriate sequence.

References

- Cross-site scripting

Vulnerability classifications

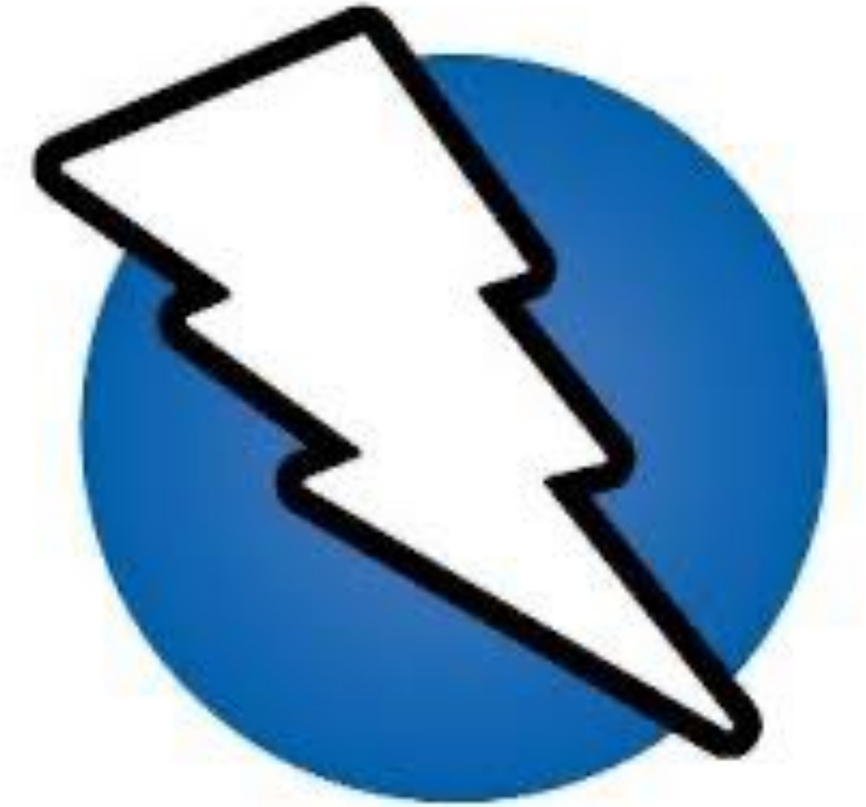
- CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- CWE-80: Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)
- CWE-116: Improper Encoding or Escaping of Output
- CWE-159: Failure to Sanitize Special Element

Typical severity

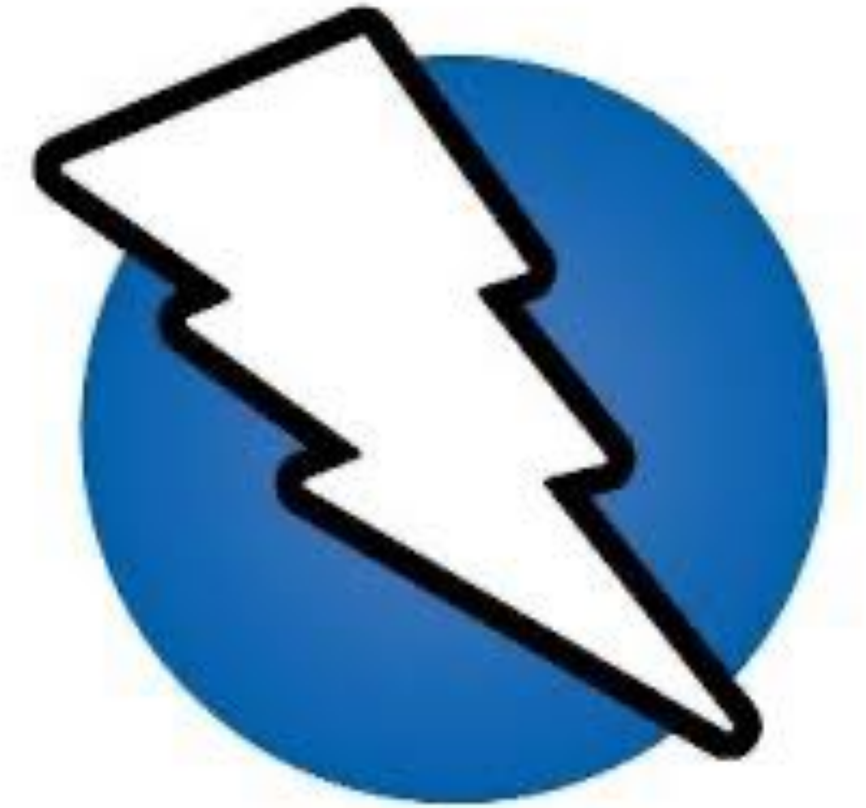
High

Type index

Owasp Zap



Owasp Zap - Live Demo



Mobile

Vulnerabilità, attacchi e contromisure nel mondo IoT

Mobile methodologies

- We can apply 4 different methodologies:
  - **Application mapping:** Application mapping pertains to the application's logic and the application's business function. Think of application mapping as gathering information about the application to be used in the next phase.
  - **Client-side attacks:** Client-side attacks pertain to data being stored in the application and how that data can be manipulated from the client side.
  - **Network attacks:** Network attacks pertain to network layer concerns such as SSL/TLS or maybe XMPP protocol data.
  - **Server attacks:** Server attacks apply to API vulnerabilities and backend server misconfigurations brought to light as a result of API testing.
- Mobile Application Security Verification Standard (**MASVS**).

Apk

- **Enjarify** is a tool for translating Dalvik bytecode to equivalent Java bytecode. This allows Java analysis tools to analyze Android applications.
  - https://github.com/google/enjarify
  - *python3 -O -m enjarify.main yourapp.apk*

Protocols

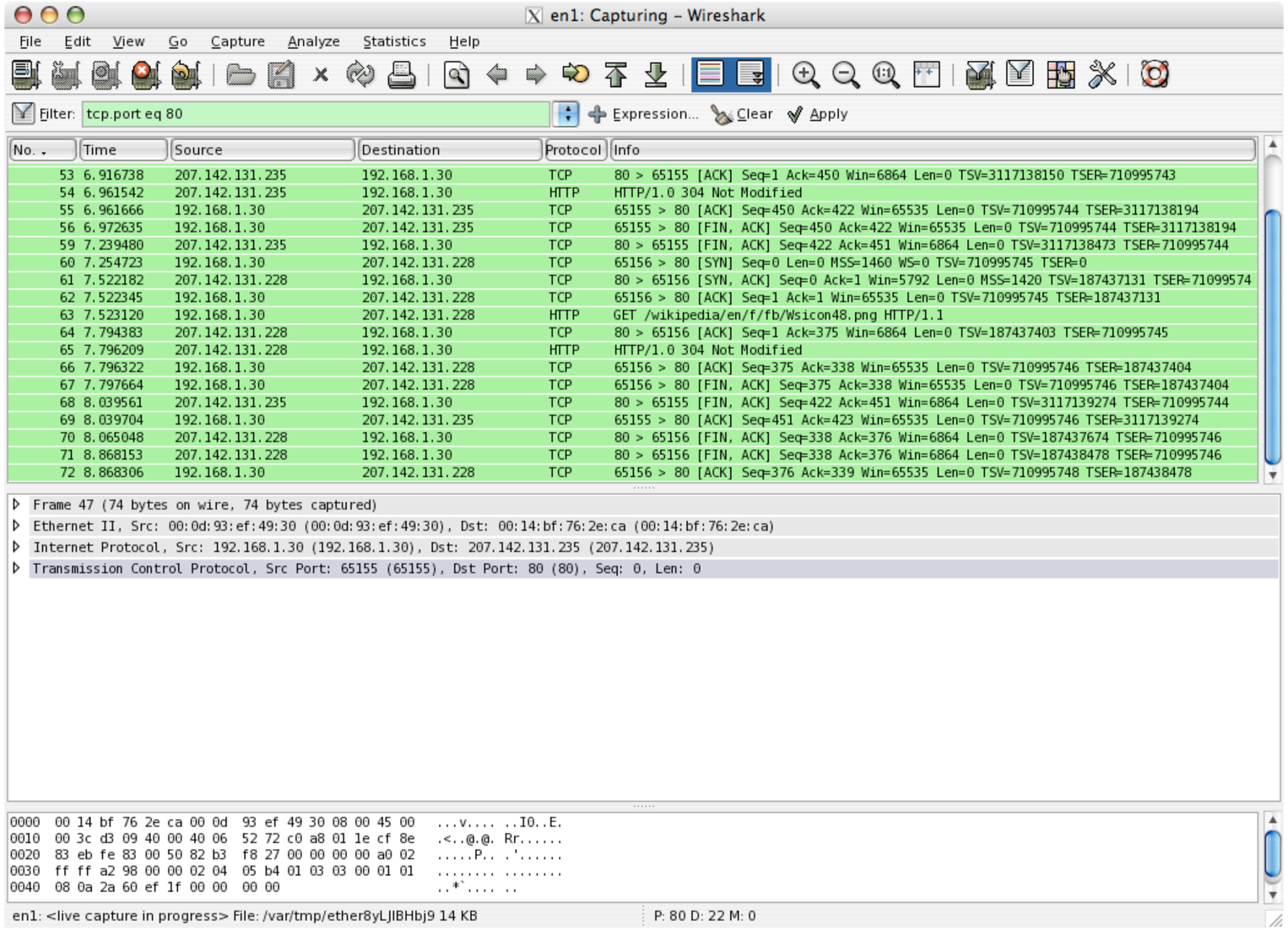
Vulnerabilità, attacchi e contromisure nel mondo IoT

Communication

- Almost all IoT devices interact with other devices to exchange informations and acts.

- It is extremely essential to know the wireless protocols used by IoT devices and the security issues affecting them, in order to effectively test IoT devices.

Wireshark


en1: Capturing – Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: tcp.port eq 80
Expression... Clear Apply
No. Time Source Destination Protocol Info
53 6.916738 207.142.131.235 192.168.1.30 TCP 80 > 65155 [ACK] Seq=1 Ack=450 Win=6864 Len=0 TSV=3117138150 TSER=710995743
54 6.961542 207.142.131.235 192.168.1.30 HTTP HTTP/1.0 304 Not Modified
55 6.961666 192.168.1.30 207.142.131.235 TCP 65155 > 80 [ACK] Seq=450 Ack=422 Win=65535 Len=0 TSV=710995744 TSER=3117138194
56 6.972635 192.168.1.30 207.142.131.235 TCP 65155 > 80 [FIN, ACK] Seq=450 Ack=422 Win=65535 Len=0 TSV=710995744 TSER=3117138194
59 7.239480 207.142.131.235 192.168.1.30 TCP 80 > 65155 [FIN, ACK] Seq=422 Ack=451 Win=6864 Len=0 TSV=3117138473 TSER=710995744
60 7.254723 192.168.1.30 207.142.131.228 TCP 65156 > 80 [SYN] Seq=0 Len=0 MSS=1460 WS=0 TSV=710995745 TSER=0
61 7.522182 207.142.131.228 192.168.1.30 TCP 80 > 65156 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1420 TSV=187437131 TSER=71099574
62 7.522345 192.168.1.30 207.142.131.228 TCP 65156 > 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0 TSV=710995745 TSER=187437131
63 7.523120 192.168.1.30 207.142.131.228 HTTP GET /wikipedia/en/f/fb/Wsicon48.png HTTP/1.1
64 7.794383 207.142.131.228 192.168.1.30 TCP 80 > 65156 [ACK] Seq=1 Ack=375 Win=6864 Len=0 TSV=187437403 TSER=710995745
65 7.796209 207.142.131.228 192.168.1.30 HTTP HTTP/1.0 304 Not Modified
66 7.796322 192.168.1.30 207.142.131.228 TCP 65156 > 80 [ACK] Seq=375 Ack=338 Win=65535 Len=0 TSV=710995746 TSER=187437404
67 7.797664 192.168.1.30 207.142.131.228 TCP 65156 > 80 [FIN, ACK] Seq=375 Ack=338 Win=65535 Len=0 TSV=710995746 TSER=187437404
68 8.039561 207.142.131.235 192.168.1.30 TCP 80 > 65155 [FIN, ACK] Seq=422 Ack=451 Win=6864 Len=0 TSV=3117139274 TSER=710995744
69 8.039704 192.168.1.30 207.142.131.235 TCP 65155 > 80 [ACK] Seq=451 Ack=423 Win=65535 Len=0 TSV=710995746 TSER=3117139274
70 8.065048 207.142.131.228 192.168.1.30 TCP 80 > 65156 [FIN, ACK] Seq=338 Ack=376 Win=6864 Len=0 TSV=187437674 TSER=710995746
71 8.868153 207.142.131.228 192.168.1.30 TCP 80 > 65156 [FIN, ACK] Seq=338 Ack=376 Win=6864 Len=0 TSV=187438478 TSER=710995746
72 8.868306 192.168.1.30 207.142.131.228 TCP 65156 > 80 [ACK] Seq=376 Ack=339 Win=65535 Len=0 TSV=710995748 TSER=187438478
Frame 47 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: 00:0d:93:ef:49:30 (00:0d:93:ef:49:30), Dst: 00:14:bf:76:2e:ca (00:14:bf:76:2e:ca)
Internet Protocol, Src: 192.168.1.30 (192.168.1.30), Dst: 207.142.131.235 (207.142.131.235)
Transmission Control Protocol, Src Port: 65155 (65155), Dst Port: 80 (80), Seq: 0, Len: 0
0000 00 14 bf 76 2e ca 00 0d 93 ef 49 30 08 00 45 00 ...v.... ..I0..E.
0010 00 3c d3 09 40 00 40 06 52 72 c0 a8 01 1e cf 8e .<..@.@. Rr.....
0020 83 eb fe 83 00 50 82 b3 f8 27 00 00 00 00 a0 02 .....P.. .'......
0030 ff ff a2 98 00 00 02 04 05 b4 01 03 03 00 01 01 ....
0040 08 0a 2a 60 ef 1f 00 00 00 00 ..*`.... ..
en1: <live capture in progress> File: /var/tmp/ether8yLJIBHbj9 14 KB
P: 80 D: 22 M: 0

Zig Bee

- ZigBee is one of the common wireless protocols used in IoT devices because of its ability to form mesh networks and perform operations with low power and resource consumption.

- What do we need:

  - **Hardware:** Atmel RzRaven USB Stick flashed with KillerBee firmware

  - **Software:** KillerBee

    - https://github.com/riverloopsec/killerbee

BLE

- BLE is designed for devices with resource and power constraints which BLE effectively solves by providing short bursts of long range radio connections, thus significantly saving battery consumption.

- What do we need:

  - **Hardware:** Dongle Bluetooth

  - **Software:** Blue Hydra or HCI Utils, Ubertooth utils, Gattacker

Countermeasures

Vulnerabilità, attacchi e contromisure nel mondo IoT

Iot devices connected to the Internet

- Does the device you are using really require internet access?
- Use **vpn** and avoid, when possible, triangulation systems or external TLS tunnels.
- Border on the devices (vlan).
- Net division between outside and inside of the network.

Metrics

- Adopt network metrics.
- Deep Inspection.
- Analysis of the network guidelines (cum grano salis).

- **Ntpong is an excellent tool.**

Security & Policy

- Protect external access to the Iot device with a **firewall** (L3 / L7).
- Do not allow access to unnecessary IOT device services.
- Always keep your device updated.

Authentication

- Use different credentials for different devices.

Iot device to avoid

- Avoid products with these characteristics:
  - not support TLS.
  - too little computing power.
  - completely depend on an external cloud.
  - rebranded products.
  - (do not support Ipv6).
- Avoid products you don't need.

Utilizzation

- When you can turn off your device.

Conclusions

Vulnerabilità, attacchi e contromisure nel mondo IoT

Iot

- The world of IoT opens up new challenges regarding the privacy and confidentiality of data.
- Pervasive presence of the IoT ecosystem.
- New privacy approaches for credentials.
- New network traffic will be required.
- Our information will be increasingly distant from us.

Tomorrow

The biggest challenge in the security industry is finding ways to defend against tomorrow's attacks today, as many devices and systems are expected to work for years or decades in the future.

Thanks for the attention!

Iot CyberSecurity

Vulnerabilità, attacchi e contromisure nel mondo IoT

Giuseppe Augiero
Email: talk@augiero.it
Website: augiero.it