Risk assessment, threat modelling and cascading threats: Analyzing IoT-enabled, cyber physical attack paths

Panayiotis Kotzanikolaou
pkotzani@unipi.gr
SecLab, Department of Informatics
University of Piraeus, Greece

COINS summer school 2021 – 14 June 2021

Speaker’s Brief Intro

- I am an Associate Professor at the Department of Informatics, University of Piraeus, Greece.
- I am currently directing the cybersecurity research lab @Dept.Informatics (https://seclab.cs.unipi.gr)
- My current research interests include:
  - CIP and Risk Assessment for CIs
  - Cascading Threats, Risk and Mitigation of relevant threats
  - IoT-enabled, cyber-physical attack path analysis
  - Resilience by design

Outline

1. Introducing the Threat Landscape
   - The traditional threat landscape of Critical Infrastructures
   - CIs and IoT: Interactions, connectivity and the new threat landscape
2. IoT-enabled attacks against CIs
   - Cyber physical attack paths against cyber-physical systems
   - Current status – Analysis of real-world incidents and PoC IoT-enabled attacks against CIs
   - Potential impact
3. Identifying and Assessing IoT-enabled Attack Paths against Critical Systems
   - Existing Risk Assessment methodologies
   - Identifying C-P attack paths
   - Assessing C-P attack paths
   - Test case validation
   - Future research

Presentation based on:

Stellios, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., & Lopez, J. (2018). *"A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services". IEEE Communications Surveys & Tutorials*, *20*(4), 3453-3495.

Stellios I., Kotzanikolaou P. and Grigoriadis C., *"Assessing IoT enabled cyber-physical attack paths against critical systems"*. Elsevier Computers and Security, Vol.107, August 2021, 102316

1. The Threat Landscape

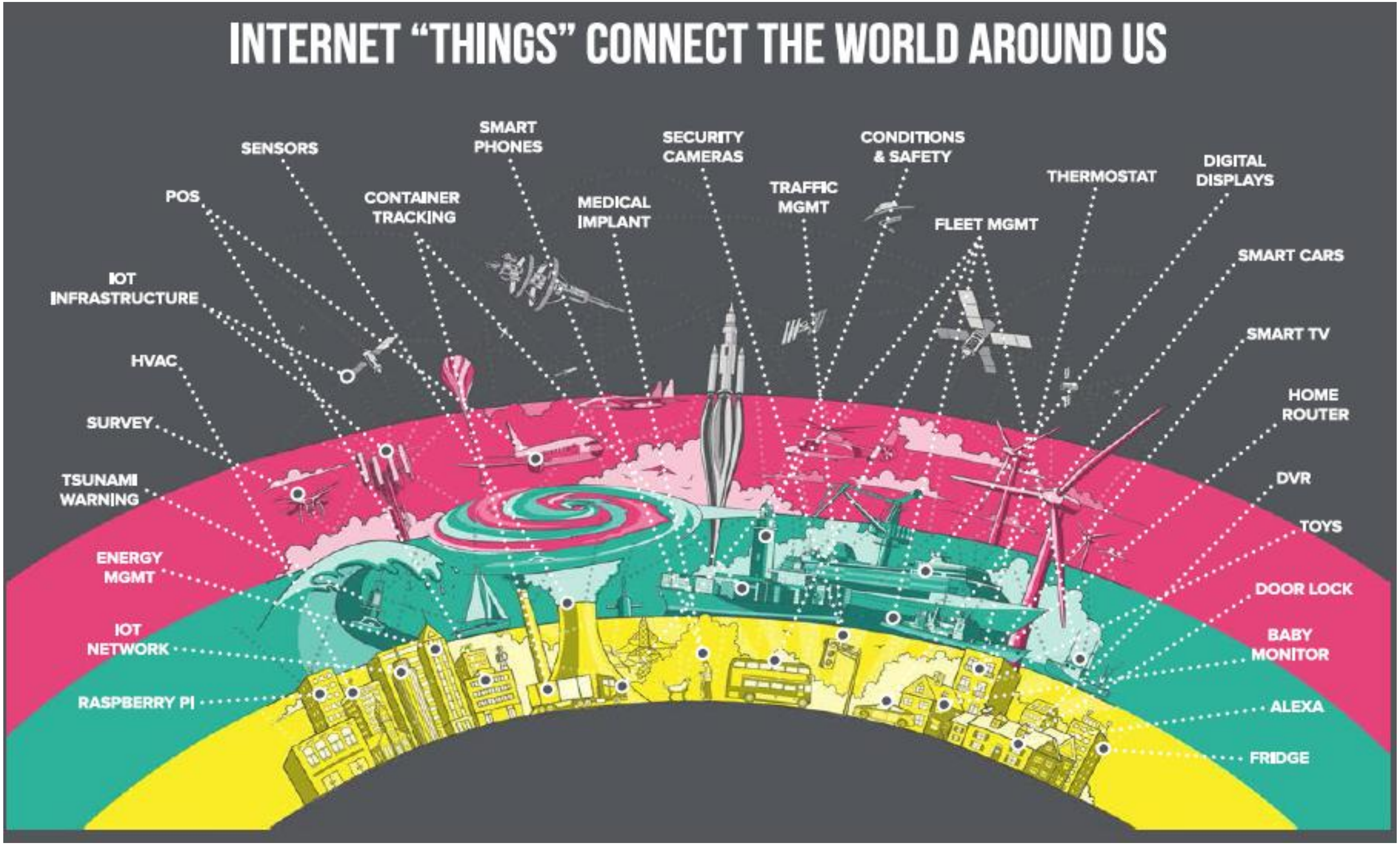


Figure source. *"The hunt for IoT: The rise of the thingbots", F5 Labs 2017 Report*

Security-related facts about IoT

Installed in Cyber-Physical systems

- Industrial systems, cars, smart grids, humans….

There are too many (and they grow very fast)

- 35.82 billion IoT devices installed worldwide by 2021
- and 75.44 billion by 2025

Technologies are not standardized

- Diversity in H/W (ARM, x86, x64,…)
- Diversity in S/W (CoAP, proprietary,…)
- Diversity in network protocols (802.15.x, 802.11.x, Ethernet, Modbus, proprietary…)

They create various connectivity paths (which are **not always obvious**)

- Local connections
- Internet connections

IoT are used as attack enablers/amplifiers against other systems

- Usually far more important

Security-related facts about Critical Infrastructures

Cyber-Physical systems installed in various sectors and supporting vital services

- **Energy** (smart grids, renewable sources etc)
- **Industry** (SCADA, production systems, control systems, …)
- **Transportation** (smart cars and smart traffic management, autonomous ships, planes, …)
- **Healthcare** (In-hospital services and systems, remote patient management, Internet of Medical Things,…)
- …

Traditional CIs

- Closed systems
- Based on proprietary systems, protocols, software
- Systems are hard to maintain, update and manage

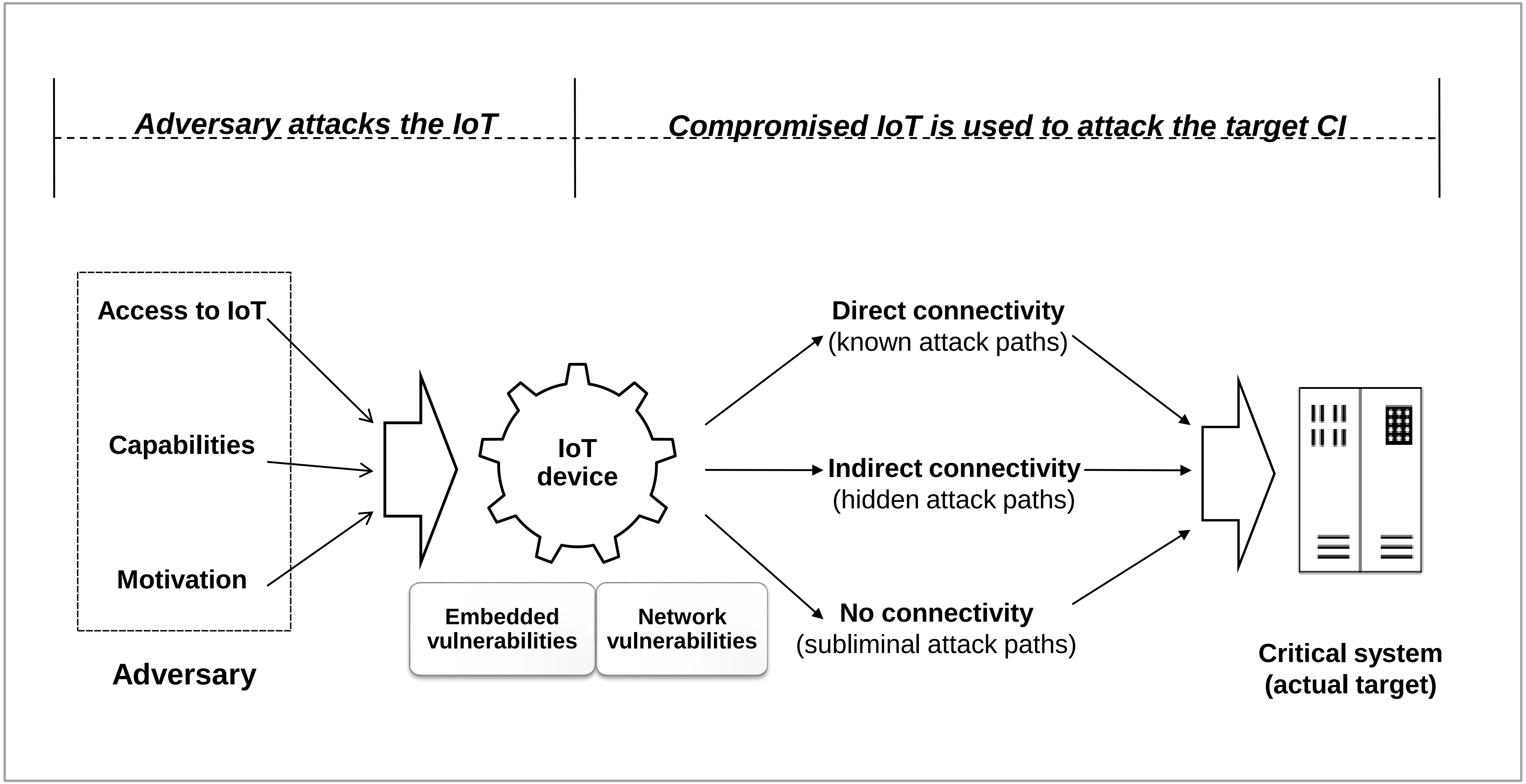
Modern CIs

- Coupled with “smart” (IoT technologies) to allow remote management, maintenance and modular design
- Interconnected systems

Security challenges

- Increased connectivity and accessibility → much higher ***exposure to remote attackers***
- Interactions among C-P systems → creation of ***novel C-P attack paths***
- Increased service inter-connectivity → increased risk of ***cascading attacks and risks***

Modeling IoT-enabled cyber attacks – A simplified approach



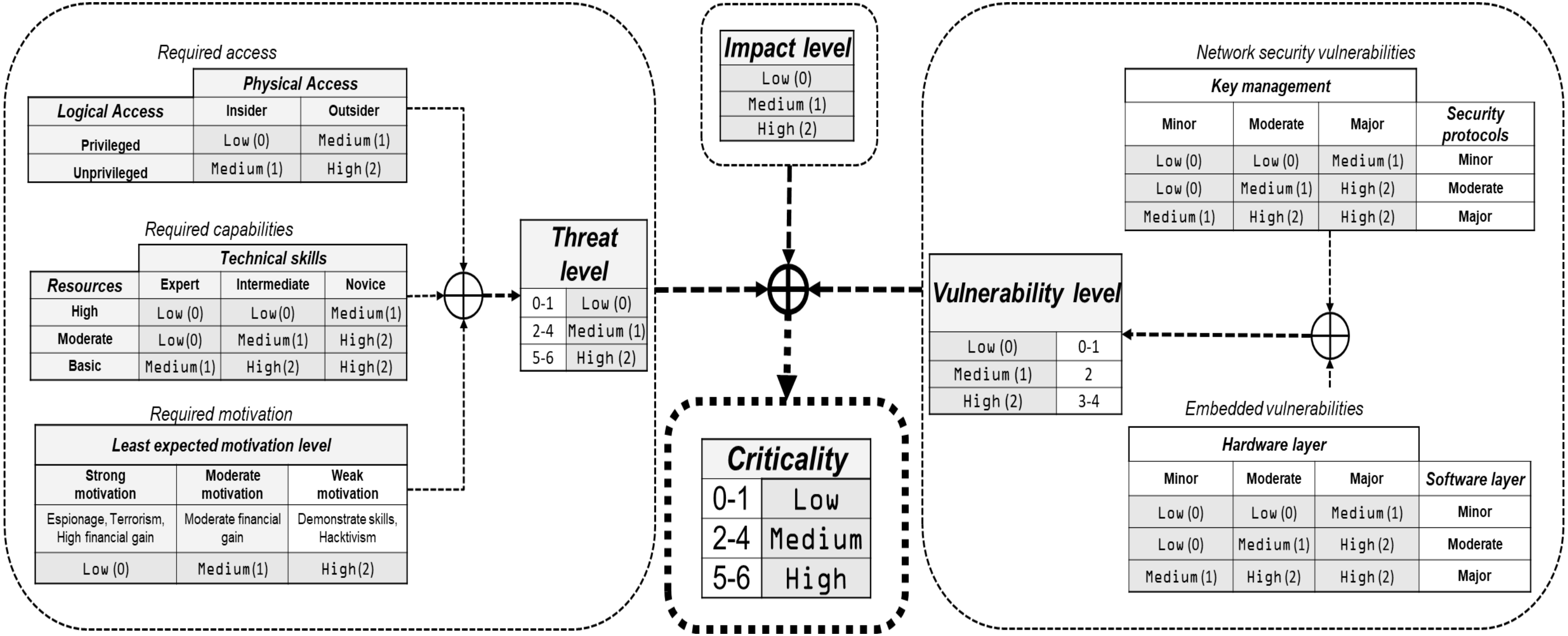
Assessing the risk of IoT-enabled Attacks: A simplified approach

- Applying a typical **Type-1** risk formula:

$$\text{Risk(Threat, Asset)} = \text{Likelihood(Threat)} \otimes \text{Vulnerability(Threat, Asset)} \otimes \text{Impact(Threat,Asset)}$$

- **Threat Likelihood:** Based on characteristics of the adversary
- **Vulnerability level:** Based on embedded and network layer vulnerabilities of the attack enablers (IoT devices)
- **Impact level:** Based on the Impact of possible targets, connected in some way with the IoT device

Assessing IoT-enabled Cyber Attacks

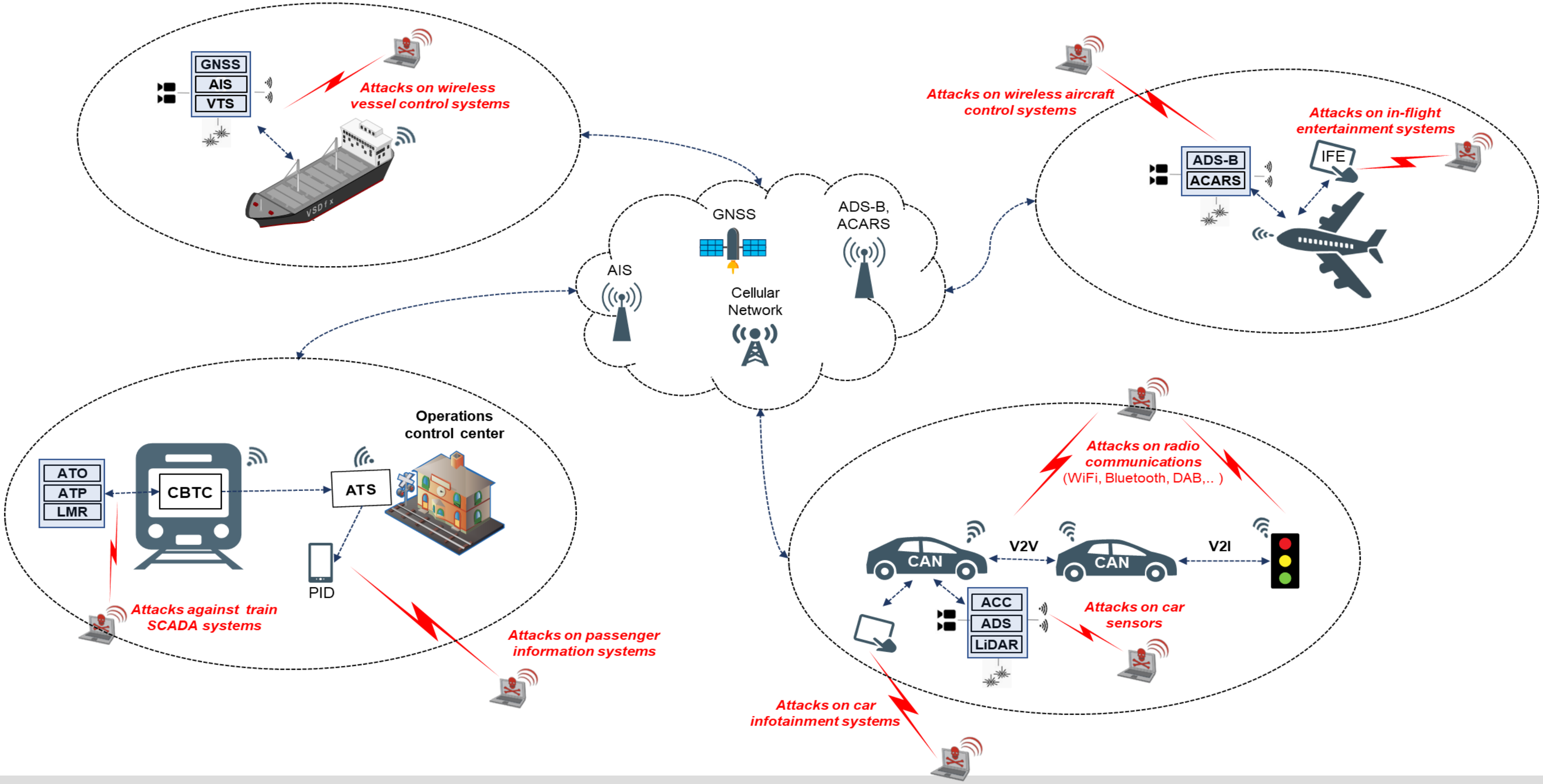


2. IoT-enabled cyber-physical attacks against Critical Infrastructures and Services

Analysis of IoT enabled attacks

- Use the risk-based methodology to assess real incidents or verified proof of concept (PoC) attacks
- We examined more than 50 recent attacks in various IoT sectors
- For each attack we describe the attack vectors and we assess their criticality level based on real/realistic data

ITS infrastructure and relative IoT-enabled attacks



Take control of a car remotely through the Internet

Attack example [1]: *Take control of cars through the Internet, by* ***abusing the car Infotainment system***

(PoC by security researchers on Cherokee Jeep, 2015)

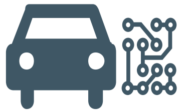
Attack vector

1. Connect to the Infotainment through an **open port** (discovered in a certain provider)
2. Remotely exploit the head unit **to install SSH and Command Line Interface to the Infotainment system**
3. Use SSH/CLI to **flash modified firmware** through the Infotainment system
4. Using the **indirect connectivity** of the IFE system (through the CAN Bus) with critical car control systems to remotely control cars.

Real damage: The manufacturer was forced to recall and patch 1.400.000 vehicles

Potential damage: harm people safety, disrupt traffic

Criticality level: **High**



Take control of traffic control lights

Attack example [2]: *Exploit* ***radio communication of traffic control systems*** *to control them*

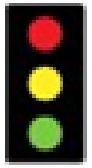
(PoC attack in real traffic control lights, 2014)

Attack vector

1. Use off-the-shelf radio equipment to communicate with traffic control systems
2. **Passively eavesdrop** communications (900 MHz and 5.8GHz)
3. Messages are **not authenticated/encrypted.** Manipulate old messages to create fake messages
4. Introduce **fake/replay messages** to control traffic control systems

Potential damage: A malicious adversary may brick traffic lights to cause traffic jams, or even cause multiple car accidents

Criticality level: **High**



Take control of plane systems via IFE

Attack example [3, 4]: *Exploit* ***In Flight Entertainment (IFE) system*** *to control of various systems (by two security researchers, while in flight, 2015, 2016)*

Attack vector

1. **Reverse engineer firmware** of an IFE system (found on the Internet)
2. Extract **hardcoded credentials** and use them to access a real IFE
3. Perform **SQL injection** attacks to control of the displays of other passengers

Potential damage: A malicious adversary may use such attacks to take control of critical systems of a plane

Criticality level: High

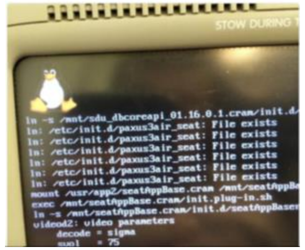
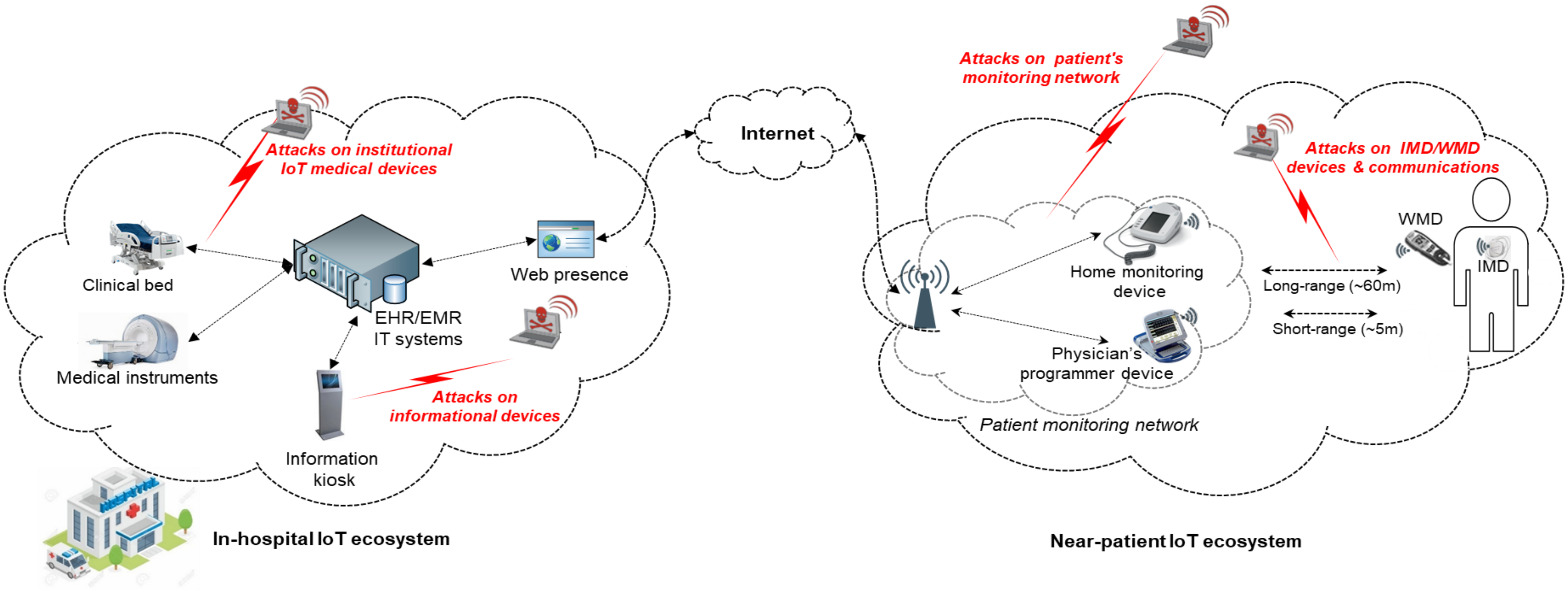


Figure source **[3]**

Healthcare infrastructure and relative IoT-enabled attacks



Manipulating implantable pacemakers

Attack example [5]: *Exploit **proprietary network protocols** to control a pacemaker (security researchers, 2017)*

Attack vector

1. **Reverse engineer proprietary network protocols** of implantable medical devices (peacemakers)
2. Use off-the-shelf equipment to bypass security controls and **remotely induce small amounts of electricity** that could potentially harm patients

Real damage: ICS-CERT issued an advisory that forced 65.000 patients to visit their doctors in order to have their devices updated

Potential damage: A malicious adversary may harm people from a distance (up to 5m)

Criticality level: High

Take control of in-hospital devices

Attack example [6]: *A real security analysis of three hospitals revealed* ***compromised in-hospital medical IoT systems*** *(security researchers, 2017)*

Attack vector

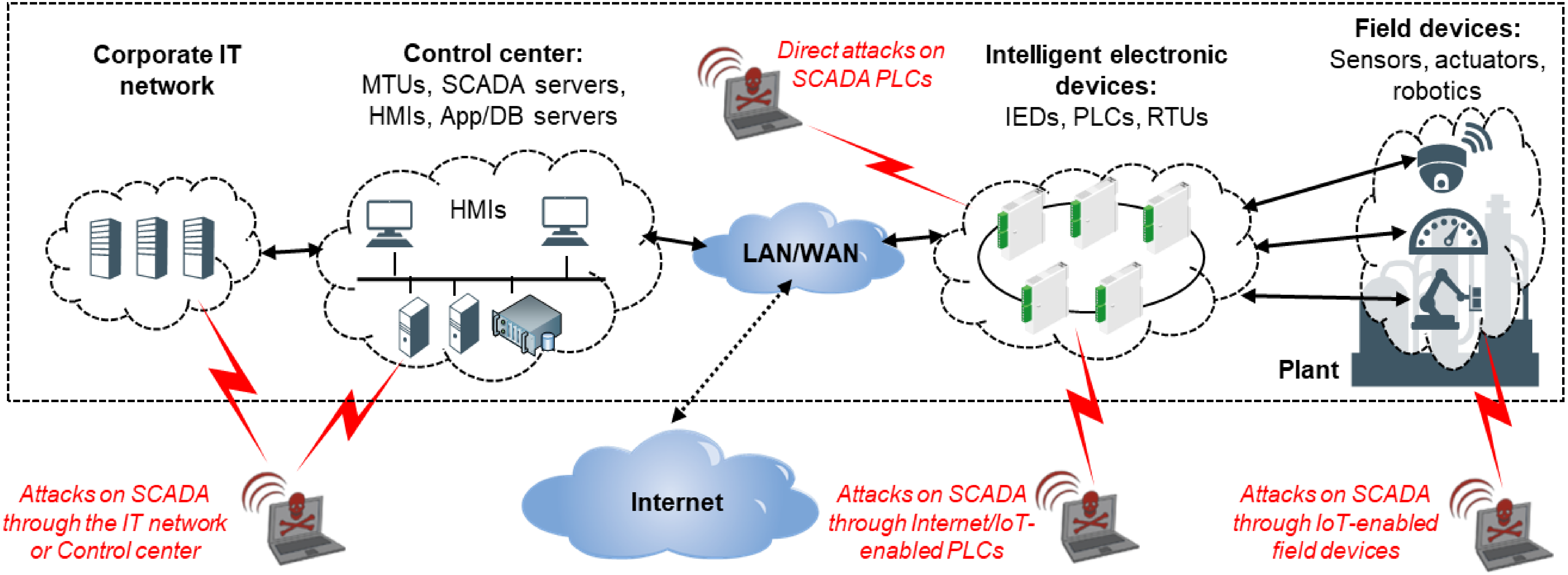
1. TrapX Research Labs in 2017 introduced **emulated IoT-enabled medical devices** inside hospitals
2. **Monitor for attacks against the emulated devices**, using special software
3. In a few days they discovered attacks against the emulated devices, that were **originating form real medical devices** within the hospital
4. Most of the malicious code found was never detected by hospital's IT stuff or the installed security systems and firewalls.

Real damage: The remediation took several weeks since the infected devices hat to be replaced

Potential (real?) damage: Use infected medical systems to gain access to medical records

Criticality level: **High**

Industrial SCADA and relative IoT-enabled attacks



Simulated water treatment plant attack

Attack example [7]: *Take control of Internet facing PLCs, by* ***creating a self-spreading cross-vendor ransomware worm (LogicLocker)***
(PoC attack by security researchers of Georgia Institute of Technology, 2017)

Attack vector

1. Locate vulnerable internet-facing PLCs **through Shodan** search engine susceptible to ransomware attack (discovered 1.500 of the model under attack)
2. Using **brute force** techniques recover the password.
3. Remotely infect PLCs with ransomware
4. **Locks the PLCs and send a ransom note** to the authorities.

Potential damage: Harm people safety, public confidence and trust.

Criticality level: High

Take control of internet connected industrial robots

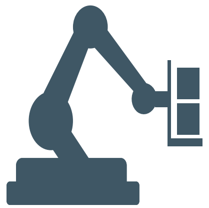
Attack example [8] : *Exploiting* ***multiple vulnerabilities such as WAN access to unfirewalled LAN ports, week authentication schemes, insecure web interfaces***

(PoC attack by security researchers of Politecnico di Milano and TRENDMICRO, 2017)

Five classes of robot-specific attacks that violates the basic operational requirements of industrial robots (accuracy, safety, integrity)

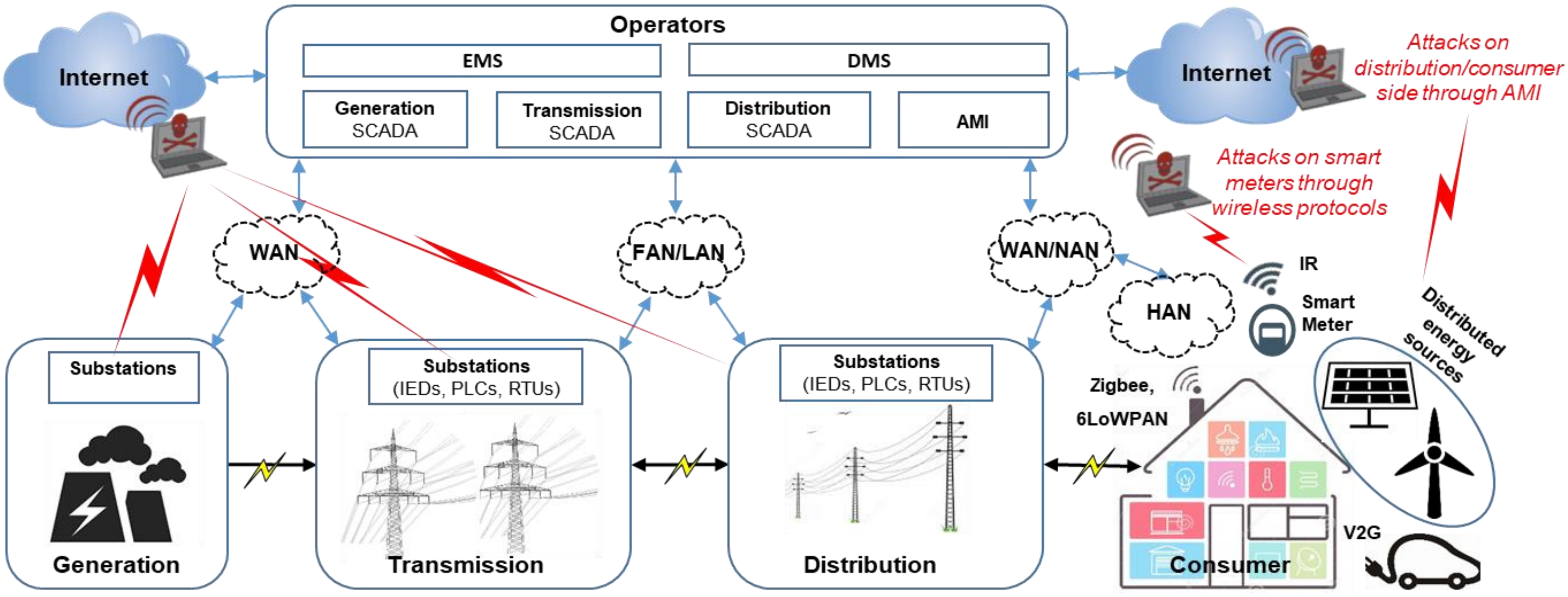
1. Control-loop parameters alteration
2. User-perceived robot state alteration
3. Actual robot state alteration
4. Calibration parameters tampering
5. Production logic tampering.

Potential damage: Harm people safety, public confidence and trust, significant economic loss.



Criticality level: **High**

Smart Grid infrastructure and relative IoT-enabled attacks



Attack Ukraine's smart Grid (part 1)

Attack example [9]: Attacks on Ukraine's smart grid transmission network.

Take control of multiple internet connected (through corporate network) circuit breakers, ***through spear-phishing campaigns*** *(2015)*

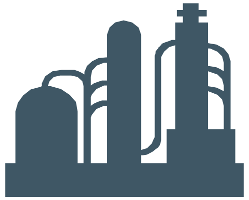
Attack vector:

1. Malware (*BlackEnergy - KillDisk*) was sent wrapped up in a word document that was attached in a phishing email impersonating a message from the Ukrainian parliament.
2. By opening the malicious word document a script run on the victims' machines, thus planting the *BlackEnergy* infection.
3. The malware compromised a VPN service that companies used to remotely access IoT-enabled equipment, and use it to gain control in multiple circuit breakers that controlled power flow in **distribution** network.

Real Damage: *230.000 people were affected*

Potential Damage: Harm public confidence, significant economic loss

Criticality level: High



Attack Ukraine’s smart Grid (part 2)

Attack example [10]: Attacks on Ukraine's smart grid distribution network (2016)

Attack vector:

1. The infection spread through spear phishing attacks.
2. The malware (CrashOverride - Win32/Industroyer) remained hidden until it was triggered.
3. The worm could be programmed to scan the victim's network, to discover potential targets, open circuits without any intervention from the attackers.
4. It included ICS protocol stacks including IEC 101, IEC 104, IEC 61850, and OPC, a wiper to delete files and processes, modules to open circuit breakers on RTUs and force them into an infinite loop thus keeping the circuit breakers open even if grid operators attempt to shut them down.

Damage: Harm people safety, public confidence and trust, significant economic loss, user discomfort.

Criticality level: High

Smart Grid (PoC attack on smart grid)

Attack example [11]: Vulnerabilities on smart meters

*Take control of multiple interconnected (through ZigBee, Cellular network) smart meters, **by exploiting embedded and network vulnerabilities** and attack the smart grid services*

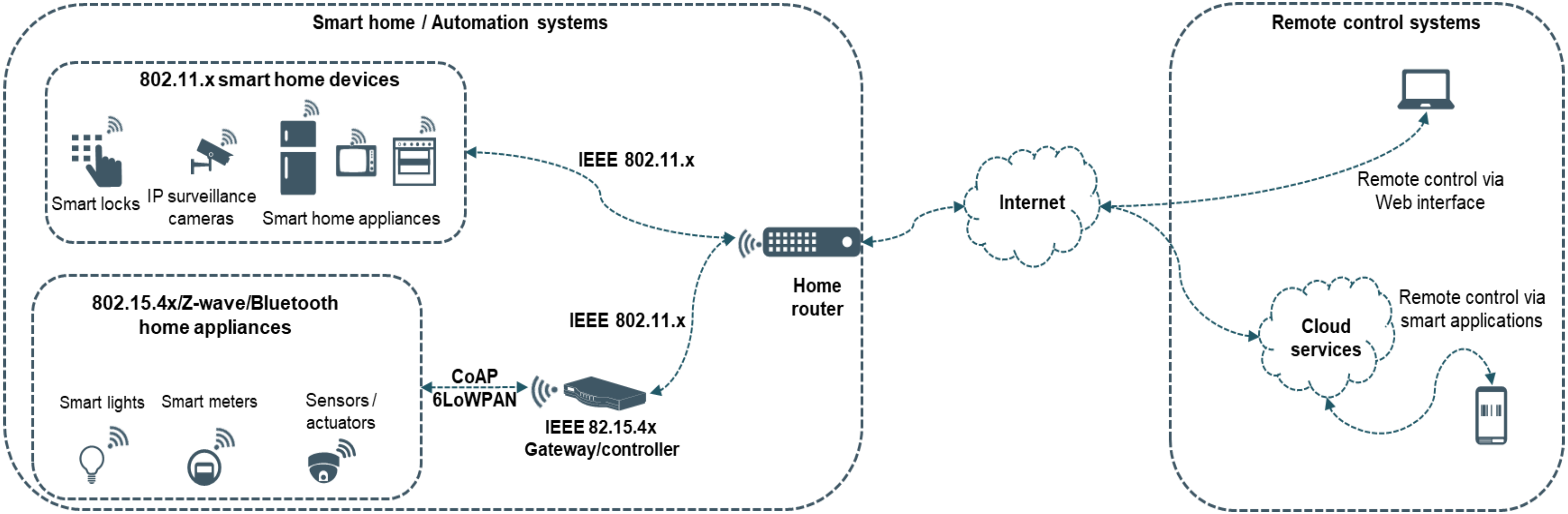
Attack vector :

1. Encryption keys derived from short (often just six-character) device names.
2. Pairing process requires no authentication, allowing an attacker to simply ask the smart meter to join the network and receive keys.
3. Hardcoded credentials, allowing administrator access with passwords as simple and guessable as the vendor's name.
4. Code simplified to work on low-power devices skipping important checks, allowing nothing more than a long communication to crash the device.

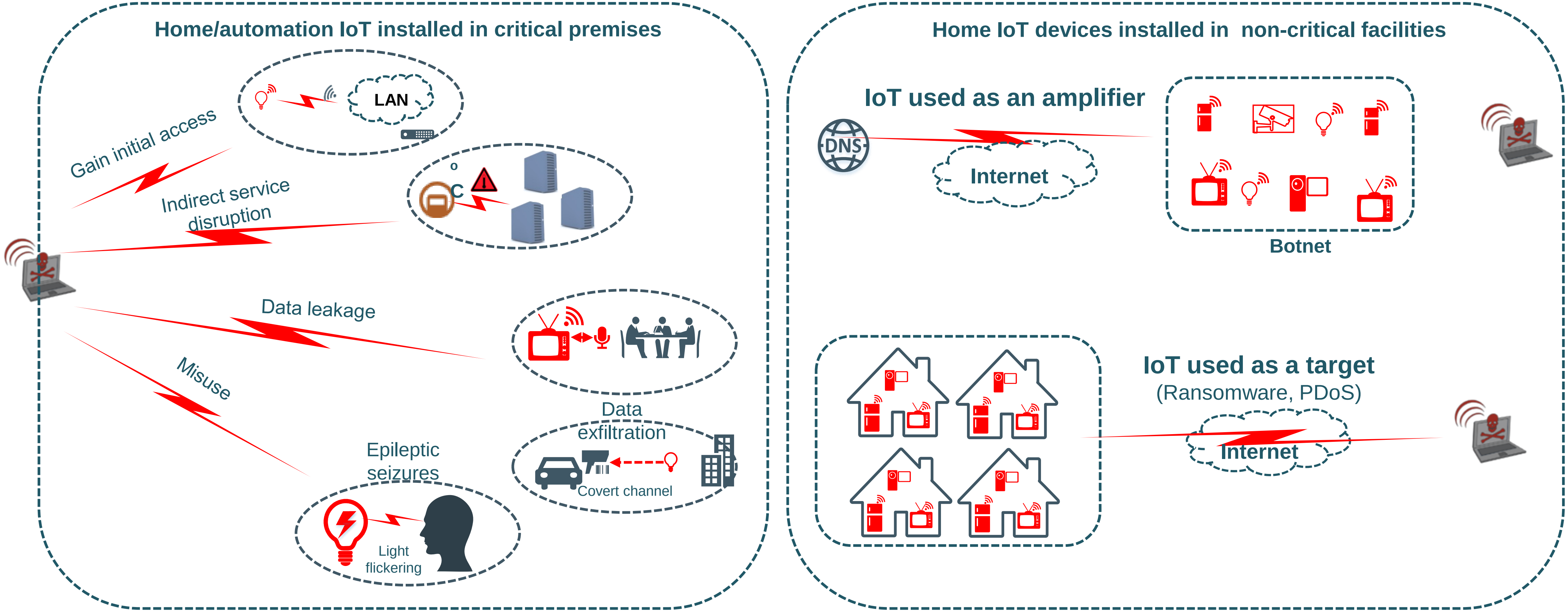
Damage: Public confidence and trust, significant economic loss, user discomfort.

Criticality level: High

Smart home infrastructure



Smart home infrastructure and relative IoT enabled attacks



Smart Lights: PoC IoT enabled attacks (IoT as a target)

Create a self-spreading worm [12,13] (PoC) :

- Researchers reversed engineered several models of **smart lighting systems** and recovered embedded sensitive information (hard-coded encryption and signing keys).
- Using off-the-shelf equipment they managed to bypass security controls and remotely control the lamps.
- Using the recovered keys the managed to create a self-propagating worm that spreads autonomously to all similar smart lighting systems. All these were possible from distances of aprox. 350 meters.
- The same group or researchers were able to create covert channels by making the smart lamps flicker in brightness levels unnoticeable to human eye. Furthermore they were able to manipulate flickering in such a way that they could cause epileptic seizures to people.

Smart home: Real IoT enabled attacks

DDoS attacks on DYN DNS services [14] (October 2016 – Real – As an amplifier):

- Thousands of unsecured IoT devices, part of a the **Mirai BotNet**, launched a coordinated DDoS attack against DNS services at a rate of 600 Gbps thus preventing customers from reaching over 1.200 domains including Amazon, Twitter, Pinterest, Reddit, GitHub, Etsy, Tumblr, Spotify, PayPal, Verizon, and Comcast for several hours.
- The infected home IoT-enabled devices had default/weak passwords and/or vulnerable OS installed.

Attacks on smart TVs [15] (January 2017 – Real – exfiltrate data):

- On March 2017 Wiki-Leaks published documents that revealed a CIA project named Weeping Angel. By placing the target TV in a *fake-off* mode they were able to record conversations in a room and then send them over the Internet to a covert server.

Mitigation controls

For the operators

- Avoid installing IoT near critical systems
- Properly segment/isolate networks (mission critical systems should always be isolated)
- Consider all attack paths (not only the obvious ones)
- Security test of IoT devices before installation
- Control physical access to IoT devices
- Control Internet access to/from IoT
- Re-examine BYOD, BYOP policies
- Favor technology diversity

For the manufacturers

- Use tamper resistant H/W
- Protect F/W update procedure
- Avoid to hardcode credentials
- Use tested APIs to develop IoT S/W
- Authenticate network communications
- Provide encryption and integrity protection of network protocols (at least optionally)
- Implement secure key management/key exchange procedures

For the regulators

- Enforce proper security controls for IoT devices
- Enforce use of security IoT in critical infrastructures

Assessing IoT-enabled Cyber Attacks: A targeted approach

- A better definition of C-P interactions
- Defining *n*-hop, C-P attack paths against critical targets
- A targeted Risk formula for IoT-enabled attack paths against critical systems
- Defining algorithms to identify and assess attack paths

3. A Method for Identifying and Assessing IoT-enabled C-P attack paths against CIS

Assessing IoT-enabled Cyber Attacks: Definitions

- **Interactions:** We define as an *Interaction* between two systems (nodes), called the *source* node *x* and the *destination* node *y* and we denote as **(*x*, *y*, type)** the directional action or 'influence' that *x* may cause to *y*, due to their *proximity* and/or *connectivity*. We define two categories of interactions: *physical* and *cyber* interactions
- **Cyber Interactions:** They include all the actions that may be triggered by the source towards the destination node, due to their cyber connectivity. In order to model cyber interactions, we make use of two characteristics: the network connectivity level and the logical access level.
- **Physical Interactions:** These include all the actions that may be triggered by *x* to *y* due to their physical proximity.
- **Attack Paths:** Let ***T*** denote the critical target system and let ***D*** denote the set of all the assets (devices) in scope. We define as an *Attack Path* against a target system T and we denote as **AP** = $(d_n \rightarrow \cdots \rightarrow d_1 \rightarrow \boldsymbol{T})$, $d\ i \in \boldsymbol{D}$ a chain of interactions, where the threat is triggered in node d_n (the entry-point system) and the actual target of the attack is the critical system ***T*** .

Table 1 – Cyber interaction types: A cyber interaction ($x \rightarrow y$) may belong to type C1–C6, based on the connectivity and the logical access of x to y.

Connectivity	Logical Access: None (no explicit access)	Logical Access: Low (user-level)	Logical Access: High (admin-level)
L2 (Local) Network	C1	C2	C3
L3 (Remote) Network	C4	C5	C6

Table 2 – Physical interactions based on the proximity between devices. The implied capabilities of the source node on the target system may involve physical tampering, manipulation of I/O interfaces or manipulation of shared-band network interfaces.

Type	Description	Interface	Examples	Common attack patterns
P1	**Physical proximity** (*x* may use a moving part and/or moving capabilities to physically reach *y*)	Remotely controlled moving parts or devices	Robotic arm, crane, wheeled device, drone	Cause destruction/obstruction.
P2	**Wireless I/O proximity** (*x* is in range with a wireless I/O interface of *y*)	Audio, Visual, Optical interfaces	Line-of-sight (LiDAR, IR), audio / video interfaces	I/O suppression/manipulation (e.g. introduce artifacts in optical sensors). Side-channel attacks (covert channels for data exfiltration).
P3	**Networks' proximity** (*x* and *y* at *different* networks that are in range)	Different, but shared-band wireless interfaces	e.g 802.11.x and 802.15.x operate at 2.4 GHz	DoS (jamming) - Packet injection attacks.

Assessing the risk of IoT-enabled Attacks: A targeted approach

- Combine typical **Type-1 + Type-4** risk formulas:

Type-1: Risk(Threat, Asset) = Likelihood(Threat) $\otimes$ Vuln(Threat, Asset) $\otimes$ Impact(Threat, Asset) (1)

Type-4: Risk(Threat, Crit.Asset) = Vuln(Crit.Asset) Impact (Threat, Crit.Asset) (2)

Proposed: Risk (Threat, ***AP***) = Likelihood(Threat, ***AP***) Vuln(Threat, ***AP***) Impact (Threat , ***T***) (3)

- Motivation:
  - Allow for **fine-grained threat/ vulnerability input** from open sources (supported by Type 1)
  - At the same time **focus on the impact of the critical target** system (supported by Type 4).

Formula reasoning

- The proposed methodology is *source driven* and *target oriented*. Our goal is to assess the risk for various threat agents that may trigger an attack at the source node of an attack path, in order to eventually affect the critical target system.
- *Asset* is replaced by an attack path ***AP*** of multiple interacting assets, where the destination of the path is the critical target system ***T***.
- **Impact is assessed based on the consequences of the *critical target T*.**
  Recall that the goal of the adversary is to harm the critical asset; the other systems in the path are used in order to extend the attack vector.
- ***Threat likelihood and vulnerability assessment consider the whole attack path AP.***
  The adversary is expected to combine any capability having on the interacting node, in order to gradually exploit all vulnerabilities within an attack path.
- The optimal adversarial strategy is to combine vulnerabilities found at the entry point system $\boldsymbol{d_n}$ with vulnerabilities found in the whole chain, to pivot (horizontally or laterally) to the ultimate target ***T.***

Assessing IoT-enabled cyber physical attack paths against critical systems –

A high level description

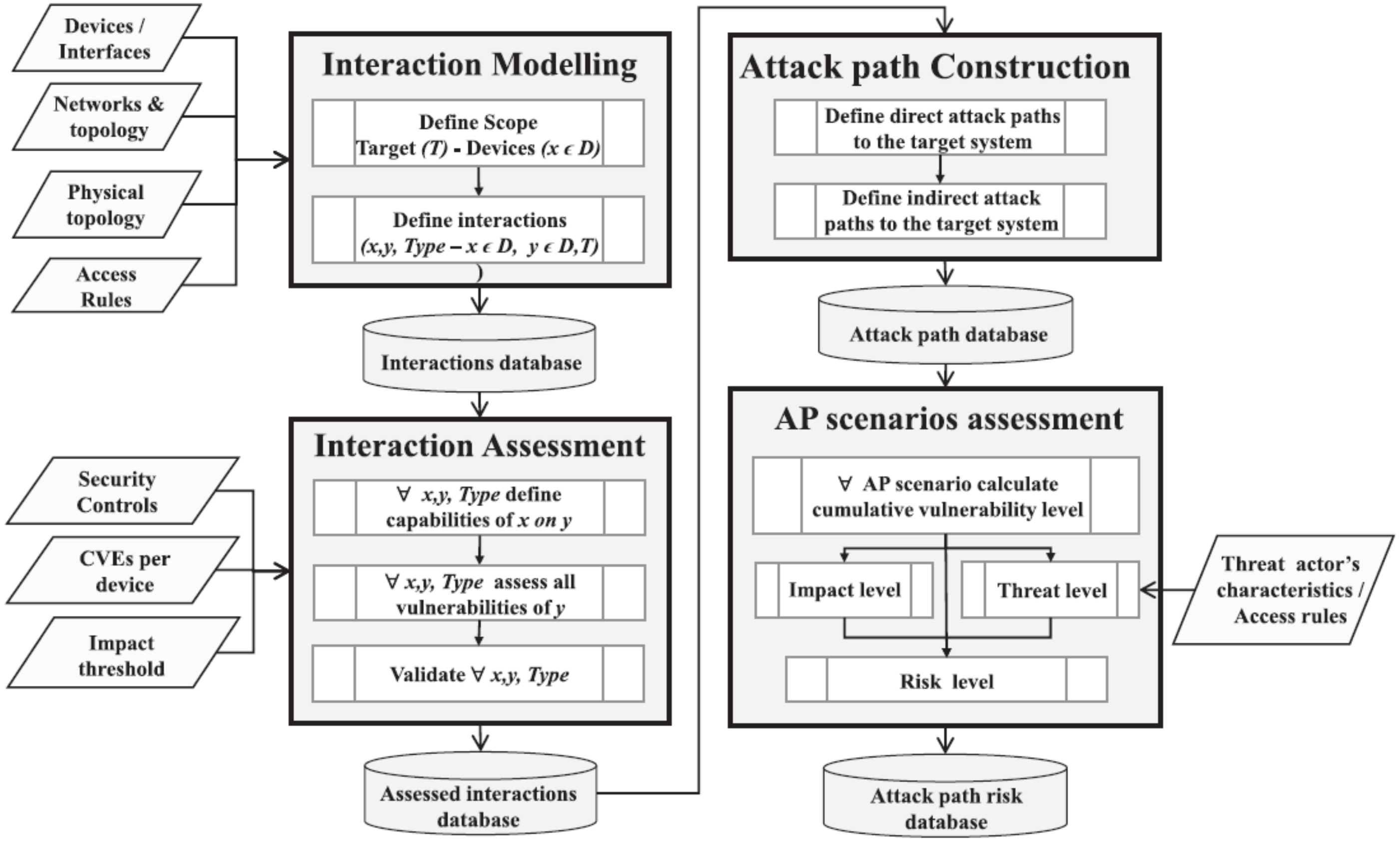
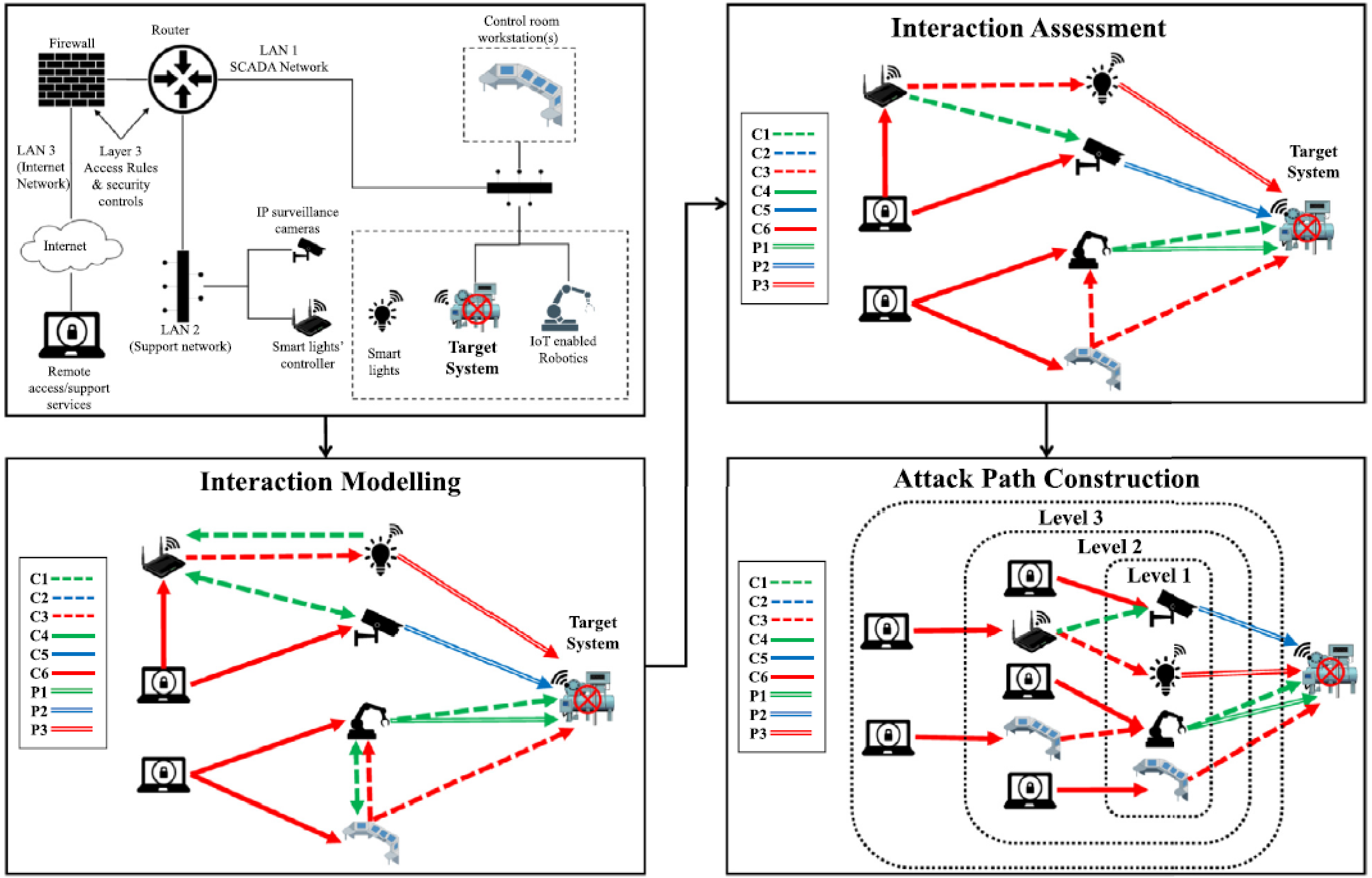


Fig. 1 – High level description of the proposed risk assessment methodology.

A graphical representation of the methodology



Phase 1 – Interaction Modeling

- First, all the direct interactions with the target system are computed to form the list L_1 (see lines 2–3 in Algorithm 1).
- Then, all the indirect interaction lists L_i , $i = 2 , \ldots , n$ are recursively computed, by exhaustively examining the potential interactions of all the source nodes in level-i interactions, but now as being destination nodes of possible interactions (lines 4–15).
- The algorithm avoids duplicating interactions already defined in previous lists, so that each interaction is defined once, in the shortest possible list. The procedure *IdentifyInteractions* is recursively called in the main algorithm.
- In the first call, since the destination of the interaction will be the target system *T*, both physical and cyber interactions will be checked. For all other calls, only the cyber interactions will be modeled.
- Since each call on IdentifyInteractions has computational cost proportional to $|D|$, the computational cost of Algorithm 1 will be proportional to $O(|D|^n)$ where n is the number of interaction lists.

Phase 1 – Interaction Modeling

Algorithm 1: Identify and model all potential interactions in $\{\mathcal{D}, \mathcal{T}\}$.

Input : $\mathcal{T}$=Target system. $\mathcal{D}$=The set of devices in scope and their corresponding interfaces. PT=Physical Topology. NT=Network Topology. AR=Access Rules.

Output: $InteractionLists[]$ = A set of lists containing all direct interactions with the target system ($\equiv \mathbb{L}_1$) as well as the devices themselves ($\equiv \mathbb{L}_i, i = 2, 3...n$)

Algorithm `ModelInteractions()`
- $i \leftarrow 1$ `// Compute` $InteractionLists[1] (\equiv \mathbb{L}_1)$
- $InteractionLists[i] \leftarrow$ **IdentifyInteractions**$(\mathcal{D}, \mathcal{T}, PT, NT, AR)$
- **while** (*TRUE*) **do**
  - $InteractionLists[i+1] \leftarrow \emptyset$
  - `// Check all devices in Level-i as 'target' of any other device, in order to construct Level-(i+1) interactions`
  - **while** ($(x, y, \text{type}) \leftarrow hasNext(InteractionLists[i])$) **do**
    - $L_x \leftarrow$ **IdentifyInteractions**$(\mathcal{D}, x, PT, NT, AR)$ `//` x `is a Level-i device`
    - $L_x \leftarrow L_x - (L_x \cap InteractionLists[i])$ `// Don't duplicate in Level-(i+1), interactions already identified in Level-i. Possible if graph has loops`
    - $InteractionLists[i+1] \leftarrow InteractionLists[i+1] + L_x$
  - **end**
  - **if** ($InteractionLists[i+1] = \emptyset$) **then**
    - `break // If no Level-(i+1) interactions exist, then exit`
  - **end**
  - $i \leftarrow i + 1$
- **end**

return ($InteractionLists[]$) `/* Interaction lists for all existing levels (`$\mathbb{L}_1, \mathbb{L}_2, ..$`) */`

Phase 2 – Interaction Assessment

- The goal of this phase is to ***filter out from further processing those interactions that are not 'mature enough' to be exploited*** by assessing their vulnerability level.
- Compute the cumulative vulnerability level (CVV), for validated interactions only.

Question 1: How to assessing whether an interaction (x, y, type) is valid or not?

- Based on ***the level of the influence that x has on y*** due to their interaction.

Question 2: How to assess the level of influence of x to y?

- Combine ***the implied capabilities of x to y due to their interaction type***….
- …along with any ***additional capabilities that x may acquire on y, by exploiting vulnerabilities*** at the destination node y of the interaction (vulnerability chaining).

Phase 2 – Interaction Assessment

Algorithm 2: Assess Identified Interactions (*AssessInteractions*)

Input : $InteractionLists[\,]$ ($\equiv \mathbb{L}_i, i = 1, 2, ...n$) : A set of lists containing all interactions produced by Algorithm 1.
$\{CVE_d\}$: Sets of CVE/CVSS (environmental) vectors $\forall\ d \in \mathcal{D}$.

Output: $AssessedLists[\,]$ ($\equiv \mathbb{AL}_i, i = 2, 3...n$) : A set of lists containing all assessed interactions.

AssessInteractions($InteractionLists[\,], \{CVE_d\}$)
for *InteractionLists[i]*, $i : 1 \ldots n$ **do**
  $AssessedLists[i] \leftarrow \emptyset$; $CVV \leftarrow \emptyset$
  while ($(x, y, \text{type}) \leftarrow hasNext(InteractionLists[i])$) **do**
    Define $IntCVSS_{base}(x, y, \text{type})$ `/* Based on Tables 3,5 */`
    $IntCVSS_{env}(x, y, \text{type}) \leftarrow$ **ApplyEnv**$(IntCVSS_{base}(x, y, \text{type}))$
      `/* As defined in Tables 4,6 */`
    if type $\in [\text{C1}, \cdots \text{C6}]$ `/* Chaining cyber interactions */`
    then
      for $CVE \in \{CVE_y\}$ **do**
        $SingleCVSS_y \leftarrow$ **SingleCVE**(CVE) `// Based on Eq.(4)`
        $ChainedCVSS_y \leftarrow$ **ChainCVE**(CVE) `// Based on Eq.(5)`
        $ValidCVSS_y \leftarrow$ **ValidCVE**$(SingleCVSS_y,\ ChainedCVSS_y)$
          `// Based on Eq.(6)`
      end
    end
    $CVV \leftarrow$ **CalcCVV**$(ValidCVSS_y,\ IntCVSS_{env})$ `/* Calculate interaction's` CVV `as described on Eq.(7) */`
    $add(AssessedLists[i], (x, y, \text{type}, CVV))$
  end
end
return $AssessedLists[i]$, $i = 1, \ldots, n$

Phase 2 – Interaction Assessment (Implied capabilities of cyber interactions)

Table 3 – Defining the implied capabilities for each of cyber interaction type as a CVSS vector.

		Exploitability Metrics					Impact Metrics		
	Type	AV	(M)AC	PR	UI	S*	(M)C	(M)I	(M)A
$IntCVSS_{base}$	C1	A	H	N	N	U	N	N	N
	C2	A	H	L	N	U	L	L	L
	C3	A	H	H	N	U	H	H	H
	C4	N	H	N	N	U	N	N	N
	C5	N	H	L	N	U	L	L	L
	C6	N	H	H	N	U	H	H	H
$IntCVSS_{env}$		(M): These metrics can be environmentally modified (See Table 4) *Scope is unchanged (U), for level 1 interactions							

Table 4 – Proposed network environmental modifiers for $IntCVSS_{env}$ vector according to the corresponding security control level.

Network Security Controls	(M)AC	Impact Modifiers		
		M(C)	M(I)	M(A)
Not defined/Weak	H → L	No effect	No effect	No effect
Moderate	H	No effect	No effect	No effect
Strong	H	H → L L → N	H → L L → N	H → L L → N

Phase 2 – Interaction Assessment (Implied capabilities of physical interactions)

Table 5 – Defining the implied capabilities for physical interactions as a CVSS-like vector.

	Type	Exploitability Metrics					Impact Metrics		
		AV	(M)AC	PR	(M)UI	S	(M)C	(M)I	(M)A
$IntCVSS_{base}$	P1	P	H	N	N	U	N	L	L
	P2	A	H	N	N	U	L	L	L
	P3	A	H	N	N	U	N	L	L
$IntCVSS_{env}$		(M): Can be modified, based on physical environment (See Table 6.)							

Table 6 – Proposed physical environmental modifiers for $IntCVSS_{base}$ vector according to the corresponding security controls for each impact metric.

Physical Security Controls	(M) AC	Impact Modifiers		
		(M)C	M(I)	(M)A
Not defined/Weak	H → L	No effect	No effect	No effect
Moderate	H	No effect	No effect	No effect
Strong	H	H → L L → N	H → L L → N	H → L L → N

Phase 2 – Interaction Assessment
(Vulnerability chaining on node *y* for each interaction)

$$\forall\, \mathtt{CVE\ of}\ d \in \mathcal{D},\ \mathtt{if\ AV{:}A/N\ \ then}\ \mathtt{CVE} \in \mathit{SingleCVSS} \qquad (4)$$

We consider all single CVSS vectors with AV:A or N.

$$\mathit{ChainedCVSS} = [AV : [N|A],\ max(AC),\ min(PR),\ max(UI),\ max(S),\ max(C, I, A)] \qquad (5)$$

Vulnerability chaining is based on the paradigm of FIRST.org (2019) which demonstrates serial exploitation of vulnerabilities for privilege escalation.

In particular, we consider the cases where the exploitation of network vulnerabilities on *y* (AV:A or AV:N) that result in basic user access or an equivalent impact of C:L/I:L/A:L is combined with high-impact vulnerabilities (AV:L) to produce a chained vulnerability CVSS vector as described in Eq.(5)

$$\text{If}\ \ IntCVSS_{env}[\text{Exploitability}] \geq CVSS[\text{Exploitability}]\ \ \text{then}\ \ CVSS \in ValidCVSS \qquad (6)$$

Each vulnerability is examined to check if it is exploitable, based on Eq.(6).

Phase 2 – Interaction Assessment
(Assessing the vulnerability of an interaction: CVV(*x*,*y*,type))

$$CVV((x,y,type)) \models V \in (ValidCVSS_y, IntCVSS_{env})\ \texttt{s.t.:}$$

$$\begin{cases} \texttt{V has max(Impact,Exploitability)} & \text{if } y = \mathcal{T} \\ (C,I,A) \geq L\ \&\ \texttt{V has max(Expl., Impact)} & \text{if } y \neq \mathcal{T} \end{cases} \quad (7)$$

Choose from all the valid CVVS vectors for the interaction (*x*,*y*,type) the one that satisfies Eq.(7).

Table 7 – Summary of all vectors utilized in interaction assessment .	
$IntCVSS_{base}$	A CVSS-like capability vector assigned on the interaction based on the interaction's type, using Table 3 (for cyber) or Table 5 (for physical interactions).
$IntCVSS_{env}$	The modified $IntCVSS_{base}$ vector based on environmental information for each particular interaction (e.g. see Tables 4 and).
$\{SingleCVSS\}$	A list of all the single CVSS vectors corresponding to vulnerabilities identified in y satisfying Eq. (6).
$\{ChainedCVSS\}$	A list of all the CVSS vectors of the chained vulnerabilities of y, computed based on Eq. (5) and satisfying Eq. (6).
$CVV((x,y,type))$	The Cumulative Vulnerability Vector of an interaction as defined on Eq. (7).

Phase 3 – Attack Path Construction

- In this phase all possible attack paths against the target system ***T*** are constructed, by exhaustively combining all the assessed interactions, produced in the previous phase.
- Attack path construction is described in Algorithm 3. First, all the assessed level-1 interactions (i.e., direct interactions with the target system ***T***) are defined by default as one-hop attack paths ($\mathbf{A}\boldsymbol{P}_{\mathbf{1}}$).
- Then all the level- i attack paths $\mathbf{A}\boldsymbol{P}_{\boldsymbol{i}}$, $i > 1$, are computed recursively using AP_{i-1} and all the assessed interaction lists up to level- i (AL_1 , . . . , AL_i), by exhaustively examining if the destination node of a level- i interaction is the initial (source) node in each level- (i−1) attack path.
- The final output is a list of lists ***AttackPaths*[*i*][*j*]** , containing all the valid chains of interactions of depth i towards the target system ***T***.
- In the case where interactions have null CVV value (computed by Algorithm 2), they are considered as invalid and are excluded from any phase of the attack path construction.
- The computational cost of Algorithm 3 will be proportional to the product of the size of all the assessed lists, i.e., $O(|AL_1| \cdots |AL_n|)$.

Phase 3 – Attack Path Construction

Algorithm 3: Attack Path Construction Algorithm

Input : $AssessedLists[i] \equiv \mathbb{AL}_1, \ldots \mathbb{AL}_n$. A set of lists containing all the *assessed* interactions between devices themselves $\in \mathcal{D}$ (Level-2,...) and against the target system $\mathcal{T}$ (Level-1)

Output: $AttackPaths[i][] \equiv \mathbb{AP}_1, \mathbb{AP}_2, \ldots \mathbb{AP}_n$. A list of lists containing chains of interactions from an initial node $\in \mathcal{D}$ against $\mathcal{T}$. $\mathbb{AP}_i$ will contain the attack paths of depth i.

Algorithm `ConstructAttackPaths()`

- **for** $(i \leftarrow 1;\ i = n;\ i \leftarrow i+1)$ `// Initialize all attack path lists.` n`:# of assessed lists`
- **do**
  - $AttackPaths[i][] \leftarrow \emptyset$
- **end**
- `// Define` $\mathbb{AP}_1$ `first. By default, all interactions` $\in \mathbb{AL}_1$ `are level-1 Attack Paths.`
- $i \leftarrow 1,\ \ j \leftarrow 1$
- **while** $(\ (x, y, Type, CVV) \leftarrow hasNext(AssessedLists[i])$ ***and*** $CVV \neq \emptyset\ )$ **do**
  - $add(AttackPaths[i][j],\ [(x, y, \texttt{Type}, \texttt{CVV})])$
  - $j \leftarrow j+1$
- **end**
- `// Recursively compute` $\mathbb{AP}_i, i \in 2, \ldots, n$ `using` $\mathbb{AP}_{i-1}$ `and` $\mathbb{AL}_i$`.`
- $i \leftarrow i+1$
- **while** $(\ (x, y, Type, CVV) \leftarrow hasNext(AssessedLists[i])$ ***and*** $CVV \neq \emptyset\ )$ **do**
  - $j \leftarrow 1,\ \ k \leftarrow 1$
  - **while** $(\ AttackPaths[i-1][j] \leftarrow hasNext(AttackPaths[i-1])\ )$ **do**
    - **if** $(\textbf{\textit{isSource}}(y,\ AttackPaths[i-1][j])\ )$ **then**
      - $add(AttackPaths[i][k],\ \textbf{append}((x, y, \texttt{Type}, \texttt{CVV}), AttackPaths[i-1][j]))$
      - $k \leftarrow k+1$
    - **end**
    - $j \leftarrow j+1$
  - **end**
  - $i \leftarrow i+1$
- **end**
- **return** $(AttackPaths[i][])$ `/* Attack paths` $\mathbb{AP}_1, \mathbb{AP}_2, ..$ `*/`

Phase 4 – Attack Path Risk Assessment

- Use Eq.(3) for assessing the risk of Attack Paths.

$$\text{Risk (Threat, } \boldsymbol{AP}\text{) = Likelihood(Threat, } \boldsymbol{AP}\text{) Vuln(Threat, } \boldsymbol{AP}\text{) Impact (Threat , } \boldsymbol{T}\text{)} \quad (3)$$

$\mathrm{Vuln}(\mathrm{Threat}, \mathcal{AP})$*: calculating the vulnerability level*

$$CVV(\mathcal{AP}, \mathrm{AV}) = [\mathrm{AV} : [\mathrm{N}|\mathrm{A}],\ \max(\mathrm{AC}),\ \max(\mathrm{PR}),\ \max(\mathrm{UI}),\ \max(\mathrm{S}),\ \mathrm{Level}_1(\mathrm{C}, \mathrm{I}, \mathrm{A})] \quad (8)$$

$\mathrm{Impact}(\mathrm{Threat}, \mathcal{T})$*: calculating the impact level*

*Based one the impact of the actual target **T***

$\mathrm{Likelihood}(\mathrm{Threat}, \mathcal{AP})$*: calculating the threat level*

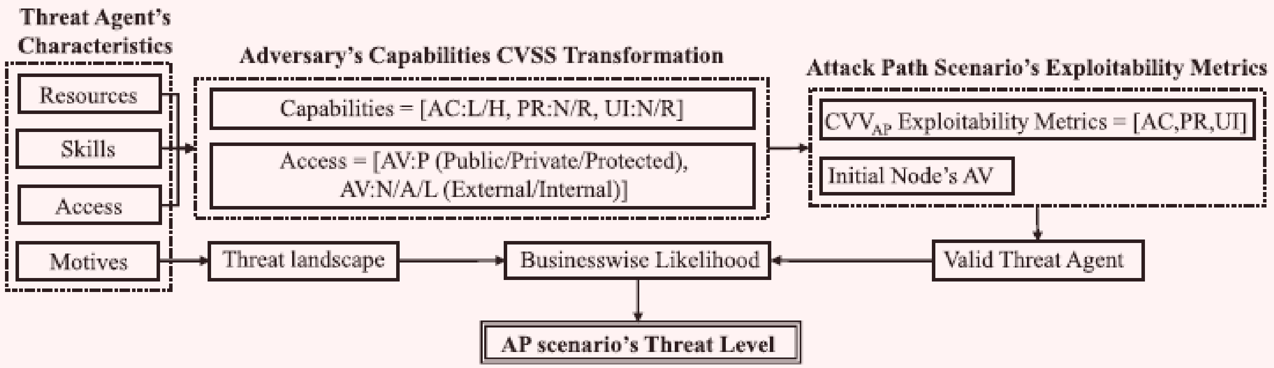


Fig. 3 – Threat level (likelihood) calculation methodology.

Phase 4 – Attack Path Risk Assessment

Table 8 – Risk calculation matrix for assessing *Risk*(*Threat*, $\mathcal{AP}$) by combining Vuln(*Threat*, AP), Likelihood(*Threat*, $\mathcal{AP}$) and *Impact*(*Threat*, $\mathcal{T}$), as defined in Eq. (3).

Risk Level

Vulnerability Level	Impact Level: Very Low					Low					Moderate					High					Very High				
Threat Level	VL	L	M	H	VH	VL	L	M	H	VH	VL	L	M	H	VH	VL	L	M	H	VH	VL	L	M	H	VH
Low	VL	VL	L	L	M	VL	L	L	M	M	L	L	M	M	M	L	M	M	M	M	M	M	M	M	H
Medium	VL	L	L	M	M	L	L	M	M	M	L	M	M	M	M	M	M	M	M	H	M	M	M	H	H
High	L	L	M	M	M	L	M	M	M	M	M	M	M	M	H	M	M	M	H	H	M	M	H	H	VH
Critical	L	M	M	M	M	M	M	M	M	H	M	M	M	H	H	M	M	H	H	VH	M	H	H	VH	VH

Risk Level: Very Low= VL, Low = L, Moderate =M, High = H, Very High = VH

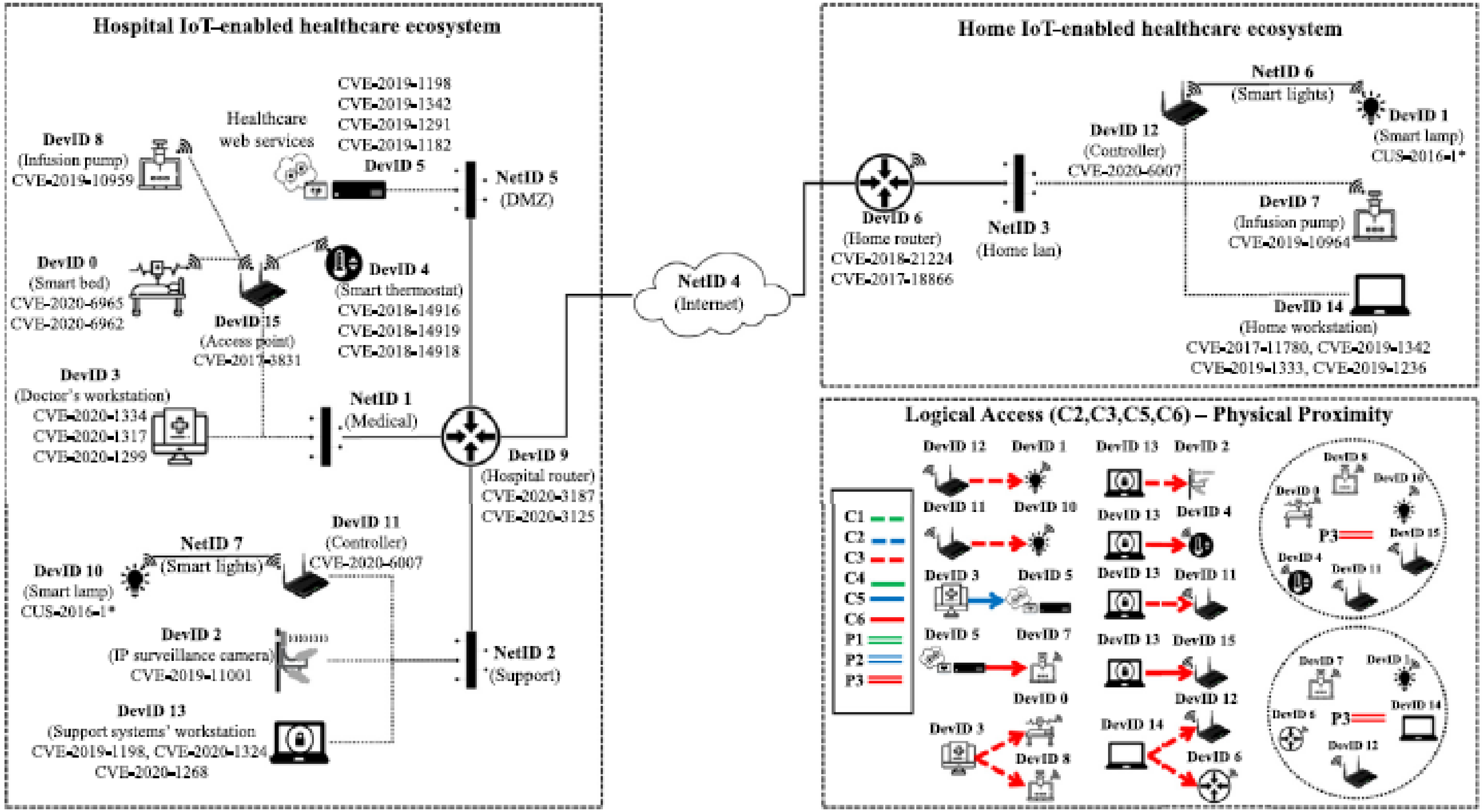
Validation of the methodology – Implementation details

- A proof-of-concept implementation was created with in python3, utilizing several libraries.
- Pandas dataframes were used to structure and analyze the required input and output data of the application.
- The AST library was used in order to split complex input data from.csv files, so they can be inserted to lists and dataframes.
- For the vulnerabilities, the CVSS/CVSSlib library was used to calculate the base score (the exploitability and impact sub scores) of the interaction CVSS vectors and the newly produced CVSS vectors.
- The CVEs were collected from the NIST database and were pulled from the json files, based on their CPE identifier.

Validation of the methodology – Test scenario

- A realistic scenario from the healthcare sector based on CVEs from real devices as critical systems and services:
  - ***On-line remote health-care services*** (Carescape B450 by ’GE healthcare’) and
  - ***Near-patient infusion pumps:*** in smart home (by ’BD Alaris’) and also in the hospital (by ’Medtronic’)
- We included various low-importance IoT devices in both environments such as ***smart lamps, thermostats and IP surveillance cameras.***
- Traditional ICT systems such as ***PCs, network routers and access points.***
- We defined logical access rules among the devices (e.g. to allow a doctor to monitor and reprogram infusion pumps via e-health services).
- For each device several well-known CVEs, or in some cases custom CVEs based on previous research were assigned.

Validation of the methodology – Test scenario



Validation of the methodology – Targeted Adversaries

Table 9 – Adversarial model for healthcare ecosystem (PoC).					
Adversaries	Capabilities	Physical/Network Access Level	Motives	Resources	Likelihood
Healthcare Rights Activist	AV:N/AC:L/PR:N/UI:N	External	1	Limited	Low
Disgruntled Healthcare Worker	AV:N,A,L/AC:L/PR:N,L/UI:N,R	Internal (Hospital)	1,2	Limited	Low
Disgruntled Healthcare Systems' Administrator	AV:N,A,L,P/AC:H/PR:N,L,H/UI:N,R	Internal/Protected (Hospital)	1,2	Moderate	Low
Business Competitor	AV:N/AC:L/PR:N/UI:N,R	External(Internet)	1	Significant	Moderate
Cyber Criminals	AV:N/AC:L,H/PR:N/UI:N,R	External (Internet)	3,4,5	High	Very High/Low
Cyber Terrorist	AV:N,A,L,P/AC:L,H/PR:N/UI:N,R	External/Internal (Hospital/Home)	1,2,4	High	Moderate/Low
Nation State	AV:N,A,L,P/AC:L,H/PR:N/UI:N,R	External/Internal (Hospital/Home)	1,2,4,5	Very High	Low

Motivation: 1=Harm Reputation, 2=Damage/Disable equipment, 3=Financial Gain, 4=Harm Patient(s), 5=Steal Patients' Data (*)Likelihood: Hospital/Home

Validation of the methodology – Results

Table 10 – Interaction modelling calculation time (per target device/total/average).

Target Device	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	sum	averg
Time (sec)	1,71	1,46	0,80	1,02	1,04	1,14	1,34	1,40	1,11	0,70	1,19	0,85	1,39	0,84	1,39	1,01	1840	1,15
Levels	3	6	4	3	3	4	6	5	3	3	4	4	6	4	6	3	N/A	4,19
Interactions	113	142	109	108	76	118	97	75	113	107	124	112	137	109	140	99	1773	12,006

Table 11 – Interactions, attack paths and attack path scenarios per interaction level for all three targets.

	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
Interactions	23 (9 Phy)	87	87	50	1	0
Assessed Interactions	19 (9 Phy)	65	47	50	1	0
Attack Paths (Cyber)	10	47	154	454	688	478
Attack Paths (Cyber-Physical)	8	24	68	171	1	0
AP Scenarios (Cyber)	46	162	514	1555	2283	1603
AP Scenarios (Cyber-Physical)	16	66	246	682	6	0

Validation of the methodology – Results

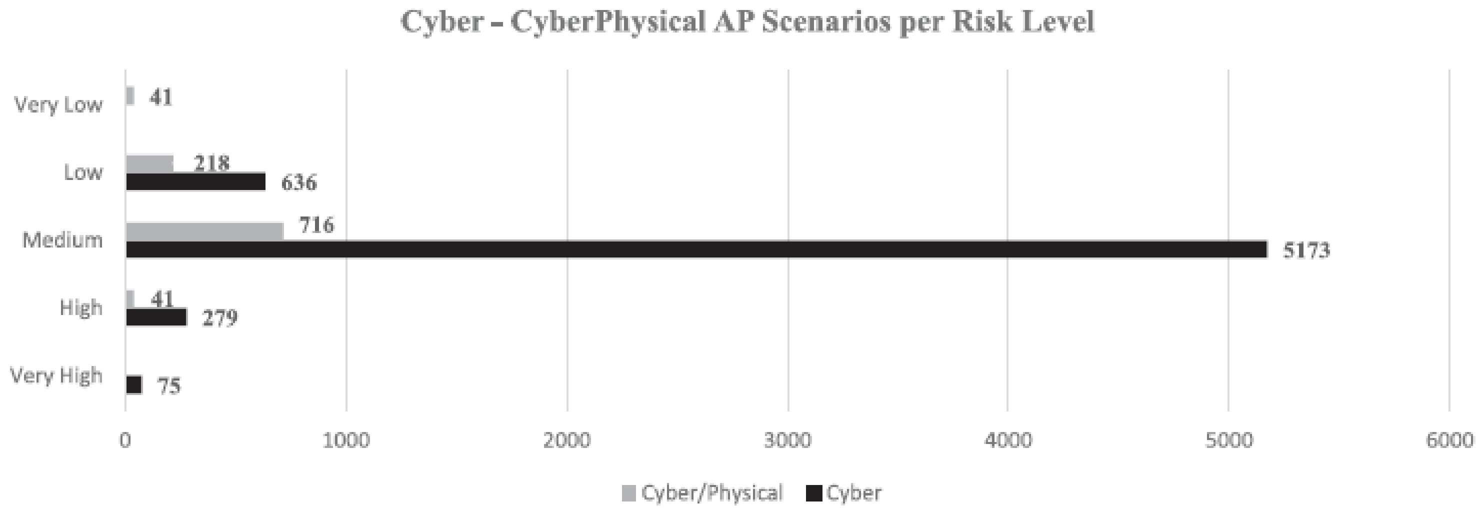


Fig. 5 – Cyber and cyber-physical attack paths scenarios per risk level.

Table 12 – Multitude of AP scenarios per node for targetIDs 5, 7 and 8.

TargetID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AP Scenarios	4857	32	4574	4927	709	2458	122	0	2002	2568	288	3199	101	4742	92	2138
As point-of-entry	315	11	762	1016	0	9	24	0	465	2568	230	423	11	562	24	759

Validation of the methodology – Results

Fig. 7 – High impact, IoT-enabled, stealthy cyber/cyber-physical AP scenarios paradigms from our test scenario.

Validation of the methodology – Risk Mitigation

- We simulated a typical patch scenario which an organization would most likely implement in order to mitigate the risks.
- First step in a typical threat remediation process: address the vulnerabilities found at the critical devices (targets).
- Next step: patch the ICT equipment such as servers, workstations and crucial network equipment.
- Final step: addressing the vulnerabilities found on IoT devices.

Validation of the methodology – Risk Mitigation

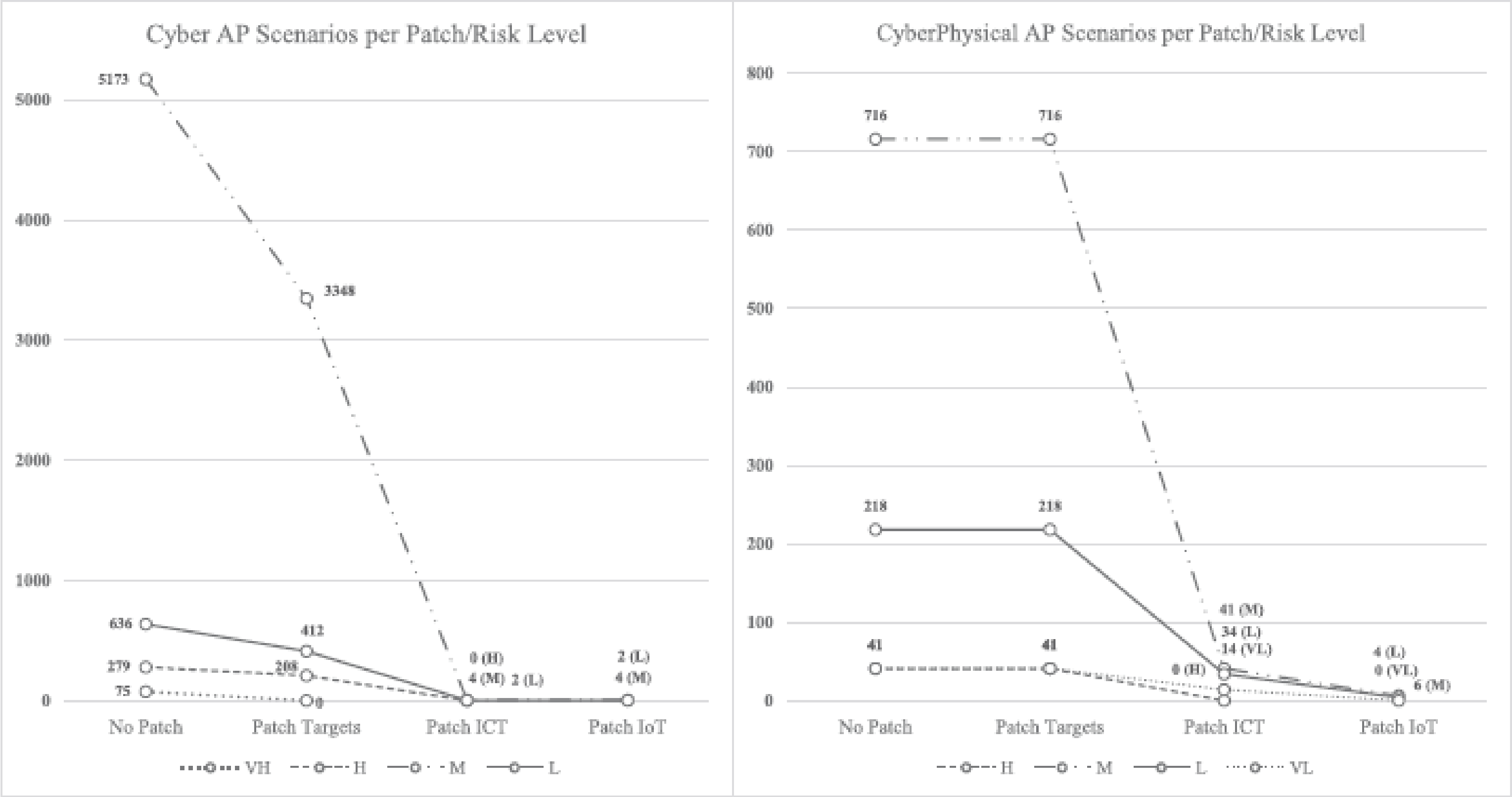


Fig. 8 – Risk level and multitude of attack path scenarios per patch level.

Assessing IoT-enabled C-P attack paths: Open Problems

- Enrichment of interaction modelling phase by including additional physical interaction types.

- Automate the interaction identification phase, by creating a cyber security ontology expressed as a knowledge graph that will improve the processing of temporal and environmental information provided by automated network scanning tools, to automatically produce network information and other stable datasets.

- A promising approach for the production of stable datasets such as the CVSS temporal and environmental scores and the adversarial (threat agent) characteristics, is the utilization of Natural Language Processing (NLP) and other Machine Learning techniques to parse and create context from existing open sources.

References

[1] C. Miller and C. Valasek, “Remote exploitation of an unaltered passenger vehicle,” Black Hat USA, 2015.
[2] C. Cerrudo, “Hacking US traffic control systems,” 2014
[3] R. Santamarta. (2016) In flight hacking system. http://blog.ioactive.com/2016/12/in-flight-hacking-system.html
[4] E. Weise. (2015) Computer expert hacked into plane and made it briefly fly sideways, according to FBI (Independent). http://www.independent.co.uk/news/world/americas/computer-expert-hacks-into-plane-and-makes-it-fly-sideways-according-to-fbi-10256145.html
[5] B. Rios and J. Butts. (2017) Security evaluation of the implantable cardiac device ecosystem architecture and implementation interdependencies. https://www.a51.nl/sites/default/files/pdf/Pacemaker%20Ecosystem%20Evaluation.pdf
[6] TrapX Research, Labs, “Anatomy of Attack: MEDJACK.2 – Hospitals Under Siege,” TrapX Investigative Report, 2016.
[7] D. Formby, S. Durbha, and R. Beyah, “Out of control: Ransomware for industrial control systems,” 2017.
[8] F. Maggi, D. Quarta, M. Pogliani, M. Polino, A. M. Zanchettin, and S. Zanero, “Rogue robots: Testing the limits of an industrial robots security,” Trend Micro, Politecnico di Milano, Tech. Rep.
[9] D. U. Case, “Analysis of the cyber attack on the ukrainian power grid,” 2016
[10] G. Andy. (2017) How an entire nation became russia’s test lab for cyberwar. www.wired.com. [Online]. Available: https://www.wired.com/story/russian-hackers-attack-ukraine/
[11] H. Alex. (2016). [Online]. https://www.theguardian.com/technology/2016/dec/29/smart-electricity-meters-dangerously-insecure-hackers
[12] E. Ronen, C. O’Flynn, A. Shamir, and A.-O. Weingarten, “Iot goes nuclear: Creating a zigbee chain reaction,” IACR Cryptology ePrint Archive, vol. 2016, p. 1047, 2016
[13] E. Ronen and A. Shamir, “Extended functionality attacks on iot devices: The case of smart lights,” in 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016, pp. 3–12.
[14] T. Greene. (2016) How the Dyn DDoS attack unfolded. [Online]. http://www.networkworld.com/article/3134057/security/how-the-dyn-ddos-attack-unfolded.html
[15] Wikileaks. (2017) Vault 7: CIA Hacking Tools Revealed – CIA malware targets iPhone, Android, smart TVs. [Online]. Available: https://wikileaks.org/ciav7p1/
[16] Stellios, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., & Lopez, J. (2018). *“A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services”.* *IEEE Communications Surveys & Tutorials*, *20*(4), 3453-3495.
[17] Stellios I., Kotzanikolaou P. and Grigoriadis C., *“Assessing IoT enabled cyber-physical attack paths against critical systems”*. Elsevier Computers and Security, Vol.107, August 2021, 102316