Vulnerabilità

Obiettivi

- Presentare nel dettaglio le principali vulnerabilità riscontrabili nei sistemi di elaborazione
- Introdurre una tassonomia delle vulnerabilità, raggruppandole in base alla loro natura, al dominio e alla sorgente
- Presentare alcuni esempi che coprono l'intero spazio della tassonomia.

Indice

- Debolezze vs. Vulnerabilità
- Tassonomia delle Vulnerabilità:
  - Natura
  - Dominio
  - Sorgente

Indice

- Debolezze vs. Vulnerabilità
- Tassonomia delle Vulnerabilità:
  - Natura
  - Dominio
  - Sorgente

“ Der Teufel steckt im Detail ”

- I sistemi informativi sono sistemi complessi
- Aspetti rilevanti per la sicurezza possono sembrare “dettagli”
- Alcuni di questi dettagli possono presentare punti deboli (debolezze), che si trasformano in vulnerabilità

Debolezza

- Caratteristica di sistemi o componenti che determina la loro *esposizione*, cioè la possibilità di perdere *Riservatezza*, *Integrità* o *Disponibilità* di asset.

Vulnerabilità

- Particolare debolezza, presente in un componente specifico di un Sistema, che può essere *sfruttata* da un aggressore per compiere azioni non autorizzate a proprio vantaggio contro la Riservatezza, l'Integrità o la Disponibilità degli asset del sistema.

Debolezza vs. Vulnerabilità

Debolezza

- È la *classe*
- Rappresenta un problema *generale*

Vulnerabilità

- È l'*istanza*
- Rappresenta un problema *specifico* di una versione *specifica* di un *componente* specifico

Debolezza vs. Vulnerabilità

Debolezza

- È la *classe*
- Rappresenta un problema *generale*
  - E.g., **CWE-122**:
    Heap-based Buffer Overflow

Vulnerabilità

- È l'*instanza*
- Rappresenta un problema *specifico* di una versione *specifica* di un *componente* specifico
  - E.g., **CVE-2019-6778**:
    In QEMU 3.0.0, tcp_emu in slirp/tcp_subr.c has a heap-based buffer overflow.

vs.

CWE™

- *Common Weakness Enumeration* è un elenco, sviluppato dalla comunità, di punti deboli comuni per la sicurezza di software e hardware. Serve come linguaggio comune, come metro di misura per gli strumenti di sicurezza e come linea di base per l'identificazione dei punti deboli, la mitigazione e gli sforzi di prevenzione.

[https://cwe.mitre.org]

CVE®

- *Common Vulnerabilities and Exposures* è un elenco di voci - ciascuna contenente un numero di identificazione, una descrizione e almeno un riferimento pubblico - per le vulnerabilità di sicurezza informatica note al pubblico

[https://cve.mitre.org]

Caveat

- Le Vulnerabilità possono impattare sia la Safety sia la Security, e quindi, la Dependability!

Safety, (Cyber)Security e Dependability

- Persone
- Ambiente

SAFETY

- Oggetti
- Computer
- Informazioni

SECURITY

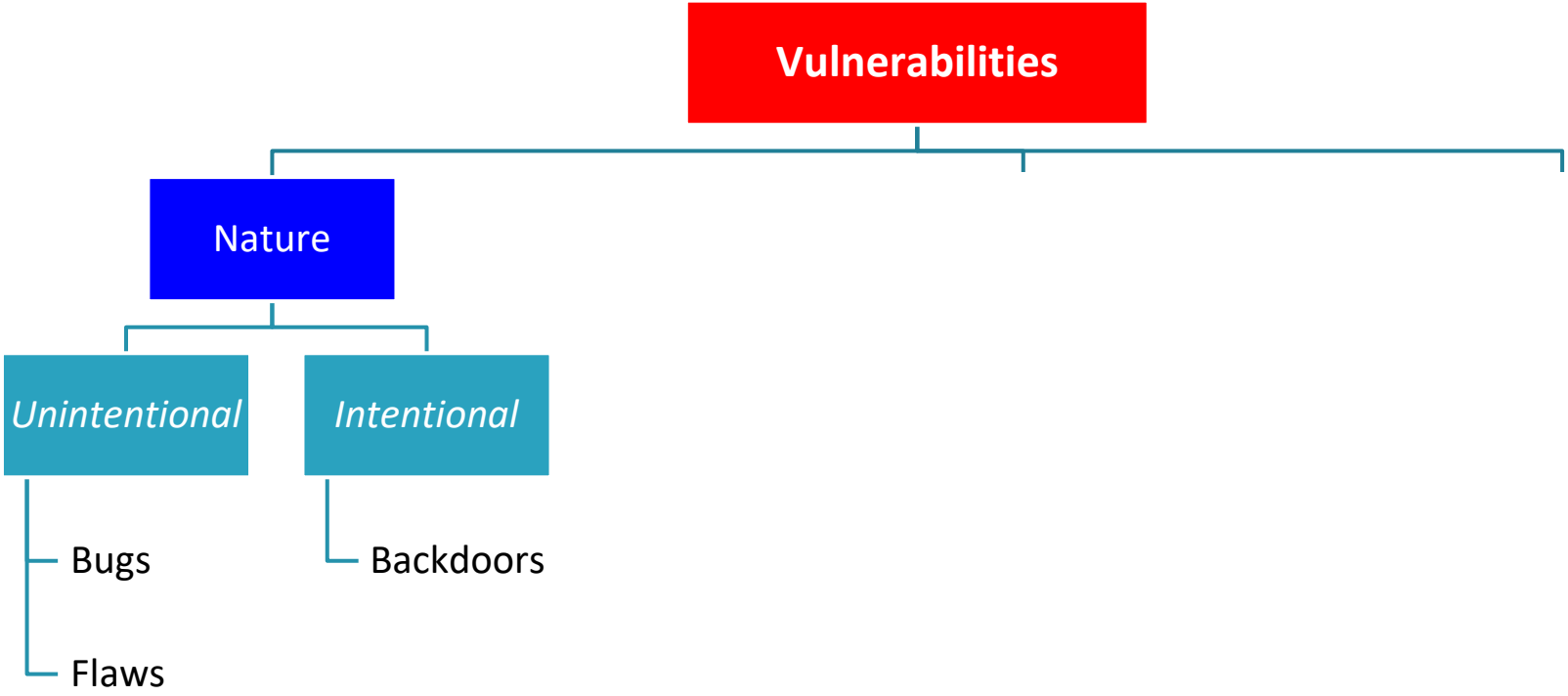
- Cyberspace

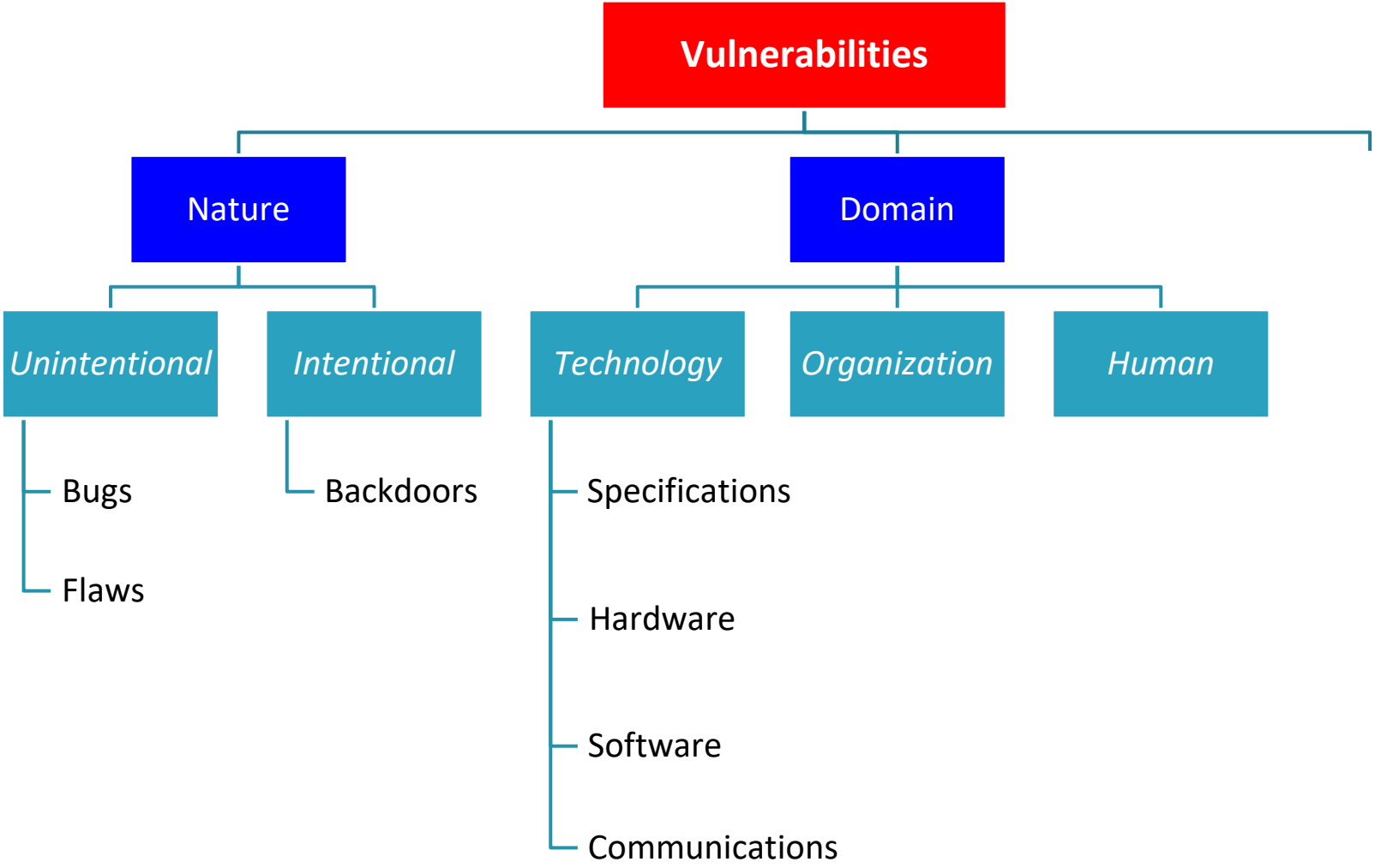
CYBERSECURITY

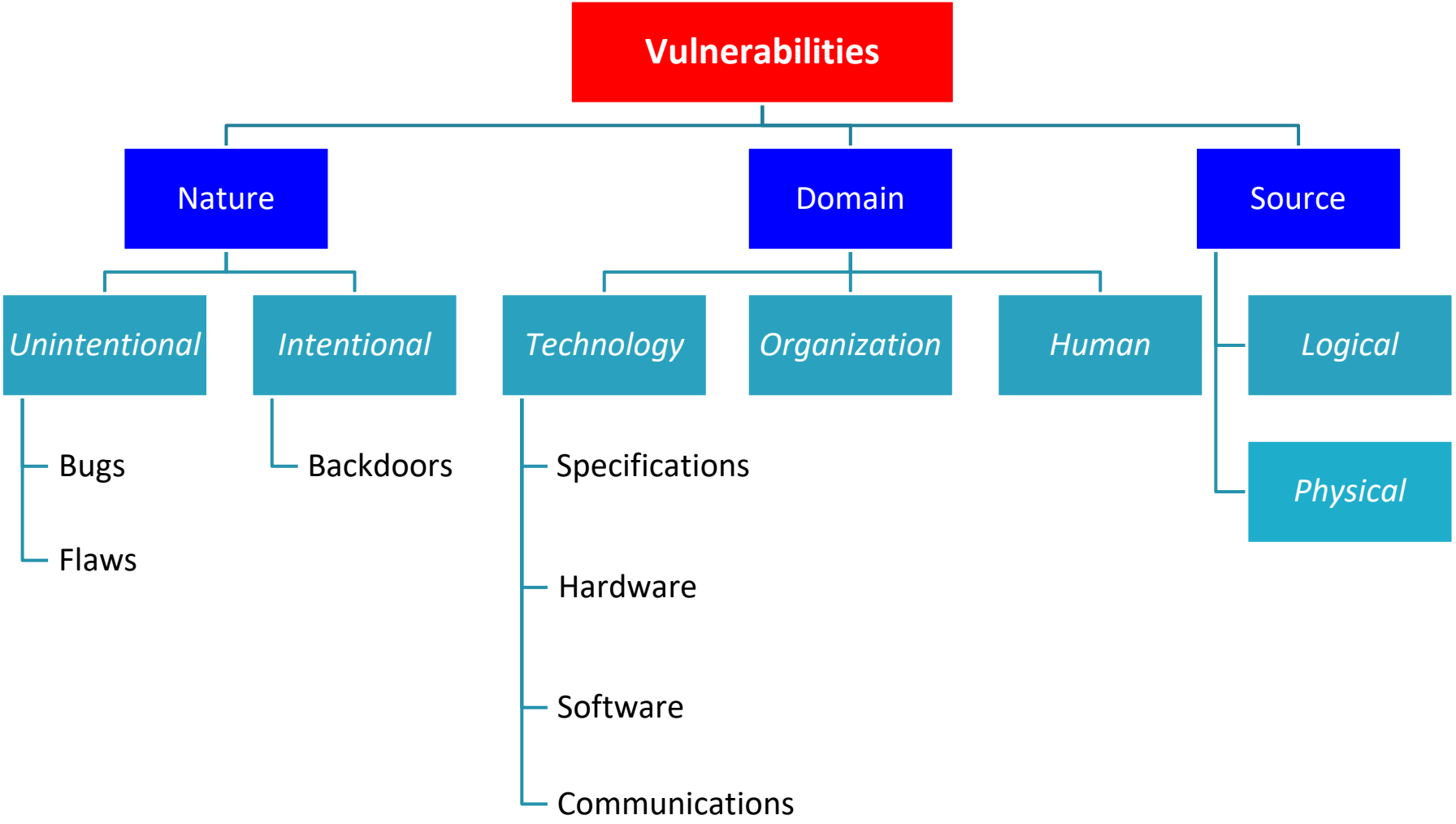
DEPENDABILITY

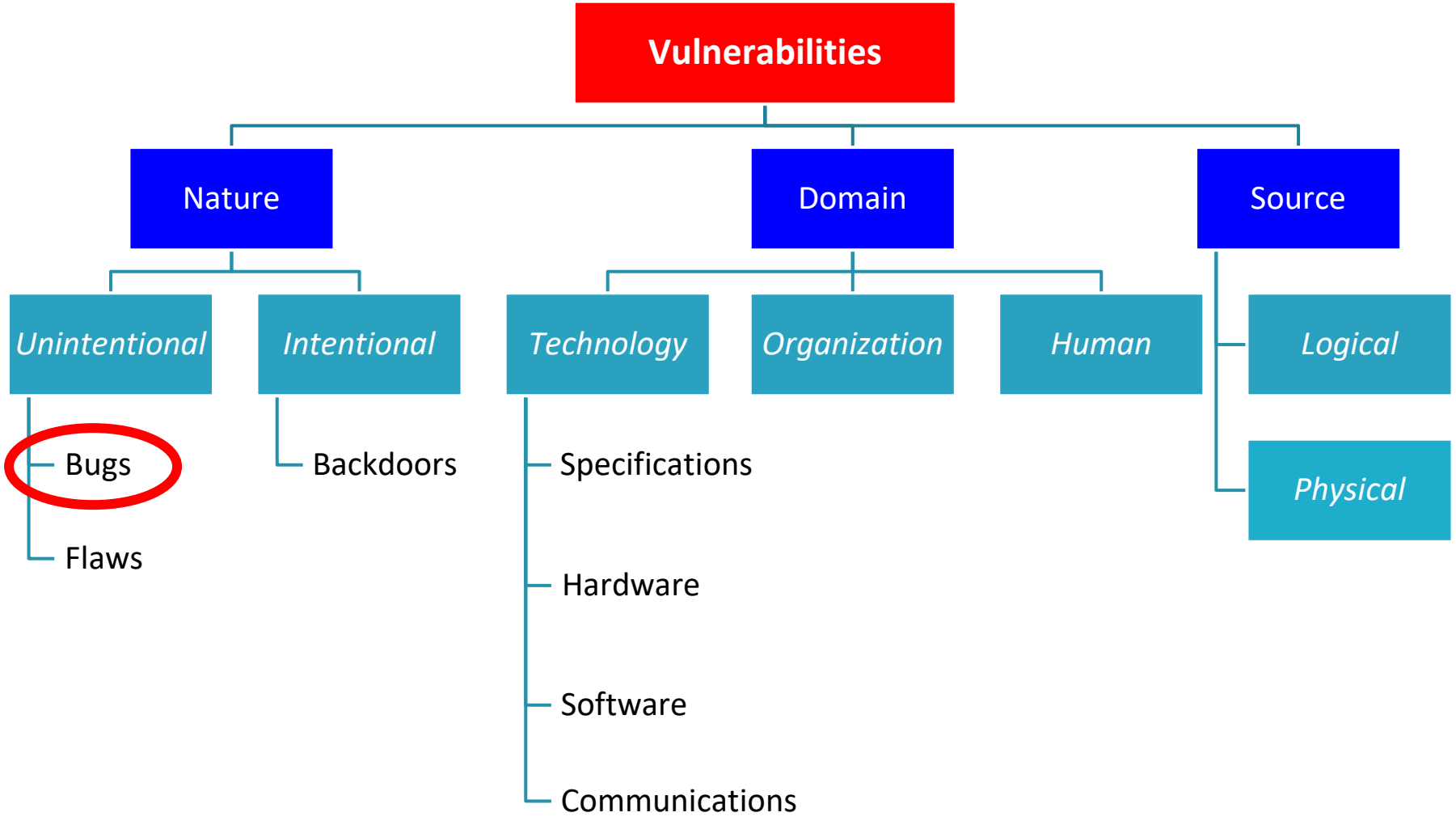
Vulnerabilità

- Possono essere raggruppate secondo diverse dimensioni ortogonali
- Nel seguito ci concentreremo sulle seguenti tre:
  - *Natura* della vulnerabilità
  - *Dominio* della vulnerabilità
  - *Sorgente* della vulnerabilità


Vulnerabilities
Nature
Unintentional
Intentional
Bugs
Flaws
Backdoors


Vulnerabilities
Nature
Domain
Unintentional
Intentional
Technology
Organization
Human
Bugs
Flaws
Backdoors
Specifications
Hardware
Software
Communications


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Specifications
Hardware
Software
Communications


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Specifications
Hardware
Software
Communications

Bug: possibili sorgenti

- I bug possono essere generate da fonti diverse, tra cui:
  - Persone
  - Procedure
  - Strumenti

Bug: possibili sorgenti

- I bug possono essere generati da fonti diverse, tra cui:
  - Persone
  - Procedure
  - Strumenti

- Durante la fase di progettazione e produzione:
  - Inesperienza di progettisti, responsabili di produzione e di collaudo:
    - Errori di progettazione
    - Carenze nelle fasi di convalida e verifica (V&V : Validation & Verification)
    - Insufficienti coperture da parte dei processi di collaudo
- Sul campo:
  - uso improprio

A proposito di “uso improprio”

- L’errore umano è una delle cause più frequenti di “failure”
- Tuttavia molti errori che sono attribuiti agli operatori spesso sono causati da
  - progettazione che non tiene in considerazione gli aspetti di usabilità (Usable Security)
  - assenza di (adeguato) addestramento
  - assenza di (adeguato) supporto alle decisioni

Bug: possibili sorgenti

- I bug possono essere generati da fonti diverse, tra cui:
  - Persone
  - Procedure
  - Strumenti
- Errori all'interno delle:
  - Regole di progettazione
  - Metodologie di progettazione
  - Metodologie di V&V
  - Metodologie di collaudo
- Mancanze di controlli di conformità rispetto a:
  - Metodologie di progettazione e V&V
  - Standard adottati ai vari i livelli

Bug: possibili sorgenti

- I bug possono essere generati da fonti diverse, tra cui:
  - Persone
  - Procedure
  - Strumenti
- Gli strumenti adottati nelle varie fasi possono essere
  - Inappropriati
  - Bacati

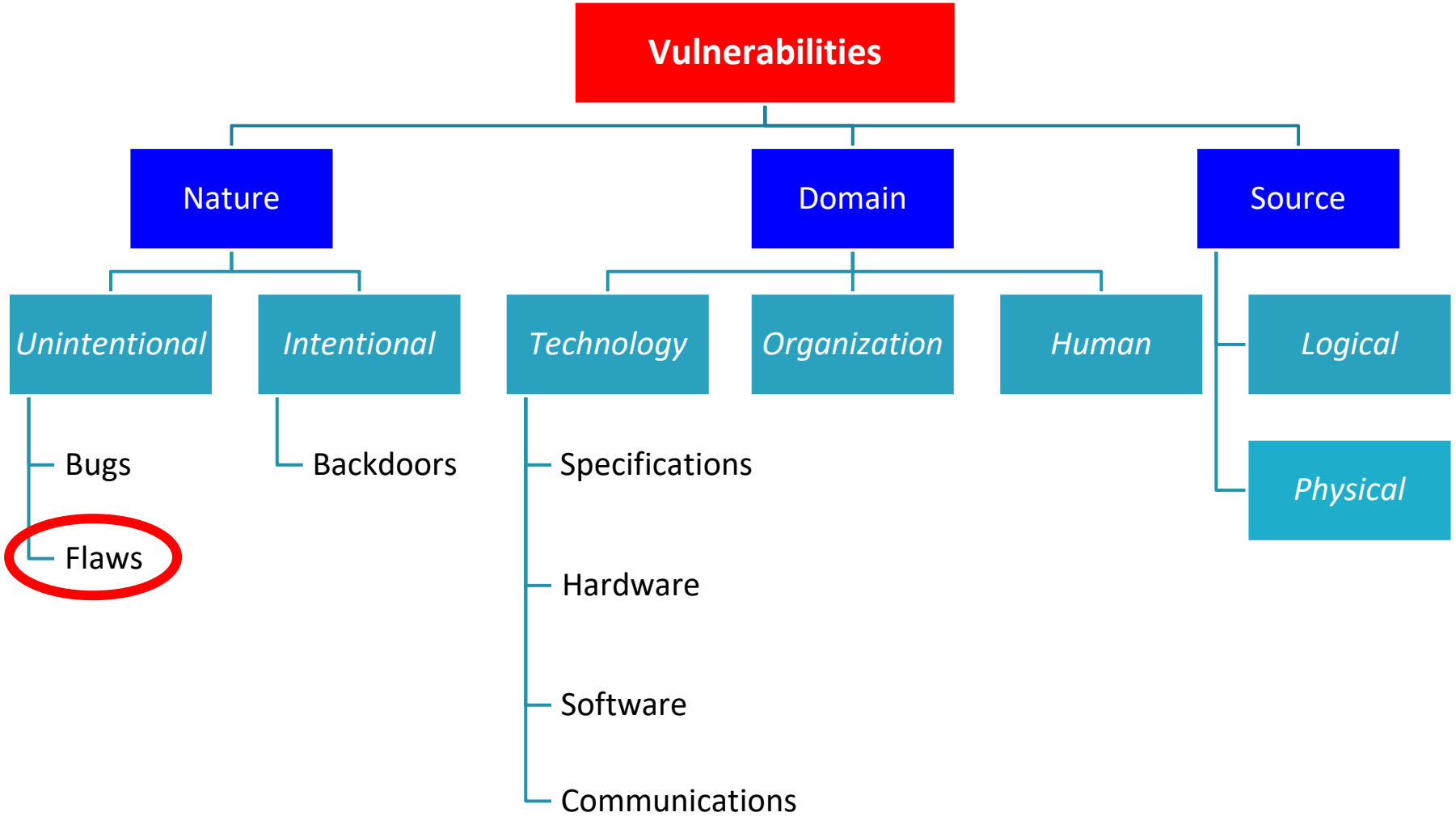
Mix molto pericoloso …

MD82 of Spanair (flight JK5022), Madrid Bajaras 20/08/2008

- Airplane was in a wrong configuration
  - Most probably due to an HW fault, the airplane before take-off was in "flight mode" instead than in "ground mode": the safety mechanism detecting the wrong position of the flaps was disconnected
- Maintenance was done considering the manual in a wrong way
  - A supposed faulty sensor was disconnected because….redundant !
  - However this missing sensor might be one of the cause of the HW fault causing the wrong configuration
- Pilots didn' t perform one of the visual checks foreseen by the pre-takeoff checklist

Bug: come rimediare

- Sono necessari ulteriori investimenti in termini di:
  - Persone
  - Procedure
  - Strumenti


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Specifications
Hardware
Software
Communications

Flaw (Difetto)

- Una caratteristica non primaria (che non costituisce una incoerenza rispetto alle specifiche) derivante da un’errata concezione del progettista che non ne ha tenuto in debito conto la potenziale pericolosità.

Flaw: cause

- Ignoranza dei problemi di sicurezza
- Analisi insufficiente dei casi di uso potenzialmente pericolosi
- Priorità all'ottimizzazione di alcune delle dimensioni dello spazio di progetto, quali, a titolo di esempio:
  - Facilità d'uso
  - Prestazioni
- …

Flaw: come rimediare

- Investimenti addizionali in termini di:
  - Persone:
    - Formazione orientata alla sicurezza
    - Formazione continua
  - Procedure:
    - Sicurezza come requisito
    - Processo di sviluppo orientato alla sicurezza
    - Security-by-Design
    - Campagne estensive di VAPT (Vulnerability Assessment & Penetration Testing)

Approfondimenti

- Software flaw:
  - Parecchi esempi nei giorni prossimi…
- Hardware flaw:
  - Un esempio è nell'Approfondimento 2

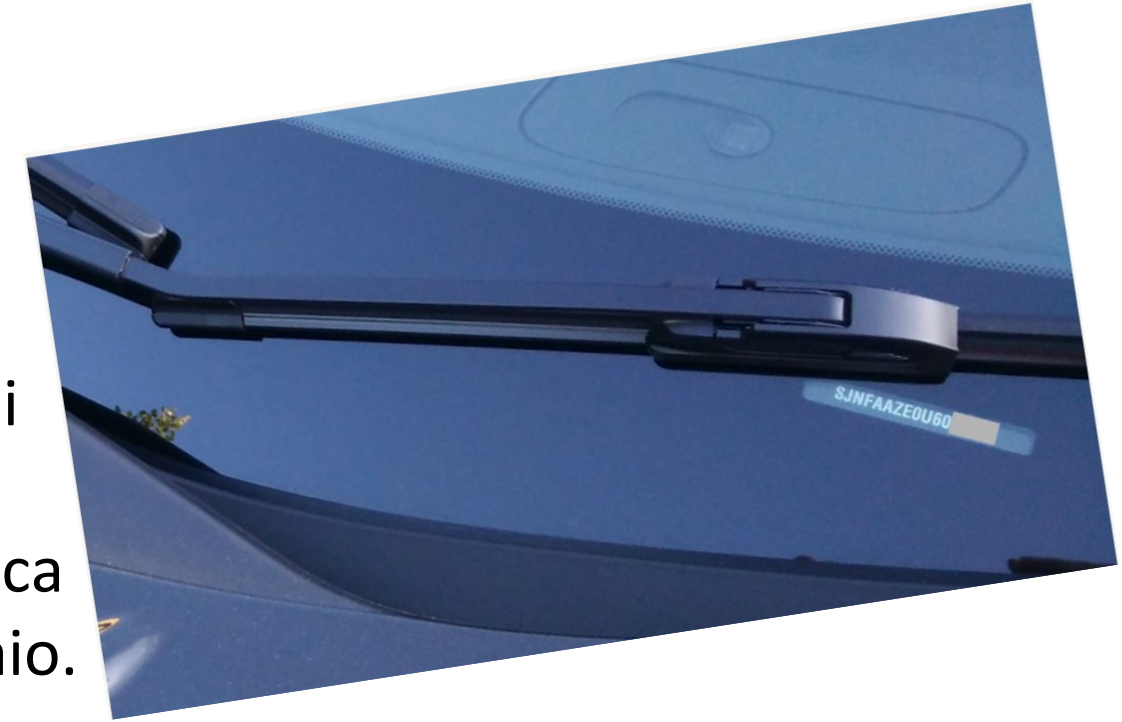
Esempio: a livello di “Sistema”

Controlling vehicle features of Nissan LEAFs across the globe via vulnerable APIs

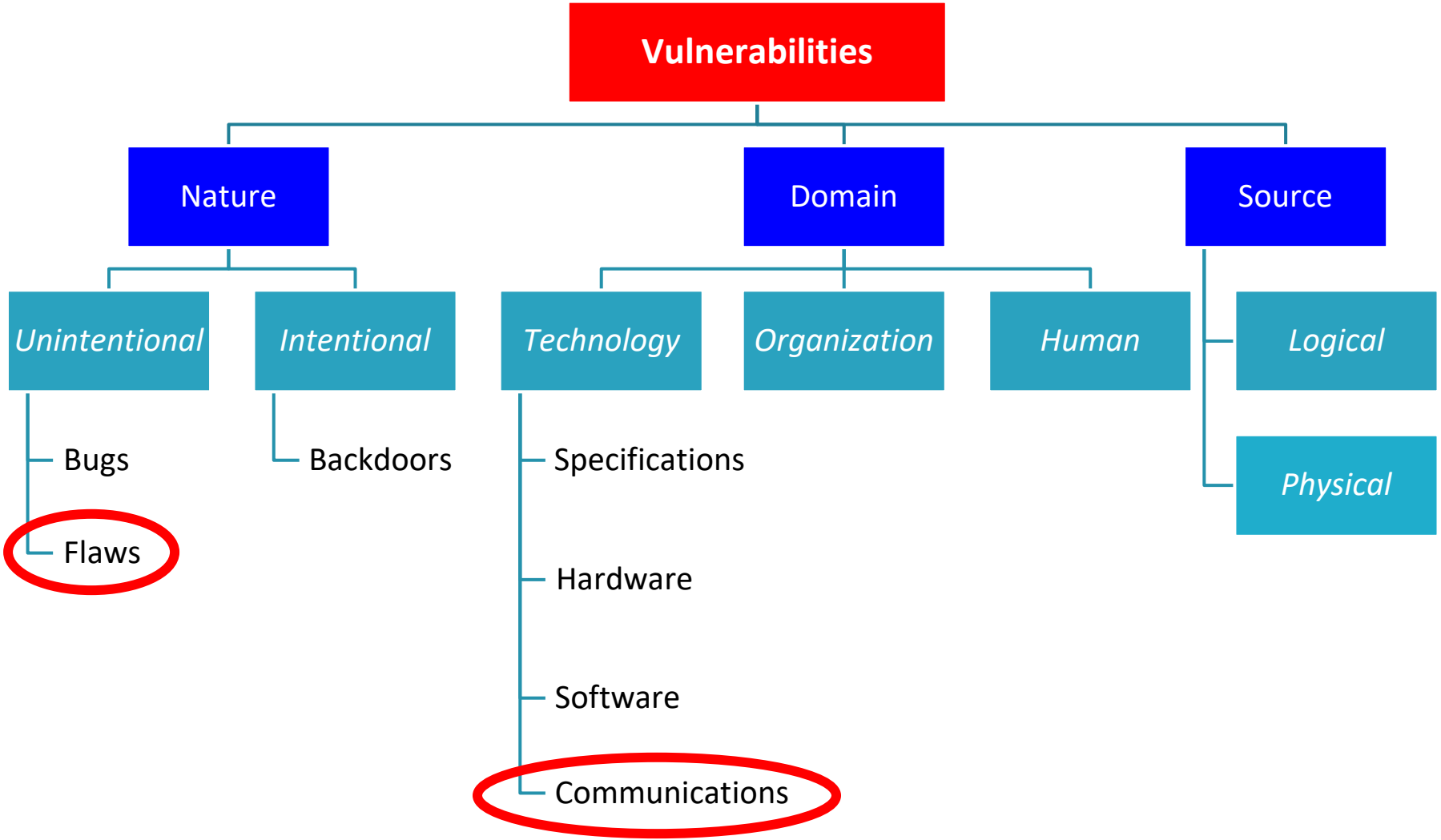
24 FEBRUARY 2016

Come

Un attacco è stato portato a termine sfruttando il numero di identificazione del veicolo, che ne identifica in modo univoco il telaio.

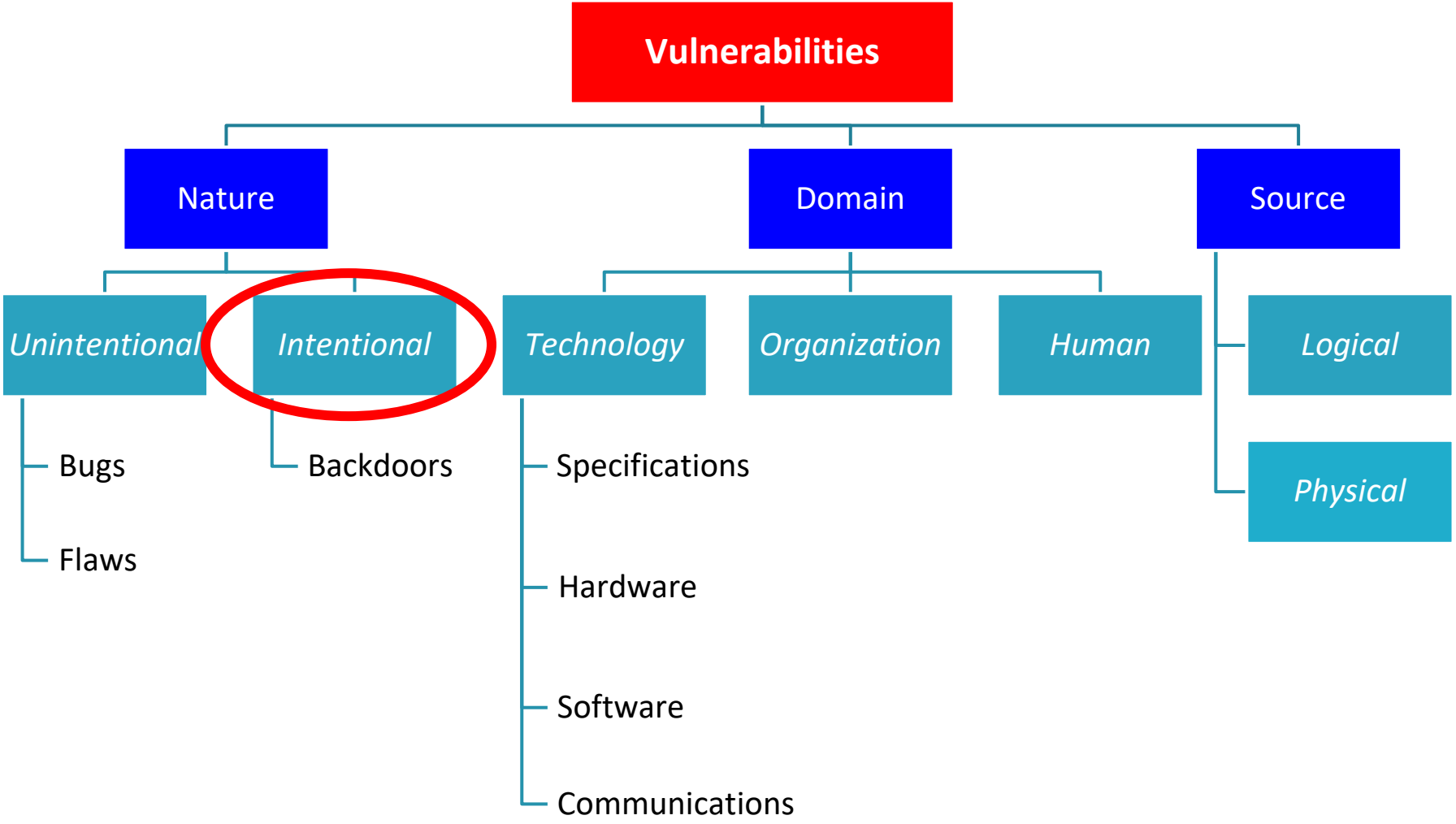


https://www.youtube.com/watch?v=Nt33m7G_42Q


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Backdoors
Specifications
Flaws
Hardware
Software
Communications

Esempi

- Protocolli di comunicazione vulnerabili
- Mancato uso della crittografia
- Maldestro uso delle crittografia
- Algoritmi crittografici non sicuri


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Specifications
Hardware
Software
Communications

Vulnerabilità intenzionali

- Quando una vulnerabilità viene inserita intenzionalmente, può essere definite come *backdoor*, in quanto la persona che la inserisce vuole garantire a se stessa (o a qualcun altro) la possibilità di un accesso o di un utilizzo successivo che esulino dall'insieme dei casi d'uso previsti.

Backdoor

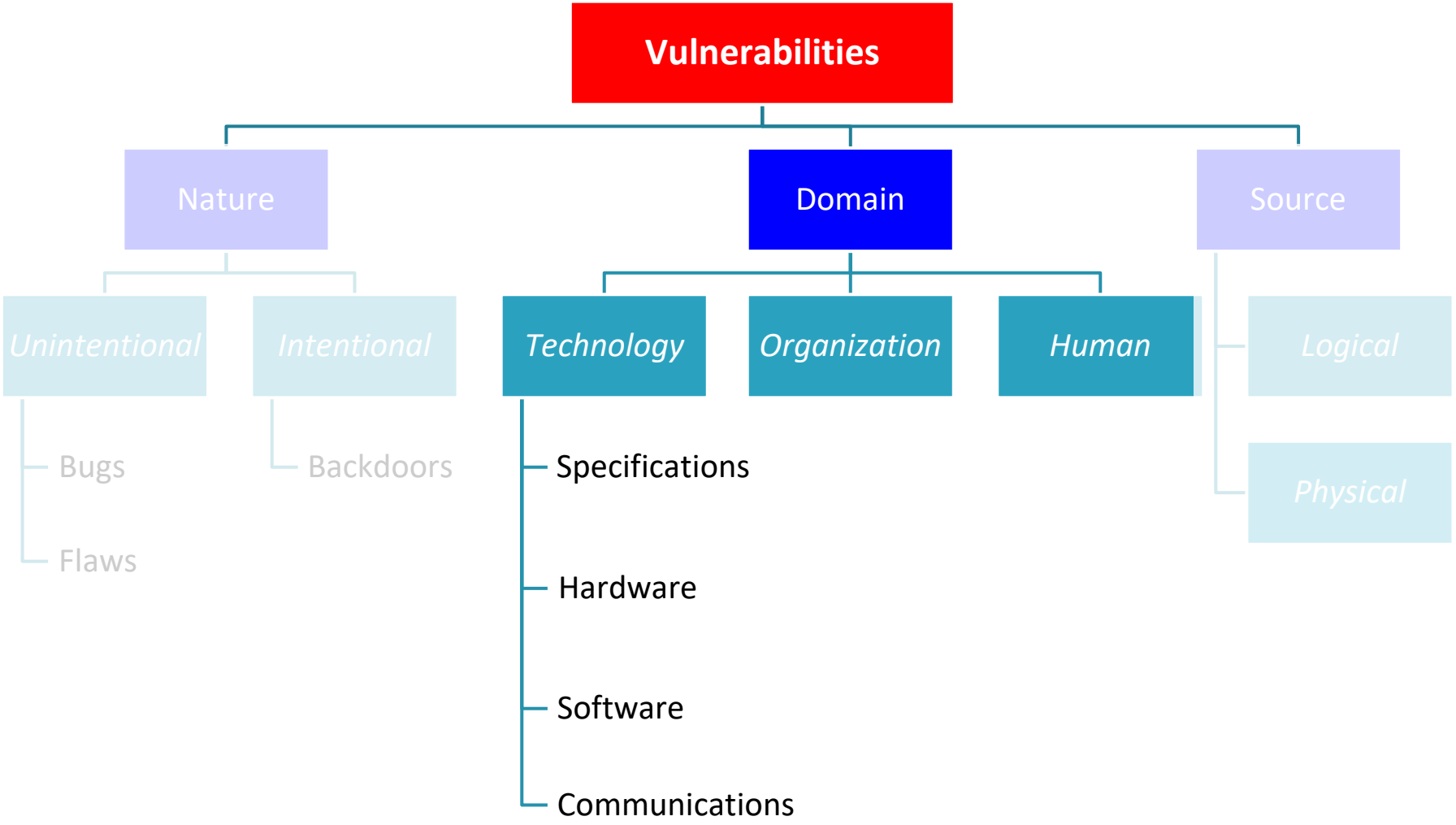
- Una backdoor è sempre una vulnerabilità, anche se i progettisti non l'hanno inserita al fine di danneggiare il sistema

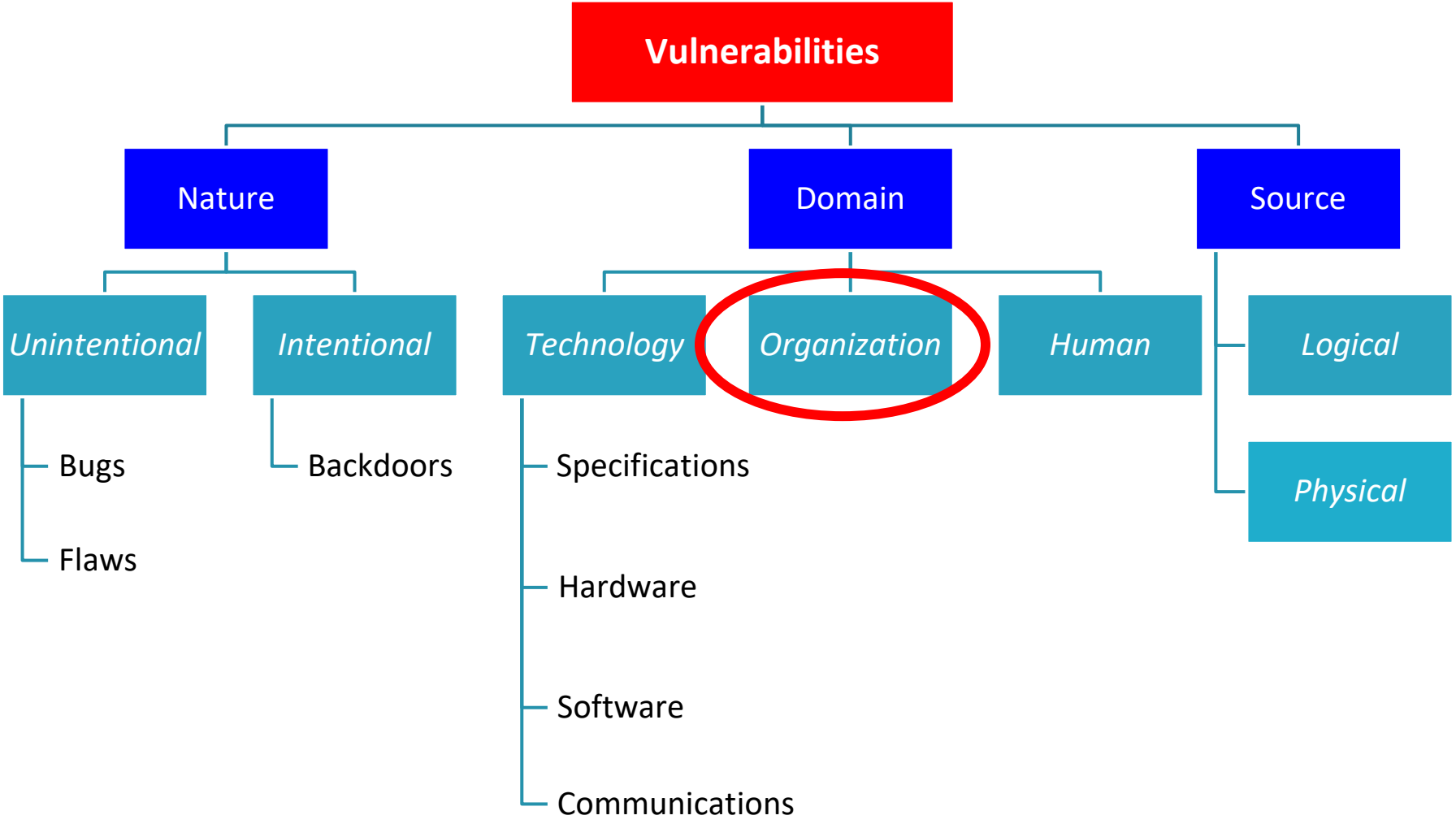
Backdoor: cause e motivazioni

- Interlocutori non fidati della catena di approvigionamento (*supply chain*)
- Elementi non fidati all'interno del processo di progettazione
- Inserimento voluto di backdoor per permettere il debug da remoto e/o la manutenzione sul campo

Backdoor: come rimediare

- Come fidarsi di OGNI anello della catena di approvvigionamento?
  - Una questione ancora aperta
  - Sono necessari ulteriori investimenti in termini di ricerca
- Assicurare la possibilità di aggiornamento e di manutenzione a distanza senza lasciare backdoor aperte


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Backdoors
Specifications
Flaws
Hardware
Software
Communications


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Specifications
Hardware
Software
Communications

Vulnerabilità a livello di organizzazione

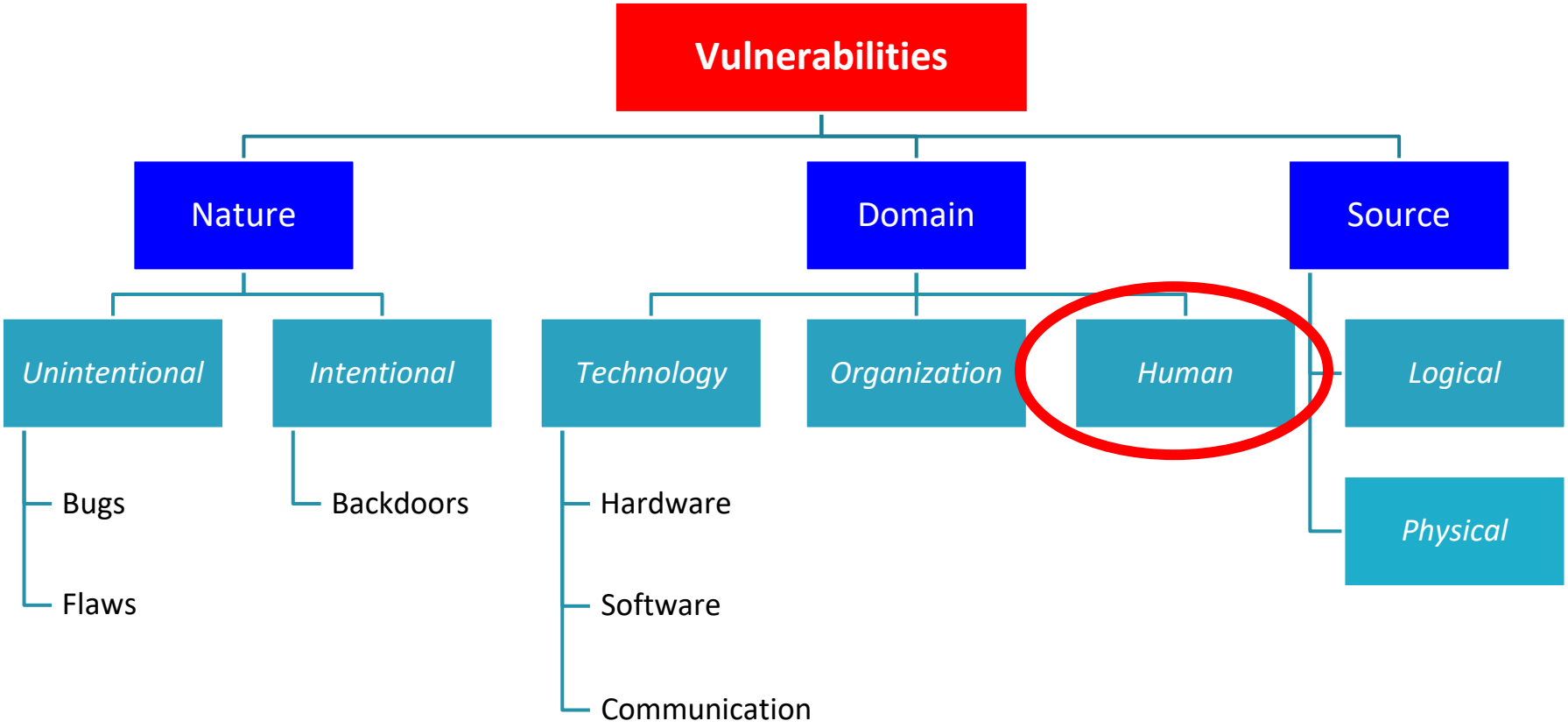
- Inadeguatezza degli aspetti organizzativi in termini di:
  - Infrastrutture di difesa
  - Rilevamento di attacchi
  - Risposte a eventuali incidenti
  - Ripristino delle attività
  - Resilienza
  - …

Vulnerabilità a livello di organizzazione

- Inadeguatezza degli aspetti organizzativi in termini di:
  - Infrastrutture di difesa
  - Rilevamento di attacchi
  - Risposte a eventuali incidenti
  - Ripristino delle attività
  - Resilienza
  - …
- Inefficacia delle strategie di sicurezza adottate, con carenze in termini di:
  - Qualità dei team
  - Best practice
  - Strumenti
  - Tecnologie

Cause

- Mancanza o carenza di infrastrutture adeguate in termini di:
  - CSIRT - Computer Security Incident Response Team
  - SOC - Security Operations Center
  - SIEM - Security Information and Event Management
- …


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Hardware
Software
Communication

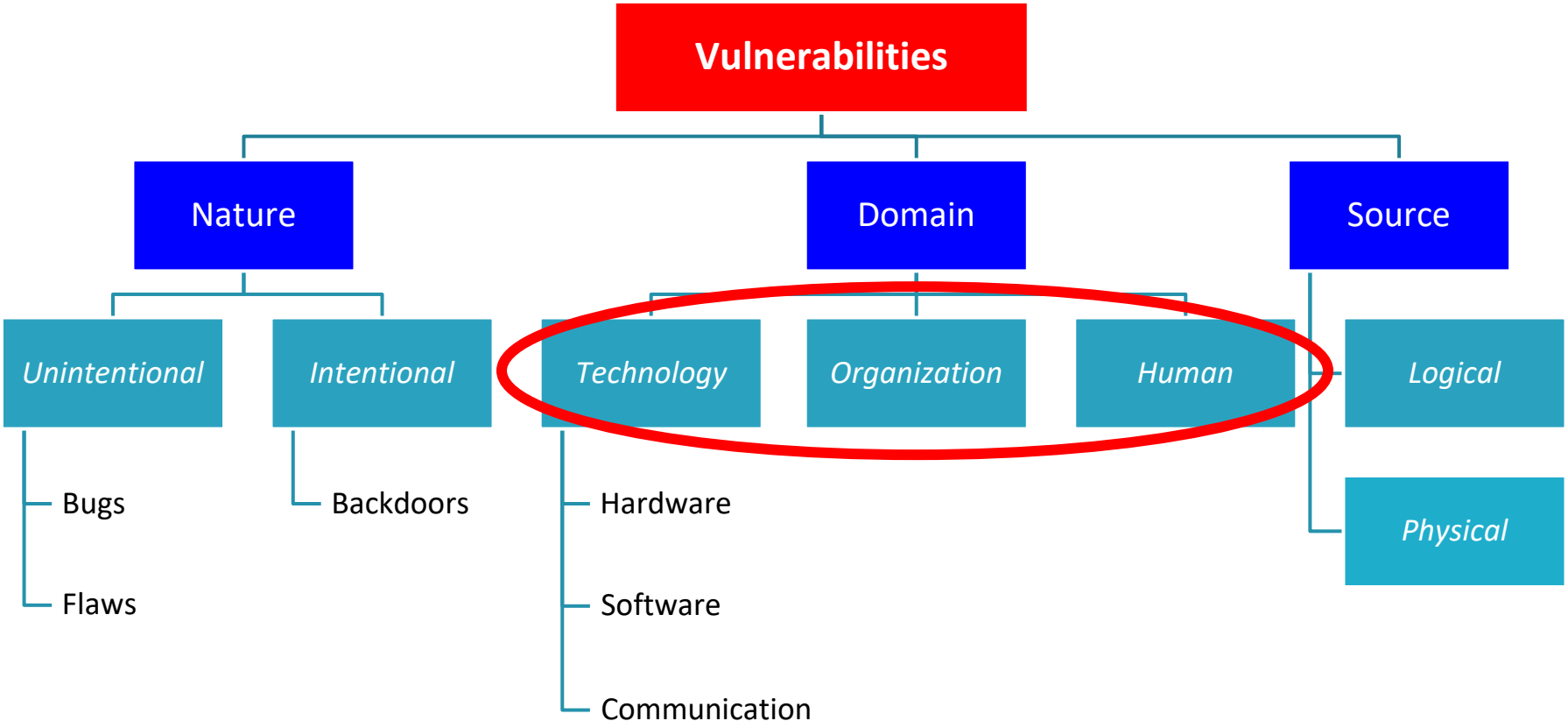
Vulnerabilità a livello umano

- Scarsa consapevolezza e cultura da parte di TUTTE le persone coinvolte
- Errata percezione dei rischi
- *Ingegneria sociale*

Vulnerabilità a livello umano

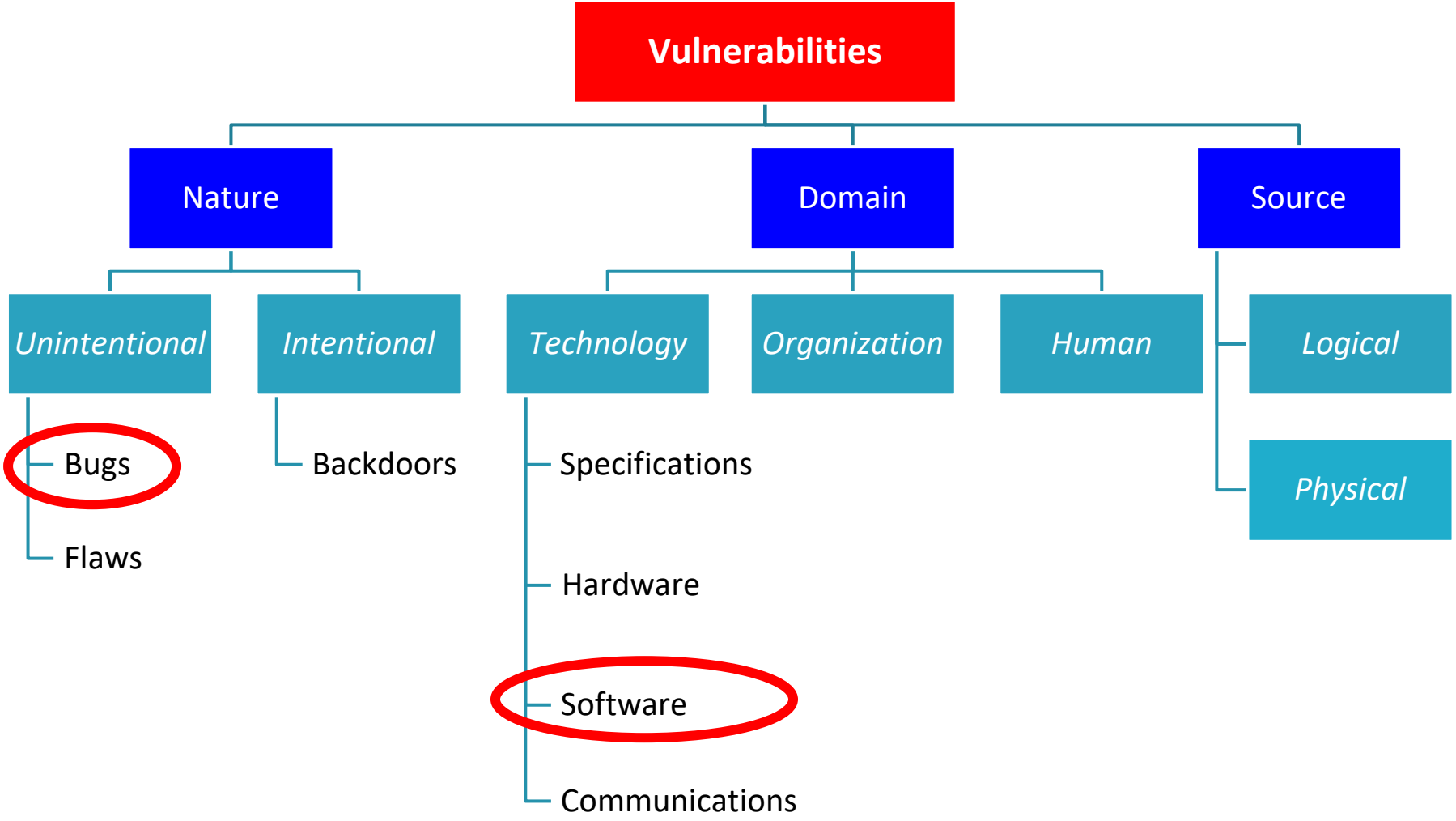
- Scarsa consapevolezza e cultura da parte di TUTTE le persone coinvolte
- Errata percezione dei rischi
- *Ingegneria sociale*

Trattata nel dettaglio nel
Il fattore umano - Social Engineering


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Hardware
Software
Communication

L’uomo è spesso l’anello debole !!

Problem Exist Between Keyboard And Chair

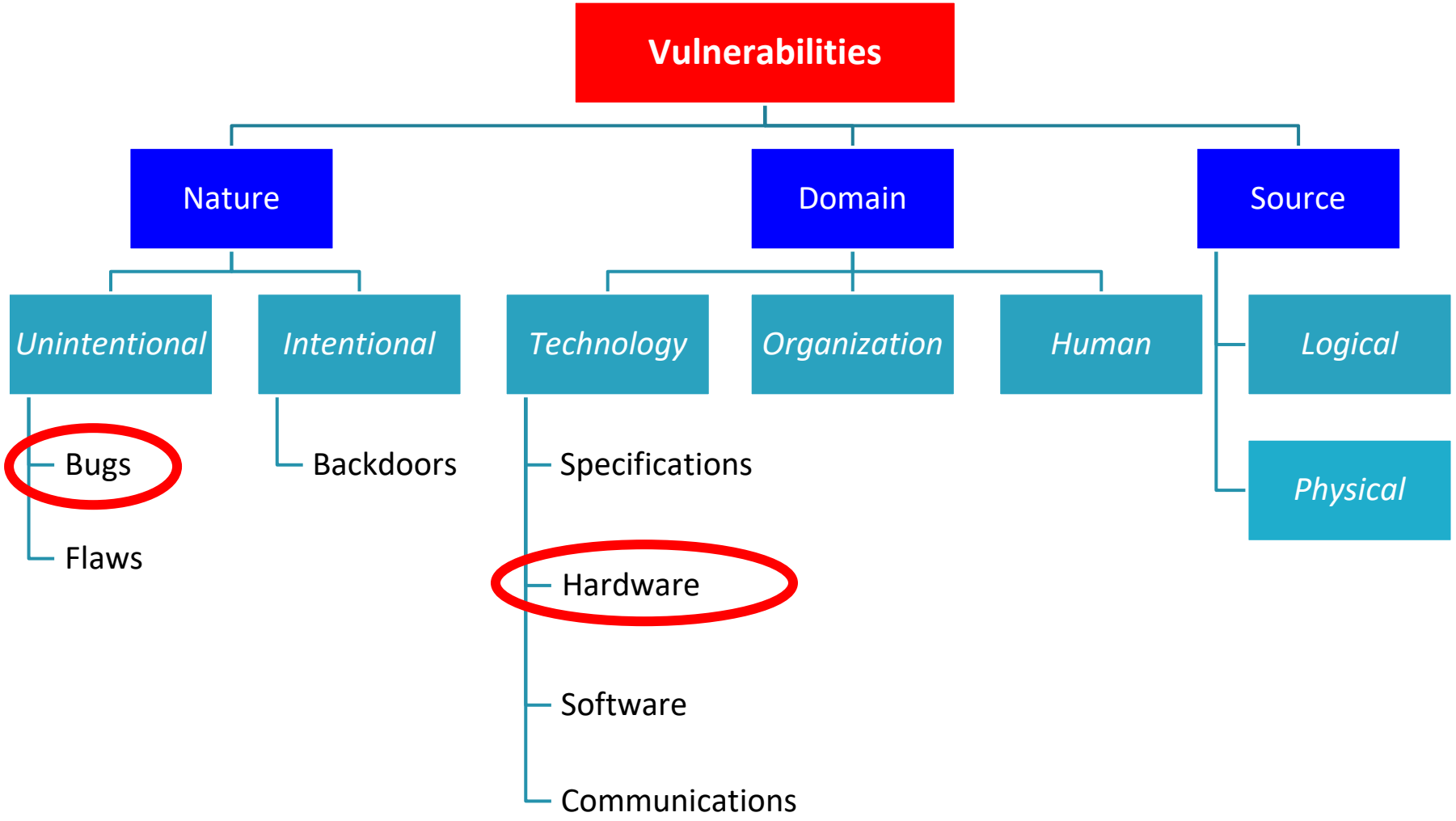

Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Specifications
Hardware
Software
Communications

Example: Ariane 6

- French Guyana,
  June 4, 1996:
  Failure of Ariane 6 due to bad specification of SW module for reuse

Example: Ariane 6

- French Guyana,
  June 4, 1996:
  Failure of Ariane 6 due to bad specification of SW module for reuse
- Cost: 800 M $


Vulnerabilities
Nature
Domain
Source
Unintentional
Intentional
Technology
Organization
Human
Logical
Physical
Bugs
Flaws
Backdoors
Specifications
Hardware
Software
Communications

Hardware Bug

- An inconsistency between a specification and its actual implementation, introduced by a mistake during the design and not detected during *Validation & Verification* (V&V) phases.

Example: “F00F Pentium P5 Bug”

- Detected in 1997 in all the Pentium P5 processors
- In the x86 architecture, the byte sequence F0 0F C7 C8 represents the instruction lock cmpxchg8b eax (locked compare and exchange of 8 bytes in register EAX) and does not require any special privilege.
- Howerer, the instruction encoding is invalid. The cmpxchg8b instruction compares the value in the EDX and EAX registers (the lower halves of RDX and RAX on more modern x86 processors) with an 8-byte value in a memory location.
- In this case, however, a register is specified instead of a memory location, which is not allowed.

Example: “F00F Pentium P5 Bug”

- Under normal circumstances, this would simply result in an exception
- However, when used with the lock prefix (normally used to prevent two processors from interfering with the same memory location), the CPU erroneously uses locked bus cycles to read the illegal instruction exception-handler descriptor.
- Locked reads must be paired with locked writes, and the CPU's bus interface enforces this by forbidding other memory accesses until the corresponding writes occur.
- As none are forthcoming, after performing these bus cycles all CPU activity stops, and the CPU must be reset to recover.
- The instruction can be exploited for a DoS attack.

Example: Cyrix coma bug

- The Cyrix coma bug is a design flaw in Cyrix 6x86, 6x86L, and early 6x86MX processors that allows a non privileged program to hang the computer.

Example: The Cyrix coma bug

- This C program (which uses inline x86-specific assembly language) could be compiled and run by an unprivileged user:

```
unsigned char
c[4] = {0x36, 0x78, 0x38, 0x36};
int main()
{
asm (
"         movl $c, %ebx\n"
"again:   xchgl (%ebx), %eax\n"
"         movl %eax, %edx\n"
"         jmp again\n"
     );
}
```

Example: The Cyrix coma bug

- Executing this program the processor enters an infinite loop that cannot be interrupted.
- This allows any user with access to a Cyrix system with this bug to perform a DoS attack.

Hardware Flaw: example

- Speculative Execution in modern processors
  - On branch instructions, both branches are executed before condition check
  - At commit time, only the correct execution is validated
- Great for performance, but …
  - Commitment does not delete completely non-valid path
  - Traces of discarded execution may leak information

Examples: Microarchitectural Flaws

- Processors do not enter an error state but reveal private information!
- They usually allow a concurrent (aggressor) program to fraudulently access private data and keys of a victim program

- Spectre (2018)
- Meltdown (2018)
- Spoiler (2019)
- Foreshadow
- ZombieLoad (2018)

Example: CWE-127

```
void getValueFromArray(int *array, int len, int index) {

    int value;

    if (index < len) {
        value = array[index];
    }
    else {
        value = -1;
    }

    printf("Value is: %d\n", value);

}
```

Example: CWE-127

```
void getValueFromArray(int *array, int len, int index) {

    int value;

    if (index < len) {
        value = array[index];
    }
    else {
        value = -1;
    }

    printf("Value is: %d\n", value);

}
```

- Check for positive value of `index` is *missing*
- Buffer Under-Read (CWE-127)
- You can read data that may be sensitive or not allowed