Security Aspects of CPSs: a dive into Risk Mitigation and Threat Modeling

September 26th, 2019

Davide ARIU

About Pluribus One

- **University of Cagliari spin-off**
  - Established in 2015 as a spin-out of the **Pattern Recognition and Applications Laboratory**
  - Holding a strong background on **AI, cyber-security, and AI-security**
    - check our blog https://www.pluribus-one.it/company/blog

- **Manufacturer of Cyber Security Solutions**
  - Our leading product, **Attack Prophecy**, is an AI-powered solution for the protection of Web Services
  - Two more solutions coming soon:
    - **AISafe DNS** for the protection of the endpoints
    - **xAV** for the protection of the Mobile Devices

attack prophecy
rewriting the rules of protection

- **Research Intensive Company**

letsCROWD ALOHA SIMARGL

Goals of the talk

- **Goal #.1** To exemplify the process through which cyber-threats affecting CPS(oS) can be modelled

- **Goal #.2** To introduce some basic concepts to those of you less familiar with cyber-security

- **Goal #.3** To provide the more curious references to study while in the way back home :-D

What does «to (cyber) attack» a CPS means?

- The video was recorded in 2010 during the Virus Bullettin conference in Vancouver
- The speaker was actually showing a demo of with trojanised Siemens SCADA system infected by **Stuxnet**.

«Stuxnet», in brief

- Generally recognized as the first example «state-sponsored» cyber-attack
  - **Apparently** organized by the U.S. government to delay Iranian's Uranium enrichment program
  - It is a «**malware**» which infected Windows machines running the **SIEMENS STEP 7** SCADA control software
    - SIEMENS Step 7 was used to control the centrifuges in the uranium enrichment facility in Natanz (IRAN)
  - Basically, what **STUXNET** did was to penetrate **ring 0** (kernel level) of the Windows Operating System. This allowed it to:
    - have the centrifuges spinning out of control, eventually leading to their destruction
    - control what the operating system reports to the user → operators realized of things going wrong very late…

Did you know…

- Uranium as it is roughly found in nature is not directly «fissile» (thus it can't be used neither in a power plant nor in a nuclear weapon)
- Of the 2 isotopes of which rough Uraniun is mostly made (U-235 and U-238), only the U-235 is «fissile».
- Being the U-235 isotope lighter then the U-238, it can be separated from it by vaporizing the uranium and making it pass through a chain of very **quickly spinning centrifuges**.
- The accurate control of the centrifuges speed is one of the most critical steps in the whole process

How STUXNET Worked

1. infection
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search
Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

3. update
If the system isn't a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.

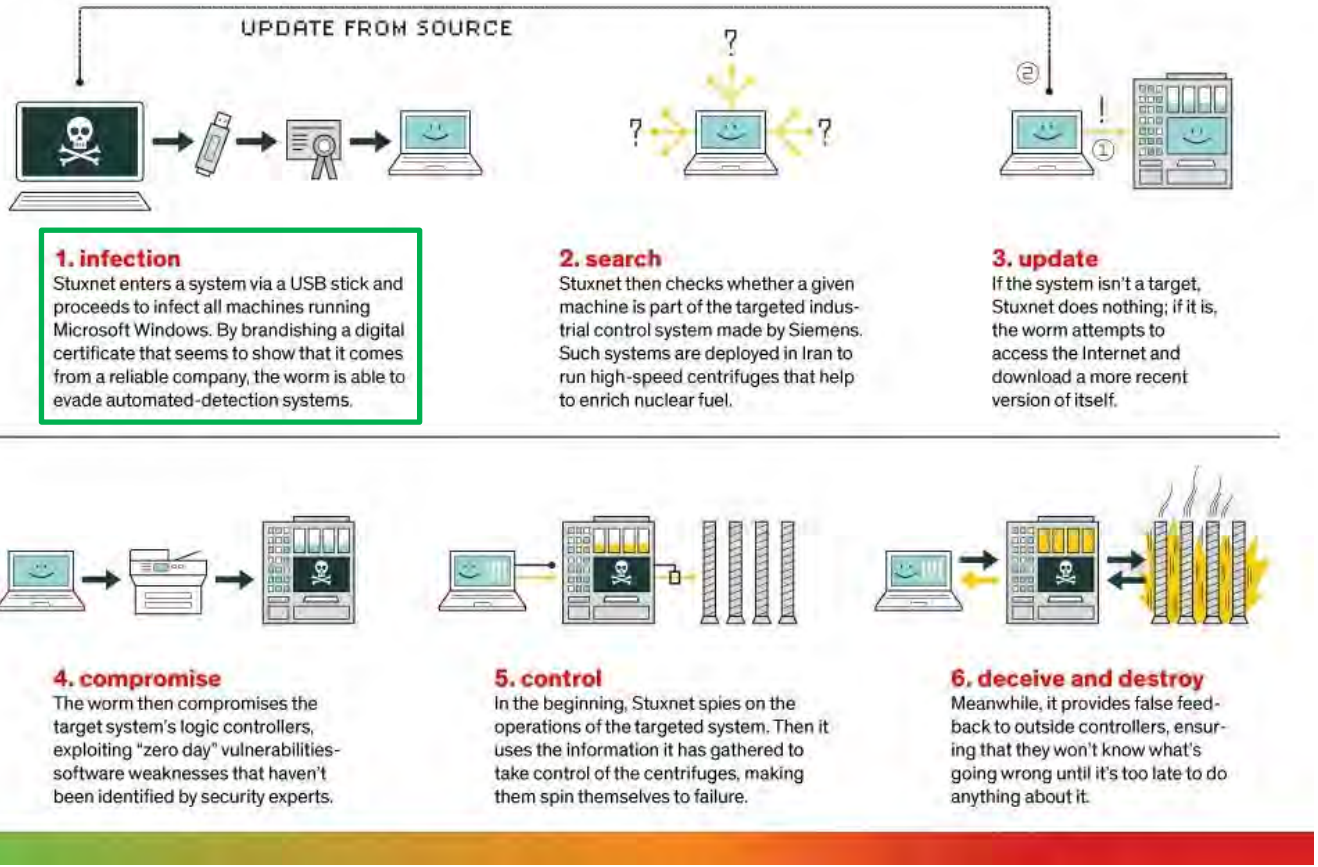
4. compromise
The worm then compromises the target system's logic controllers, exploiting "zero day" vulnerabilities-software weaknesses that haven't been identified by security experts.

5. control
In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

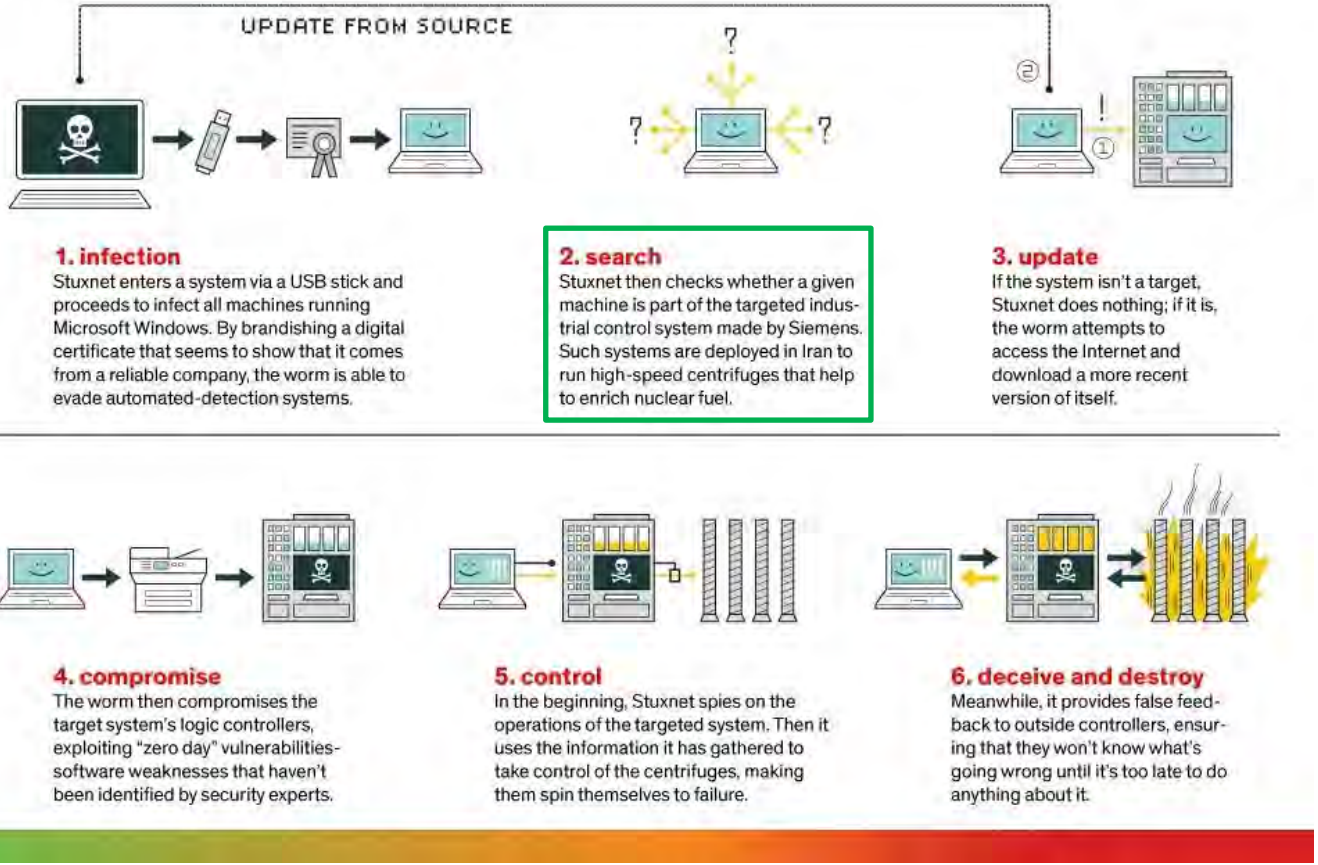
6. deceive and destroy
Meanwhile, it provides false feedback to outside controllers, ensuring that they won't know what's going wrong until it's too late to do anything about it.

Let’s quickly enumerate the most evident threats….

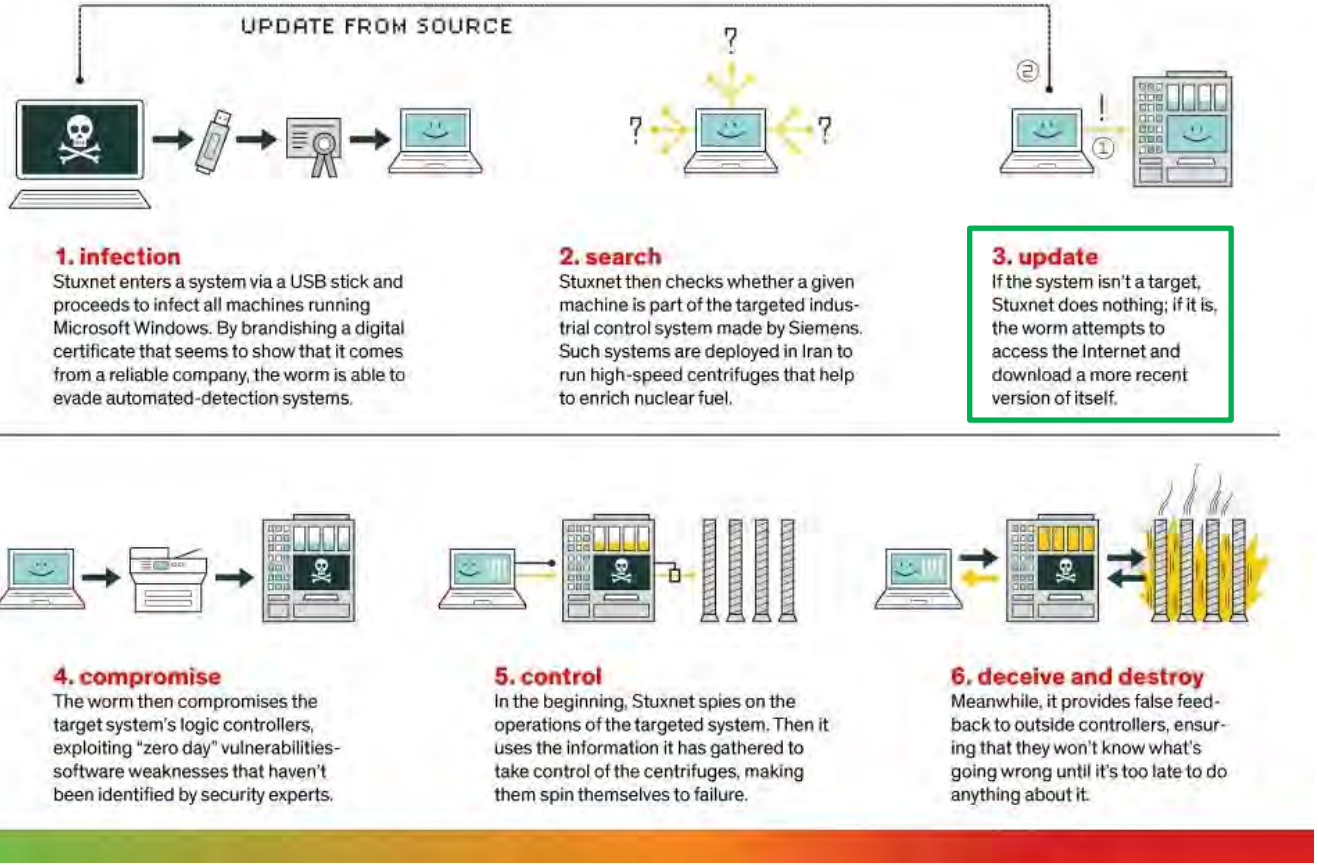
Threat #1 – The USB stick



Threat #2 – Unupdated system



Threat #3 – The Internet connection



General considerations from the Stuxnet example

- Cyber-threats affecting a system, including a CPS, might be of different kind.

But despite the specific domain in which the system operates:

  - the larger the system is
  - the higher the number of components it includes
  - the higher the number of vulnerabilities and issues with such components
  - the higher the number of people operating on the system
  - **the higher the number of threats**

- **Challenge:**
  - Similarly for what it would be done to prevent or address issues of other kind (e.g. energy consumption, performance, etc.) it is necessary, also for cyber-security related issues, to adopt a systematic approach, since the early design stages of the CPS.

  - How can we model the CPS and thus *enumerate*, *analyse*, *evaluate*, and eventually *mitigate* the threats possibly affecting it?

Risk Mitigation and Threat Modeling - 1

- Cyber-security is basically about **risk mitigation**

- Risk can be defined as the combination of the probability of an event and its consequences[1].
  - Or risk can be evaluated as the product of **hazardous event** and the **frequency**, or probability of occurrence.

"*... For operational plans development, the combination of **threats**, **vulnerabilities**, and **impacts must be evaluated in order to identify important trends and decide where effort should be applied to eliminate or reduce threat capabilities;** eliminate or reduce vulnerabilities; and assess, coordinate, and deconflict all cyberspace operations...*"

- **Hazardous events (or threats)** practically **impact** the ***assets***, that is (not exhaustive list):
  - all the subparts a CPS is made of
  - all the pieces of information or material it handles or produces
  - also the people working on the system (safety)

[1]ISO Guide 73:2009 - Risk Management -- Vocabulary
[2]The National Strategy for CyberSpace Operations – Office of the Chairman, Joint Chiefs of Staff, U.S. Department of Defense

Risk Mitigation and Threat Modeling - 2

You can check the «*NIST Special Publication 800-30: Guide for conducting Risk Assessments*» for a comprehensive analysis of the Risk Assessment subject.

- Thus to calculate the risk we need to:
  1. List all the **assets** connected with a CPS
  2. List all the **threats** connected with each **asset** and for each **threat** evaluate the **impact** on the **asset**
  3. For each **asset** and for each **threat** (**impact**) affecting it, evaluate the **likelyhood** the **threat** occurs

Likelyhood \ Severity	**Insignificant** (e.g. no lost time at work)	**Minor** (e.g. some lost time at work)	**Moderate** (e.g. significant lost time at work)	**Major** (e.g. unable to return at work)	**Catastrophic** (e.g. death)
Rare (<3% chance)	Low	Low	Low	Moderate	Moderate
Unlikely (3-10% chance)	Low	Moderate	Moderate	High	High
Moderate (10-50% chance)	Low	Moderate	High	Extreme	Extreme
Likely (50-90% chance)	Moderate	High	Extreme	Extreme	Extreme
Almost Certain (>90% chance)	Moderate	High	Extreme	Extreme	Extreme

Risk Mitigation and Threat Modeling - 2

You can check the «*NIST Special Publication 800-30: Guide for conducting Risk Assessments*» for a comprehensive analysis of the Risk Assessment subject.

- Thus to calculate the risk we need to:
  1. List all the **assets** connected with a CPS
  2. List all the **threats** connected with each **asset** and for each **threat** evaluate the **impact** on the **asset**
  3. For each **asset** and for each **threat** (**impact**) affecting it, evaluate the **likelyhood** the **threat** occurs

Likelyhood \ Severity	**Insignificant** (e.g. no lost time at work)	**Minor** (e.g. some lost time at work)	**Moderate** (e.g. significant lost time at work)	**Major** (e.g. unable to return at work)	**Catastrophic** (e.g. death)
Rare (<3% chance)	1	3	5	12	15
Unlikely (3-10% chance)	3	7	9	25	35
Moderate (10-50% chance)	5	9	25	50	60
Likely (50-90% chance)	9	25	50	60	80
Almost Certain (>90% chance)	12	35	60	80	100

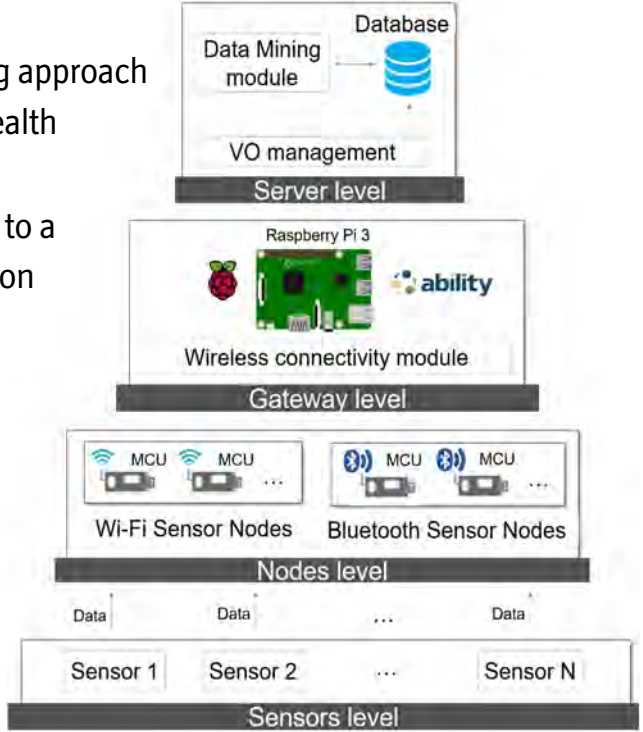
Threat Modeling

- Threat modelling generally helps with these two points:
  1. List all the **assets** connected with a CPS
  2. List all the **threats** connected with each **asset** and for each **threat** evaluate the **impact** on the **asset**
     - Some approaches (which are more risk-oriented) allow to implement the third point
  3. For each **asset** and for each **threat** (**impact**) affecting it, evaluate the **likelyhood** the **threat** occurs

- In order to achieve the goal threat modelling drives to:
  - **create an abstraction of the system**
  - **identify profiles of potential attackers**, including their goals and methods
  - **enumerate** the potential **threats** that may arise

A toy example

- In order to illustrate how to proceed, we will apply a basic threat modelling approach to a very simple system, developed in the context of a research project on e-Health
- **Overall goal of the system:**
  - To collect clinical data through wearable sensors and to send them to a remote platform to enable constant monitoring and advanced analytics on patients data
- **Four logical layers**
  - Wearable sensors layers (e.g. sensors applied onto a t-shirt)
  - Nodes receiving sensors data
  - A board hosting the nodes and operating as a gateway → Raspberry Pi3 running a custom Linux distro
  - The remote application (Web-based API)
- **Actors**
  - Patients
  - Medical Staff



Step #1 – Understanding how the system works

- In order to start modeling threats, it is first important to understand how the system works:
  - Who are the **actors**
  - Which are the **components** involved

- **E.g. User stories** can be used (other requirements gathering/elicitation approaches are fine as well)
  - **US#1. Patient Registration**
    - The **patient** in the ward receives from the **hospital staff** the **monitoring node**, then wears and activates it. The **hospital staff** inserts in the system the patient's personal data, anamnestic data, other relevant information, and **link the activated device** to the **patient.** The **hospital staff** repeats the procedure for **non-wearable devices**. **The system** is ready to use and **starts collecting data.**
  - **US#2. Ordinary data acquisition**
    - The **patient** stays in the ward. The **system** collects data, activating the transmission from **non-wearable devices** when the **patient** interacts with **a sensor** (for example, a pressure sensor on a bed triggers the transmission when he is lying down/rising up), while the collection of data for the **wearable node** is done with a temporal resolution and a level of detail based on the connection and the type of (critical/non-critical) monitoring.

Step #2 - Representing the System with Data Flow Diagrams

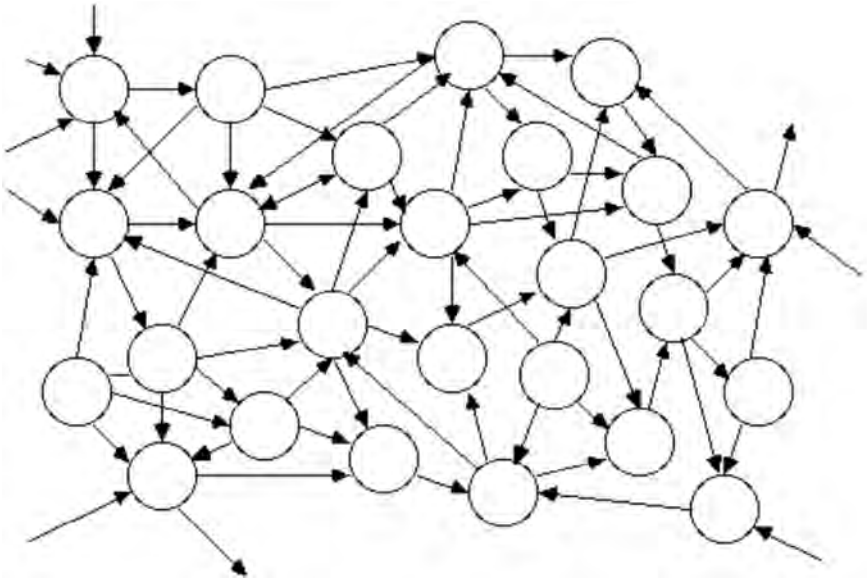
- **Data Flow Diagrams**[1] **(DFDs)** are a modelling tool (defined in the context of systems engineering) that allows to highlight the functional processes of a system, with a particular focus on showing what and where the data flows move.

- The main parts of DFDs are the following:
  - **Entities** (rectangle): external components that interact with the system.
    - In this specific example, and since it helps in highlighting several data flows, some internal components (such as sensor nodes) are considered as Entities;
  - **Process** (circle): a component that transforms an input stream into a different output; the process name indicates the action accomplished;
  - **Data flow** (line): the data moving from a component (an entity, a process) to another;
  - **Data store** (rectangle without side edges): a database, file or similar;
  - **Authentication** (red dashed line): It summarizes the various exchanges of data typical of the authentication process.

[1]E. Yourdon, Just Enough Structured Analysis, Chapter 9., 2006

Step #2 - Representing the System with Data Flow Diagrams

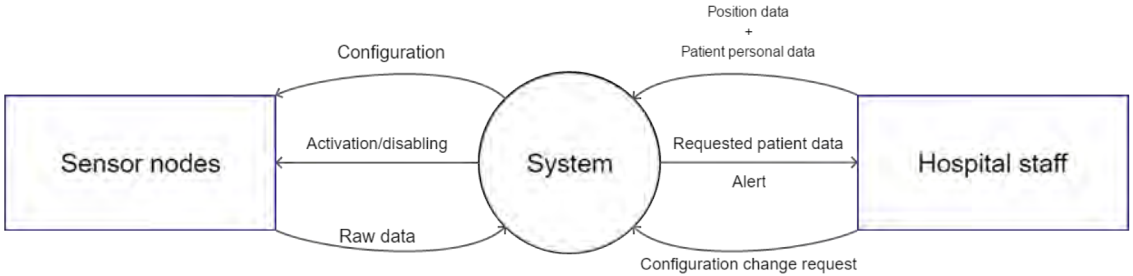
- If the system is intrinsically complex and has dozens or even hundreds of functions to model, the risk is to have a DFD like the one below.
- **How to avoid it?**



Step #2 - Representing the System with Data Flow Diagrams

- The answer is to organize the overall DFD in a series of levels so that each level provides successively more detail about a portion of the level above it.
  - This is analogous to the organization of maps in an atlas:
    - we would expect to see an overview map that shows us an entire country, or perhaps even the entire world;
    - subsequent maps would show us the details of individual countries, individual states within countries, and so on.

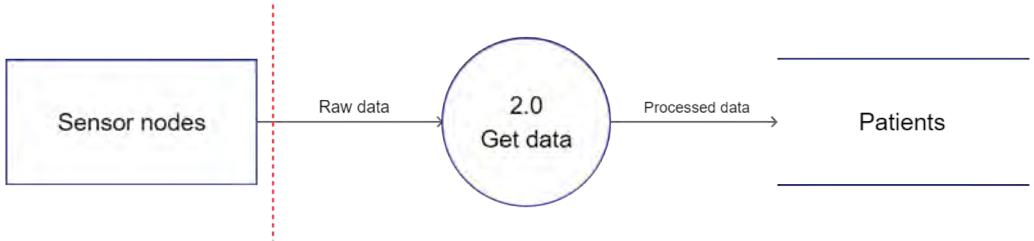
- The DFD representing the whole system is called «**context diagram**»



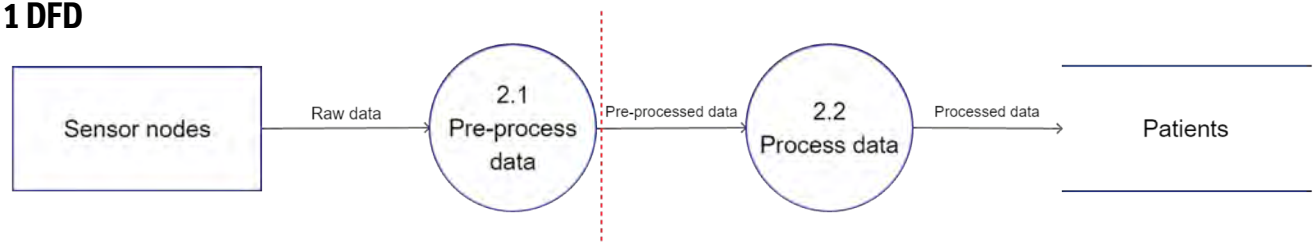
Step #2 - Representing the System with Data Flow Diagrams

- Then, separated DFD can be used to model the different subparts of the system
  - Different «levels» of the DFD can be defined, depending on the granularity of the representation

- **Level 0 DFD**



- **Level 1 DFD**



Step #2 - Representing the System with Data Flow Diagrams: final outcome

- What modelling the system with DFD allowed us to do, is:
  - To comprehensively **list** all the **assets** in the system
  - To comprehensively **list** all the **data flows** (data is also an asset, of course)
  - To represent **interactions** between the **Entities** (which are both human-beings as well as devices) and the **rest of the system**
    - While modelling the threats, we should in principle assume they are not trustworthy…

- **Now, it is time to list the threats affecting all of them**

ELEMENT TYPE	NAME (NUMBER)
Entity	Hospital staff Wearable sensor node Non-wearable sensor node Gateway
Process	Insert patient (1.1) Associate node (1.2) Create configuration (1.3) Pre-process data (2.1) Process data (2.2) View data (3.0) Update patient (4.1) Analyse data (5.0)
Data Store	Patients Nodes
Data Flow	Patient personal data Node configuration Configuration change request Node activation/disabling Position data Node association Raw data Pre-processed data Processed data Data visualization request Requested patient data Node data Alert

Step #3 – Applying the STRIDE Threat Modeling

- **STRIDE** is a TMM adopted from Microsoft since 2002 and was included into the Secure Development Lifecycle
- **STRIDE** is a mnemonic, where each letter represent a particular category of threat to model

Threat		Property Violated	Threat Definition
S	Spoofing identify	Authentication	Pretending to be something or someone other than yourself
T	Tampering with data	Integrity	Modifying something on disk, network, memory, or elsewhere
R	Repudiation	Non-repudiation	Claiming that you didn't do something or were not responsible; can be honest or false
I	Information disclosure	Confidentiality	Providing information to someone not authorized to access it
D	Denial of service	Availability	Exhausting resources needed to provide service
E	Elevation of privilege	Authorization	Allowing someone to do something they are not authorized to do

Step #3 – Applying the STRIDE Threat Modeling

- From the list of threats foreseen by the STRIDE methodology, it emerges that:
  - Not all the threats apply to all the element types
    - **E.g.1** Spoofing definitely apply to Entities and Processes, but hardly to Data Stores
    - **E.g.2** Elevation of privileges only applies to Processes
  - A matrix like the one below allows to do it

Element type	Spoofing (S)	Tampering (T)	Repudiation (R)	Information Disclosure (I)	Denial of Service (D)	Elevation of Privilege (E)
Entity	X		X			
Process	X	X	X	X	X	X
Data Store		X	X	X	X	
Data Flow		X		X	X	

Step #4 – Listing the actual Attack Patterns

- The mapping between the **Element types** and the **Attack categories** we did so far still does not highlight the real instances of attack

- Let's consider the Spoofing attack which applies to Entities
  - When it comes to Hospital staff, Spoofing means that somebody authenticated itself on the system using somebody else credentials
  - When it comes to sensors, it might mean that a rogue sensor has been connected to the system and is sending fake data

- Thus:
  - Conceptually the attack is similar (somebody or something is pretending to be somebody else)
  - The real **Attack Pattern** which is observed is of course completely different, and also is the **mitigation measure**
  - A great help in listing **AttackPatterns** is provided by the **CAPEC** (**C**ommon **A**ttack **P**attern **E**numeration and **C**lassification) framework - https://capec.mitre.org

CAPEC-151: Identity Spoofing

Attack Pattern ID: 151
Abstraction: Meta

Status: Stable

Presentation Filter: Basic

Description

Identity Spoofing refers to the action of assuming (i.e., taking on) the identity of some other entity (human or non-human) and then using that identity to accomplish a goal. An adversary may craft messages that appear to come from a different principle or use stolen / spoofed authentication credentials. Alternatively, an adversary may intercept a message from a legitimate sender and attempt to make it look like the message comes from them without changing its content. The latter form of this attack can be used to hijack credentials from legitimate users. Identity Spoofing attacks need not be limited to transmitted messages - any resource that is associated with an identity (for example, a file with a signature) can be the target of an attack where the adversary attempts to change the apparent identity. This attack differs from Content Spoofing attacks where the adversary does not wish to change the apparent identity of the message but instead wishes to change what the message says. In an Identity Spoofing attack, the adversary is attempting to change the identity of the content.

Relationships

The table(s) below shows the other attack patterns and high level categories that are related to this attack pattern. These relationships are defined as ChildOf, ParentOf, MemberOf and give insight to similar items that may exist at higher and lower levels of abstraction. In addition, relationships such as CanFollow, PeerOf, and CanAlsoBe are defined to show similar attack patterns that the user may want to explore.

Nature	Type	ID	Name
MemberOf	C	156	Engage in Deceptive Interactions
MemberOf	C	403	Social Engineering
MemberOf	C	512	Communications
MemberOf	C	513	Software
ParentOf	S	89	Pharming
ParentOf	S	98	Phishing
ParentOf	S	194	Fake the Source of Data
ParentOf	S	195	Principal Spoof
ParentOf	S	473	Signature Spoof

Prerequisites

The identity associated with the message or resource must be removable or modifiable in an undetectable way.

Mitigations

Employ robust authentication processes (e.g., multi-factor authentication).

CAPEC-154: Resource Location Spoofing

Attack Pattern ID: 154
Abstraction: Meta

Status: Stable

Presentation Filter: Basic

Description

An adversary deceives an application or user and convinces them to request a resource from an unintended location. By spoofing the location, the adversary can cause an alternate resource to be used, often one that the adversary controls and can be used to help them achieve their malicious goals.

Relationships

The table(s) below shows the other attack patterns and high level categories that are related to this attack pattern. These relationships are defined as ChildOf, ParentOf, MemberOf and give insight to similar items that may exist at higher and lower levels of abstraction. In addition, relationships such as CanFollow, PeerOf, and CanAlsoBe are defined to show similar attack patterns that the user may want to explore.

Nature	Type	ID	Name
MemberOf	C	156	Engage in Deceptive Interactions
MemberOf	C	403	Social Engineering
MemberOf	C	512	Communications
MemberOf	C	513	Software
MemberOf	C	515	Hardware
ParentOf	S	159	Redirect Access to Libraries
ParentOf	S	616	Establish Rogue Location

Prerequisites

None. All applications rely on file paths and therefore, in theory, they or their resources could be affected by this type of attack.

Mitigations

Monitor network activity to detect any anomalous or unauthorized communication exchanges.

To recap:

- During the past 25 minutes we described a very simplified approach to threat modelling, organised in **4** steps
  - **STEP 1 – Understand the System (e.g. User Stories)**
  - **STEP 2 – Conceptualize and Represent the System (e.g. Data Flow Diagrams)**
  - **STEP 3 – Map the threats on the assets (using STRIDE)**
  - **STEP 4 – Identify the real Attack Patterns (using the CAPEC framework) and the corresponding mitigation measures**
- In a real scenario, it would be also useful to organize the measures in a **mitigation plan (STEP 5)** (and of course to implement it).

- **Regarding the mitigation plan:**
  - Sometimes (Often…), the mitigation plan is implemented «informally»
  - A cost & benefit analysis should be performed
  - The sooner the threat modelling is performed, the smaller the cost of mitigation[1]
    - Mitigating a threat identified in the design phase usually costs much less then doing it when the system is already in production

[1]Gary Mc. Graw, Software Security, O'Really, 2006

Beyond STRIDE and the (toy) example provided

- The one we have (quickly) seen today is actually not the only approach to Threat Modelling

- At least 12 different Threat Modelling methodologies do exist in the literature[1], each one with its own focus:
  - **STRIDE** (and Associated Derivations)
  - **PASTA**
  - **LINDUUN**
  - **CVSS**
  - **Attack Trees**
  - **Persona non Grata**
  - **Security Cards**
  - **hTMM**
  - **Quantitative Threat Modeling Method**
  - **Trike**
  - **VAST Modeling**
  - **OCTAVE**

[1]N. Shevchenko, T.Chick, P. O'Riordan, T.P. Scanlon, C. Woody, *Threat Modeling: a Summary of Available Methods, 2018*

Differences Between Threat Modelling Methods

	Focus/Perspective	Easy to Use	Easy to Learn	Documentation	Automation	Mitigation
STRIDE	Defender	Medium	Medium	Very Good	Yes	Yes
PASTA	Risk	No	No	Very Good	No	Yes
LINDUUN	Assets/Data	No	Medium	Good	No	Yes
CVSS	Scoring	No	No	Good	Yes	No
Attack Trees	Attacker	Yes	Medium	Good	No	No
Persona non Grata	Attacker	Yes	Yes	Some	No	No
Security Cards	Attacker	Yes	Yes	Very Good	No	No
hTMM	Attacker/Defender	Medium	Medium	Good	No	No
Quantitative Threat Modeling Method	Attacker/Defender	No	No	Some	No	No
Trike	Risk	Medium	Medium	Good for v1	No	Yes
VAST Modeling	Attacker	Medium	Medium	Very Good	Yes	Yes
OCTAVE	Risk/Organization	No	No	Good	No	Yes

[1]N. Shevchenko, B.R. Frye, C. Woody, *Threat Modeling for Cyber-Physical System-of-Systems: Methods Evaluation, 2018*

Persona non Grata

- As a threat modeling method, Persona non Grata (PnG) focuses on the motivations and skills of hu- man attackers.
  - It characterizes users as archetypes that can misuse the system and forces analysts to view the system from an unintended use point of view
- Tends to detect only a certain subset of threat types
- This technique fits well into the agile approaches, which incorporates personas.

CVSS – Common Vulnerability Scoring System

- Often used in combination with other methods, the Common Vulnerability Scoring System (CVSS) is a method that "capture[s] the principal charac- teristics of a vulnerability, and produce[s] a numerical score reflecting its severity".
  - A CVSS score is computed based on values assigned by an analyst for each metric (an online calculator is available)
  - Possible inconsistencies produced by different judging «experts»

<table>
<tr><th colspan="2">Base Metric Group</th><th>Temporal metric Group</th><th colspan="2">Environmental Metric Group</th></tr>
<tr><td><u>Exploitability Metrics</u></td><td><u>Impact Metrics</u></td><td rowspan="2">Exploit Code Maturity</td><td rowspan="5">Modified Base Metrics</td><td>Confidentiality Requirements</td></tr>
<tr><td>Attack Vector</td><td>Confidentiality Impact</td><td rowspan="2">Integrity Requirement</td></tr>
<tr><td>Attack Complexity</td><td>Integrity Impact</td><td rowspan="2">Remediation Level</td></tr>
<tr><td>Privileges Required</td><td>Availability Impact</td><td rowspan="2">Availability Requirement</td></tr>
<tr><td>User Interaction</td><td></td><td rowspan="2">Report Confidence</td></tr>
<tr><td colspan="2">Scope</td></tr>
</table>


THANKS
Come AGAIN

Questions?

Pluribus One S.r.l.
Via Vincenzo Bellini 9, Cagliari (CA), Italy
Via Emilio Segrè, 17, Elmas (CA), Italy

info@pluribus-one.it
www.pluribus-one.it

@pluribus_one

davide.ariu@pluribus-one.it

https://www.linkedin.com/in/davidearIu/

@davidearIu