Network Protection for Energy Control Systems

CSP004_C_E

PRESENTATION BY:

- **CRISTINA ALCARAZ**
  UNIVERSITY OF MALAGA, SPAIN

Acknowledgement

- *Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.*
- *Project Agreement no. 101083594*

Topic-4: Advanced Protection for Energy Control Networks

Overview

- Intrusion detection techniques and systems
- Advanced monitoring systems
- Final remarks
- References and sources

Topic-4: Advanced Protection for Energy Control Networks

Overview

- **Intrusion detection techniques and systems**
- Advanced monitoring systems
- Final remarks
- References and sources

Intrusion detection in power control networks

- The National Institute of Standards and Technology (NIST) defines '**intrusion detection**' as:
  - "*The process of monitoring the events occurring in a computer system or network and analysing them for signs of possible incidents*"
  - "*A security service that monitors and analyses network or system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner*" – NIST SP 800-82 Rev 2 about "Guide Industrial Control Systems (ICS) Security"
  - "*Sensing and analysing system events for the purpose of noticing (i.e., becoming aware of) attempts to access system resources in an unauthorized manner*" - RFC 4949
- The European Union Agency for Cybersecurity (ENISA) in "*Appropriate security measures for smart grids*" also considers the importance of '**prevention/detection**' as part of incident management processes (SM5.1) and secure networks communications (SM10.2) in Smart Grids

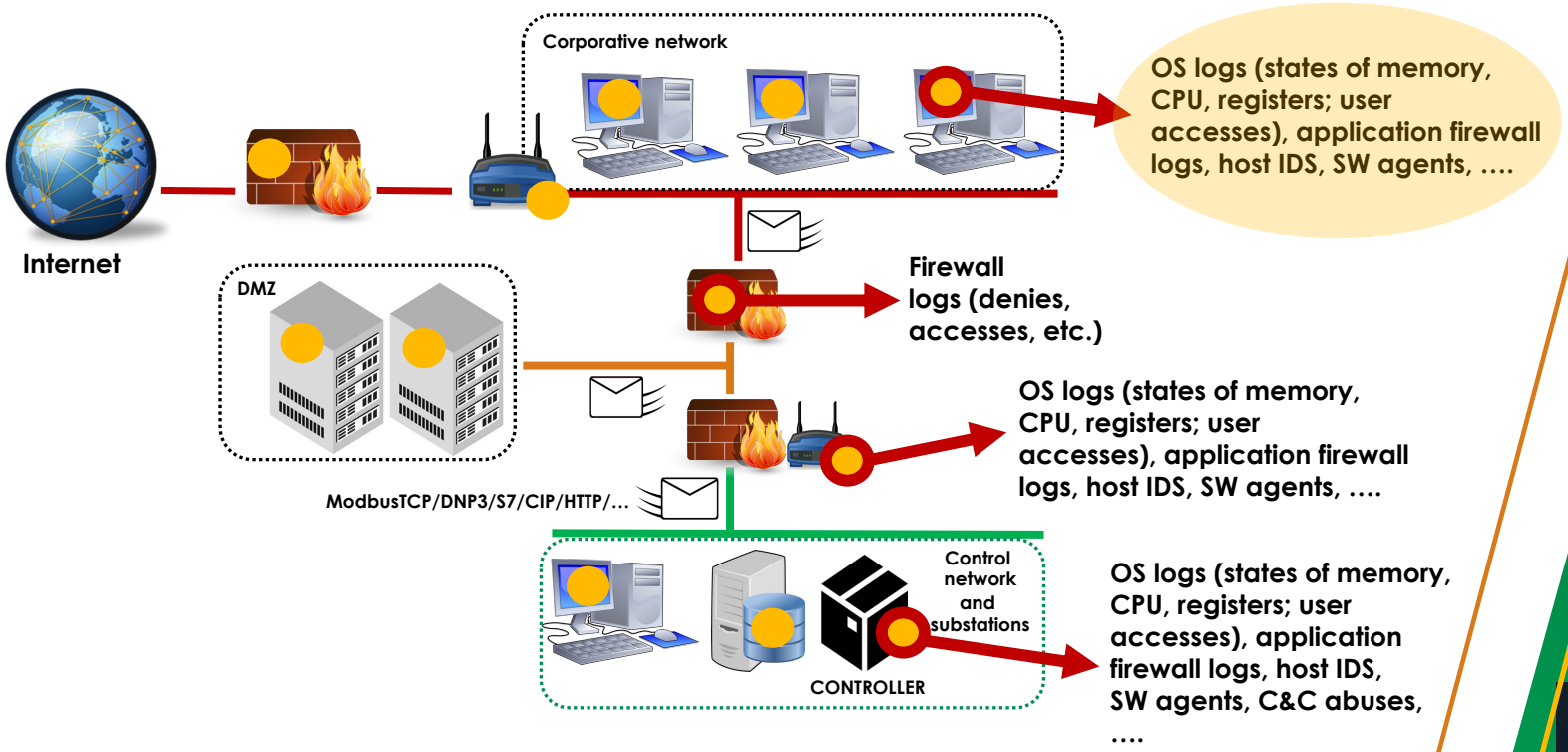
As can be appreciated, these three definitions have in common the words: '**monitoring**' and '**analysis**' **of events**, both at the network level and at the endpoint level (e.g. HMIs, controllers, sensors, servers, gateways, routers, etc.).

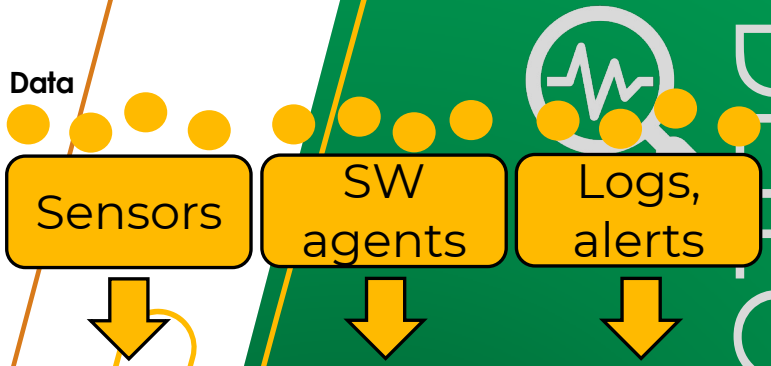
SM10.2: explicitly "*Where technically feasible, intrusion detection systems and intrusion prevention systems (signatures based or behavioural) are implemented on smart gird information systems (including SCADA components) and/or network (especially in field devices networks)*"

Source: NIST, "Intrusion detection", Computer Security Resource Centre, 2024
URL: https://csrc.nist.gov/glossary/term/intrusion_detection
Source: NIST, "Guide Industrial Control Systems (ICS) Security", 2023
URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf
Source: IETF, Internet Security Glossary, 2007
URL: https://datatracker.ietf.org/doc/html/rfc4949
Source: ENISA, "Appropriate security measures for smart grids. Guidelines to assess the sophistication of security measures implementation", 2012
URL: https://www.enisa.europa.eu/publications/appropriate-security-measures-for-smart-grids

Detection components

- The **main components** of an IDS are:
  - **Data sensing sources**:
    - Goals: to perceive data for detection and decision making
    - Resources: sensors (e.g. physical sensors, smart meters), software agents, logs (firewalls, OS, accesses…), alerts, etc.

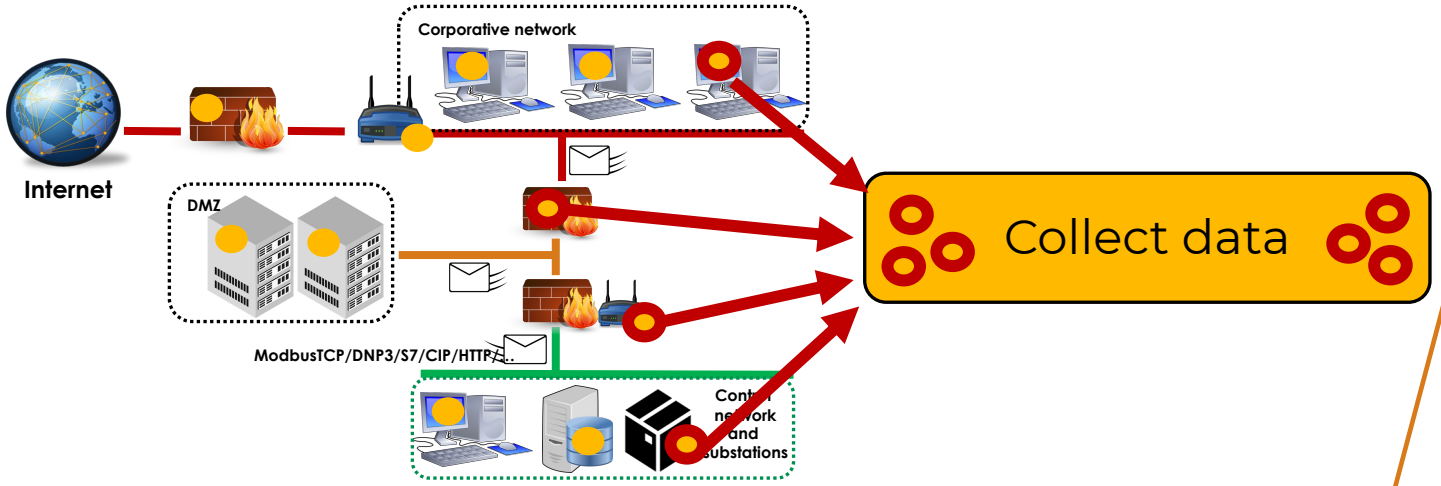


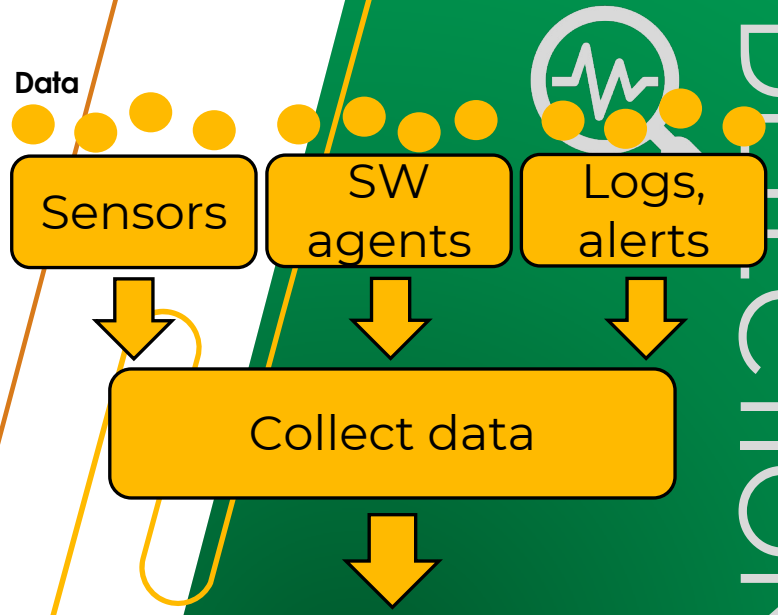


DETECTION TECHNIQUES

Detection components

- The **main components** of an IDS are:
  - **Data sensing sources**:
    - Goals: to perceive data for detection and decision making
    - Resources: sensors (e.g. physical sensors, smart meters), software agents, logs (firewalls, OS, accesses…), alerts, etc.
  - **Data collection sources**:
    - Goal: to collect data from perception sources for detection and decision making
    - Resources: centralised, distributed or decentralised systems

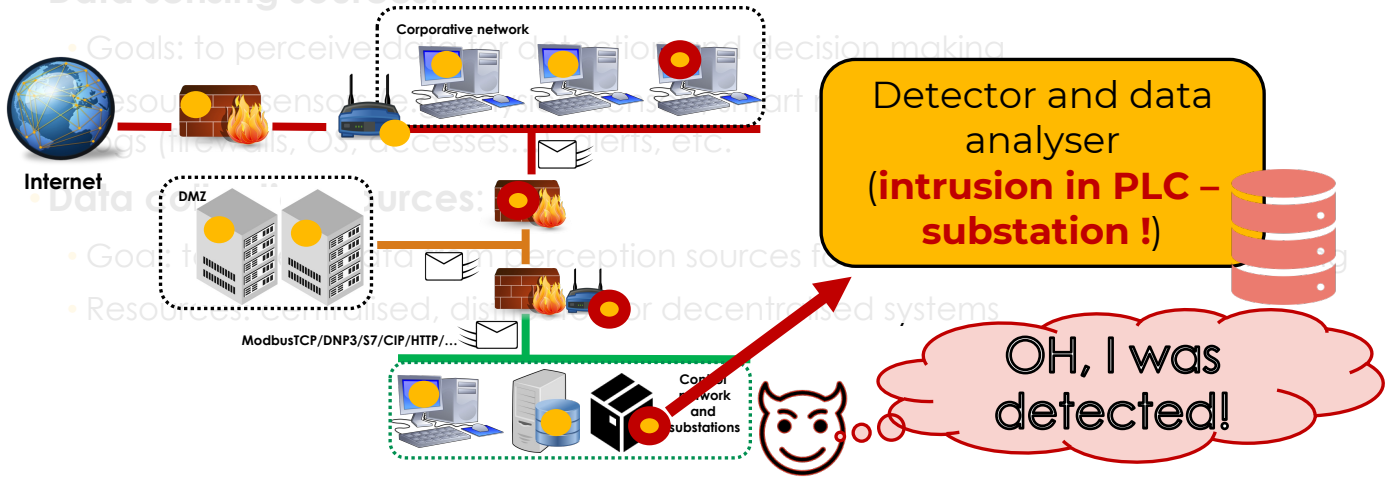




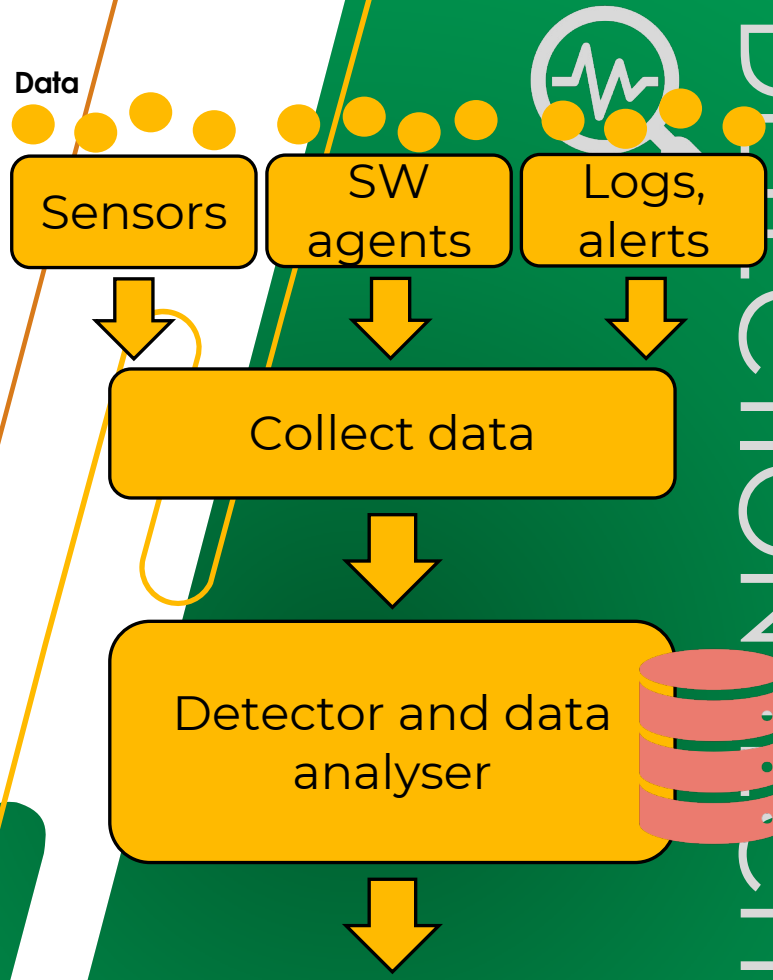
DETECTION TECHNIQUES

Detection components

- The **main components** of an IDS are:



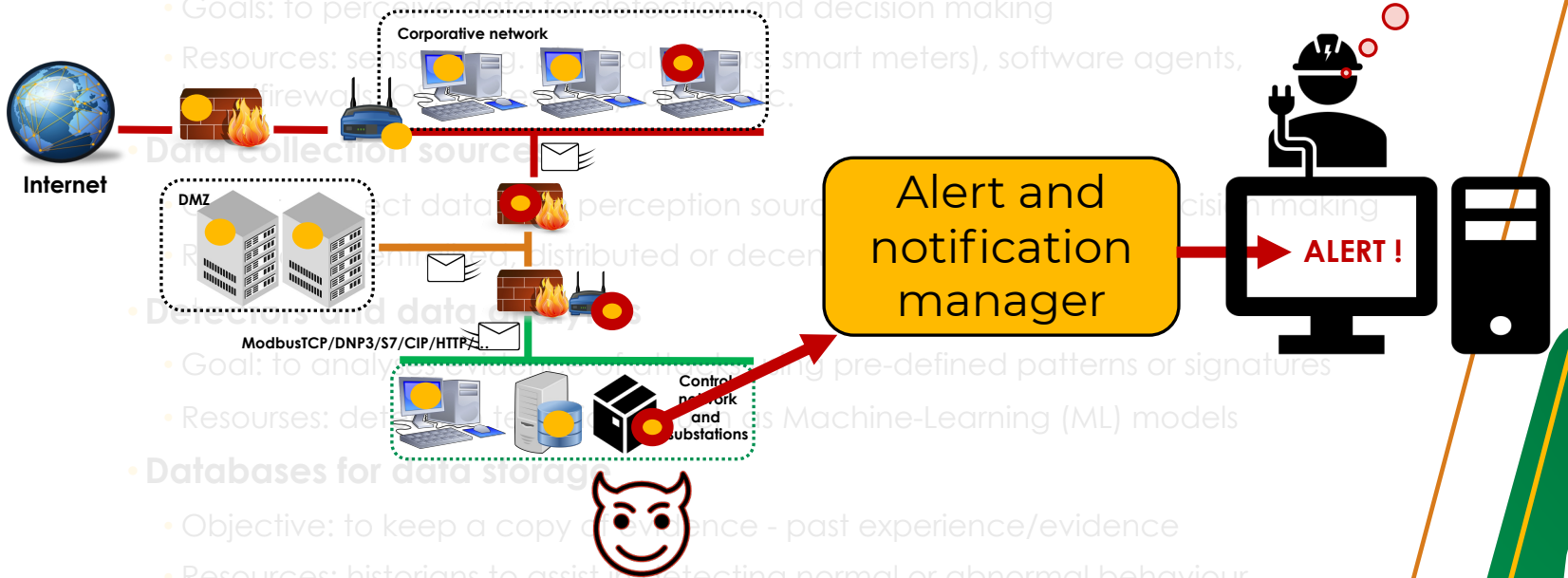
- **Detectors and data analytics**
  - Goal: to analyses evidence of attacks using pre-defined patterns or signatures
  - Resourses: detection techniques such as Machine-Learning (ML) models
- **Databases for data storage**
  - Objective: to keep a copy of evidence - past experience/evidence
  - Resources: historians to assist in detecting normal or abnormal behaviour



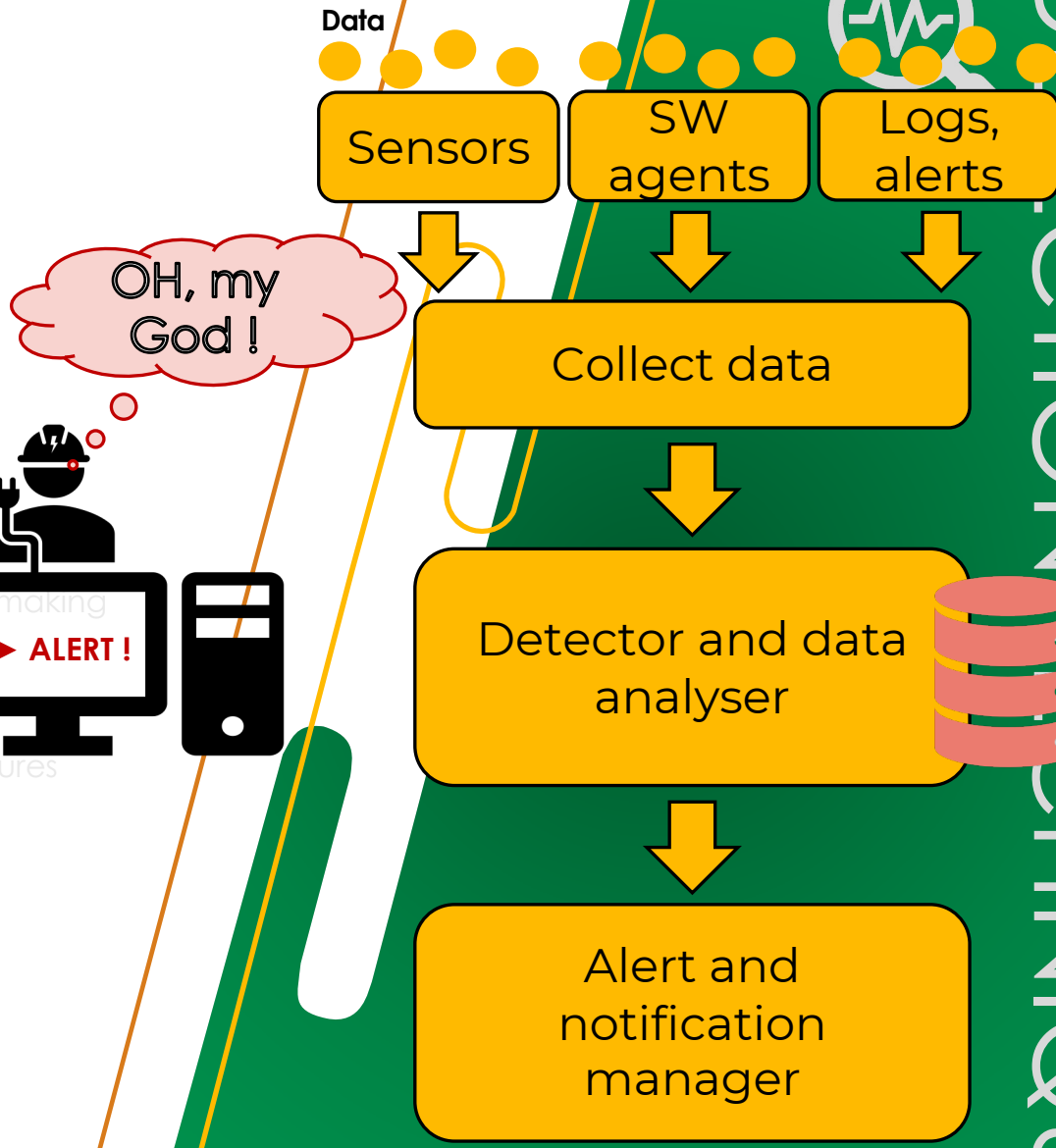
DETECTION TECHNIQUES

Detection components

- The **main components** of an IDS are:
  - **Data sensing sources:**
    - Goals: to perceive data for detection and decision making
    - Resources: sensors (e.g. alarms, smart meters), software agents, firewalls, logs, etc.
  - **Data collection sources**
    - Goal: to collect data from perception sources for decision making
    - Resources: centralized, distributed or decentralized
  - **Detectors and data analysers**
    - Goal: to analyse data using pre-defined patterns or signatures
    - Resourses: detection techniques such as Machine-Learning (ML) models
  - **Databases for data storage**
    - Objective: to keep a copy of evidence - past experience/evidence
    - Resources: historians to assist in detecting normal or abnormal behaviour

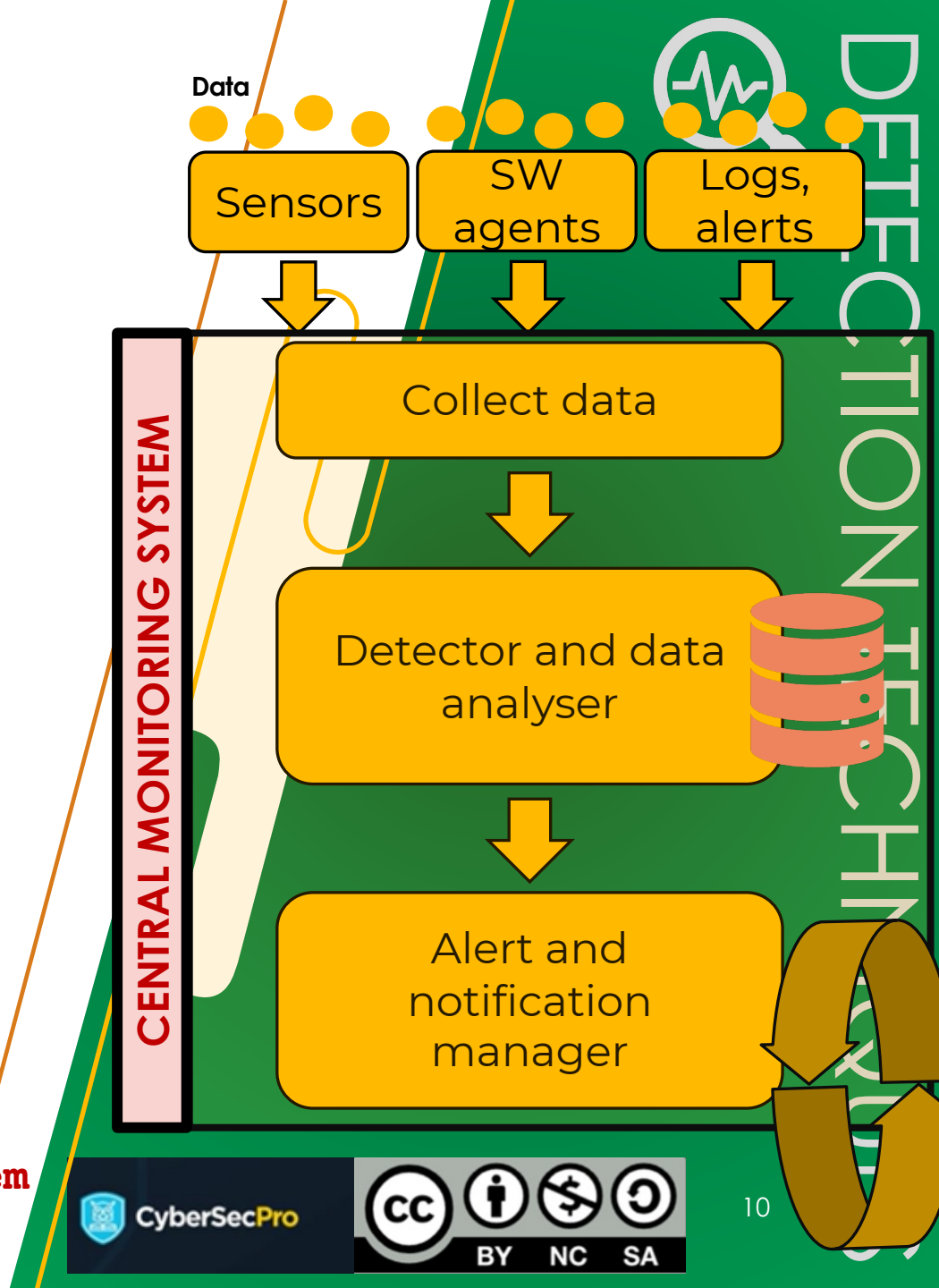


- **Alert and notification managers**
  - Objective: to notify the respective managers of the situation.
  - Resource: alarms and reports (via SMS, email, specific app, etc.)



Detection components

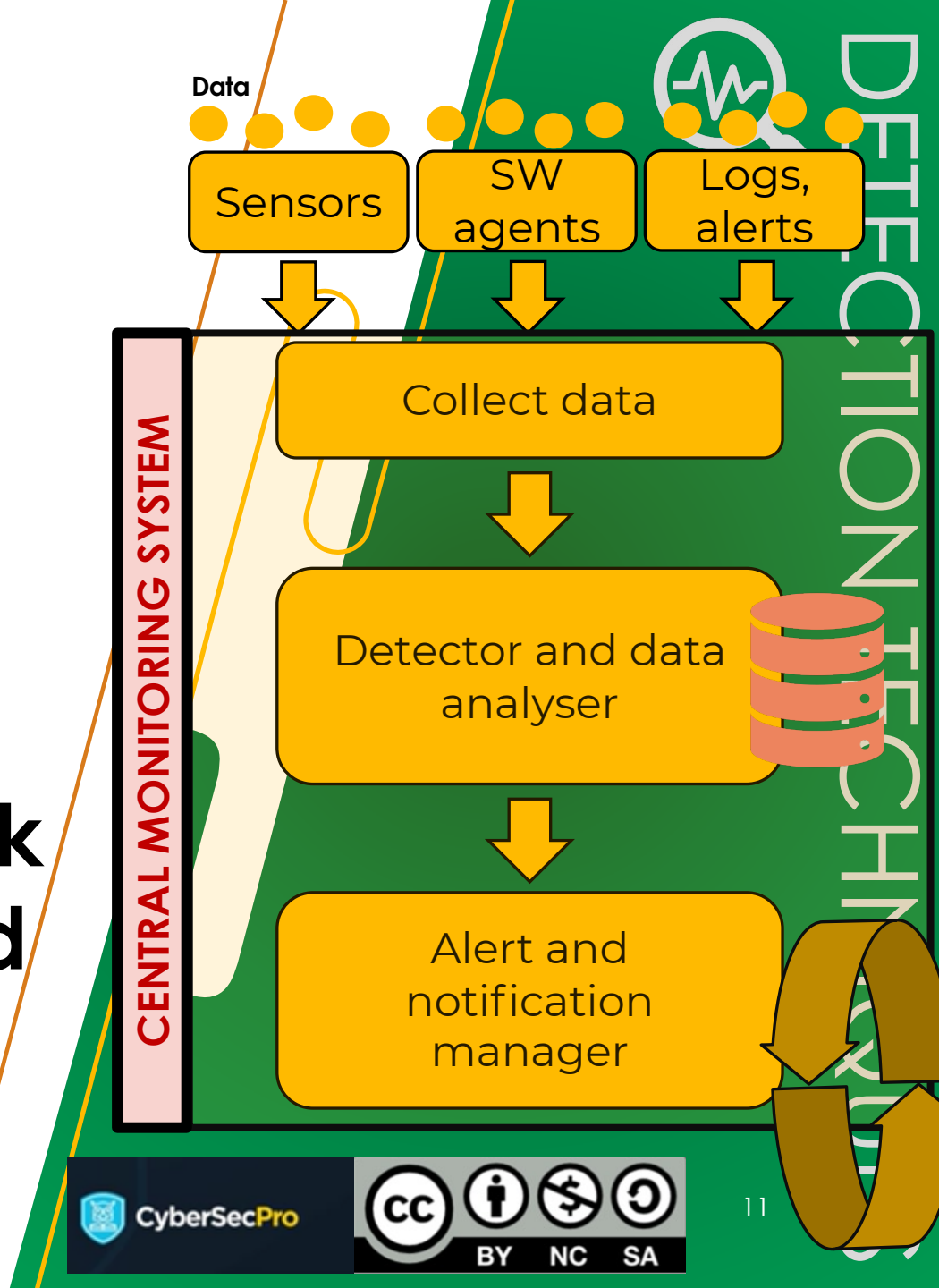
- The **main components** of an IDS are:
  - **Data sensing sources**:
    - Goals: to perceive data for detection and decision making
    - Resources: sensors (e.g. physical sensors, smart meters), software agents, logs (firewalls, OS, accesses…), alerts, etc.
  - **Data collection sources**:
    - Goal: to collect data from perception sources for detection and decision making
    - Resources: centralised, distributed or decentralised systems
  - **Detectors and data analytics**
    - Goal: to analyses evidence of attacks using pre-defined patterns or signatures
    - Resourses: detection techniques such as Machine-Learning (ML) models
  - **Databases for data storage**
    - Objective: to keep a copy of evidence - past experience/evidence
    - Resources: historians to assist in detecting normal or abnormal behaviour
  - **Alert and notification managers**
    - Objective: to notify the respective managers of the situation.
    - Resource: alarms and reports (via SMS, email, specific app, etc.)
- RESULT: **an automatic centralised monitoring system**



Detection components

Therefore … what are we *'looking for'* with this detection?

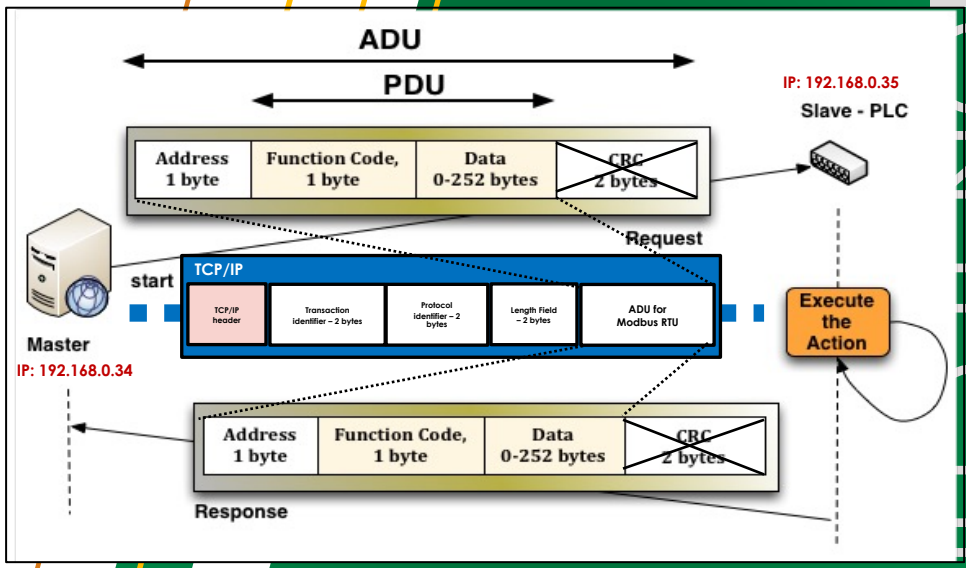
Clear: Anything that shows signs of intrusion or cyber-attack against the control network and its main resources!



Symptoms of intrusion in control networks

- Some symptoms of intrusion in control networks and substations may, for example, be:

IT networks	OT networks
Replay packets, selective forwarding, etc.	Abuses of C&C instructions to a particular operational device
Datagram malformations	Unexpected timing packets
Abusive port scanning	CPU overflow, memory overflow, or interruption of processes or services in the controllers, HMIs or SCADA servers
Anomalous packets with false messages, modifications to packets or irregularities in the fields	Invalid C&C instructions, e.g., invalid function codes
CPU overflow, memory overflow, or interruption of processes or services in servers or routers	Desynchronisation in the control (between the master and the slave)
Irregularity in data processing and management times	Irregularity of the packets and fields
…	…



DETECTION TECHNIQUES

Accuracy and techniques of detection

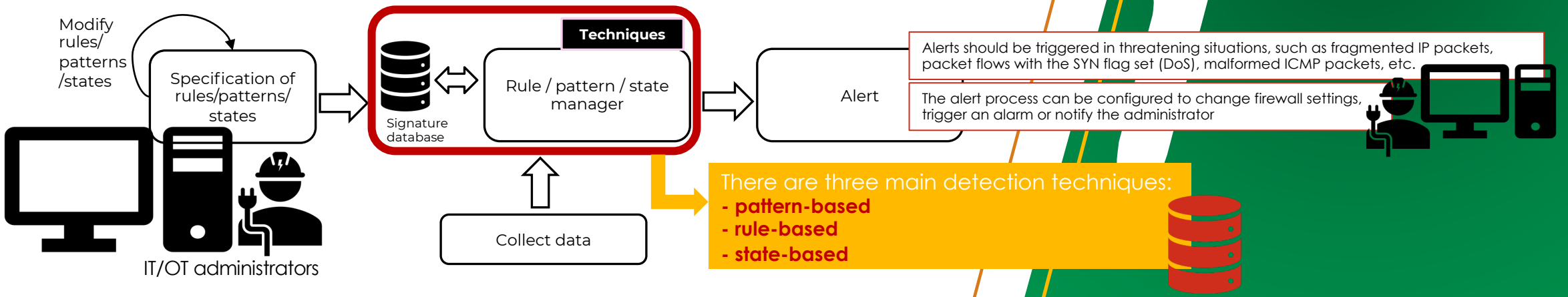
- The level of accuracy of IDSs can be diverse:

Response of the IDS	True (must be high)	False (must be low)
POSITIVE	True-Positive	False-Positive
NEGATIVE	True-Negative	False-Negative

  - A properly configured IDS will produce:
    - High number of true positives and true negatives, and
    - Low number of false positives and false negatives
- These levels of accuracy depend on the techniques applied, such as:
  - **Signature-based IDS**
  - **Specification-based IDS**
  - **Statistical anomaly / Anomaly-based IDS**
  - **Hybrid IDS**

Signature-based IDS

- This technique protects against known **WELL-KNOWN** threats
  - It is the simplest way to detect intrusion signs
- Its approach is mainly based on **'attack signatures'**
  - That is, attack patterns that correspond to a "known" threat
  - Therefore, they must be specified by IT-OT administrators according to known attack vectors
- As discussed above, the model is typically based on four essential components



Signature-based IDS

Pattern-based	Rule-based	Based on states

• They look for attack patterns according to packet headers and their contents, or according to system calls and based on logs	• Definition of attack rules:
	  • *If this AND this AND this occur → signature 1*
	  • *If this AND this AND this occurs → signature 2*
	• The resulting system is usually an expert system
		• It is based on the definition of expressions that determine the system state and state transitions
		• Example Petri nets:
		  • *If I am in state 1 (at t=1 and related to the login failure) AND I am in state 1 again (at t=2) → brute force*

• Advantages: fast pattern-matching (search in database)	• Advantages: fast rule matching (search in database)	• Advantages: fast event management
Disadvantages: • High level of knowledge to define rules • Regular maintenance of the rules and their databases, as well as the maintenance of attack signatures • Not accurate in the face of new attacks (**zero-days / APTs**)		Disadvantages: • Maintenance of signature databases • Not accurate in the face of new attacks (**zero-days / APTs**)

A major drawback for energy control networks where the number of APTs increases considerably

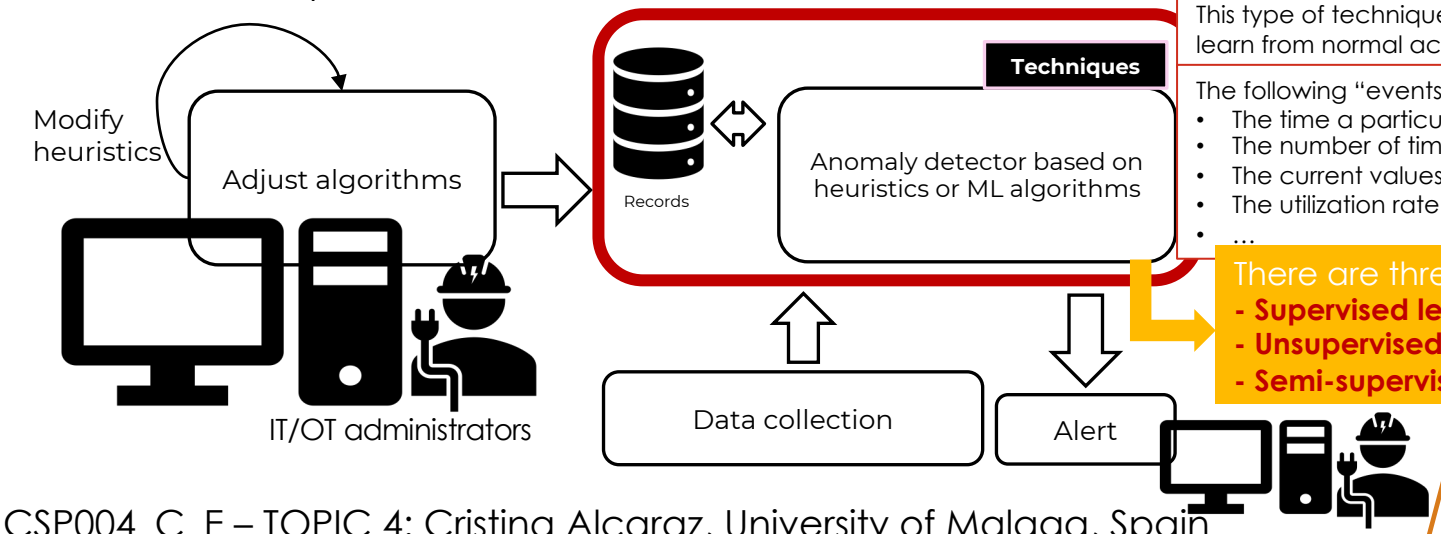
Specification-based IDS

- A specification-based IDS is also based on attack signatures, but these signatures are specific to the type of hardware/software specification, such as:
  - Type of protocol (Modbus/TCP, DNP3.0, IEC-101, IEC-104…)
  - Type of devices (sensor, HMI, controller, …)
  - Type of Operating Systems (Windows 10/11, Windows XP, Kali Linux, Parrot, Fedora, …)
  - …
- Therefore, this technique is equivalent to a signature-based IDS, and in terms of:
  - Functionality
  - Required maintenance of "patterns / rules" in databases (the signatures)
  - **Susceptibility to unknown threats, such as zero-days attacks and APTs**

Anomaly-based IDS

- When there is no guarantee of a continuous update of attack signatures or the system is critical, it is advisable to apply anomaly-based IDSs
- This type of IDS aims to **identify significant deviations** from the "normal behavior" of an IT/OT device or a network:
  - Everything that is not "normal" is classified as "anomalous", but
    - *What is normal or anomalous ?*
    - *Where are the limits of normality? – a causal noise is threatening event ?*
  - The classification is based on heuristics or rules, rather than signatures / attack patterns

This is something that may occur more intensely in operational control environments, as the complexity and heterogeneity of the (IT-OT) environment may not help in the detection process



This type of technique may require the IT/OT administrator to put the IDS into "learning mode" so that it can learn from normal activity

The following "events" are some examples of quantifiable events that can be used to detect anomalies:
- The time a particular event occurs
- The number of times a particular event occurs in a period of time
- The current values of system variables
- The utilization rate of system resources
- …

There are three main detection techniques:
- Supervised learning
- Unsupervised learning
- Semi-supervised learning

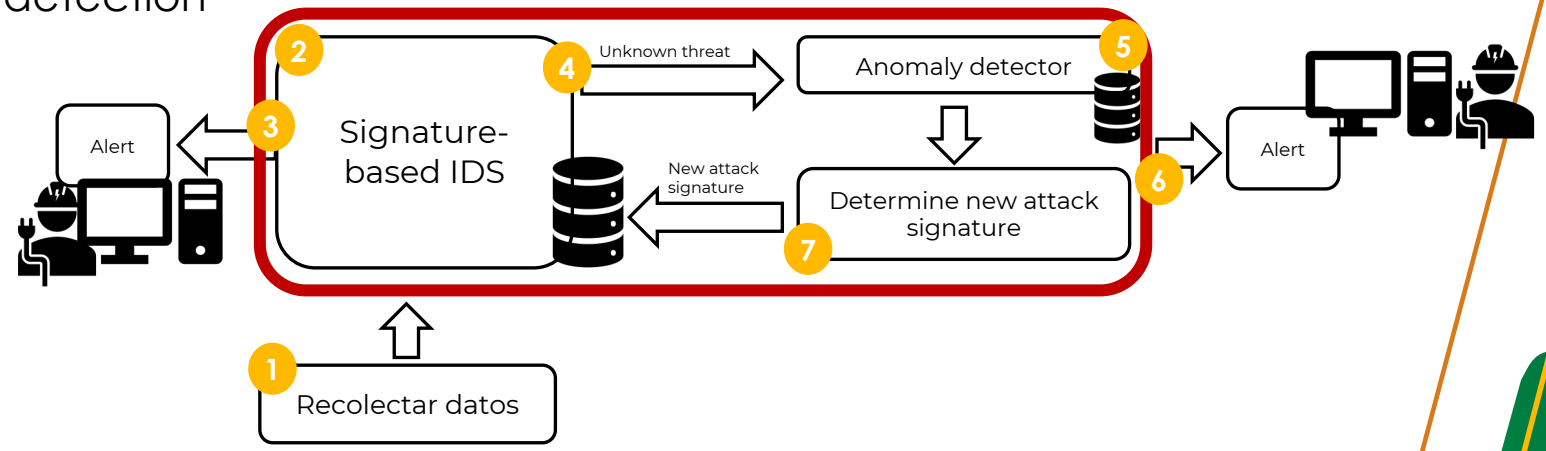
DETECTION TECHNIQUES

Anomaly-based IDS

Supervised learning	Unsupervised learning	Semi-supervised learning
• Algorithms that learn from both values and relationships and according to a set of desired or pre-determined examples/samples • Training phase required to associate input values with output values - *labelling* • Examples of algorithms: classification (e.g. decision trees, by distances to centroid, Support Vector Machine (SVM), ...) and regression	• Algorithms that increase their knowledge according to the available data/samples. For example, by applying similarity-based or association-based technique • No training phase required • Examples of algorithms: clustering (e.g. k-means), association, ...	• Consists of a mixture of supervised and unsupervised ML models • A training phase is required for *labelling*, but not necessarily for all input values
• Advantages: increased accuracy in the detection phase	• Advantages: more autonomy and less dependence on humans	• Advantages: the advantages of the two previous ones
• Disadvantages: Dependence on a training phase and on human supervision	• Disadvantage: less accurate detection	• Disadvantages: the disadvantages of the two previous ones

Hybrid-based IDS – the ideal approach to energy control systems

- Hybrid-based IDSs combine detection techniques (anomaly-based and signature/specification-based) to find the most optimal detection



FEATURES	Signature-based IDS	Anomaly-based IDS	Hybrid IDS
Advantages	• **Low false positive rate**, as there is the ability to **detect known attacks**, as long as the rules are correctly defined • **Fast detection** if rules are well defined	• **Low false negative rate**, as there is capacity to **detect unknown or novel attacks** → **APT or zero-days attacks**	• **Reduces both false positive rate and false negative rate**
Disadvantages	• Maintenance of detection rules • Maintenance of attack signature databases • **High false negative rate**; therefore, there is a high inability to detect unknown attacks if the signature databases are not updated → **APT or zero-days attacks**	• Maintenance of historical records • Adjustment of models/detection techniques • **High false positive rate**; therefore, it is necessary to be precise in the technique to avoid constant false alarms	• Maintenance of detection rules • Maintenance of attack signature databases • Maintenance of logs • Adjustment of detection models/techniques

Data or attributes for detection and contextual awareness

- There are many **types of data and attributes** that facilitate detection:
  - **At network level - Network IDS (NIDS)**
  - **At host level – Host IDS (HIDS)**
- This data can come from many **IT-OT sources or elements**:
  - IT-OT nodes: IP, MAC, ports, industrial protocol, etc.
  - Operating Systems: Windows/Linux/Unix... and version
  - Types of applications used in the control scenario
  - Type of network and its characteristics: network size, number of hops (TTL), protocols, etc.
- With this data, we can **create specific logs** that record:
  - IPs (source and destination)
  - TCP/IP Protocols (TPC, UPD, ICMP...)
  - Operational protocols (ModbusTCP, DNP3, OPC-UA, S7, etc.)
  - Event types and alerts, including priorities and severities
  - Connection or session ID, open services, percentage of erros SYN, timestamp, …
  - User connected and authenticated
  - Number of bytes transmitted, memory consumed, CPU used …
- In turn, these data can be used to manage a **variety of statistics** related to:
  - The network and the content of the packets (e.g. TCP/IP-specific protocols or industrial protocols)
  - Active services in controllers, SCADA servers, HMIs, routers, etc.
  - The hosts including controllers, sensors and actuators

If all these data are well defined and clearly linked to each other, it is possible to explain an anomaly or a threatening situation by pointing out **WHO, WHERE, WHEN, HOW and WHAT**

All this information in turn helps to meet the objectives of **INDUSTRY 5.0** in terms of **RESILIENCE + HUMAN CENTRALITY**

DETECTION TECHNIQUES

Automatic capabilities and types of IDS

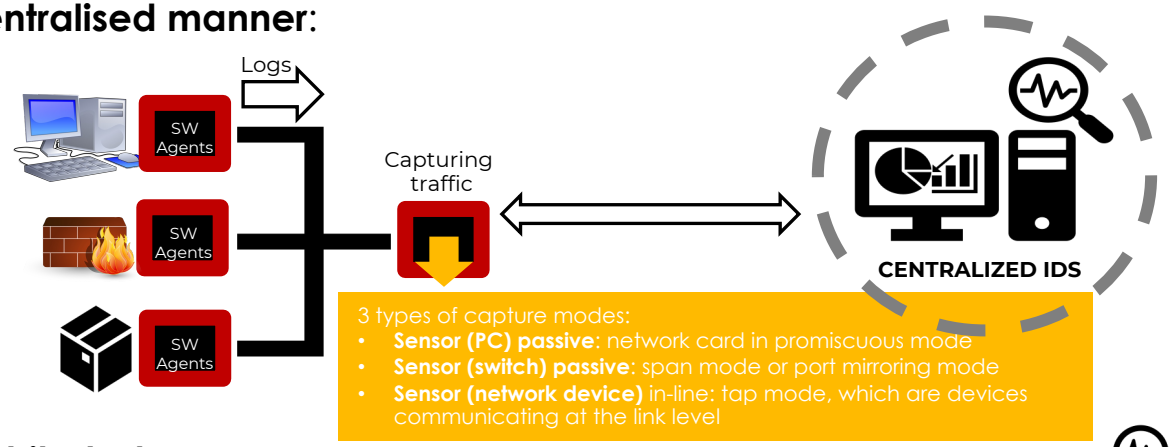
- What is more, based on the previous information, it is possible to **automate response actions** according to:
  - *Reaction thresholds*, which establish what is normal from what is not normal
    - Thresholds usually specify a maximum acceptable level, such as, for example: 5 failed connection attempts in 10 seconds, 5 file reads in 2 seconds, …
  - *Blacklist" and "whitelist"*
    - Each list can contain information about a specific node, such as IP, MAC, port, protocol, etc.
    - These lists are useful to manage the goodness level of a given node, which prevents the system from possible connections
- These automatic capabilities can be performed by:

Network-based IDS (NIDS)	Host-based IDS (HIDS)	Hybrid IDS
Monitors and analyses network traffic to detect anomalous actions or intrusion attempts	IDS that resides on the system and monitors it for anomalous actions or intrusion attempts	It is based on the combination of a host-based IDS and a network-based IDS, with their combined advantages and disadvantages as well
Its implementation requires a network card, SW agents or intermediary elements in the network to capture traffic passing through the network	• Its implementation requires SW agents capable of identifying any activity or event in the system • Make use of event logs such as EventLog Analyzer (Windows) or Sylog (Linux/Unix) to precisely capture and collect host-specific attributes	

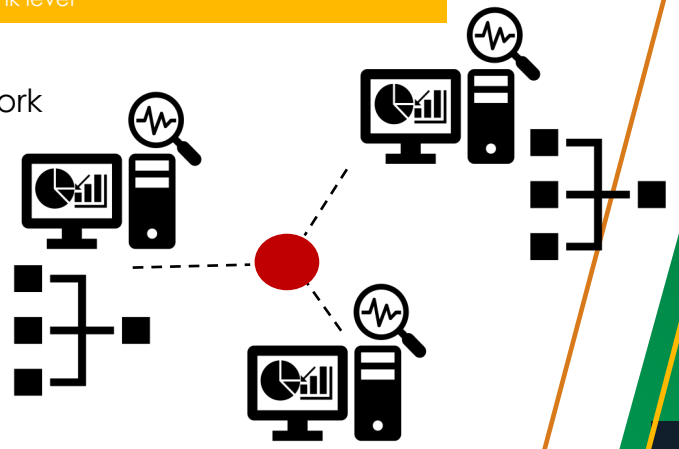
	HIDS	NIDS
Advantages	• They are able to manage and associate users and system events • They are able to analyse incoming and outgoing traffic from a host, including encrypted traffic	• They are capable of handling network traffic from more than one device • They are inexpensive to detect and are often easy to configure or deploy
Inconveniences	• The IDS is effective as long as the device is available or is not attacked by a DoS attack • May require local resources (e.g. databases/records) to detect anomalies or intrusion • Constant maintenance of IDS rules required	• **Not able to decrypt** encrypted network traffic • **High false positive rate** due to large number of data sources • Constant maintenance of IDS rules required • Difficulty in analysing (and in real time) collapsed networks or under DoS attacks

Automatic capabilities and types of IDS

- We have already seen that an IDS can be either a host or a network type, whose main nodes can collect and analyse data in:
  - **A centralised manner**:



  - **A distributed manner**:
    - There are several IDSs distributed in the network and generate events
    - The events can be sent to a central system, such as Security Information and Event Managements (SIEMs)
    - The central system correlates the events detected by the different IDSs and finally alerts the IT-OT administrator



Intrusion Detection Systems - TOOLS

IDS	Features
Snort	**NIDS, HIDS; https://www.snort.org/snort3**
Suricata	NIDS, HIDS; https://docs.suricata.io/en/latest/#
Zeek/Zeek-IDS/Bro-IDS	Network analysis framework; https://zeek.org
OpenWIPS-ng	Wireless IDS; https://openwips-ng.org
Security Onion	Monitoring solutions; https://securityonionsolutions.com
Hogzilla IDS	Network anomaly detection; https://ids-hogzilla.org
OSSEC	HIDS; https://www.ossec.net

- Snort is **a signature-based IDS** that can work as
  - HIDS when installed in a host
  - NIDS when installed in another location or filtering device in the network
- It is open-source NIDS/HIDS and can be used as IDPS (if it is in inline mode)

Intrusion Detection Systems - TOOLS

IDS	Features
Snort	NIDS, HIDS; https://www.snort.org/snort3
Suricata	**NIDS, HIDS; https://docs.suricata.io/en/latest/#**
Zeek/Zeek-IDS/Bro-IDS	Network analysis framework; https://zeek.org
OpenWIPS-ng	Wireless IDS; https://openwips-ng.org
Security Onion	Monitoring solutions; https://securityonionsolutions.com
Hogzilla IDS	Network anomaly detection; https://ids-hogzilla.org
OSSEC	HIDS; https://www.ossec.net

- The Suricata project and code is owned and supported by the Open Information Security Foundation (OISF)
- It is an **open-source signature-based NIDS/HIDS**
- Its engine is capable of managing:
  - Intrusion detection system (IDS)
  - Inline intrusion prevention (IPS)
  - Network security monitoring (NSM)
  - Offline .pcap processing
- With standard input and output formats
  - such as YAML and JSON
- It can be integrated with other monitoring tools such as SIEMs
  - such as Splunk, Logstash/Elasticsearch, Kibana, and others
- Its syntax is similar to SNORT !

SNORT: Configuration and detection

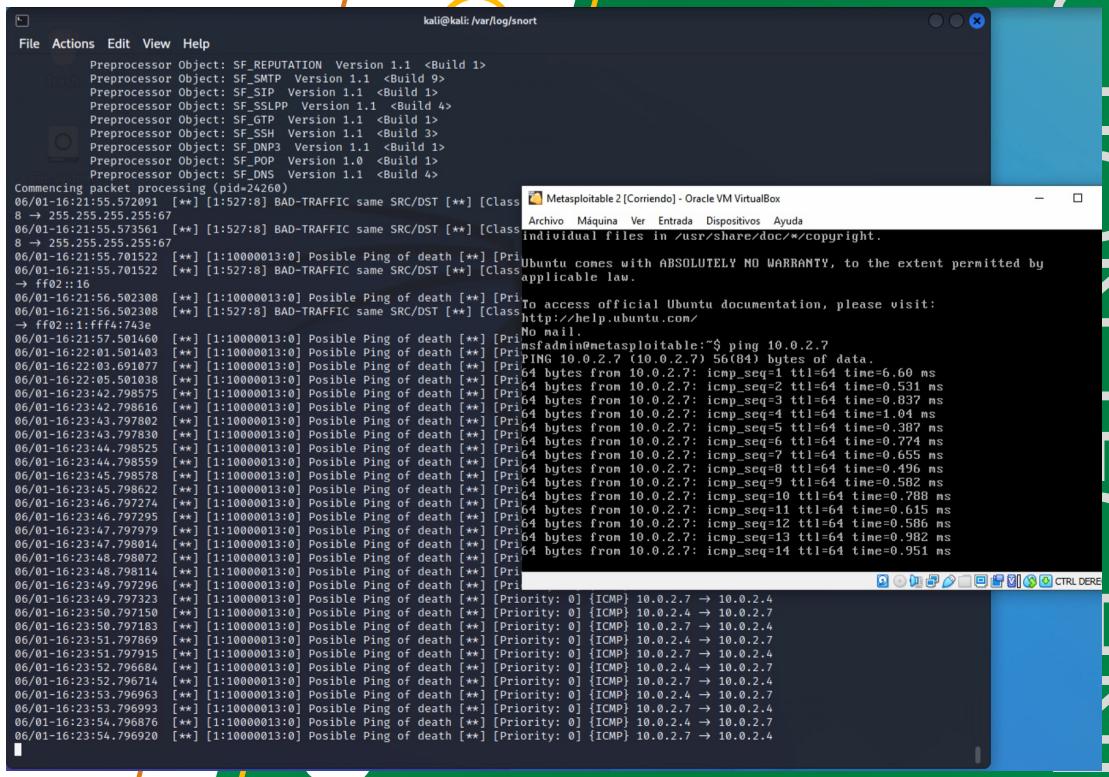
- **Installation on Linux:**
  - **$ apt-get update**
  - **$ apt-get install snort**
- **Snort configuration:**
  - Edit the file **/etc/snort/snort.conf** with nano or gedit
  - In this file, it is possible to configure the network variables, such as the IP of a network
    - **$HOME_NET**: represents the IP addresses of the internal network
    - **$EXTERNAL_NET**: represents the IP addresses of external networks
  - Specify the location of the rules
    - **var RULE_PATH .../rules**
  - Specify that they will be stored in local-rules:
    - ***# include $RULE_PATH/local.rules***
    - ***# include $RULE_PATH/app-detect.rules***
    - ***# include $RULE_PATH/attack-response.rules***
    - ***# include $RULE_PATH/backdoor.rules***
    - ***# include $RULE_PATH/bad-traffic.rules***
    - ...
- **Test and run Snort:**
  - **$ sudo service snort start / stop**
  - **$ sudo snort -T -c /etc/snort/snort.conf**

SNORT: Configuration and detection

- Run Snort without displaying alerts on screen, just log them to /var/log/snort/snort/snort.log.**XXXX**
  - **$ sudo snort -c /etc/snort/snort.conf -l /var/log/snort/**

- Run Snort displaying the alerts on screen and log them to /var/log/snort/snort/snort.log.**XXXX**
  - **$ sudo snort -d -l /var/log/snort/ -h 10.0.0.0/24 -A console -c /etc/snort/snort.conf**

- Important: Snort logs are stored encrypted. To view their contents, you must access the corresponding log (.XXXX)
  - **$ sudo snort -r /var/log/snort/snort.log.XXXX**



SNORT: Configuration and detection - *WHO, WHERE and WHAT*

- A rule in SNORT is represented as follows:

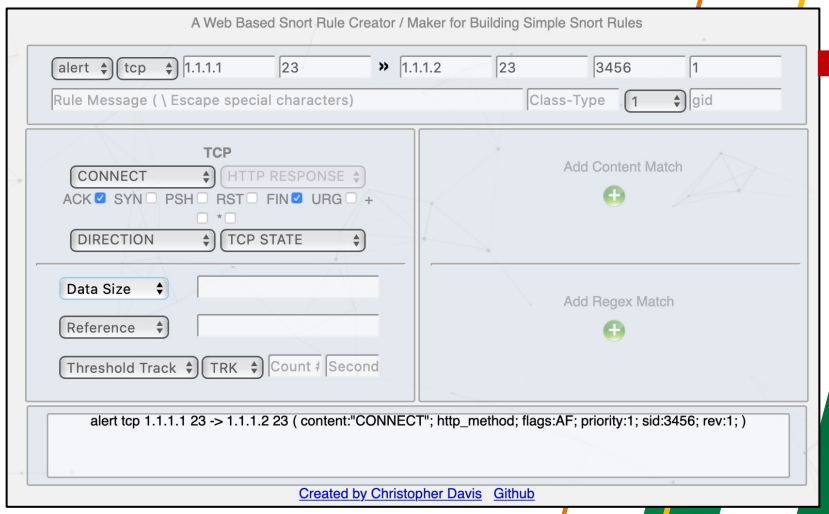
```
alert tcp $HOME_NET 2562 -> $EXTERNAL_NET any
(msg:"ALERTA"; content:"I LOVE YOU"; sid:2009;)
```

WHERE (->, <-, <>) WHO WHAT

- Structure of the rule:
  - **Action (red colour):** alert, pass (stop inspection of the packet), drop (drop the packet and warn) and reject (send an error)
  - **Header (green colour)**: protocol, IP addresses and ports
    - IP addresses can be declared as $HOME_NET and $EXTERNAL_NET
    - Both variables represent an IP range that is specified in the *snort.conf* configuration file
  - **Rules options (blue colour)**: msg (message to be represented/saved in log), sid: the ID of the rule which must be unique, etc.
- It is recommended to access the official SNORT documentation to explore all possible options:
  - The installation of Snort 3 (it is complex): https://docs.snort.org/start/installation
  - Snort configuration: https://docs.snort.org/start/configuration
  - Structure of the SNORT rules: https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm; https://docs.snort.org/rules/; https://docs.snort.org/rules/options/
  - Examples: https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm

SNORT: Configuration and detection

- There are online analysers with static rules that help future experts to get started in the field
  - **Snorpy**: a web-based Snort rule creator, provided by C. Davis
    URL: http://snorpy.cyb3rs3c.net



It is a simple way to design rules in Snort

- **Snort Analyzer**, provided by asecuritysite, allows to analyse rules based on raw existing traffic captures (.pcap)
  URL: https://asecuritysite.com/forensics/snort?fname=bit.pcap&rulesname=bit.rules

It is a simple way to understand how Snort works

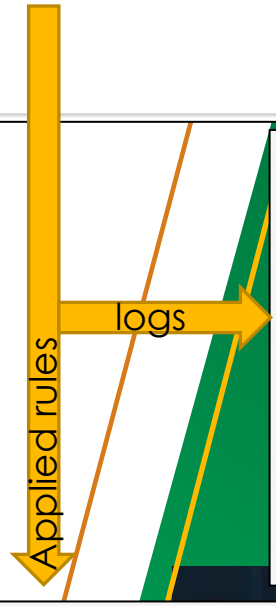
Source: C. Davis, "Snorpy", 2024.
URL: http://snorpy.cyb3rs3c.net
Source: Buchanan, William J (2024). Snort Analyser. Asecuritysite.com
URL: https://asecuritysite.com/forensics/snort

SNORT: Configuration and detection

- With **Snort Analyzer**, it is possible to analyse the results

- The steps are simple
  - Invoke the capture in Wireshark, e.g.: FTP
  - Choose the Snort rules associated to FTP
  - Click on "Determine"

- In the panels below, it is possible to read the outputs of the SNORT logs and the applied rules



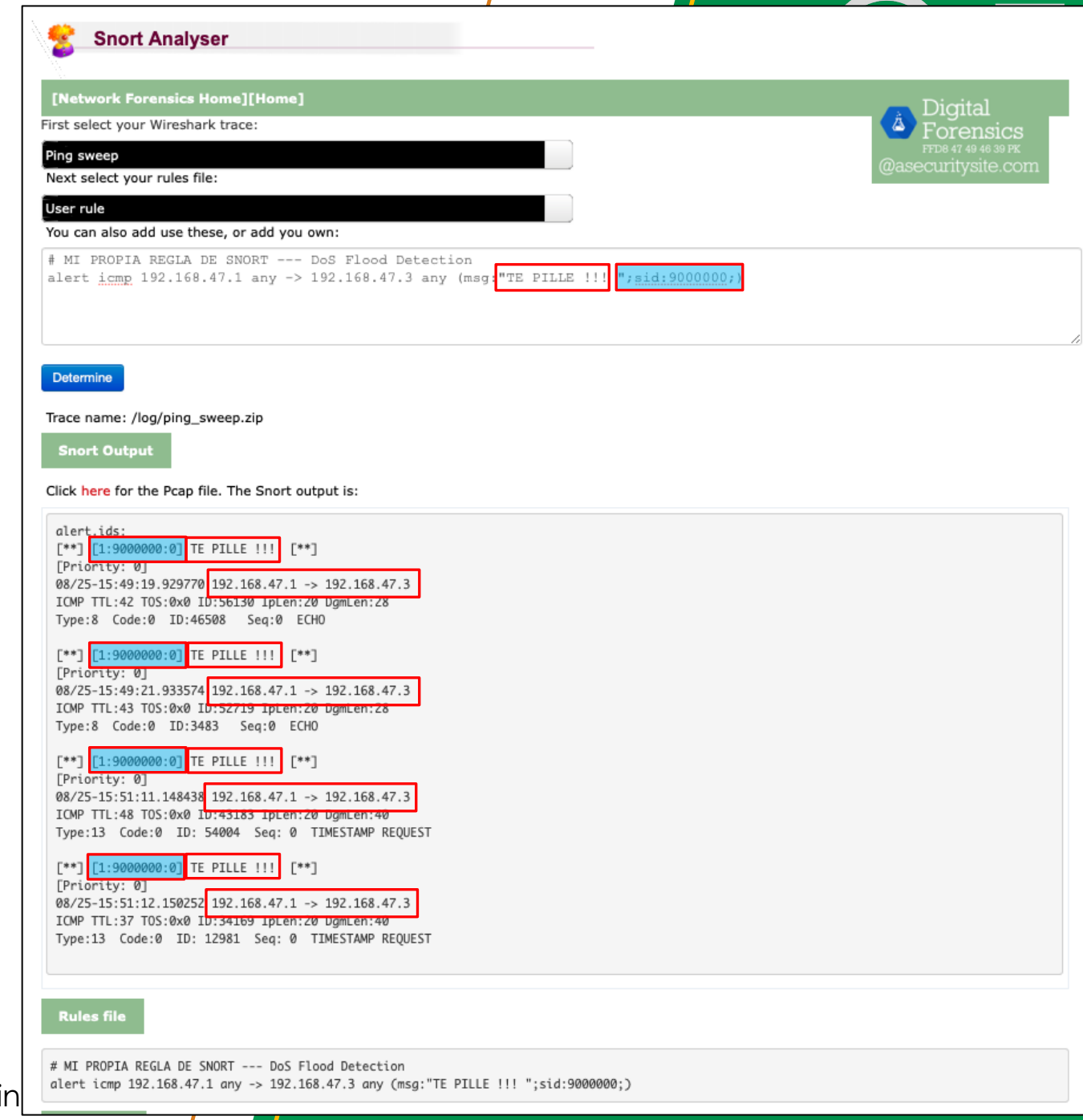

```
alert.ids:
[**] [1:9000005:1] FTP Connection [**]
[Priority: 0]
08/31-20:24:40.417691 192.168.47.1:49430 -> 192.168.47.134:21
TCP TTL:128 TOS:0x0 ID:16588 IpLen:20 DgmLen:52 DF
******S* Seq: 0x4372316F  Ack: 0x0  Win: 0x2000  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP WS: 2 NOP NOP SackOK

[**] [1:9000005:1] FTP Connection [**]
[Priority: 0]
08/31-20:25:00.774487 192.168.47.1:49440 -> 192.168.47.134:21
TCP TTL:128 TOS:0x0 ID:16620 IpLen:20 DgmLen:52 DF
******S* Seq: 0x32065348  Ack: 0x0  Win: 0x2000  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP WS: 2 NOP NOP SackOK
```

```
alert tcp any any -> any 21 (flags:S;msg:"FTP Connection";sid:9000005;rev:1;)
alert tcp any 21 -> any any (msg:"FTP Bad login"; content:"530 User "; nocase; flow:from_server,established; sid:491; rev:5;)
```

Source: Buchanan, William J (2024). Snort Analyser. Asecuritysite.com. https://asecuritysite.com/forensics/snort
URL: https://asecuritysite.com/forensics/snort?fname=bit.pcap&rulesname=bit.rules

SNORT: Configuration and detection

- With **Snort Analyzer** we can define **our own SNORT rules** based on the provided .pcap captures
  The steps are simple
  - Select a traffic capture and download it
  - Analyse the capture with Wireshark
  - Define the corresponding Snort rule
    - *E.g. using Snorpy first ☺!*
  - select the "**User Rule**" option for the rules
  - Click on "Determine"
- The panels below detail the SNORT log outputs and the rules applied
- A tutorial (video) can be found at https://www.youtube.com/watch?v=XuaG61hTHK4&t=386s



Source: Buchanan, William J (2024). Snort Analyser. Asecuritysite.com. https://asecuritysite.com/forensics/snort
URL: https://asecuritysite.com/forensics/snort?fname=bit.pcap&rulesname=bit.rules

Homework: Wireshark, Snorpy, Snort Analyser and SNORT

- **Task1**: Considering the **online "Snort Analyser":**
  1. Test at least 3 different SNORT rules with their corresponding .pcap captures, e.g. "Ping Sweep" and "ICMP Rules"
  2. Use "Snort Analyser" again, but this time it is necessary to select a .pcap traffic capture and download it for local analysis
     - Based on that .pcap capture and using **Wireshark and Snorpy**, analyse and design at least 3 rules in Snort
     - Using these three rules (and through the "User Rule" option of **Snort Analyser**), visualise the success of detection
     - Subsequently, analyse the resulting logs/logs and verify that they are indeed associated with the rules predefined in Snort
- **Task2**: Considering the **SNORT** tool:
  1. Install SNORT on one of the virtual machines (in GNS3) or on your local machine
  2. Repeat Task 1, but this time using SNORT

OPTIONAL PRACTICAL TASKS

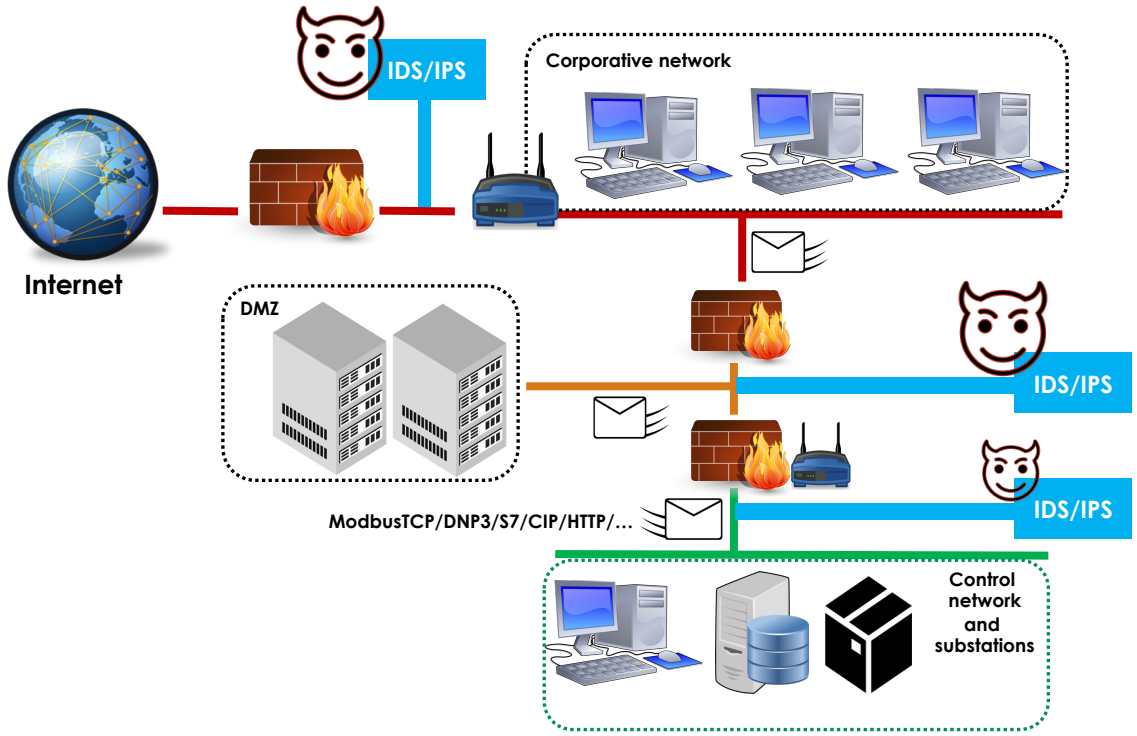
Intrusion prevention through detection + response

- The prevention of an intrusion prevention system (IPS) relies not only on the above detection mechanisms, but also on the ability to respond in a timely manner
- The response can be varied, such as:

Type of IPS	Prevention / response	Details
HIPS (HIDPS)	Closing TCP sessions	• Closing the established connection between a client-server / master-slave (in power control networks)
HIPS (HIDPS)	Using the firewall properties	• In-line IPS sensors offer firewall capabilities that can be used to discard or block network traffic • SNORT's DROP and REJECT commands are only effective if they are configured online
HIPS (HIDPS)	Bandwidth limitation	• In-line IPS sensors could limit the percentage of bandwidth between a client-server / master-slave (in power control networks)
HIPS (HIDPS)	Disinfection of malicious code	• Online IPS sensors can sanitise part or all of the contents of a package by (i) re-packaging payloads into new packages and removing the infected portion, (ii) removing infected attachments from email messages
HIPS (HIDPS)	Reconfiguration of network systems	• IPS sensors can update configurations of firewalls, routers and switches, so that they could reconfigure themselves to block certain types of activity
NIPS (NIDPS)	Interruption of execution of SW processes	• Prevent certain suspicious code/applications from running
NIPS (NIDPS)	Interruption of incoming/outgoing traffic	• Prevent certain packets from being processed on or out of the host
NIPS (NIDPS)	Disinfection of incoming traffic	• Sanitise part or all of the contents of an incoming parcel
NIPS (NIDPS)	Activation of the host firewall	• Stopping unauthorised access or security policy violations
NIPS (NIDPS)	System reconfiguration on the host	• Update firewall configurations so that they reconfigure themselves to block certain types of activity
…	…	• …

Vulnerable IDS/IPS: new security risks and challenges

- Both IDSs and IPSs are susceptible to specific attacks, especially those based on **stealthy actions or based on APT-type attacks**
- Attacking IDS/IPS:



Impersonate a legitimate network node and poison evidence
Scan the ports from time to time (stealthily) for penetration, so that they are not considered as serious anomalies
Deny the IDS/IPS actions to interrupt the monitoring processes
Mislead IDS with fake attacks to camouflage real attacks • That is, to produce "fake" attacks from multiple locations or IPs, in order to generate large volumes of alarms. In turn, the attacker attacks the system from one location or IP to do damage • In this situation, and due to the large volume of alarms, the IT-OT administrator will not be able to discern "real" malicious actions from non-real ones

Source: Seguridad en Sistemas Informáticos Detección de intrusión
URL: https://www.tlm.unavarra.es/pluginfile.php/11611/mod_resource/content/0/clases/08_SSI-monitorizacion1.pdf

Connect with CyberSecPro:
How to register and other practical information

1. Website: www.cybersecpro-project.eu
2. X (Twitter): https://twitter.com/CyberSecPro_eu
3. LinkedIn: https://www.linkedin.com/company/cybersecpro-euproject/

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

Project Agreement no. 101083594

Thank you

If you have any questions, please do not hesitate to contact:

- Cristina Alcaraz
  Associate Professor
  University of Malaga
  alcaraz@uma.es