ANASTACIA

Advanced Networked Agents for Security and Trust Assessment in CPS/IoT Architectures

D2.2

Attacks Threats Analysis and Contingency Actions

This deliverable presents the results of the first 14 months of work concerning the task 2.2.

Distribution level	PU
Contractual date	28.02.2018 [M14]
Delivery date	28.02.2018 [M14]
WP / Task	WP2 / T2.2
WP Leader	CNR
Authors	E. Cambiaso (CNR), I. Vaccari (CNR), E. Punta (CNR), S. Scaglione (CNR), S. Bianchi (SOFT), A. Zarca (UMU), R. Trapero (ATOS), P. Sobonski (UTRC), D. Rivera (MONT)
EC Project Officer	Carmen Ifrim carmen.ifrim@ec.europa.eu
Project Coordinator	Softeco Sismat SpA Stefano Bianchi Via De Marini 1, 16149 Genova – Italy +39 0106026368 stefano.bianchi@softeco.it
Project website	www.anastacia-h2020.eu

ANASTACIA has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 731558 and from the Swiss State Secretariat for Education, Research and Innovation.

Table of contents

PUBLIC SUMMARY

Concerning the ANASTACIA platform, it is important to analyse and protect in an accurate way the underlying infrastructure, in order to avoid malicious users to perpetrate malicious activities on the network. In this context, it is crucial to analyse in deep the threats an attacker may exploit for malicious activities. Also, it is important to classify the attacker's aims and which means he can adopt.

The aim of this document is to address four selected use cases, focused on security aspects of IoT and smart devices, and addressing the following attacks:

1. Distributed Denial of Service,
2. SQL injection,
3. 0-day exploit, and
4. malware.

For each attack, the document reports applicable detection and mitigation actions, in order to identify the threat and react to it. Such activities often require a protection plan that involves the entire life cycle of the components, from the design and implementation, to a continuous protection plan to be continuously adopted on the components, e.g., through signature update activities executed to update the detection system to the latest discovered threats.

Also, the document reports detailed information about unique protection components implemented by ANASTACIA partners, and adopted for the development of the platform.

Finally, focusing on two specific threats discovered during the study, we analyse the panorama of attacks, in order to identify emerging threats in the cyber-security context.

1 INTRODUCTION

1.1 AIMS OF THE DOCUMENT

The aim of this document is to analyse in deep the selected security use cases considered for the ANASTACIA platform. For each use case, the document reports detailed information about the exploited threat, hence proposing detection and mitigation approaches to be implemented and deployed on the ANASTACIA platform. The document also reports how such proposals are managed by ANASTACIA, by exploiting unique tools, methodologies and knowledge provided by project partners. Finally, the document reports innovative threats discovered while during the study and proposes the evaluation of such menaces for future work on the topic.

1.2 APPLICABLE AND REFERENCE DOCUMENTS

This document refers to the following documents:

- ANASTACIA Deliverable D1.2 “User-centred Requirements Initial Analysis”
- ANASTACIA Milestone MS12a “Monitoring component service specified and agreed by the board”
- ANASTACIA Milestone MS12c “Monitoring agent service specification”
- ISO/IEC 27002 Information technology - Security techniques

1.3 REVISION HISTORY

Version	Date	Author	Description
0.1	02/10/2017	I. Vaccari	Initial Draft
0.2	09/01/2018	E. Cambiaso	Enhanced use cases description
0.3	12/02/2018	E. Cambiaso	Enhanced the quality of the document
0.4	23/02/2018	E. Punta	Improved ANASTACIA detection tools and conclusions section
0.5	26/02/2018	E. Cambiaso	Produced the final version of the document

1.4 ACRONYMS AND DEFINITIONS

Acronym	Meaning
BMS	Building Management System
BYOD	Bring-your-own-device
C&C	Command and Control
DBMS	Database management system
DDoS	Distributed Denial of Service
DoS	Denial of Service
IDS	Intrusion Detection System
IoT	Internet of Things
IPS	Intrusion Protection System

MEC	Mobile Edge Computing/Multi-access Edge Computing
MitM	Man-in-the-Middle
SDA	Slow DoS Attack
SDN	Software-defined networking
SQL	Structured Query Language
SQLi	SQL injection

2 OVERVIEW OF ATTACK CATEGORIES

The aim of this section of the document is to define and identify relevant threats against a network infrastructure, by focusing on the core components of the ANASTACIA scenarios, and considering in particular Internet of Things (IoT) and Software-Defined Networking (SDN) environments. Different types of threats will be presented, with the aim of enhancing the development of the project, by identifying relevant security threats and to define the right protection approach to deploy on the system.

Although communication networks have brought great technological innovation, they always attracted malicious users. Therefore, network security assumes a crucial role for every ICT based system. In this context, Figure 1 reports a basic classification of well-known network attacks.

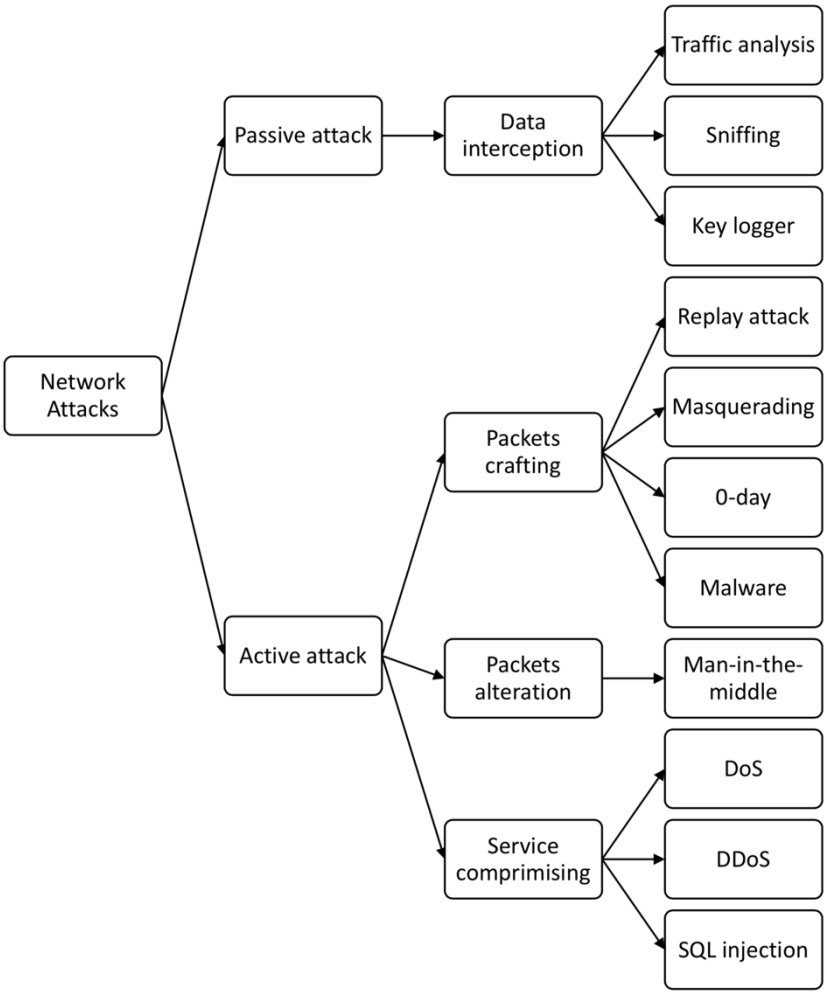

Figure 1: Categorisation of network attacks

The classification is based on two main categories of threats: *active* and *passive* attacks. Concerning active attacks, a malicious client actively injects or alters a network message in order to exploit some sort of vulnerability affecting the targeted host or network. This type of attacks is considered very complex and its prevention is not easy to be accomplished. Instead, relatively to passive attacks, the aim of the attacker is to obtain the information without actively communicate on the network.

In general, each of these attacks aims to introduce delays on the network or to steal sensitive information from the targeted systems. In order to detect and mitigate these threats, specific approaches can be implemented to avoid/reduce the possibility of the exploitation. A good starting point is to analyse international standards on cyber security, such as the ISO/IEC 27002. For instance, analysing requirements reported in chapter 9.1.2 “Control access to networks and network services”, the standard suggests to analyse if VPN or wireless network policies are defined or if network usage is monitored. By following the standard, it is possible to identify sensitive actions to take on sensitive networks or nodes. The adoption of standards like the ISO/IEC 27002 helps network and system administrators to keep the infrastructure safe and secure from possible cyber and physical attacks on the system.

2.1 ACTIVE ATTACKS

Active attacks allow an attacker to interact on the network, for example by sending packets to network devices or by accessing their services [Uma, 2013]. The main objective of such threats is to damage the network or the entire infrastructure depending on the extent of the attack. Concerning active network attacks, three different activities are involved: fabrication/craft of novel packets, modification/alteration of existing packets, services interruption. In the second case (packets alteration), the attacking host usually have to be placed between the two entities involved in the transmission. According to Figure 1, we will now analyse the most known and most harmful active attacks.

2.1.1 Packets Crafting

In this case, new packets are crafted in order to target the victim.

- **Replay Attack**
  A replay attack is intended to postpone or replay the transmission of a package to get a victim's disservice or to obtain information that it would not have access to. An attacker acquires data that he previously had no access to and uses them for his (malicious) purposes. For instance, by repeating a connection packet seizing some sort of resource on the victim, it would be possible to seize all the available resources, hence creating a disservice.
- **Masquerading**
  During a masquerading attack, the attacker assumes the identity of another user of the system to gain access to specific information. It is a technique used by a malicious user to pretend to be an authorized person to gain access to confidential information (e.g., by executing some sort of privilege escalation) in an illegal way.
- **Malware**
  Malware are malicious files or software running on infected hosts. The malware category includes several kinds of malicious programs such as computer viruses, worms, trojan horses, spyware, and ransomware. These programs aim to attack users' devices for different malicious reasons. For example, they can steal user sensitive data, encrypt data to request an unlock ransom, or directly delete them to cause damage to the victim.
- **Zero-day vulnerabilities**
  Zero-day vulnerabilities (also known as "0-days") concerns the exploitation of unknown software vulnerabilities never appeared in networks before. Because of this, their knowledge is extremely limited, usually only to a restricted number of malicious users (even not knowing/communicating among them). In virtue of this, since most of the times even the software producer is not aware of the vulnerability, appropriate patches are not

available and the affected system is vulnerable. Until appropriate patches are deployed on the vulnerable systems, hosts afflicted with such vulnerabilities are exposed to cyber-attacks that may even cause serious damage to the system.

2.1.2 Packets Alteration

In this case, the attacker (physically or logically) places the malicious host between the two nodes of the communication.

- **Man-in-the-Middle (MitM)**
  A man-in-the-middle attack is implemented to access private data exchanged in a communication session or to modify packets thus violating session integrity. This attack is executed in real-time, which means that the attack occurs during the communication session between two network devices. Data can be read, edited and stored when the attacker is able to access the session. The attacker will know the contents of the message before the intended recipient receives it or changes the message along the path [Welch, 2003]. The attacker could adopt different well-known techniques (e.g. by exploiting the HTTP protocol [Callegati, 2009], or by perpetrating ARP poisoning attacks [Nam, 2010]), could put himself in the middle of the communication between two hosts pretending to be the respective recipients of the session.

2.1.3 Service Compromising

In this case, the aim of the attacker is to compromise availability or integrity of the target.

- **Denial of Service (DoS)**
  In a denial of service attack, an attacker exploits the network connection to make the services offered by the victim unavailable, by simply flooding the victim with several packets (e.g., flooding [Huici, 2009], amplification and reflection DoS [Wei, 2013]), or by exploiting some sort of vulnerability (e.g., low-rate [Kuzmanovic, 2003] or exploit based DoS [Muscat, 2016]). Denial of service attacks cause significant damage each year, making it essential to implement and develop innovative techniques for detection and protection against this attack. In order to develop innovative protection techniques, a thorough knowledge of the dynamics of the attack is required. Being a well-known attack with vast potentials, it is considered one of the most dangerous cyber-attacks [Hussain, 2013].
- **Distributed Denial of Service (DDoS)**
  A Distributed Denial of Service (DDoS) threat is a simultaneous attack executed by different coordinated nodes against commonly targeted services offered by the victims. The services under attack can be classified in *primary victims*, where the targeted service is the one that the attacker wishes to make inaccessible, and *third victims*, where third-party hosts or services are exploited to execute the attack against the primary victims (real targets). Instead, the use of *secondary victims* during a DDoS attack provides the attacker the ability to exploit (usually infected) zombies/bots to amplify the attack power by remaining anonymous [Specht, 2004].
- **SQL Injection (SQLi)**
  Structured Query Language (SQL) injection is a computer attack that involves the injection of malicious SQL code to target a web application directly connected to a database management system (DBMS) and to access/steal or inject illegitimate data. During this attack, the attacker usually crafts a portion of the SQL statement by passing it to the server into an HTTP request, in order to alter the initial query and gain access to the database [Sadeghian. 2013].

2.2 PASSIVE ATTACKS

This type of attack does not involve the attacker's interaction with the network devices. Often in these attacks, hackers only care about staying hidden and reading and saving the information of interest exchanged by the various devices on the network. Their goal is to steal information without interacting and remaining hidden.

2.2.1 Data Interception

Passive attacks focus on intercepting system or network communications.

- **Traffic analysis**
  Traffic analysis is a process of intercepting and analysing packets exchanged in a network in order to infer the exchanged content. This kind of threat can also be executed if analysed packets are encrypted and decryption is not possible [Aiello, 2013]. In general (but not always), more packets are exchanged on the network, more information can be extrapolated from the captured traffic.
- **Sniffing/Eavesdropping**
  In general, if network communications occur in plain text, hence exchanged data are not encrypted, it is possible for a malicious user to intercept exchanged information and process them. In this case, it may be required to the attacker to place the malicious host between the two nodes of the communication (see MiTM attack description in Section 2.1.2). For instance, this is possible for a network administrator, by using mirroring ports of network switches, or for an insider threat, by placing a tap on the network. The interception action is generally referred as sniffing or spoofing. The ability of an attacker to monitor the network is generally one of the main problems that users have to deal with, especially if unknown networks (e.g. public access points) are adopted, since, without enabling strong and effective encryption algorithms, data can be read and stored by malicious users.
- **Keylogger**
  Keyloggers runs in the background on the infected system, recording key press and executed commands. Keyloggers can be software based or physical devices attached between the keyboard and the motherboard of the target. Concerning software based keyloggers, once data are stored, they are hidden in particular memory areas for later retrieval, or directly sent in background to the attacker on the Internet. Once the malicious payload is retrieved, the attacker may find passwords or other sensitive data that could be used to compromise the system, for personification, or for social engineering attacks.

3 ANALYSED ATTACK SCENARIOS

The main contribution described in this deliverable document concerns different selected scenarios and attacks, considered at the current stage of the development of the ANASTACIA platform. In order to implement an efficient protection system to detect and mitigate an attack to the network, it is important to analyse the considered threat, its functioning, and how to detect and mitigate it to protect the system.

According to Table 1, four use cases were extrapolated from the ANASTACIA D1.2 deliverable document, describing a wide range of possible attack scenarios. Particularly, the three use cases concern the Building Management Systems (BMS) context, while one is related to the Mobile Edge Computing/Multi-access Edge Computing (MEC) context. By deeply analysing the selected use cases, related attacks, their functioning and their aim were deeply investigated to define appropriate protection methods to be adopted and deployed.

D1.2 USE CASES (Attacks)	How to Detect? WP4-WP5	How to Mitigate? WP2-WP3
UseCase_MEC.3 DoS or DDoS attacks with Ping-ICMP through Smart Cameras or IoT devices.	SNORT as Monitoring Agent. MMT Tool for Montimage. Two detection mechanisms provide more reliability for XL-SIEM of ATOS.	Virtual Firewall for NFV. IP Tables with Netconf.
UseCase_BMS.3 Remote attack to building management system (i.e. SQL injection towards SCADA)	MMT Tool for Montimage.	SDN-ONOS isolates IPv6 addresses
UseCase_BMS.4 A hacker manipulate a critical temperature sensor to trigger the fire and evacuation alarms	Data analysis for UTRC.	IoT Controller requests to stop IoT device. SDN isolates IPv6 addresses. Virtual Honeynet.
UseCase_BMS.2 Insider attack to a fire suppression system	IoT infrastructure detect attacks with invalid credential for actions that require authentication or authorization	AAA Architecture for NFV DTLS channel protection SDN isolates IPv6 addresses.

Table 1: Selected Use Cases for first interaction

According to the description reported in ANASTACIA D1.2 deliverable, the selected attack scenarios cover a wide range of possible threats, as reported in Table 2, where for "well known" we mean that the considered attack is deeply investigated in literature and appropriate protection systems may be applied.

USE CASE	WELL KNOWN	0-DAY	TARGET	ATTACK TYPE
UseCase_MEC.3	•		IoT device	Denial of Service
UseCase_BMS.3	•		Database	SQL Injection
UseCase_BMS.4		•	Temperature sensor data	0-day
UseCase_BMS.2	•		Fire alarm panel	Malware

Table 2: Selected Use Cases for first interaction

The next sections of this chapter focus on these use cases, describing them in detail in order to define innovative protection systems to be adopted. Before describing them, it is important to

define an appropriate severity rank to be adopted to score a specific threat. Particularly, our rank derives from the distinction between critical and non-critical attacks, in conjunction with the targeted entity, that may in general be a (non-)sensitive host or network (in function of its impact on the network of the impairment of the host/network), the entire network, or human beings. By following this approach, Table 3 reports the proposed severity rank, considering that 1 is the lower, 10 is the higher.

ATTACK SEVERITY	NON-SENSITIVE		SENSITIVE		ENTIRE NETWORK	HUMAN BEINGS
	HOSTS	SUBNETS	HOSTS	SUBNETS		
NON-CRITICAL	1	2	3	4	-	-
CRITICAL	5	6	7	8	9	10

Table 3: Proposed severity rank (10 is higher)

For each of the considered levels, Table 4 reports instead a sample attack, in order to provide a real example of threat assigned to each of the specified ranks.

Rank	Attacks category	Attack target sample
1	Non-critical for non-sensitive hosts	Adware installation on a device connected to the public wireless network
2	Non-critical for non-sensitive subnets	Adware spread on the public wireless network
3	Non-critical for sensitive hosts	Lowering performance on public servers
4	Non-critical for sensitive subnets	Lowering performance on the server network
5	Critical for non-sensitive hosts	Compromission of a host connected to the public wireless network
6	Critical for non-sensitive subnets	Compromission of a public wireless network
7	Critical for sensitive hosts	Compromission of a production server host
8	Critical for sensitive subnets	Compromission of the subnetwork including production servers
9	Critical for the entire network	Power interruption for the entire organization facilities
10	Critical for human beings	Compromission of fire suppression systems capabilities

Table 4: Example of attacks for each of the proposed rank

3.1 USE CASE MEC.3

This scenario is focused on the execution of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks through smart cameras and IoT devices belonging to the targeted network. The aim of the attacker is in this case to create disservice on the network and make the smart camera service unavailable to its intended users.

3.1.1 Attack description

In the cyber-security panorama, Denial of service (DoS) attacks are considered a serious threat, since their aim is to compromise connectivity capabilities of an entire network or internal nodes/hosts. DoS attacks can be executed at every layer of the ISO/OSI stack, from the physical layer (e.g., by physically accessing a network server and removing the power cord) to the application one (e.g., by communicating with the listening application in order to make them crash).

A DoS attack can be executed autonomously by a single attacking host. In this case, if a stateful protocol is adopted, the attack may easily be identified and mitigated from the targeted node or network, since the source IP address usually does not change during the attack, hence, its communications/incoming packets can be blocked, thus making the threat ineffective. Nevertheless, in order to bypass such limit, an attacker may execute a simultaneous and coordinated attack from several different nodes/hosts, willing or not to participate to the malicious activity, thus executing a distributed DoS attack. This approach can be adopted also to increase the overall attack bandwidth, especially for volumetric attacks[1].

Usually, it is quite easy to implement and run a denial of service attack, due to the vastness of tools available on the Internet. In general, it should be highlighted that, analogously to many computer and network attacks, DoS threats should not only be considered for malicious activities, but for benign ones too. For instance, a legitimate network administrator may legitimately accomplish a DoS attack against non-production services in order to evaluate the ability to respond under stress conditions. The results may in this case be extremely important to analyse the capabilities of the server to effectively manage flash crowd events.

For our scenario, a DoS is accomplished by a malicious user with malicious goals. Although a denial of service attack could make it possible to dismantle an entire building or organization network, the use case is focused on an attack against a smart camera system. Although the severity rank of the attack is lower than in case of a target to the entire network, it should be considered that in this case the attack may be the first step of a more accurate plan (e.g. involving physical access to the building).

Figure 2 depicts the analysed scenario. It is possible to notice that the attacker accesses the ANASTACIA network to target the smart IP camera, which is directly connected to the network.

[1] In the following, although we will refer generically to the "denial of service" term, a distributed attack will be considered.

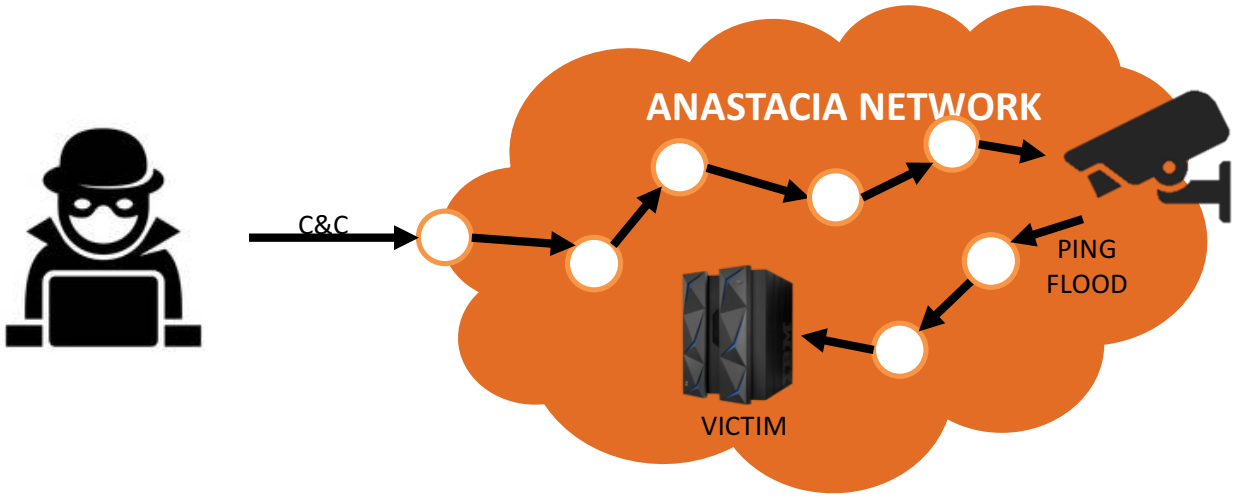

Figure 2: Representation of the MEC.3 scenario

In this scenario, an attacker, external at the network, controls a set of internal nodes/zombies and instructs them to execute a *ping flood DoS* attack on the network. In this case the attacking hosts are compromised IoT devices and smart cameras, that flood the targeted host/victim with a large amount of ping requests packets. The victim is therefore induced to consume its resources, in order to reply to each received request. During a successful attack, it is expected that after some minutes, all targeted hosts are unable to communicate with other network nodes. Therefore, in case a protection plan is not deployed, as for other DoS attacks, the attacked hosts become useless. In legitimate situations, ping messages are based on the ICMP protocol and they are used to check reachability of a remote host. In a legitimate situation, a ping packet, usually sized less than 100 bytes, is sent every second to the host. Moreover, the attack is characterized by the ability to spoof packet source IP address (since it makes use of a stateless protocol). Therefore, it is trivial for the attacker to execute an (apparent) DDoS attack, since in this case the victim would receive packets from many different sources, although a single host is perpetrating the attack.
Figure 3 reports a sample execution of the attack. The attacker starts the threat by sending ICMP echo/ping request to the targeted IP camera in order to make it unavailable on the network, temporarily or indefinitely.

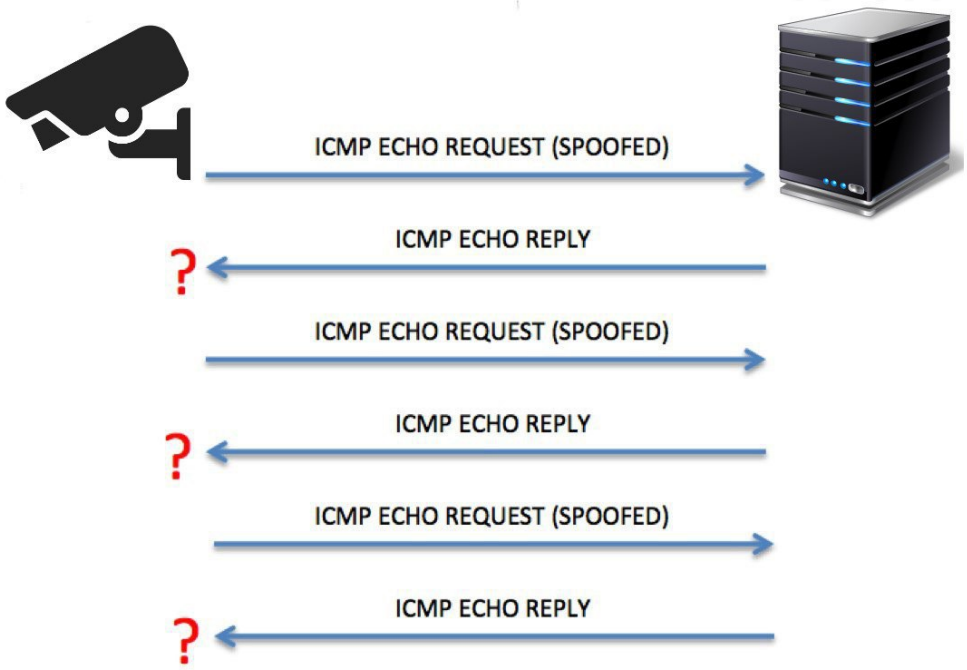

Figure 3: Representation of a ping DoS attack

3.1.2 Protection approach

By analysing how the ping flood DoS attack works, being source IP spoofing easy to do for the attacker, a protection system may not bind on the source of the attack. For simplicity, let's suppose the attacker changes the source IP address of the attack every minute. In this case, the protection system would successfully block the first source of attack, but at the first IP address change from the attacker (after 60 seconds from the begin of the attack), a new attack will be identified by the victim, involving a different source address and leading to another potential DoS. Hence, the protection system may not be effective in this case. Let's imagine if the attacker changes the source IP address of the attack every packet (e.g. by continuously changing the attack source/smart camera/IoT host in order to prevent detection). In this case, the protection system would be totally ineffective and useless. Therefore, in this case, it is important to bind on the destination address (the internal targeted host), instead of the source address of the packets.
In general, two possible protection approaches can be adopted: from one side, it may be possible to block all ping packets, hence blocking the attack at the source. Nevertheless, although proper evaluation of such solution will be dedicated in the development of the project, since some legitimate ping requests may be needed for other network nodes, this approach may not be adopted (or adopted partially, by allowing ping packets receiving only from specific hosts, although they may be under the control of the enemy). The other protection approach involves packets limiting. In this case, considered in the following (since the first approach is trivial to deploy), a filter is applied to limit the incoming flow of ping packets involving a specific host or network.

3.1.3 Detection plan

In order to protect the ANASTACIA infrastructure by a ping flood DoS attack, the proposed protection system should be binded on the destination address of the targeted system, to counter IP spoofing and DDoS attacks, although it is important to highlight that in case of continuous attacks this solution may lead to inject large traffic volumes on the network (hence, a network block of the (internal) attacking hosts may be applied as well). Also, as previously described, the

proposed solution limits the number of packets involving a single host allowed on the network. Particularly, the proposed solution allows a maximum of 10 packets every 5 seconds. By exceeding such limit, an alert is triggered. In this way, there is a maximum bandwidth consumption allowed (without triggering any alert) of about 200 bytes per second (that is almost equal to 1.6Mbps). Such trade-off allows in average two different legitimate clients to simultaneously send ping packets to the destination host, plus it detects DoS or DDoS on the victim, since the maximum allowed bandwidth is extremely low, compared to the bandwidth needed to successfully lead a DoS.
For instance, if the Snort intrusion detection and prevention system [Roesch, 1999] is deployed on the network, it is possible to deploy a Snort rule on network taps integrated in ANASTACIA, in order to detect a ping DoS attack attempt, by using a rule similar to the following one:

```
alert icmp any any -> any any (itype:8; detection_filter:track by_dst, count 10, seconds 5;
priority:7; msg:"Ping DoS attempt"; sid:100121)
```

The proposed rule, generically valid for the entire network, should be in practice binded to a specific host/network, in order to assign a proper ranking/priority value.

3.1.4 Mitigation plan

After the attack is detected, the ANASTACIA platform has to react to the threat, by deploying a mitigation plan. Particularly, in this case a mitigation plan is followed in order to interrupt the attack, thus making the smart IP camera able to properly communicate on the network, independently from the fact the detection alert was triggered when the camera was able to communicate (hence, before the DoS is reached) or not (hence, under the DoS). The mitigation plan can be deployed in different ways. For instance, if Snort is adopted, it is possible to alter the Snort rule reported in Section 3.1.3 to use the drop directive (instead of alert), to directly drop potentially malicious packets matching the filter. Nevertheless, since such solution requires the network tap including Snort to intercept and validate each packet passing through the node (instead, if the alert based trigger is adopted, the tap may access mirrored traffic), it may lead to efficiency and performance issues.
Another solution that may be adopted involves the iptables firewall available on Linux based operating systems. In this case, it is possible to execute the following commands to block the attack.

```
iptables -A INPUT -p icmp -m limit --limit 2/second --limit-burst 2 -j ACCEPT
iptables -A INPUT -p icmp -j DROP
```

It should be noted that these solutions are just representative solutions. For instance, in practice, the reaction/mitigation component of ANASTACIA may receive the alert from the monitoring/detection component, interpret it to identify the targeted node, hence deploy a specific rule able to limit only the traffic directed to the targeted host. Later, in order to reduce the number of rules installed on the system, the same component may restore the situation to the original state (hence, deleting the generated rule).

3.2 USE CASE BMS.3

This scenario is relative to a situation where a malicious user targets an energy micro-grid by exploiting the network nodes in order to violate a SCADA database, by executing a SQL injection attack.

3.2.1 Attack description

SQL injection attacks represents a well-known serious threat for web applications [Halfond, 2006]. By executing such threats, an attacker is potentially able to retrieve or alter database information. Indeed, web applications vulnerable to SQL injection attacks may allow an attacker to gain complete access to the adopted databases. Usually, databases are directly accessed by web servers in order to access structured data from the (web) user interface. SQL injection attacks exploit vulnerabilities affecting web pages, often deriving from bad code quality. A simple example of SQL injection is reported in Figure 4, where the attacker exploits a vulnerability affecting HTTP requests in order to retrieve all the users data stored in the database.

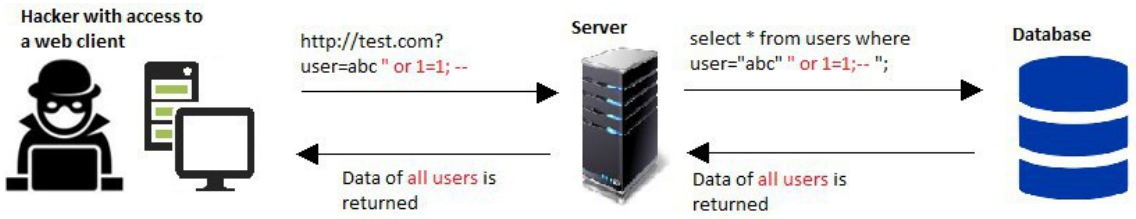

Figure 4: Sample SQL Injection attack

In the analysed BMS.3 scenario of ANASTACIA, according to Figure 5, an external attacker exploits a web page vulnerability to inject SQL malicious code in order to access or manipulate the SCADA database. Such exploitation is based on the generation of the query by using unfiltered inputs provided by the user. The targeted device is a database management system and the attacker's aim may be threefold:

- **To alter/tamper the database content:** in this case, the malicious payload includes, e.g., a concatenation of queries, in the following form (underlined text identifies the input payload provided by the attacker):
  SELECT * FROM users WHERE name = 'foo';DROP TABLE users; SELECT * FROM users WHERE '1' = '1';
- **To bypass access restrictions (in order to accomplish privilege escalation):** in this case, the malicious payload includes, e.g., a concatenation of a logical condition always true bypassing required checks (underlined text identifies the input payload provided by the attacker):
  SELECT * FROM users WHERE password = '' OR '1'='1';
- **To access/steal sensitive data:** in this case, the malicious payload includes, e.g., a concatenation of a logical condition always true enlarging the number of resulting records (underlined text identifies the input payload provided by the attacker):
  SELECT * FROM customers WHERE id = '' OR '1'='1';

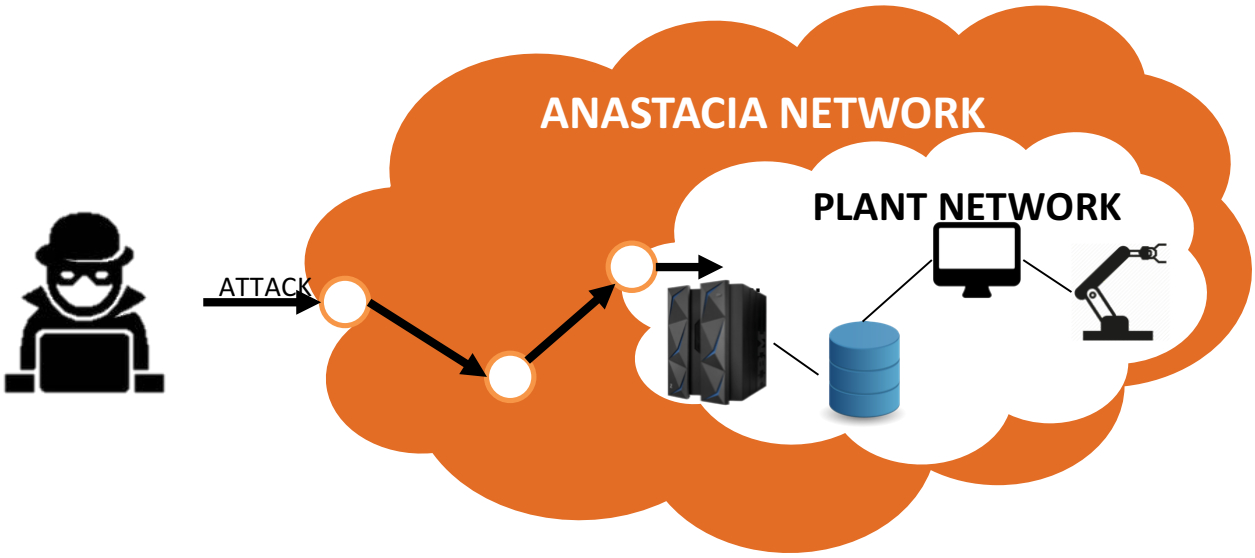

Figure 5: Representation of the BMS.3 scenario

3.2.2 Protection approach

Concerning SQL injection attacks, it is worth mentioning that nowadays it is possible to easily mitigate the threat, for instance by applying input sanitization [Mui, 2010]. Nevertheless, since such implementation depends on computer programmers and network administrators activities, it is important to consider potential bad code or misconfigurations that may expose the system to SQL injection attacks. In order to protect the system from SQL injection attacks, different protection approaches could be implemented: a summary of the proposed protection plan is reported in Table 5.

Proposed activity	Goal
Access control	• Reduce the possibility of attacks • Log network and database accesses
Logs inspection	• Identify the intrusion • Identify the source of the attack (e.g. the “former employee”) • Identify the attack vector
General protection	• Prevent data theft and tampering • Prevent hosts compromising • Prevent application compromising
Restore capabilities	• Block and avoid future compromising • Make the attack ineffective

Table 5: Proposed protection activities for the BMS.3 scenario

3.2.3 Detection plan

In order to detect a SQL injection attack on the network, based on the attacker’s aims described in Section 3.2.1, logs from three principal components should be monitored: *database*, *network*, and *application server*. Further details are reported in Table 6.

Attacker's aim	Processed logs	Detection aim
Alter the DB content	Database	To identify unexpected (non-)queries
Bypass access restrictions	Database	To identify unexpected queries
	Network	To identify unwanted network accesses
	Application server	To identify unwanted accesses to the application
Access/steal sensitive data	Database	To identify generic and unexpected queries
	Network	To detect anomalous/large 1-to-1 traffic

Table 6: Proposed detection plan for the BMS.3 scenario

The main idea is to monitor logs from the three components, in order to analyse performed queries and network and application accesses. In this way, if an attacker tries to exploit the network infrastructure through a SQL injection attack, the malicious activity can be detected by the ANASTACIA platform.

3.2.4 Mitigation plan

In order to efficiently mitigate SQL injection attacks, three separated approaches are suggested. The first one involves a *design time mitigation plan*, natively implemented in the system before it goes into production. The second one involves *run-time mitigation*, executed after the attack is detected by the detection component of ANASTACIA and able to mitigate the identified attack. The last approach involves instead *continuous mitigation*, in order to keep the system updated in view of novel threats. More details of these approaches are reported in Table 7.

Mitigation approach	Activities
Design time mitigation	• Access Control Lists/Permissions (both network and database) • Application vulnerabilities checks • Database backup/restore procedures (e.g. from logs)

Run-time mitigation	• Database level:
	  ○ Block/validate the client's credentials
	  ○ Execution of database restore procedures
	• Network level:
	  ○ Block/validate the network client
	  ○ Block of source IP addresses
	• Application server level:
	  ○ Block/validate the application client
	  ○ Block of source IP addresses (already applied at the network level)

Continuous mitigation	• Continuous maintenance needed, to counter novel threats

Table 7: Proposed reaction plan for the BMS.3 scenario

The proposed mitigation plan implements these three different approaches at three different levels, in order to protect the infrastructure and the network devices from SQL injection attacks.

3.3 USE CASE BMS.4

In this scenario, a malicious user exploits the system in order to manipulate some critical temperature sensors, to trigger fire and evacuation alarms. The attacker exploits in this case a zero-day vulnerability to bypass signature-based intrusion detection systems.

3.3.1 Attack description

A zero-day vulnerability (0-day) is exploited by an attacker that makes use of unknown vulnerabilities on the system to target it [Bilge, 2012; Endorf, 2004]. Indeed, unlike well-known vulnerabilities, "known" by the system and often mitigated, a zero-day attack is unknown to the targeted system, usually attacked in such way for the first time. Since the vulnerability is discovered for the first time during the execution (if it is detected), there may not be known solutions or patches able to efficiently protect the system.

A well-known example of zero-day vulnerability is WannaCry [Mohurle, 2017], a ransomware [O'Gordman, 2012] similar to CryptoLocker [Liao, 2016], whose main objective is to block user access to the host, encrypt sensitive data on the disk and ask a ransom to the user, in order to allow data recovery/decryption.

Considering the ANASTACIA selected scenario, a hacker, external to the network, exploits a zero-day vulnerability to remotely target a sensitive device, in order to access the entire network and attack the infrastructure. According to the ANASTACIA deliverable D1.2 document, the aim of the attacker is to tamper or craft novel temperature packets, in order to trigger fire and evacuation alarms or deactivate building elevators. A simple schema of the scenario is reported in Figure 6.

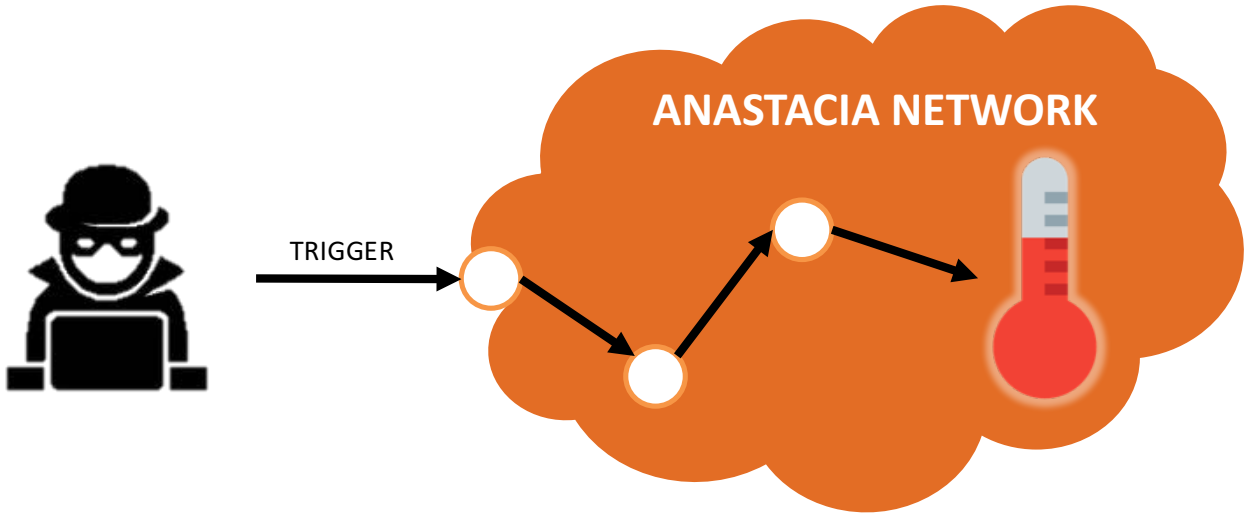

Figure 6: Representation of the BMS.4 scenario

3.3.2 Protection plan

Concerning zero-day vulnerabilities, due to their novelty, there is not a common and general protection and mitigation plan that can be adopted in order to defend a system. Nevertheless, it is well known that zero-day threats can be identified through anomaly based intrusion detection systems [Alazab, 2011; Endorf, 2004; Song, 2013]. Concerning mitigation, it is important to consider that common guidelines can be deployed on the network, in order to prevent zero-day attacks, or, at least, to limit their effects. For the analysed scenario, three protection approaches are proposed:

- **To isolate source addresses from the network:** although each attacker communication is interrupted, such approach may not be efficient in case of distributed attack, since the

attacking source is not a single IP address on the network, but composed by multiple cooperating addresses. In this case, although an isolation of the destination address would make the attack ineffective, it would lead on a denial of service attack, hence, it should be avoided. Moreover, it is important to consider that the source address isolation may affect an (infected) host that is supposed to continuously communicate with the targeted sensor. In this case, the isolation may generate new problems on the network.

- **To deploy a virtual honeynet:** in this case, targeted sensors are not real sensors, but, instead, simulated devices built to fool the attacker and protect real sensor hosts. By exploiting a honeynet, it is possible to accomplish detailed analysis of the exploited zero-day vulnerabilities. The virtual honeynet may be instantiated only after an attack is detected, in order to limit resources usage.
- **To apply sensor data and communication access control:** in this case, it is possible to implement access control for communications involving sensor data transmissions. For instance, client-sensor authentication may be deployed (even in conjunction with sensor-client authentication).
- **To continuously scan vulnerability to counter novel threats:** due to the nature of 0-day attacks, it is required to continuously scan the network to assess the exposure to novel potential threats.

3.3.3 Detection plan

In order to identify the zero-day threat, it is important to deploy an Intrusion Detection System (IDS). IDS can be categorized into *anomaly detection* and *misuse detection* (or *signature based detection*) systems [Cambiaso, 2016]: while anomaly detection systems flag as anomalous each activity that significantly deviates from normal usage profile, misuse detection systems profile well-known menaces extrapolating attack signatures characterizing an intrusion. Being in this case the payload carried out by the attack potentially unknown, no signature can be adopted, due to the lack of knowledge about the threat.

Anomaly based detection systems may adopt algorithms and properties belonging to different research branches, such as statistics [Debar, 1999], machine learning [Tsai, 2009], neural networks [Mukkamala, 2002], or game theory [Alpcan, 2003]. For the context of the selected use case, the aim is to identify anomalous temperature values from the sensor, e.g., by replicating the sensor and comparing the different detected values, or by computing mean and variance of expected values, hence comparing values obtained at run-time. While the first case may, e.g., also identify broken sensors easily, in the latter case, as for other possible cases, a training phase is required to train the algorithm to classify the boundaries of legitimate situations, in order to identify the thresholds able to discriminate between a legitimate and an anomalous situation.

3.3.4 Mitigation plan

The proposed approach is similar to the plan described for the use case BMS.3 (see Section 3.2.4), composed by three separated approaches, *design time mitigation*, *run-time mitigation*, and *continuous mitigation*. More details of these approaches are reported in Table 7.

Mitigation approach	Activities
Design time mitigation	• Sensor data and communication access control • Honeynet simulating a network of sensors

Run-time mitigation	• Deploy a virtual honeynet
	  ○ Anomalous packets are automatically redirected to the honeynet
	• Isolate the attacker addresses
	  ○ Block packets sent from the attacker
	  ○ Connect the target to a secure network to protect it

Continuous mitigation	• Continuous maintenance needed, to counter novel threats

Table 8: Proposed reaction plan for the BMS.4 scenario

The proposed mitigation plan implements these three different approaches in order to protect the infrastructure and the network devices from 0-day threats.

At design time, involving the infrastructure implementation period, appropriate protection systems should be developed, in order to monitor sensitive network sensors and to deploy honeynets simulating a sensor network. Hence, at run-time, it is possible to instantiate a honeynet in case a zero-day attack is found, to redirect the threat to non-sensitive nodes (apparently sensitive for the attacker) and to better analyse the exploited threat. Also, IP address isolation of the attacking hosts can be adopted to block packets received from the malicious nodes. Finally, due to the continuous appearance of novel threats, it is important to maintain the system secure over the time, through continuous mitigation activities.

3.4 USE CASE BMS.2

In this scenario, an insider injects a malware on the network in order to target a fire alarm application system with the aim to control a fire suppression system.

3.4.1 Attack description

The selected use case is relative to an insider threat, e.g. an unhappy employee targeting his own company for some sort of revenge. This kind of threats is extremely dangerous, since insiders typically have advanced knowledge on the targeted system and access to restricted areas. For the selected use case, the malware is spread by using different attack vectors, such as USB infection of a building operation workstation of the malicious employee, or by exploiting wireless connectivity to access the network and spread the malware. Figure 7 depicts the first case: for our aim, such scenario allows us to deploy better security systems (since the operation workstation is controlled by the entity, while wireless devices may be controlled by the insider, e.g., in a BYOD context [Miller, 2012]).

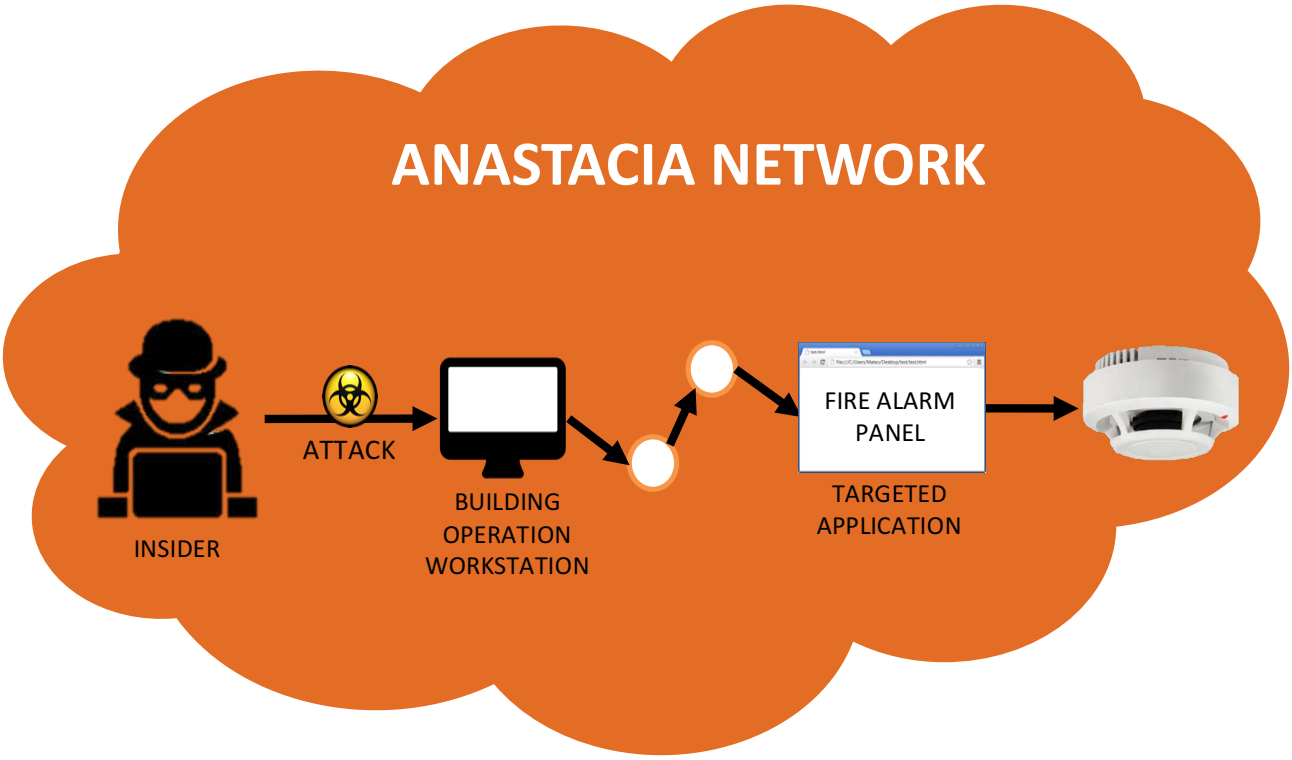

Figure 7: Representation of the BMS.2 scenario

For the depicted scenario, the malware is spread via network using a computer internal to the infrastructure of the targeted organization. The malware exploits an unpatched application of the fire suppression system to access sensitive sensors.

3.4.2 Protection approach

The protection plan proposed for this use case is based on the protection and analysis at three different layers: *network*, *host* and *application*. Indeed, in order to protect a system by malware spreads on the network, it is important to apply a multi-layer protection.

- **Network level protection:** based on network traffic analysis to identify and drop malicious packets, plus access control to limit network access to internal hosts (e.g. by applying MAC

address filtering to avoid mobile devices to connect to the network, or, at least, to make them join only a separate isolated network)

- **Host level protection:** based on controlling hosts and limit privileges and activities users can execute:
  - Installation and update of antivirus and anti-malware software
  - Limit user privileges on the host, to avoid, e.g., installation of custom and unsigned software
  - Block of USB plugging capabilities, to avoid host infections from USB drives and devices (this protection is not efficient in case of a custom hardware device adopted by the insider, in case of malware download from the network, or wireless attack)
- **Application server protection:** based on continuous vulnerabilities patching on both the system and its nodes and exposed applications

3.4.3 Detection plan

The attack detection plan proposed for the selected scenario is based mainly on logs inspection, accomplished by the ANASTACIA monitoring component. By adopting this approach, it is possible to have a complete vision of the current state of the system in order to identify the attack in time, for proper mitigation. Logs of different nature should be analysed, according to Table 9.

Analyzed log	Sample of detected activity
Network logs	Send of malicious packets, connected devices, etc.
Hosts logs	Installation of new software, USB plugging, etc.
Protection software logs	Virus detected on the host
Access logs	Access to the application server, users authenticated on the network, etc.
Application logs	Access to the fire alarm panel
IoT devices logs	Status changes

Table 9: Proposed detection plan for the BMS.2 scenario

Concerning access control activities, it is important to highlight that some sort of action rules/boundaries may also be implemented, to avoid unexpected and unpredicted behaviours of the insider threat. For instance, if it is supposed that employees work during day light, access control rules could be deployed to block employee traffic on sensitive nodes during night. In general, based on the scenario, date-time constraints, or multi-permission actions, it is possible to design and implement some sort of “advanced” access control rules.

3.4.4 Mitigation plan

The proposed approach is similar to the plan described for the use case BMS.3 (see Section 3.2.4), composed by three separated approaches, *design time mitigation*, *run-time mitigation*, and *continuous mitigation*. More details of these approaches are reported in Table 10.

Mitigation approach	Activities

Design time mitigation	• Host level:
	  ○ Evaluate the possibility to block USB plugging on sensitive hosts
	• Application level:
	  ○ Application server vulnerabilities checks

	• Network level:
	  ○ Application server vulnerabilities checks
	  ○ MAC address filtering
	  ○ External devices block/isolation

Run-time mitigation	• Network level:
	  ○ Block/validation of user accounts
	  ○ Validation of IoT devices credentials on the network
	  ○ Block of source IP addresses
	• Application level:
	  ○ Block/validation of application user accounts
	  ○ Block of source IP addresses (already applied at the network level)
	• System restore:
	  ○ To block malicious actions in time (or, at least, to limit the damages)

Continuous mitigation	• Continuous maintenance needed, to counter novel threats

Table 10: Proposed reaction plan for the BMS.2 scenario

Particularly, at design time, it is important to evaluate the possibility to block USB plugging on sensitive hosts, although, as mentioned above, this solution does not protect the system if different attack vectors are adopted by the insider. Instead, at run-time, it is important to execute system restore procedure to restore the state of the system to a secure situation prior to the attack, or, at least, to limit the damages of the attack.

4 ATTACKS DETECTION IN ANASTACIA

Concerning ANASTACIA and, in particular, the monitoring components, the identified methodologies will be addressed through the adoption of an architecture composed by both sensors and processing unities. The aim of the sensors is to live-capture sensitive information and forward them to the processing unities, in order to evaluate if an attack is running or not, also extrapolating detailed information about the identified threat. ANASTACIA monitoring components are based on three unique technologies provided by the partners of the project. In particular, the *Montimage Monitoring Tool (MMT)*, the *XL SIEM*, and innovative *UTRC agents*. Each of these tools will be described in detail in the following.

4.1 MONTIMAGE MONITORING TOOL (MMT)

4.1.1 General Description

The Montimage Monitoring Tool (MMT) has been designed as a modular approach to analyse network traffic and extract protocols metadata. The tool makes use of the Deep Packet/Flow Inspection (DPI/DFI) techniques to extract the required metadata that will be used by other modules of the tool with analysis and testing purposes. In addition, this tool also allows collecting statistics about the analysed flows that facilitates monitoring the performance, the security of the network, and operation troubleshooting.

In general terms, MMT is a collection of modules that work in a coordinated manner to analyse networks protocols and streams by checking two types of properties: “Security rules” and “Attacks”. The former ones describe an expected behaviour of the application or protocol under test; by non-respecting an MMT-Security rule, an abnormal working can be detected. The latter ones describe an attack behaviour of any nature – an attack model, a vulnerability or a misbehaviour. In this case, respecting a MMT-Security rule indicates an abnormal behaviour that might indicate the presence of an attack.

The principal modules of the solution are presented below, and a general view of the workflow between them is presented in Figure 8.

- MMT Extract: This module is the library that implements the DPI/DFI techniques to analyse the packets and extract information related to the protocols including (but not limited to) protocol field values, network and applications QoS parameters and KPIs. The handlers and protocol-dependent extractor functions must be implemented as a plugin of this engine to implement the protocols under study, which allows at the same time an easy extension of this library.
- MMT Probe: The Probe module is a standalone application that makes use of the Extraction Engine – via APIs of this module – to extract the data from the network interface. This tool can be configured to control the source of the packets to analyse (online or offline analysis) and the output of the extracted data (CSV files, Redis communication channels, or others).
- MMT Security: This module is the core component to specify and monitor security and attack rules. The core functionality of this module allows correlating network and application events to detect operation and security incidents. The rules used by MMT are expressed in XML files, which are then compiled to work with the MMT software. In

addition, these tools allow using the rules either in form of plugins of the Probe module or as standalone packet analyser application.

- MMT Operator: Finally, the MMT Operator module is a web-based platform that actively tracks the information reported by the probe on an event bus. This information is later displayed to the monitoring agent to in the web browser. This solution has been conceived to constantly display the statistics of the data computed by, principally, the Probe and Security modules.

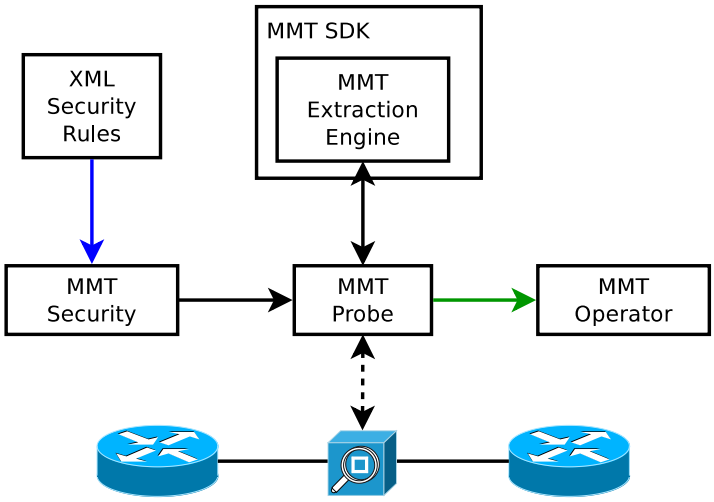

Figure 8 Modular overview of the Montimage Monitoring Tool

The modules interact actively between them to provide a complete security monitoring solution as shown in Figure 8.

All the aforementioned modules are implemented using a modular approach. In this sense, it is possible to extend them in order to offer support for new protocols (extracting the information contained in the flows concerned by the newly-implemented protocol) and novel attack detections by means of defining new attack definitions and security rules that will be actively monitored on the analysed traffic.

4.1.2 Capabilities

The DPI/DFI technique used by MMT Probe allows accessing the raw packets that are traversing the network. In this sense, the Probe is capable of identifying different communication protocols at different layers of the IP stack, extracting information from each one of them. The following list is not exhaustive, and it gives an example of the recognized protocols, but MMT Probe is capable of recognizing a long list of protocols:

- Layer 2-related protocols: Ethernet (mac addresses, payload size)
- Layer 3-related protocols: IPv4, IPv6 (IP addresses, Fragments, flags of the packet, among others)
- Layer 4-related protocols: TCP (port numbers, sequence/acknowledgement numbers, control bits, window size), UDP (port numbers, packet length)
- Upper layers protocols: RTP (sequence numbers, timestamps, synchronization information, etc.), HTTP, and many more.

As mentioned before, the list of recognized protocols is not extensive, and only provides some examples of the protocols from whom the MMT Probe is capable of extract information. In addition, since the Extraction Engine is part of the MMT SDK, it is possible to extend the support to

new protocols by correctly defining the structure of the new protocol and how to correctly extract the data. The documentation of MMT SDK gives the guidelines how to implement new protocols, which are compiled and loaded as plugins of the MMT Probe software.
Once the agent was able to extract the data, the probe is also capable of aggregating and filtering the extracted data, generating reports regarding at different levels of the network stack. A few examples of these reports are the following:

- System info reports: These reports contain information about the resource usage of the machine running MMT Probe, such as the usage memory and the CPU load.
- General statistical reports: containing information about the amount of data.
- Protocols and Applications statistics report (with no session): If these reports are activated, MMT Probe will periodically report the amount of detected protocols and their corresponding traffic statistics.
- Flow report (Protocol with session): A general-purpose report that is used with each flow detected in the network. It contains information about the IP and MAC addresses, port numbers, number of packets involved, the volumes of data involved in the flow, among other statistics. Depending on the protocols detected in the upper layer, this report can have extensions:
    - HTTP report: It contains information extracted from the HTTP headers of the detected flow, such as the User agent, the server response time, accessed URL, number of requests associated with this flow, etc.
    - SSL report: If an application uses an encryption protocol, the amount of information that can be extracted is limited, however, MMT is capable of detecting the Application Family (Web, P2P, etc.), the content type (text, video, etc.), among other fields
    - RTP Report: When a multimedia application uses RTP to stream the content, MMT is capable of computing the packet loss rate, the packet burstiness and jitter.
    - FTP Report: For FTP protocol, MMT can give information about the user name used in the session, their password, the file name, etc.

For more information about the specification of security properties please refer[2] to ANASTACIA MS12a. In addition, more information regarding the usage and configuration of the MMT-Probe can be found[3] in ANASTACIA MS12c.

4.1.3 MMT General Workflow

The general workflow of the MMT solution is shown in Figure 9.

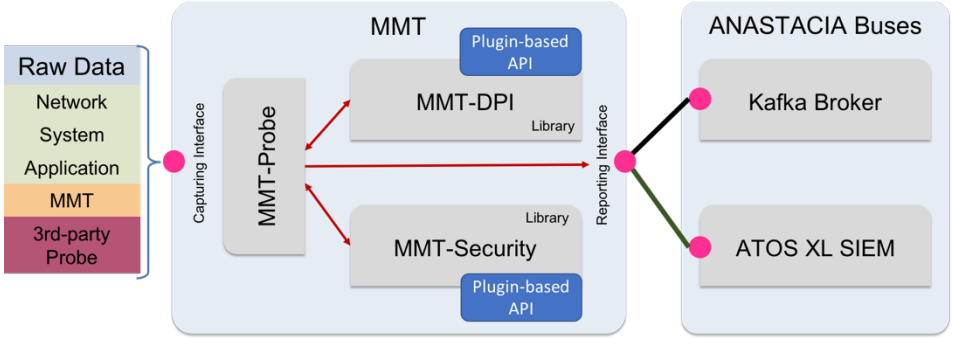

Figure 9 MMT General Workflow

[2] Please note that the mentioned milestone document is not public.
[3] Please note that the mentioned milestone document is not public.

The Montimage Monitoring Too has been conceived with a modular approach. In this sense, the principal parts of the solution interact with each other in order to evaluate the security properties under analysis

The MMT-Probe binds itself to a capturing interface in order to have complete access to the packets processed on the network card under study. Once MMT-Probe have access to the packets, it extracts the information contained in them by using DPI techniques. This functionality is provided by a library called MMT-DPI, which is in charge of parsing the structure of the involved protocols and extracting the information contained in them. The MMT-DPI is a plugin-based, extensible library, offering the possibility to enlarge the MMT data extraction capabilities to new protocols. To this end, MMT-DPI provides an API (part of the MMT-SDK) that allows defining new protocols and their structure. The definition is then compiled in form of MMT plugins that can be easily integrated into the MMT-DPI Library.

Once the information has been extracted from the protocols, it is passed to the MMT-Security Library, the module that will be in charge of correlating the extracted information in order to detect attacks and security issues. The MMT-Security Library has also been conceived in an extensible way, providing an API (also part of the MMT-SDK) that allows the definition of new security properties. These properties are compiled into plugins that are loaded and executed by the MMT-Security module.

Once the security properties (rules) are tested, they generate verdicts about the tested properties. These verdicts (in addition with statistical data about the connections detected in the network) are transmitted to the visualisation interface, using the MMT Reporting Interface.

In the particular case of ANASTACIA, the reported information will be fed to the ANASTACIA buses: a Kafka Broker (in a JSON-based format) and directly to the XL SIEM Tool (in a syslog format).

This design allows the direct correlation of the MMT-verdicts in the XL-SIEM tool, and also use this information and the statistic of the network by other detection tools that will be integrated in the platform.

4.2 XL-SIEM

The Atos XL-SIEM solutions provide with cross-level cybersecurity event and information management capabilities. Different types of security systems can be integrated, correlating events across multiple layers and identifying anomalies in real-time. To this end, the Atos XL-SIEM can be deployed upon a distributed approach, thus reducing the overhead upon the operation of the infrastructure to protect and increasing the resilience of the security infrastructure itself.

The XL-SIEM is built based on three main parts:

- **XL-SIEM dashboard**: provides with a graphical user interface to be used by system administrators for configuration, visualization of events, alarms, reports and decision support assistance.
- **XL-SIEM server**: provides with correlation capabilities based on the events compiled and on thresholds defined for the generation of alarms. It is the core of the Atos XL-SIEM solution, providing with a vision of the current status of the security within an infrastructure.
- **XL-SIEM agent**: provides with a decentralized way to compile and distribute events generated by many different types of sensors, unifying the format of the events in order to be interpreted correctly by the XL-SIEM agent.

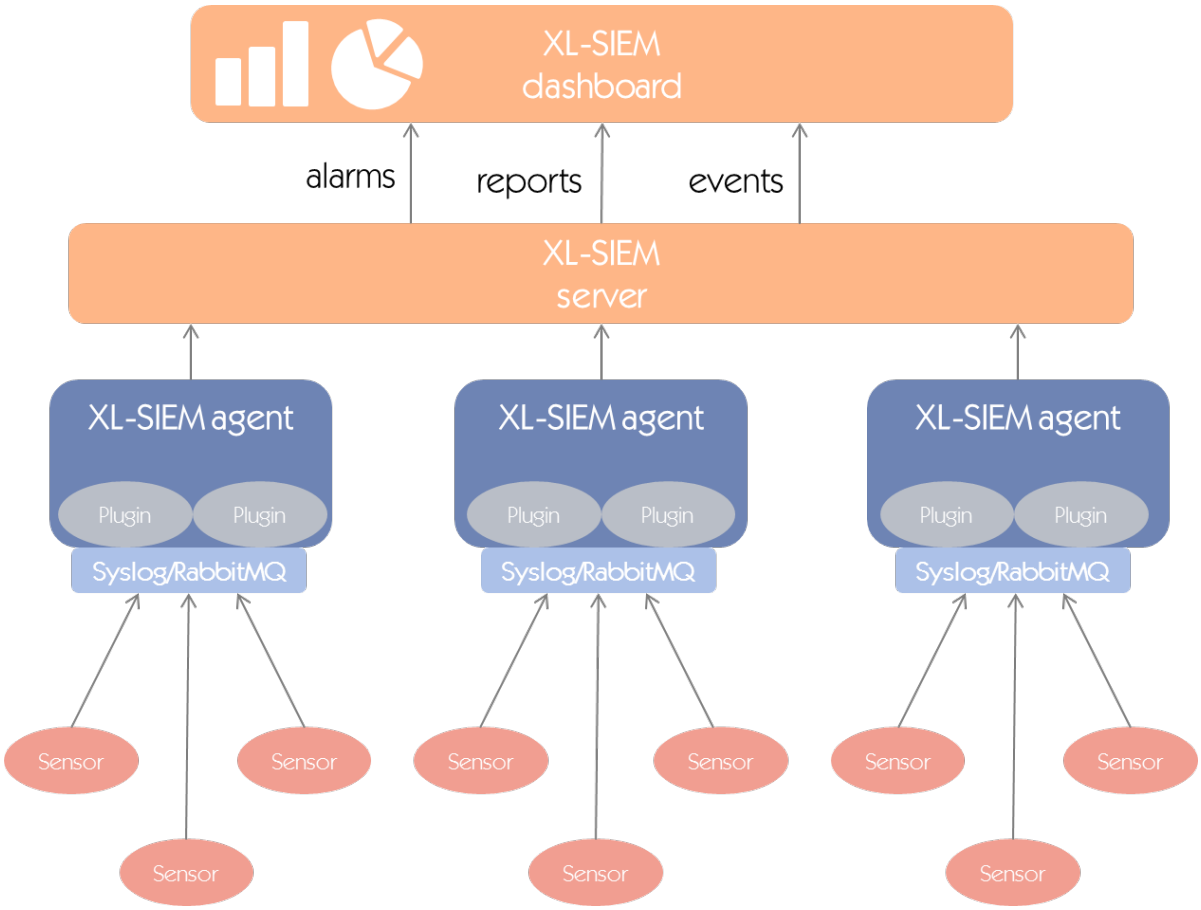

Figure 10. XL-SIEM agents, server and dashboard deployment layout

XL-SIEM agents provide with a flexible approach for the compilation of security related events. XL-SIEM agents are based on plugins which interpret the information received from sensors, normalize it and submit it to the XL-SIEM server. There are a wide variety of plugins already developed for many well-known security solutions and sensors, including (to name just a few): Snort/Suricata (network based intrusion detection), Arpwatch (ARP activity monitor), Ntop (network usage monitor), Kismet (wireless intrusion detection system), OSSEC (host based intrusion detection system), Fprobe (network traffic), pads (Passive Asset Detection System), tcptrack (Monitor TCP connections on the network) or openVAS-Client (the client part of the OpenVAS Security Scanner), nagios3 (Network/systems status monitoring daemon), p0f (Identify remote systems passively).

Beside the predefined list of available plugins, the XL-SIEM provides with a flexible way to adapt new sources of information. New events from sensors are received either from a RabbitMQ message broker or through Remote syslog. In both cases, the event is received and processes by the corresponding plugin.

Plugins parses the log, looking for a match among the message formats defined in the plugins. The information contained in the event is processed and extracted prior to be sent to the XL-SIEM server.

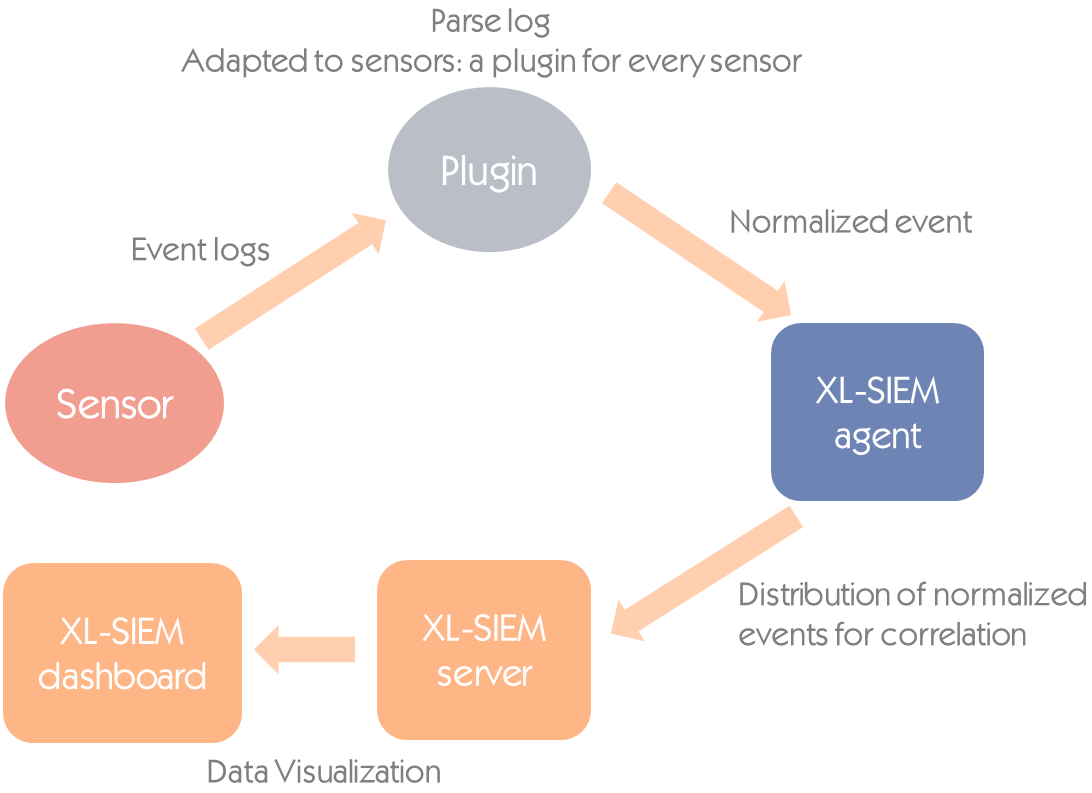

Figure 11. Information flow for processing security events

The following information is included in the normalized event sent from the XL-SIEM agent to the XL-SIEM server:

- **Date**: Timestamp of the event received
- **Sensor**: Name of the agent submitting the event
- **Triggered Signature**: Text describing the event received
- **Category and Subcategory**: Type of event based on the plugin processing the event
- **Data Source Name**: Name identifying the type of event for the plugin processing the event
- **Product Name (optional)**: Name of the product related to the plugin processing the event
- **Source Address**: IP address of the sensor producing the event
- **Source Port (optional)**: Port used to send the event
- **Destination Address**: IP address of the entity receiving the event (generally the XL-SIEM agent)
- **Protocol**: Protocol used to transmit the event
- **User defined data (1..n)**: Custom user data containing additional information included in the event

For every event processed by the XL-Agent a unique event ID is assigned. Additionally, for every event there is a preliminary analysis which results in several properties:

- **Priority**: This parameter determines the importance of the event processed, which is used for the XL-SIEM server to assign more resources to its processing.
- **Reliability**: This parameter determines how trustworthy is the information contained in the event. The reliability level is based on the sensor producing it, which is set by the system administrator depending on the importance of the sensor or the infrastructure being monitored.
- **Risk**: This parameter determines the security threat that the processed event might entail.

The flexibility of the XL-SIEM model allows the integration of the Montimage Monitoring Tool and the UTRC agents as additional sources of information, using their events or alerts as an additional input when correlating events and generating more accurate alarms.

4.3 UTRC AGENTS

UTRC Agents are used to build a data-driven model based on collected operational data of the machines that will be continuously monitoring and analysing newly collected data in order to detect if a severe deviation from expected behaviour can be noticed, that could be caused by an attack.

The functioning of the UTRC Agent can be described as anomaly-based intrusion detection and it can be divided in two phases:

- **Offline**: when it builds and learns the model, based on collected and processed data to represent the normal system behaviour. Offline data processing flow has been illustrated on Figure 12.

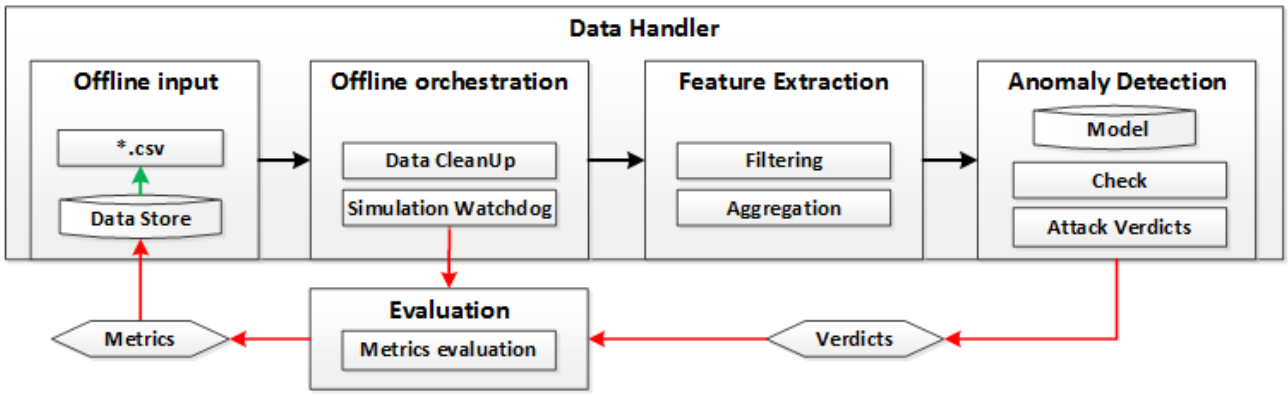

Figure 12. Offline data handler processing flow.

- **Online**: when the built model is used to continuously monitor and evaluate the newly collected data in order to state if there are any signs of an attack taking place from the point of the of sensor data. This phase of operation was shown on Figure 13.

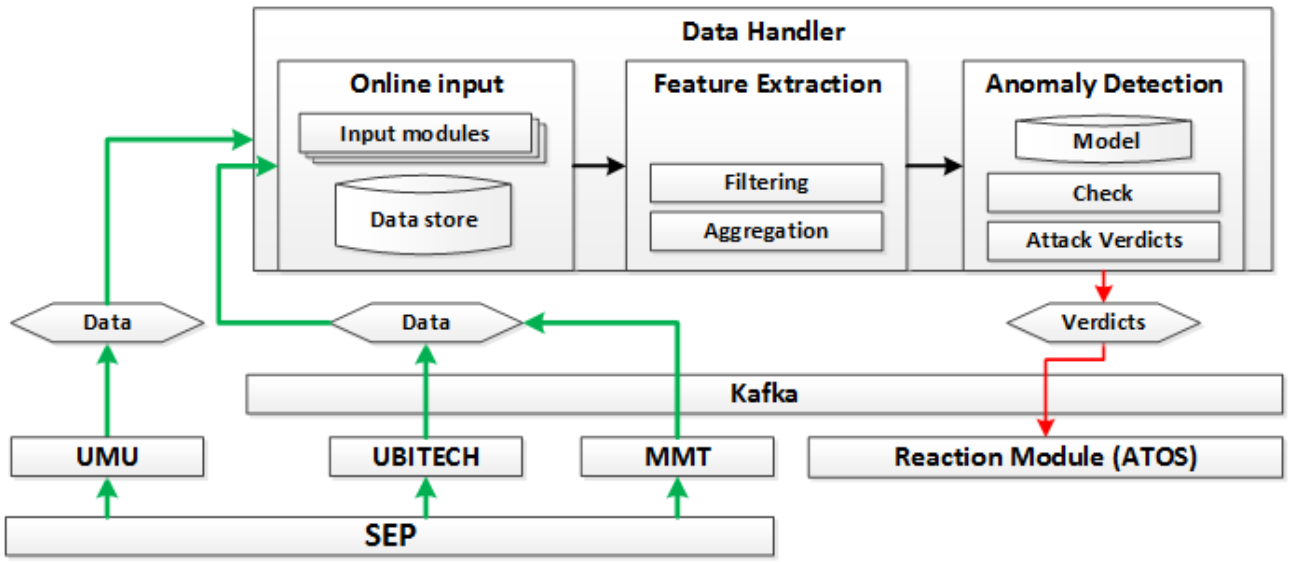

Figure 13. Online data handler processing flow.

Initially, the agent undergoes an offline phase, also referred to as training period, when a data-driven model is built. This model consists of features and a set of relations among them to capture normal system behaviour. By system behaviour we mean collection of system states, where a state is defined by the attributes of the features of the model that are derived from the operational data. The agent collects operational data from the physical IoT devices and performs cleaning, aggregation, filtering. Feature extraction is performed on this data, capturing system behaviour over time, through identifying relations among one or more features. These features themselves already describe the monitored systems state but in order to better capture global behaviour relations between features are also created. From the collection of features relations are built between them that capture the normal and already observed system states. A threshold is learned that is used as a measurement to state whether a new system state is considered normal or not. In case of anomaly the threshold is also capable to provide a measurement of deviation of expected and actual system state. After initial constructing this model it is evaluated and updated based on its performance until it reaches a specified performance.

After learning the model that represent the behaviour of the system it is used in the online phase where the operational data is continuously monitored, collected and processed. The Agent collects the available operational data, maps it to the features used by the model and feeds them accordingly. The model then decides if the received system state can be classified as normal behaviour. When a system state derived from a collection of operational data is deviating significantly from the previously observed behaviour, the agent flags the specified state as an anomaly and reports it with the explanation to XL-SIEM component**.** The agent is able to provide further details what sensor or collection sensors caused the deviation and how severe it is, that is, how much it deviated from the expected system behaviour.

5 ADDITIONAL CONSIDERATIONS

Concerning the considered threats, four use cases have been selected, by considering important attacks able to compromise the security of the entire infrastructure. Although the mentioned threats are extremely serious and represents nowadays relevant threats every system and network administrator has to face, it is important to consider novel threats also, able to perpetrate attacks by adopting last generation attack methodologies and technologies. During the development of ANASTACIA, hard work was dedicated to the study of last generation threats, and, in particular, two different types of attack were investigated, relative to the MEC.3 and BMS.4 use cases.

5.1 SLOW DENIAL OF SERVICE ATTACKS

In particular, the MEC.3 use case concerns a (distributed) denial of service attack against smart IoT devices. Although this scenario includes the important Ping flood DoS attack, in the last years, the DoS arena of cyber-threats evolved. Generally speaking, there are several approaches that can lead to a denial of service (for instance, physical attacks, exploit based threats, flooding DoS like Ping DoS, etc.). Over all these types of threats, we analyzed the emerging category of Slow DoS Attacks (SDA), also known as application layer or Low-Bandwidth Rate DoS, that represent the second generation of network based DoS attacks [Cambiaso, 2017]. Particularly, the first generation of DoS attacks, including Ping/ICMP DoS and SYN flood attacks, is based on “flooding” the victim with large amount of data, until its resources are overwhelmed and a DoS is reached. Differently, a Slow DoS attack is able to lead a DoS on the victim by adopting a low fraction of attack bandwidth [Cambiaso, 2013]. In virtue of this, the resources employed to successfully perpetrate Slow DoS Attacks are reduced, hence making them a more dangerous threat, since they could be executed even from non-performing devices. Concerning slow DoS threats, their characteristics make their traffic appear to the victim particularly similar to legitimate traffic, since the exchanged payload is compliant to the protocols of the ISO/OSI stack (e.g. to the HTTP protocol, in case of web server exploitation). Although this characteristic makes it hard to identify a running attack [Aiello, 2013], some works on this context try to detect a SDA by reducing detection times [Aiello, 2014]. During our study on the topic, we have discovered and modelled a novel Slow DoS Attack called SlowComm [Cambiaso, 2017]. The attack exploits a vulnerability on most server applications that limits the number of simultaneous threads on the host. Unlike flooding DoS threats, which aim to overwhelm some network capabilities of the victim host, SDA often adopt a smarter approach, seizing all the available connections with the application listening daemon and exploiting specific timeouts [Cambiaso, 2013]. Compared to older types of DoS attacks, this approach requires extremely low amounts of attack bandwidth, as the number of connections a server is able to simultaneously manage at the application layer is sensibly lower that the number it is able to manage at the transport layer of the ISO/OSI model.

In particular, SlowComm sends a large amount of slow (and endless) requests to the server, saturating the available connections at the application layer on the server inducing it to wait for the (never sent) completion of the requests. As an example, we refer to the HTTP protocol, where the characters sequence \r\n\r\n represent the request end: SlowComm is supposed to never send these characters to the victim, by forcing it to an endless wait. Additionally, the request is sent slowly, by fragmenting it in multiple packets, hence applying timeouts between each packet send. Similar behavior could be adopted for other protocols as well (e.g., SMTP, FTP, etc.). As a consequence, by applying this behavior to a large amount of connections with the victim, a DoS

state may be reached. By executing the attack on test environments, we observed that it is successful and a DoS is reached on the victim, although future enhancements may focus on maintaining it for long times (without perpetrate a new attack). In virtue of this, the proposed attack should be considered extremely dangerous for ICT infrastructures.

5.2 IoT Security Considerations

Concerning instead the BMS.4 use case, focused on IoT critical temperature sensors exploitation, during the development of the project, we have worked to study in deep the IoT security context. In particular, our work on the topic investigates security aspects of Internet of Things networks. We focused on ZigBee, a communication protocol ensuring low power consumption and characterized by low data transmission rates. We found important security issues related to a ZigBee based system and, potentially, to other similar protocols. Particularly, we identified [Vaccari, 2017] the possibility to send *Remote AT Commands*, AT meaning "attention", to a connected sensor, in order to reconfigure it, for instance, by making it join a different malicious network, hence to forward captured data to a malicious entity. In particular, AT Commands are specific packets, historically used by old generation modems to interface with the device and today adopted by radio modules such as XBee [Boonsawat, 2010], ESP8266 [Thaker, 2016], or ETRX3 [Dao, 2017] to configure device parameters like connection type, network identifier, device name on the network, or destination address. AT Commands are today implemented on many devices of different nature, providing different functionalities and hence commands. The proposed threat focuses on XBee based sensors exploitation. XBee modules support remote send of AT Commands (Remote AT Commands). Since these packets belong to the (IEEE 802.15.4) MAC layer, they are interpreted by the XBee module automatically, hence, being this interpretation demanded to the device firmware and being such firmware closed/provided by the manufacturer, Digi International, it is not possible to avoid implicit Remote AT Commands interpretation. The proposed attack exploits Remote AT Command functionality to reconfigure the sensor, potentially, for malicious aims. XBee supports several AT Command packets[4]. For our aim, we evaluated feasibility of the proposed threat by using ATID commands against targeted sensors, in order to reconfigure identifier of the joined network on the attacked device. By executing the attack on test environments, we observed that the attack is successful and it is able to target a single node without affecting the other nodes of the network, hence resulting extremely precise and potentially difficult to identify. Attack identification is also extremely difficult due to the number of packets sent by the attacker, resulting minimum. The proposed attack should therefore be considered particularly dangerous, since it may compromise the security of an entire IoT network with minimum effort for the attacker.

Although our studies on low-rate DoS attacks and IoT sensors security, including the proposals of novel threats operating in these context, are not directly related to the selected use cases, they represent an important result for the protection of the entire infrastructure, since it is well known in the cyber-security field that studying novel threats, and, possibly, implementing novel attacks, is a fundamental activity to reduce the gap between white and black hackers, entities working in cyber-protection and cyber-criminal, and it is needed in order to enhance the overall protection of ICT systems.

[4] More information can be found at the following address:
https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf

6 CONCLUSIONS

In this document, the security use cases selected for the ANASTACIA platform were analysed in depth. For each use case, detailed information on the exploited threat was provided, thus proposing detection and mitigation approaches to be implemented and deployed on the ANASTACIA platform. The document also showed how these proposals are managed by ANASTACIA, exploiting unique tools, methodologies and knowledge provided by project partners. Furthermore, the document presented the innovative threats discovered during the study and the evaluation of these threats for future work on the subject.

First, an overview of the possible categories of attacks was presented in order to define and identify the relevant threats against a network infrastructure, focusing on the main components of the ANASTACIA scenarios and considering in particular Internet of Things (IoT) and Software-Defined Networking (SDN) environments. Different types of threats were presented, with the aim of improving project development, identifying the relevant security threats and defining the right protection approach for implementation on the system.

Communication networks represent a great technological innovation and always attract malicious users. Therefore, network security assumes a crucial role for any ICT-based system. A basic classification of well-known network attacks is based on two main categories of threats, namely active and passive attacks. As for active attacks, a malicious client actively injects or alters a network message to exploit some sort of vulnerability that affects the targeted host or network. Active attacks allow an attacker to interact on the network, such as sending packets to network devices or accessing their services. The main objective of these threats is to damage the network or the entire infrastructure depending on the scope of the attack. With regard to active network attacks, three different activities are involved: fabrication/craft of new packets, modification/alteration of existing packets, services interruption. This type of attacks is considered very complex and prevention is not easy to achieve.
Instead, as for passive attacks, the attacker's goal is to obtain information without actively communicating on the network. This type of attack does not involve the attacker's interaction with network devices. Often in these attacks, hackers only care about staying hidden and reading and saving the information of interest exchanged by the various devices on the network. The goal is to steal information without actively interacting and remaining hidden.
In general, each of these attacks aims to introduce delays on the network or to steal sensitive information from the targeted systems. In order to identify and mitigate these threats, specific approaches can be implemented to avoid/reduce the possibility of exploitation. In this document, it was emphasized that a good starting point is the analysis of international standards on cyber security, such as ISO/IEC 27002. Following the standard, it is possible to identify sensitive actions to be undertaken on sensitive networks or nodes. The adoption of standards such as ISO/IEC 27002 helps network and system administrators keep the infrastructure safe and protected from possible cyber and physical attacks on the system.

In this ANASTACIA D2.2 deliverable document, different attack scenarios were selected, analysed and considered in the current phase of the development of the ANASTACIA platform. The analysis of the considered threat and its functioning is crucial to implement an efficient protection system to detect and mitigate an attack on the network.

Four use cases were extrapolated from the ANASTACIA D1.2 deliverable document, which described a wide range of possible attack scenarios. One of the selected use cases is related to the Mobile Edge Computing/Multi-access Edge Computing (MEC) context, while three use cases concern the Building Management Systems (BMS) context, in particular: DoS or DDoS attacks with Ping-ICMP via Smart Camera or IoT devices (UseCase_MEC.3); remote attack to building management system, i.e. SQL injection towards SCADA (UseCase_BMS.3); a hacker manipulate a critical temperature sensor to trigger the fire and evacuation alarms (UseCase_BMS.4); insider attack to a fire suppression system (UseCase_BMS.2).

The selected use cases were analysed deeply. The related attacks, how they work and their aim, were examined to define the appropriate security methods to be adopted and implemented. According to the description reported in ANASTACIA D1.2 deliverable document, the selected attack scenarios cover a wide range of possible threats, ranging from “well known” to zero-day. An appropriate severity rank was defined and adopted to score a specific threat. In particular, the defined rank considers the fundamental distinction between critical and non-critical attacks, in conjunction with the targeted entity, that may in general be a (non-)sensitive host or network (in function of its impact on the network of the impairment of the host/network), the entire network, or human beings.

Innovative attacks detection and protection systems were proposed and described through the use cases and classified according to the severity rank of the specific threat. In ANASTACIA platform, the identified methodologies have been addressed through the adoption of an architecture composed by both sensors and processing unities, as for monitoring components. The sensors have the aim to live-capture sensitive information and forward them to the processing unities, in order to evaluate if an attack is running or not, also extrapolating detailed information about the identified threat. ANASTACIA monitoring components are based on three unique technologies provided by the partners of the project. In particular, the Montimage Monitoring Tool (MMT), the XL SIEM, and innovative UTRC agents.

Regarding the threats considered, four use cases were selected, considering important attacks that could compromise the security of the entire infrastructure. Although the threats mentioned are extremely serious and represent nowadays relevant threats, which every system and network administrator faces, it is important to consider new threats, capable of perpetrating attacks using the latest generation of attack technologies and methods. As a final remark, it is important to underline that during the development of ANASTACIA, an important activity was dedicated to the study of the latest generation threats.

7 REFERENCES

[Aiello, 2013] Aiello, M., Cambiaso, E., Scaglione, S., & Papaleo, G. (2013, July). A similarity based approach for application DoS attacks detection. In *Computers and Communications (ISCC), 2013 IEEE Symposium on* (pp. 000430-000435). IEEE.

[Aiello, 2014] Aiello, M., Cambiaso, E., Mongelli, M., & Papaleo, G. (2014, October). An on-line intrusion detection approach to identify low-rate DoS attacks. In *Security Technology (ICCST), 2014 International Carnahan Conference on* (pp. 1-6). IEEE.

[Alazab, 2011] Alazab, M., Venkatraman, S., Watters, P., & Alazab, M. (2011, December). Zero-day malware detection based on supervised learning algorithms of API call signatures. In *Proceedings of the Ninth Australasian Data Mining Conference-Volume 121* (pp. 171-182). Australian Computer Society, Inc.

[Alpcan, 2003] Alpcan, T., & Basar, T. (2003, December). A game theoretic approach to decision and analysis in network intrusion detection. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on* (Vol. 3, pp. 2595-2600). IEEE.

[Bilge, 2012] Bilge, L., & Dumitras, T. (2012, October). Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 833-844). ACM.

[Boonsawat, 2010] Boonsawat, V., Ekchamanonta, J., Bumrungkhet, K., & Kittipiyakul, S. (2010, May). XBee wireless sensor networks for temperature monitoring. In *the second conference on application research and development (ECTI-CARD 2010), Chon Buri, Thailand*.

[Callegati, 2009] Callegati, F., Cerroni, W., & Ramilli, M. (2009). Man-in-the-Middle Attack to the HTTPS Protocol. *IEEE Security & Privacy*, *7*(1), 78-81.

[Cambiaso, 2013] Cambiaso, E., Papaleo, G., Chiola, G., & Aiello, M. (2013). Slow DoS attacks: definition and categorisation. *International Journal of Trust Management in Computing and Communications*, *1*(3-4), 300-319.

[Cambiaso, 2016] Cambiaso, E., Papaleo, G., Chiola, G., & Aiello, M. (2016). A Network Traffic Representation Model for Detecting Application Layer Attacks.

[Cambiaso, 2017] Cambiaso, E., Papaleo, G., & Aiello, M. (2017). Slowcomm: Design, development and performance evaluation of a new slow DoS attack. *Journal of Information Security and Applications*, *35*, 23-31.

[Dao, 2017] Dao, V. L., & Hoang, V. P. (2017, October). A smart delivery system using Internet of Things. In *Integrated Circuits, Design, and Verification (ICDV), 2017 7th International Conference on*(pp. 58-63). IEEE.

[Debar, 1999] Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer Networks*, *31*(8), 805-822.

[Endorf, 2004] Endorf, C., Schultz, E., & Mellander, J. (2004). *Intrusion detection & prevention* (pp. 1-247). Emeryville, CA: McGraw-Hill/Osborne.

[Halfond, 2006] Halfond, W. G., Viegas, J., & Orso, A. (2006, March). A classification of SQL-injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering* (Vol. 1, pp. 13-15). IEEE.

[Huici, 2009] Huici, F., Niccolini, S., & d'Heureuse, N. (2009, November). Protecting SIP against very large flooding DoS attacks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE* (pp. 1-6). IEEE.

[Hussain, 2013] Hussain, A., Heidemann, J., & Papadopoulos, C. (2003, August). A framework for classifying denial of service attacks. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (pp. 99-110). ACM.

[Kuzmanovic, 2003] Kuzmanovic, A., & Knightly, E. W. (2003, August). Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications* (pp. 75-86). ACM.

[Liao, 2016] Liao, K., Zhao, Z., Doupé, A., & Ahn, G. J. (2016, June). Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin. In *Electronic Crime Research (eCrime), 2016 APWG Symposium on* (pp. 1-13). IEEE.

[Miller, 2012] Miller, K. W., Voas, J., & Hurlburt, G. F. (2012). BYOD: Security and privacy considerations. *It Professional*, *14*(5), 53-55.

[Mohurle, 2017] Mohurle, S., & Patil, M. (2017). A brief study of wannacry threat: Ransomware attack 2017. *International Journal*, *8*(5).

[Mui, 2010] Mui, R., & Frankl, P. (2010). Preventing SQL injection through automatic query sanitization with ASSIST. *arXiv preprint arXiv:1009.3712*.

[Mukkamala, 2002] Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on* (Vol. 2, pp. 1702-1707). IEEE.

[Muscat, 2016] Muscat, I. (2016). Web vulnerabilities: identifying patterns and remedies. *Network Security*, *2016*(2), 5-10.

[Nam, 2010] Nam, S. Y., Kim, D., & Kim, J. (2010). Enhanced ARP: preventing ARP poisoning-based man-in-the-middle attacks. *IEEE communications letters*, *14*(2).

[O'Gordman, 2012] O'Gorman, G., & McDonald, G. (2012). *Ransomware: A growing menace*. Symantec Corporation.

[Roesch, 1999] Roesch, M. (1999, November). Snort: Lightweight intrusion detection for networks. In *Lisa* (Vol. 99, No. 1, pp. 229-238).

[Sadeghian, 2013] Sadeghian, A., Zamani, M., & Manaf, A. A. (2013, September). A taxonomy of SQL injection detection and prevention techniques. In Informatics and Creative Multimedia (ICICM), 2013 International Conference on (pp. 53-56). IEEE.

[Specht, 2004] Specht, S. M., & Lee, R. B. (2004, September). Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures. In ISCA PDCS (pp. 543-550).

[Song, 2013] Song, J., Takakura, H., Okabe, Y., & Nakao, K. (2013). Toward a more practical unsupervised anomaly detection system. *Information Sciences*, *231*, 4-14.

[Thaker, 2016] Thaker, T. (2016, March). ESP8266 based implementation of wireless sensor network with Linux based web-server. In *Colossal Data Analysis and Networking (CDAN), Symposium on*(pp. 1-5). IEEE.

[Tsai, 2009] Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, *36*(10), 11994-12000.

[Uma, 2013] Uma, M., & Padmavathi, G. (2013). A Survey on Various Cyber Attacks and their Classification. IJ Network Security, 15(5), 390-396.

[Vaccari, 2017] Vaccari, I., Cambiaso, E., & Aiello, M. (2017). Remotely Exploiting AT Command Attacks on ZigBee Networks. *Security and Communication Networks*, *2017*.

[Wei, 2013] Wei, W., Chen, F., Xia, Y., & Jin, G. (2013). A rank correlation based detection against distributed reflection DoS attacks. *IEEE Communications Letters*, *17*(1), 173-175.

[Welch, 2003] Welch, D., & Lathrop, S. (2003, June). Wireless security threat taxonomy. In Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society (pp. 76-83). IEEE.