D2.7. CIPSEC Framework Final Version

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-2019
Actual submission date: 26-April-2019
© 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 action | Innovation |
| Start date | May 1st, 2016 |
| Duration | 36 months |
| Website | www.cipsec.eu |

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This deliverable contains original unpublished work except where clearly indicated otherwise.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 Research Executive Agency (REA) or the European Commission.

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

<table>
<thead>
<tr>
<th>Editor</th>
<th>Manos Athanatos (FORTH)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contributors</strong></td>
<td><strong>Reviewers</strong></td>
</tr>
<tr>
<td>Leonidas Kallipolitis</td>
<td>AEGIS</td>
</tr>
<tr>
<td>Rubén Trapero Burgos, Jesús Villalobos Nieto, Antonio Álvarez Romero, Rodrigo Díaz Rodríguez</td>
<td>ATOS</td>
</tr>
<tr>
<td>Ovidiu Michaila, Vlad Ursu</td>
<td>BD</td>
</tr>
<tr>
<td>Haim Nachmias, Yana Fesh</td>
<td>COMSEC</td>
</tr>
<tr>
<td>Barbara Lunel</td>
<td>CSI</td>
</tr>
<tr>
<td>Jasmin Cosic</td>
<td>DB</td>
</tr>
<tr>
<td>Pascal Papagrigoriou, Sifniadis Panagiotis</td>
<td>EMP</td>
</tr>
<tr>
<td>Manos Athanatos, Christos Papachristos, Sotiris Ioannidis</td>
<td>FORTH</td>
</tr>
<tr>
<td>Ferrán Rodríguez, Manel Sanz and Paco Sánchez</td>
<td>HCPB</td>
</tr>
<tr>
<td>Jordi Forné, José Estrada, Eva Marín Tordera</td>
<td>TUD</td>
</tr>
<tr>
<td>Apostolos Fournaris</td>
<td>UOP</td>
</tr>
<tr>
<td>Denis Guilhot, Alejandro Lampropoulos, Ignasi Garcia-Mila</td>
<td>WOS</td>
</tr>
</tbody>
</table>

Document history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>05/09/2018</td>
<td>Manos Athanatos (FORTH), Antonio Álvarez (ATOS)</td>
<td>Table of contents</td>
</tr>
<tr>
<td>0.1</td>
<td>04/10/2018</td>
<td>Manos Athanatos(FORTH)</td>
<td>Table of contents revised</td>
</tr>
<tr>
<td>0.2</td>
<td>17/10/2018</td>
<td>Manos Athanatos(FORTH), Antonio Álvarez (ATOS)</td>
<td>Table of contents revision after the GA discussions.</td>
</tr>
<tr>
<td>0.3</td>
<td>17/10/2018</td>
<td>Haim Nachmias (COMSEC)</td>
<td>Table of contents revisions (section 3)</td>
</tr>
<tr>
<td>0.4</td>
<td>13/11/2018</td>
<td>Rubén Trapero (ATOS), Jesús Villalobos (ATOS), Antonio Álvarez (ATOS)</td>
<td>Input to sections 2.1 and 2.2.1</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Author(s)</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>0.4</td>
<td>12/12/2018</td>
<td>Manos Athanatos (FORTH), (WOS), (UPC), (UoP)</td>
<td>Incorporate all provided input and revised ToC.</td>
</tr>
<tr>
<td>0.5</td>
<td>22/12/2018</td>
<td>Manos Athanatos (FORTH), Leonidas Kallipolitis (AEGIS)</td>
<td>More input incorporated</td>
</tr>
<tr>
<td>0.6</td>
<td>11/02/2019</td>
<td>Denis Guilhot (WOS), Panagiotis Sifniadis (EMP), Yana Fesh (COMSEC), Haim Nachmias (COMSEC), Eva Marin Tordera (UPC), Manos Athanatos (FORTH)</td>
<td>Updates to sections 3, 6, 9</td>
</tr>
<tr>
<td>0.7</td>
<td>20/02/2019</td>
<td>José Estrada, Jordi Forné (UPC)</td>
<td>Update to section 3.3.7 Input to section 3.5.11</td>
</tr>
<tr>
<td>0.7</td>
<td>21/02/2019</td>
<td>Antonio Álvarez (ATOS), Rubén Trapero (ATOS)</td>
<td>Update figure 1 in section 2.1. Updated final figure in section 2.1. Minor corrections in the executive summary. Removing second figure of section 2.1, which was redundant, as requested. Corrections and updates in section 2.1. Minor corrections in 2.2.1.1. Change in figure in section 2.2.1.2. Update figure in section 2.2.1.5. Minor corrections in section 2.2.2. Minor corrections in section 2.2.6. Updated figure in section 2.2.6.1.2. Updated figure in section 2.2.8. Input to section 2.2.1.4. Input to section 2.3.1. Minor correction in section 2.3.4. Input to section 2.3.5. Refinement of section 3.2.1.2, 3.2.1.3. Contribution to section 4. Some comments added on the right margin addressed to partners</td>
</tr>
<tr>
<td>0.7</td>
<td>14/03/2019</td>
<td>Haim Nachmias (COMSEC)</td>
<td>Updates to section 3, minor styling fixes thought the document</td>
</tr>
<tr>
<td>0.7</td>
<td>18/03/2019</td>
<td>Panagiotis Sifniadis (Empelor)</td>
<td>Updates to Empelor specific content, marking of comments as done.</td>
</tr>
<tr>
<td>0.7</td>
<td>18/03/2019</td>
<td>Haim Nachmias (COMSEC)</td>
<td>Updates to section 3</td>
</tr>
<tr>
<td>0.7</td>
<td>19/03/2019</td>
<td>Leonidas Kallipolitis (AEGIS)</td>
<td>Addressed remaining comments, updates in section 4, fixed figure references</td>
</tr>
<tr>
<td>0.7</td>
<td>19/03/2019</td>
<td>José Estrada, Jordi Forné (UPC)</td>
<td>UPC new input and comments resolved.</td>
</tr>
<tr>
<td>0.7</td>
<td>20/03/2019</td>
<td>Ovidiu Michaila, Vlad Ursu (BD)</td>
<td>BD new input and comments resolved.</td>
</tr>
<tr>
<td>0.7</td>
<td>20/03/2019</td>
<td>Haim Nachmias (COMSEC)</td>
<td>Added content to section 2.3.2, minor updates to section 3 and section 8.1.1.</td>
</tr>
<tr>
<td>0.7</td>
<td>21/03/2019</td>
<td>Antonio Álvarez (ATOS), Rubén Trapero (ATOS), Rodrigo Díaz (ATOS)</td>
<td>Comment added in section 2.2.1.4, 2.2.3. Minor update in section 4. Input to section 8.2.</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Authors/Comments</td>
<td>Details</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>0.7.10</td>
<td>22/03/2019</td>
<td>Apostolos Fournaris (UoP)</td>
<td>Updated Up contribution on the Deliverable using the provided comment</td>
</tr>
<tr>
<td>0.7.11</td>
<td>22/03/2019</td>
<td>Denis Guilhot(WOS)</td>
<td>updates in sections: 2.2.7, 2.3.5 and 3.2.1</td>
</tr>
<tr>
<td>0.8</td>
<td>29/03/2019</td>
<td>Manos Athanatos</td>
<td>Final consolidated version.</td>
</tr>
<tr>
<td>0.8.1</td>
<td>03/04/2019</td>
<td>Antonio Álvarez (ATOS), Jasmin Cosic (DB), Barbara Lunel (CSI)</td>
<td>Updating figures 1, 16, 44, 45. Updating section 2.3.1.1. Input to section 2.3.1.3. Minor refinement in section 2.3.5</td>
</tr>
<tr>
<td>0.8.2</td>
<td>11/04/2019</td>
<td>Ilias Spais, Leonidas Kalipolitis (AEGIS)</td>
<td>First Internal review(QA1)</td>
</tr>
<tr>
<td>0.8.3</td>
<td>16/04/2019</td>
<td>Antonio Álvarez (ATOS)</td>
<td>Comments dropped in the front cover, executive summary. Update of figure 1. Fixing figure 3. Comment addressed regarding figure 4. Comment addressed in section 2.2.1. Comment addressed in section 2.2.1.1. Input to section 2.2.1.6. Comments addressed in section 2.2.8. Update of figure 44. Input to section 2.3.1.2.</td>
</tr>
<tr>
<td>0.8.4</td>
<td>16/04/2019</td>
<td>Manos Athanatos(FORTH)</td>
<td>Integrated first internal review comments and changes.</td>
</tr>
<tr>
<td>0.8.5</td>
<td>17/04/2019</td>
<td>Christian Schlehuber (DB)</td>
<td>First Internal review(QA2)</td>
</tr>
<tr>
<td>0.8.6</td>
<td>17/04/2019</td>
<td>Antonio Álvarez (ATOS)</td>
<td>Fixings in several figures.</td>
</tr>
<tr>
<td>0.8.7</td>
<td>22/04/2019</td>
<td>Antonio Álvarez (ATOS), Rubén Trapero (ATOS)</td>
<td>Several comments addressed.</td>
</tr>
<tr>
<td>0.9</td>
<td>24/04/2019</td>
<td>Denis Guilhot(WOS), Ovidiu Michaila(BD), Apostolos Fournaris (UoP), Manos Athanatos(FORTH), Christian Schlehuber (DB)</td>
<td>Final version. Final Comments addressed. Final input to 5.2.</td>
</tr>
<tr>
<td>0.10</td>
<td>25/04/2019</td>
<td>Ruben Trapero (ATOS)</td>
<td>Final quality check</td>
</tr>
<tr>
<td>1.0</td>
<td>26/04/2019</td>
<td>Antonio Álvarez (ATOS)</td>
<td>Version for submission</td>
</tr>
</tbody>
</table>
Index

Glossary .................................................................................................................. 7
List of Tables ........................................................................................................... 12
Executive Summary ................................................................................................. 13

1 Introduction ........................................................................................................... 15
  1.1 Purpose and scope of the document ............................................................... 15
  1.2 Structure of the document .............................................................................. 15
  1.3 Relationship to other project outcomes ......................................................... 16
  1.4 Methodology .................................................................................................. 16

2 The Final integrated CIPSEC Framework ................................................................ 17
  2.1 The integration environment .......................................................................... 17
  2.2 CIPSEC Tools ................................................................................................ 21
      2.2.1 Anomaly detection reasoner ................................................................. 21
      2.2.2 Data anonymization and privacy (UPC) ................................................. 31
      2.2.3 Network Intrusion Detection System .................................................. 32
      2.2.4 Gravity Zone antimalware solution ..................................................... 32
      2.2.5 Identity Access – Secocard ................................................................. 36
      2.2.6 Honeypots ............................................................................................ 37
      2.2.7 Integrity Management – Hardware Security Module ............................ 45
      2.2.8 DoSSensing jamming detector ............................................................. 47
  2.3 Services .......................................................................................................... 51
      2.3.1 Contingency plans ............................................................................... 51
      2.3.2 Vulnerability assessment ...................................................................... 56
      2.3.3 Updating and Patching ......................................................................... 58
      2.3.4 Forensics service .................................................................................. 62
      2.3.5 Training Service .................................................................................... 68

3 CIPSEC Final Prototype ....................................................................................... 77
  3.1 Overview ........................................................................................................ 77
  3.2 Prototype Architecture .................................................................................... 78
      3.2.1 Offline Deployment ............................................................................... 78
      3.2.2 Cloud Deployment ............................................................................... 81
  3.3 CIPSEC Framework Tools of the Prototype .................................................. 83
      3.3.1 ATOS XL-SIEM & IDS ....................................................................... 83
      3.3.2 UPC Dashboard ..................................................................................... 83
      3.3.3 Bitdefender GravityZone ...................................................................... 84
      3.3.4 FORTH’s Honeypot ............................................................................. 84
      3.3.5 AEGIS Forensics ............................................................................... 87
      3.3.6 WOS DoSSensing ............................................................................... 88
      3.3.7 UPC Data Privacy ................................................................................ 89
      3.3.8 Empelor’s Secocard ............................................................................ 89
      3.3.9 UoP HSM ............................................................................................ 90
  3.4 Prototype Simulators ....................................................................................... 91
      3.4.1 Rapid SCADA ....................................................................................... 91
      3.4.2 PLC Simulator ..................................................................................... 91
      3.4.3 Medical Web Site ................................................................................ 92
3.5 Demonstration Scenarios

3.5.1 Network Scan

3.5.2 Vulnerability Exploit

3.5.3 Modbus Man-in-the-Middle

3.5.4 Malware by USB

3.5.5 Downloading Malware

3.5.6 Uploading Malware

3.5.7 Denial of Service by CPU Load

3.5.8 Wireless Jamming Simulation

3.5.9 Logging in with Smart Card

3.5.10 Message Integrity Failure Protection Using the HSM

3.5.11 Anonymizing Cyber Security Logs

4 CIPSEC Framework Innovations

5 CIPSEC final evaluation settings and assessment summary

5.1 Final Evaluation Settings

5.2 Framework Assessment Summary

6 Conclusions

7 Appendix

7.1 CIPSEC Unified Dashboard

7.1.1 CIPSEC Dashboard User Guide

7.1.2 Technical aspects and installation

7.2 Deployment recommendations
## Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>ARP</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>AVT</td>
<td>AEGIS Visualization Toolkit</td>
</tr>
<tr>
<td>BD</td>
<td>BitDefender</td>
</tr>
<tr>
<td>BEST</td>
<td>BitDefender's Endpoint Security Tools</td>
</tr>
<tr>
<td>CCM</td>
<td>Counter with Cbc-Mac</td>
</tr>
<tr>
<td>CERT</td>
<td>Computer Emergency Response Team</td>
</tr>
<tr>
<td>CI</td>
<td>Critical Infrastructure</td>
</tr>
<tr>
<td>CIPI</td>
<td>Critical Infrastructure Performance Indicators</td>
</tr>
<tr>
<td>CSIRT</td>
<td>Computer Security Incident Response Team</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed Denial of Service</td>
</tr>
<tr>
<td>DMZ</td>
<td>Demilitarized Zone</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>DR</td>
<td>Disaster Recovery</td>
</tr>
<tr>
<td>ECDH</td>
<td>Elliptic-curve Diffie–Hellman</td>
</tr>
<tr>
<td>ECDSA</td>
<td>Elliptic Curve Digital Signature Algorithm</td>
</tr>
<tr>
<td>EPP</td>
<td>EndPoint Protection</td>
</tr>
<tr>
<td>FEA</td>
<td>Field Element Area</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-Programmable Gate Array</td>
</tr>
<tr>
<td>FVT</td>
<td>Forensics Visualization Tool</td>
</tr>
<tr>
<td>GSoC</td>
<td>Google Summer of Code</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>GZ</td>
<td>Gravity Zone</td>
</tr>
<tr>
<td>HA</td>
<td>High Availability</td>
</tr>
<tr>
<td>HMAC</td>
<td>keyed-Hash Message Authentication Code</td>
</tr>
<tr>
<td>HMI</td>
<td>Human-machine Interface</td>
</tr>
<tr>
<td>HSM</td>
<td>Hardware Security Module</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>HVI</td>
<td>HyperVisor memory Introspection</td>
</tr>
<tr>
<td>ICS</td>
<td>Industrial Control System</td>
</tr>
<tr>
<td>ICS/SCADA</td>
<td>Industry Control Systems/Supervisory Control and Data Acquisition</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPS</td>
<td>Intrusion Prevention System</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MISP</td>
<td>Malware Information Sharing Platform</td>
</tr>
<tr>
<td>MitM</td>
<td>Man-in-the-Middle</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MongoDB</td>
<td>Mongo Database</td>
</tr>
<tr>
<td>MySQL</td>
<td>My Structured Query Language</td>
</tr>
<tr>
<td>NIDS</td>
<td>Network Intrusion Detection System</td>
</tr>
<tr>
<td>OAuth</td>
<td>Open Authentication</td>
</tr>
<tr>
<td>OT</td>
<td>Operation Technology</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>PoC</td>
<td>Proof of Concept</td>
</tr>
<tr>
<td>REST</td>
<td>RePresentational State Transfer</td>
</tr>
<tr>
<td>RTU</td>
<td>Remote Terminal Unit</td>
</tr>
<tr>
<td>SCADA</td>
<td>Supervisory Control and Data Acquisition</td>
</tr>
<tr>
<td>SDR</td>
<td>Software Defined Radio</td>
</tr>
<tr>
<td>SIEM</td>
<td>Security Information and Event Management</td>
</tr>
<tr>
<td>SoC</td>
<td>System on Chip</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>SVN</td>
<td>Apache Subversion</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual Machine</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>XMPP</td>
<td>eXtensible Messaging and Presence Protocol</td>
</tr>
<tr>
<td>XSS</td>
<td>Cross-site scripting</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Detailed diagram of the CIPSEC integration environment</td>
</tr>
<tr>
<td>2</td>
<td>Phases for integration of the CIPSEC framework</td>
</tr>
<tr>
<td>3</td>
<td>Fully distributed approach for integration</td>
</tr>
<tr>
<td>4</td>
<td>Hybrid distributed approach for integration</td>
</tr>
<tr>
<td>5</td>
<td>Local deployment (offline)</td>
</tr>
<tr>
<td>6</td>
<td>Multitenancy approach for event processing</td>
</tr>
<tr>
<td>7</td>
<td>Policies configured per pilot</td>
</tr>
<tr>
<td>8</td>
<td>Correlation bolts configured per pilot</td>
</tr>
<tr>
<td>9</td>
<td>The Anomaly Detection Reasoner GUI is embedded in the dashboard</td>
</tr>
<tr>
<td>10</td>
<td>Single sign on implemented by the XL-SIEM</td>
</tr>
<tr>
<td>11</td>
<td>Deployment at the XL-SIEM to integrate the Data Anonymization Tool</td>
</tr>
<tr>
<td>12</td>
<td>Updating and patching process in the Atos XL-SIEM Agent</td>
</tr>
<tr>
<td>13</td>
<td>Updating and patching tool to upload new tools versions</td>
</tr>
<tr>
<td>14</td>
<td>Updating and patching service running at the Atos XL-SIEM agent</td>
</tr>
<tr>
<td>15</td>
<td>Updating and patching panel at the CIPSEC dashboard</td>
</tr>
<tr>
<td>16</td>
<td>Interface and communications of the Cyberagent</td>
</tr>
<tr>
<td>17</td>
<td>Types of Secocard events defined at the XL-SIEM agent</td>
</tr>
<tr>
<td>18</td>
<td>Types of HSM events defined at the XL-SIEM agent</td>
</tr>
<tr>
<td>19</td>
<td>Event types defined at the XL-SIEM server for Secocard events</td>
</tr>
<tr>
<td>20</td>
<td>Event types defined at the XL-SIEM server for HSM events</td>
</tr>
<tr>
<td>21</td>
<td>Directives for Secocard related alerts</td>
</tr>
<tr>
<td>22</td>
<td>Directives for HSM related alerts</td>
</tr>
<tr>
<td>23</td>
<td>Secocard and HSM events received by the XL-SIEM</td>
</tr>
<tr>
<td>24</td>
<td>Secocard and HSM related alerts triggered by the XL-SIEM</td>
</tr>
<tr>
<td>25</td>
<td>Interface of the backend of the anonymization tool with the XL-SIEM and the dashboard</td>
</tr>
<tr>
<td>26</td>
<td>Interface of the NIDS sensor and the Cyberagent</td>
</tr>
<tr>
<td>27</td>
<td>The Gravity Zone antimalware solution GUI is embedded in the dashboard</td>
</tr>
<tr>
<td>28</td>
<td>Interface and communications of the backend of the Gravity Zone Antimalware solution</td>
</tr>
<tr>
<td>29</td>
<td>Interface of the Gravity Zone agent with the backend of the Gravity Zone server</td>
</tr>
<tr>
<td>30</td>
<td>Roadmap of BitDefender’s solutions</td>
</tr>
<tr>
<td>31</td>
<td>Interface of the Secocard with the CyberAgent</td>
</tr>
<tr>
<td>32</td>
<td>Interfaces and communications of the backend of the honeypot server</td>
</tr>
<tr>
<td>33</td>
<td>Communication between the honeypot and the backend of the honeypot server</td>
</tr>
<tr>
<td>34</td>
<td>The honeypot GUI embedded in the dashboard</td>
</tr>
<tr>
<td>35</td>
<td>Integration of Honeypots' Dashboard in Unified Dashboard</td>
</tr>
<tr>
<td>36</td>
<td>Home Page of FORTH’s Honeypots Visualization Tool</td>
</tr>
<tr>
<td>37</td>
<td>Top IP/Port Statistics Page of FORTH’s Honeypots Visualization Tool</td>
</tr>
</tbody>
</table>
Figure 105. Single-Sign-On Architecture ......................................................... 127
Figure 104. Partners menu in the CIPSEC Dashboard. .................................... 126
Figure 103. Updating and Patching service in the CIPSEC Dashboard. ............... 125
Figure 102. Training courses in the CIPSEC Dashboard. ............................... 124
Figure 101. Contingency plan in the CIPSEC Dashboard. ............................... 123
Figure 100. Managing Vulnerability Assessment reports ............................... 122
Figure 101. Managing Vulnerability Assessment reports ............................... 122
Figure 102. Training courses in the CIPSEC Dashboard. ............................... 124
Figure 103. Updating and Patching service in the CIPSEC Dashboard. ............... 125
Figure 104. Partners menu in the CIPSEC Dashboard. .................................... 126
Figure 105. Single-Sign-On Architecture ......................................................... 127
List of Tables

Table 1. XL-SIEM agent installed on hybrid approach ................................................................. 20
Table 2. Structure of the contingency plan template used in CIPSEC ........................................... 51
Table 3. General courses for cybersecurity topics ........................................................................... 69
Table 4. Specialized courses on Critical Infrastructures ................................................................. 70
Table 5. Courses focused on the CIPSEC platform ....................................................................... 70
Table 6. Offline prototype virtual machines .................................................................................... 80
Table 7. Offline prototype virtual machines on the additional host .................................................. 80
Table 8. Other devices connected to the offline prototype ............................................................ 80
Table 9. Online prototype virtual machines ................................................................................... 82
Table 10. Evaluation Results for Transportation pilot ................................................................. 107
Table 11. Evaluation Results for Health pilot .................................................................................. 108
Table 12. Evaluation Results for Environmental pilot .................................................................... 109
Table 13. Affordability analysis with yearly figures ....................................................................... 110
Executive Summary

This deliverable describes the final integrated CIPSEC Framework summarizing all the activities that took place in Task 2.5 “From the prototype to the final CIPSEC security framework”. It describes the final Integrated Framework, as it was developed throughout the duration of CIPSEC project (Section 2). It discusses the core components of the framework developed (Section 2.2), the interconnection between the different components (Section 2.1), the final deployed services (Section 0) as well as the adjustments to each tool through the lessons learned from the deployment to the different pilot Infrastructures (Section 2.2) and the final evaluation settings with an overall assessment summary of the framework (Section 5).

Building upon the feedback received from the deployment of the Initial version of the CIPSEC framework to our pilots, all the evaluation and validation results, the advances in technologies and the proposed tools, this document will describe the final version of CIPSEC along with the adjustments that were made to the prototype and each incorporating component (Section 3).

Moreover, a detailed user guide of the unified CIPSEC dashboard (Section 7.1), along with the deployment recommendations and assets road mapping is included in the document. This information can assist end users, providing a specific methodology, on how to adjust the CIPSEC framework to their case. The flexibility of CIPSEC can be exploited by all types of Critical Infrastructures, not only use cases from Transportation, Health and Environmental Monitoring sectors, which are the specific pilots of the CIPSEC project.

The results presented in this report, provide a summary of those presented in D2.1[M9], D2.2[M18], D2.6[M24], D4.2[M27], D4.4[M36], an enhancement the results of D2.4[M18], D2.5[M24] and a description of the work of Task 2.5 – “From the prototype to the Final CIPSEC security Framework” (Sections 2, 3), thus concluding the work WP2 – “Development of the CIPSEC security framework for Critical Infrastructure environments”

In order to assist the reader to navigate more easily in the document we have included a small description of the major developments presented in this deliverable:

The Integration Environment

The final version of CIPSEC framework is presented, which addresses CIPSEC architecture that was presented in D2.2 and consolidated in the initial integration of the framework reported in D2.5. It provides the integration of the different tools provided by the CIPSEC partners, with the Atos XL-SIEM being the central entity that acts as the anomaly detector reasoner. More details in Section 2.1

CIPSEC Tools

A detailed analysis and presentation of the tools that comprise the final integrated CIPSEC framework is present in this section, along with the integration process to the common GUI, samples configurations and the road-mapping of each tool presenting how its evolution after the pilots’ deployment and the validation process during the last year of the project. More details in Section 2.2

Services

The final version of each of the five CIPSEC services is presented in detail. Moreover, the integration process and interconnection with the unified CIPSEC framework is present for each service, as well as, how the services are present in the common unified dashboard. More details in Section 0

Final CIPSEC prototype Overview

The CIPSEC prototype is a dedicated environment of the CIPSEC framework, which also includes critical infrastructure simulators and attacking tools. It is mainly installed on a laptop. This prototype was proven an invaluable tool for the validation, refinement and demonstration of CIPSEC framework. More details in Section 3.1

Prototype Architecture

The prototype architecture was developed to be versatile and supporting two flavours of the prototype that are in-place and operated independently: Offline deployment prototype that supports a fully local installation of the whole framework with no Internet connectivity; and the Cloud deployment prototype that supports remotely deployed and operated tools in each responsible partner cloud environment. The local components are connected to the Internet and integrated with the remote components. More details in Section 3.2
CIPSEC Framework Tools (in the prototype)
The CIPSEC Framework products that form the prototype components for both offline and cloud deployments, are presented. Furthermore, the configurations and the adjustments needed in order to fit the two prototype architectures are presented, along with integration of the common UI in the prototype deployments. More details in Section 3.3

Prototype Simulators
A number of Critical Infrastructure systems simulators have been integrated in the prototype in order to simulate various components, corresponding to different CI sectors, that are protected by the CIPSEC framework. This allows to perform attack simulations that demonstrated the efficiency of CIPSEC into a secure simulated environment. Additionally, these simulations allow us to test new features and measure the performance of CIPSEC. More details in Section 3.4

Demonstration Scenarios
The demonstration scenarios section describes how to execute various testing scenarios and how each scenario is detected by the tools. The demonstration scenarios allow to assure the framework works as intended. Further demonstrations can be performed ad-hoc, according to specific requirement or to demonstrate specific capability. More details in Section 3.5

CIPSEC Framework Innovations
The innovative features that have been developed by the solution partners in the final eighteen months of the project, are discussed in this section. Some of the innovation items have been brought forward during the actual integration of the solutions into the complete framework, whilst others are offered by the final framework itself. Details for these all these innovations can be found in Section 4

Final Evaluation Settings
The CIPSEC evaluation was part of WP4 and is the last in a series of phases that amongst others include design, development, deployment and integration. In order to better evaluated the efficiency and correct operation of CIPSEC unified framework a series of tests were devised and introduced in D4.1, revised in D4.2 and D4.3 during the tests to the pilots’ premises. Through that procedure the evaluation settings, that were used to evaluate the complete CIPSEC framework, were derived and reported in D4.4 and are also summarized in Section 5.1

Framework Assessment Summary
The complete report on the tests performed at each of the test sites (Rail, Hospital, Environment) can be found in Deliverable 4.3 “Prototype Demonstration: Field trial results”. The main findings and the summary of all the evaluations’ process and the results can be found in Section 5.2

User guide of the unified CIPSEC dashboard
The CIPSEC dashboard is a web interface that aggregates all the tools of the CIPSEC framework into one simple user-friendly interface. From this interface, it is possible to control all the functionalities of the CIPSEC framework, have a quick view of the system status and in case of any threat, the dashboard provides a quick look at all the relevant data related to the detected threat and a deep view and analysis of the data using the various embedded tools of the framework. More details in Section 7.1

Deployment Recommendations
The deployment recommendations of the three main possible deployments of the CIPSEC Framework, namely: public cloud deployment, on premise deployment and hybrid deployment are presented. The recommendations to be taken into account before choosing the right approach for your infrastructure are presented in detail in Section 7.2
1 Introduction

1.1 Purpose and scope of the document

Deliverable D2.7 is the output of the last task of Work Package 2 namely Task 2.5 of CIPSEC description of action. It consists of two main parts: i) a single public report of the activities carried out in T2.1, T2.3 and T2.4 including assets road mapping (section 2), CIPSEC prototype description (section 3) and final evaluation settings (section 5) and ii) the final version of the CIPSEC platform (SW – release) which is also described in section 2. Additionally, the final CIPSEC Framework Innovations are reported in section 4. Finally, in the Appendix and more specifically in section 7.1 we provide a user guide of the Unified Dashboard and in section 7.2 general deployment recommendations.

This document is building on top of the work reported in D2.5 “Final version of CIPSEC Unified Architecture and Initial Version of the CIPSEC Framework Prototype” and describes the optimizations that took place, after the integration of the framework to the pilots and the validation of the framework, delivering the final optimised version of the CIPSEC framework. In order to do so we took into account the work of WP3 and the reports of the integration process to each pilot. Moreover, any issues that arose during the validation process fuelled new optimization for the individual tools and the Unified CIPSEC Framework.

The document presents the final release of the framework prototype and all software components it is composed of, along with the road mapping the optimizations that each component integrated. First, the core components of the final integrated CIPSEC framework are presented; the road map for each of the component is presented focusing on the optimizations that took place in the last phase of the project. All the details of the final framework are extensively described in section 2.

The final prototype architecture was updated with respect to D2.5 and two approaches were proposed in order to build a prototype that is able to operate in different scenarios (offline and online architecture). New capabilities were also included in the prototype addressing the respective architecture design. By incorporating validation tools the prototype became versatile and able to demonstrate the complete range of CIPSEC’s features. Section 3 describes the final prototype thoroughly.

After the description of the solution and the final prototype, the Innovations of the whole framework are discussed in section 4. The main innovation actions were reported on M18, but during the last year of the project and through the evaluation process a number of innovative features have been developed by different partners and integrated to the final version of the framework.

In section 5 the final settings and the methodology that were used during the evaluation of the framework, that took part in WP4, are summarized. Along with the evaluation settings, a summary of the evaluation tests and the results that were collected during the evaluation process that took place in each of the three pilot cases is also presented.

In summary, this document concludes the development actions that took place within CIPSEC and describes in detail how the framework has evolved throughout the project. Moreover, it refers to the evolution and capabilities of the final version of CIPSEC’s prototype, the summary of the evaluation activities of the framework, a summary of the CIPSEC’s innovation and finally a User Guide for the Unified Dashboard and Deployment recommendations for the whole framework.

1.2 Structure of the document

The structure of the document is the following:

- Executive summary stating at a glance the scope of the document and the main topics covered.
- Section 1 introduces the work reported.
- Section 2 documents the final version of the integrated CIPSEC framework, focusing on the various optimizations with respect to that was published in D2.5 (M24).
- Section 3 describes the Final version of the prototype, the components that are included and the capabilities

CIPSEC. Enhancing Critical Infrastructure Protection with innovative SECurity framework
• Section 4 describes the Innovation of CIPSEC framework
• Section 5 summarizes the evaluation settings and the evaluation of the framework, building on top of D4.1, D4.2 and D4.4
• Section 6 provides the conclusion of the document.
• Section 7 includes an Appendix which describes the CIPSEC Unified Dashboard and the Deployment Recommendations.

1.3 Relationship to other project outcomes

This document is the final outcome of Work Package 2. The work described here, builds upon Work Package 2 and more specifically the work done in tasks T2.1, T2.3 and T2.4. It builds on top of the results obtained in T2.1, T2.3, T2.4 and includes the work done in T2.5. Provides the final results of T2.1, T2.3 and T2.4. It is based on the latest version and development status that is reported in D2.5, updating the status of the framework and providing its final version. Apart from the new development and features that were develop within T2.5 a newer more versatile version of the prototype is described leveraging the work that was initially included in D2.5

This deliverable and the respective Task T2.5 received extensive input from WP3, which describes the integration of the framework to CIPSEC’s pilots. Using this input each component and the framework as a whole was optimised to achieve a higher readiness level and better modularity in order to be applicable to more critical sectors.

Another important aspect of this deliverable is evaluation results and the final evaluation settings. Utilizing the work performed in WP4 and more specifically D4.1, D4.3 and D4.4 we are able to report a summary of the evaluation results that were produced during the final validation of the framework that occurred during the last year of the project.

Furthermore, the Innovations of the CIPSEC framework are included in the current document enhancing the work reported in D2.3, deriving new information from T2.1 and T2.5 and the optimizations occurred in the final phase of the project.

1.4 Methodology

As this deliverable summarizes and reports the optimizations that were incorporated into the final CIPSEC framework we had to take into account all the previous work that has been completed in WP2, WP3 and WP4. Using the structure of D2.5 and the methodology used to define CIPSEC’s reference architecture, we reported all the optimizations and the current functionalities of the individual components as well as the refined final integrated version of the framework.

Most of the details of each of CIPSEC’s components already exist in past deliverables (D2.2, D2.3, D2.5). Thus, in this document we are focusing on the nature of the optimizations, new features and the process that each tool underwent in order to be harmonically be integrated into the final unified framework. Moreover, we are focusing on the usage of framework as a whole, the specifications and settings of each component. This document reports all the achievements and the development tasks that executed during the last phase of the project, including new capabilities, innovations, evaluation results and user guide.

All this information has been elaborated and consolidated in this document, with the corresponding revision of the Coordinator, WP Leaders and Project Management Board members (where all partners are represented). The document follows the quality assurance process described in Deliverable D6.1 ‘Project Management Strategy. Project Handbook’.
2 The Final integrated CIPSEC Framework

2.1 The integration environment

Figure 1 represents the final version of CIPSEC framework. CIPSEC framework addresses CIPSEC architecture that was presented in D2.2 and consolidated in the initial integration of the framework reported in D2.5. Figure 1 highlights the integration of the different tools provided by the CIPSEC partners, with the Atos XL-SIEM being the central entity that acts as the anomaly detector reasoner. XL-SIEM consists of two different components:

- The XL-SIEM agent comprises a cyberagent component, which receives events and logs from the rest of the tools, and a NIDS sensor that analyses network traffic and reports to the cyberagent, the relevant events detected. It also contains an updating and patching agent to support the updating and patching service, which allows to update the cyberagent to a new version.

- The XL-SIEM server, which provides with correlation activities based on the events received and stores events and alerts generated in a database. It also provides with a control panel, a GUI that allows to administrate the XL-SIEM and visualize events and alerts. This GUI is part of the CIPSEC Dashboard which is presented in section 7.1.

For the rest of the tools (Secocard, HSM, honeypots, DoSSensing Gravityzone), the main interaction occurs with the ATOS cyberagent, in order to report events based on their respective acquisition activities. Certain tools, such as the WOS’s jammer detector or the FORTH honeypots, also include a GUI for their management and activity visualization, integrated in the CIPSEC dashboard, too.

The forensics capabilities are provided within the AEGIS network. A set of Nagios and Netflow sensors retrieve information from the assets deployed in the infrastructure. These sensors report the generated events to the cyberagent, which normalizes and forwards them to the XL-SIEM server for its storage into the events database. The Aegis Visualization tool elaborates on that information, making available advanced visualization capabilities that are provided by their visualization tool.

Similar to Atos XL-SIEM agent, AEGIS, FORTH and Bitdefender tools include an Updating and Patching agent. Every updating and patching agent communicates with the Updating and Patching manager, which keeps track of the status of every tool and manages their update in case there is a new version available.

A unified dashboard, is developed which integrates the different tools, control panels from the various tools and all the available services. This integration, of tools and services, is based on the usage of OAuth, which allows the usage of Single Sign On capabilities to access all framework’s components.

The rest of the services (training, vulnerability and contingency) have also been added to the integrated dashboard, being able to access to specific reports and information for every pilot directly from the dashboard implemented by UPC.
In order to facilitate the integration of the different tools, an incremental approach has been followed, where the different components were running remotely or locally depending on the integration stage.

**Figure 1. Detailed diagram of the CIPSEC integration environment**

In order to facilitate the integration of the different tools, an incremental approach has been followed, where the different components were running remotely or locally depending on the integration stage.
The integration procedure was organized in three phases:

- **Phase 1: Fully distributed approach.** During this phase the initial integration of the tools was made. In order to simplify the process every tool was running on every partner’s premises, exposing interfaces and servers to the public Internet for testing purposes. To this end, an instance of the XL-SIEM server and an XL-SIEM agent was running on the cloud over the same machine. The syslog server port (514) was made available to receiving events sent from the partners’ tools. The SSH port was also opened for accessing to the HTTPS based XL-SIEM dashboard. In the case of the AEGIS Forensics Visualization Tool, an additional interface was required to allow accessing to the events database at the XL-SIEM (mysql, port 3306).

- **Phase 2: Hybrid distributed approach.** During the second phase a hybrid approach was followed. In this phase the integration with the pilots were made. Also, at this stage, some tools were installed on pilot’s premises (such as FORTH honeypots and AEGIS agents), while others remained located on their own premises (such as Worldsensing DoSSensing jammer detector tool) and others followed a hybrid approach (such as Bitdefender with local antivirus running on pilot premises while the Gravity Zone control panel was running on their premises). To allow the interoperability of different tools running on premises or in the cloud, a mixed approach was also deployed in the XL-SIEM. Two agents were deployed per pilot: one running on premises and another one running in the cloud. Tools installed on premises reported their events to the local XL-SIEM agents while tools running in the cloud reported their events to the corresponding XL-SIEM agent running in the cloud. In total, there are three local XL-SIEM agents running on premises (one per pilot) and three Atos XL-SIEM agents running in the cloud (one per pilot). The local XL-SIEM agents, running on premises, use the syslog standard port (514) to receive events from the tools. The XL-SIEM agents running on cloud are different virtual machines running over the same host, therefore the syslog standard ports of every agent are forwarded to the ports 5006, 5007 and 5008 of the host machine (being 5006 the port for the HCPB pilot, 5007 for the CSI pilot and 5008 for the DB pilot).
For the sake of simplicity, optimization of resources and testing purposes, the same instance of the XL-SIEM server was used to manage the events from the three pilots in this phase. To guarantee the isolation of the different pilots it was implemented a multi tenancy approach for the XL-SIEM server, which allows for separating events from different pilots. Three users were configured at the XL-SIEM, each associated to a pilot. The multitenancy isolation allows that every user is able to see just the events of its pilot. Also, in order to guarantee a proper correlation, not mixing events from different pilots, just events from the same pilots are correlated when detecting security anomalies and triggering alerts. Tool X refers to any tool in the deployment of a specific pilot which is sending logs using the syslog interface.

Table 1. XL-SIEM agent installed on hybrid approach

<table>
<thead>
<tr>
<th>Pilot</th>
<th>XL-SIEM agent on premises</th>
<th>XL-SIEM agent on cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCPB</td>
<td>xlsiem-clinic</td>
<td>xlsiem-clinic-c</td>
</tr>
<tr>
<td>CSI</td>
<td>xlsiem-csi</td>
<td>xlsiem-csi-c</td>
</tr>
<tr>
<td>DB</td>
<td>xlsiem-db</td>
<td>xlsiem-db-c</td>
</tr>
</tbody>
</table>

Figure 4. Hybrid distributed approach for integration
2.2 CIPSEC Tools

2.2.1 Anomaly detection reasoner

Several improvements have been made to the Anomaly Detection Reasoner during the final stages of the framework integration and with respect to what was reported in D2.5. The changes and improvements have been done in several aspects, namely covering multitenancy, support for Single Sign On, integration with the Data Anonymization Tool, inclusion of the tool in the Updating and Patching Service, as well as the development of new plugins and rules for processing the logs coming from Secocard and HSM devices.

2.2.1.1 Multitenancy support

As part of the integration activity done in Phase 2 (hybrid approach) it was required to support multitenancy at the XL-SIEM for the isolation of event processing coming from different pilots. With the goal of optimizing resources and making the integration flexible, we went for a shared correlation engine and different event processing per pilot. Figure 6 represents the approach followed to support multitenancy at the XL-SIEM. The correlation engine that the XL-SIEM uses to correlate events and detect incidents is shared by all pilots. This allows to optimize computational resources. The XL-SIEM just instantiates a new correlation bolt\(^\dagger\) that uses the

\(^\dagger\) In the multitenant approach, event correlation is done per pilot, not mixing events coming from different pilots when correlating them. To be able to achieve this, new instances of the XL-SIEM correlator which are called **correlation bolts**
correlation engine shared by all pilots, correlating events coming from the pilots and not mixing events from different pilots to generate alerts.

![Diagram](image)

**Figure 6. Multitenancy approach for event processing**

For the implementation of this multitenancy several modifications were required at the XL-SIEM:

- **Isolation policies**

  Policies allow to differentiate between events coming from different agents. Therefore, different policies were added to manage events coming from the agents associated to the different pilots. Figure 7 shows the summary of the policies configured for the XL-SIEM, being clinicPolicy, csiPolicy and policyDb the policies configured for the HCPB, CSI and DB pilots respectively.

![Policy Configuration](image)

**Figure 7. Policies configured per pilot**

- **Isolated correlation**

  In the multitenant approach, event correlation is done per pilot, not mixing events coming from different pilots when correlating them. To be able to achieve this, new instances of the XL-SIEM correlator (called correlation bolts) were created, having one correlator per pilot. Every correlator is linked to the policy configured for every pilot to guarantee that the correlator is considering just the events that fulfil with the indicated policy (this is, the events from the corresponding pilot). It is also possible to configure different correlation rules per correlator. This is useful in the case of different security sensors running in different pilots. For example, the Secocard and HSM devices are not used at the HCPB pilot, so it is not necessary to consider the correlation rules that use their
events when configuring the correlator bolt of the hospital pilot, while for the DB pilot it is necessary. Figure 8 shows a summary of the correlation bolts configured for the pilots, where the policies and used by them are specified.

<table>
<thead>
<tr>
<th>Correlation Bolts Configured Per Pilot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>corCICPSEC</td>
</tr>
<tr>
<td>corCICPSEC</td>
</tr>
</tbody>
</table>

Figure 8. Correlation bolts configured per pilot

- Visualization isolation

In the hybrid approach the XL-SIEM server is running in the cloud and shared among the different pilots. The XL-SIEM dashboard is also shared among them. Therefore, a mechanism to isolate the information from different pilots is required. A user-based mechanism was included in the XL-SIEM in order to support isolated dashboards. Three users were included in the XL-SIEM, one per pilot. Every user was associated to a context, understanding the context as the events sent by the agents deployed for to the same pilot and the alerts generated by correlating them. To support this, the field “organization”, configured at the XL-SIEM agent, was used to identify the pilot. The normalized events that the XL-SIEM agent sends to the XL-SIEM server include this field, which is used by the XL-SIEM server to identify the pilot that is sending the event. The XL-SIEM keeps a link between the organization (configured at agents) and the context associated to the users. This link is used at the dashboard to identify the events and alerts to show, filtering only those that matches organization with the context of the user logged in the system.

2.2.1.2 Single sign on support

To securely integrate the XL-SIEM dashboard with the UPC dashboard, OAuth2 authentication has been implemented. With OAuth2 it is possible to authorize UPC dashboard to access to XL-SIEM by embedding its content.

Figure 9. The Anomaly Detection Reasoner GUI is embedded in the dashboard
This process consists on sending from the client computer a token request to the Authorization Server when the user logs in UPC dashboard. This request contains the username, the password, a client id and a client secret. The server returns an access token and the scope. This data is forwarded to XL-SIEM which takes this data and sends a validation request to the Authorization Server to validate the received authorization token. If the validation is successful, the client gets logged in XL-SIEM dashboard through UPC dashboard, being able to navigate through the different pages in the application, such as the “Dashboard Overview”, the “SIEM Events” panel, “Alarms” panel, etc.

2.2.1.3 Compatibility with the Data Anonymization Tool

The Data Anonymization Tool, developed by UPC, retrieves alerts generated by the Anomaly Detection Reasoner, anonymizing them and storing it in a MISP - Open Source Threat Intelligence Platform (previously known as Malware Information Sharing Platform). The XL-SIEM supports several mechanisms to export alerts (export to file, email, messaging queues, etc). Among these mechanisms a message queue based on RabbitMQ was chosen to publish alerts and allow the Data Anonymization Tool to retrieve them. The capability of publishing and subscribing to alerts in real time was the main reason for using this mechanism. This would allow the real-time anonymization as long as new alerts are generated at the XL-SIEM. RabbitMQ is compatible with several approaches to organize producer and consumer queues. Among these mechanisms the fan-out approach allows to publish messages to an exchange queue while consumer would need to create their queues and attach them to the exchange queue available at the server. The RabbitMQ server will replicate the alerts published at the exchange queue and will forward them to the queues that are attached to the exchange queue. Figure 11 depicts the deployment implemented for exporting alerts to the Data Anonymization Tool. A RabbitMQ server was deployed at Atos. The XL-SIEM was attached to the exchange queue eu.cipsec.dw_input, where alerts are exported and consumed by the Data Anonymization Tool from UPC. Details of the Data Anonymization Tool and its integration with the MISP will be given in Section 2.2.2.

---

1 https://olympus.epsevg.upc.edu/oauth2/validation
2 https://www.misp-project.org/
2.2.1.4 Compatibility with the updating and patching service

The updating and patching service has been included in the Atos XL-SIEM agent. The mechanism used to update the tool relies on three main steps, as depicted in Figure 12.

![Diagram](image)

**Figure 12. Updating and patching process in the Atos XL-SIEM Agent**

Three elements of the CIPSEC framework are used during the Updating and Patching service:

- **Data Anonymization Tool**
- **RabbitMQ server**
- **XL-SIEM server**
• An update tool, provided by Bitdefender, which allows to upload the installer of new version of the XL-SIEM agents. The updates for the XL-SIEM agent is packaged as a .deb installer. New updates are uploaded to the CIPSEC update tool, using the GUI represented in Figure 13.

![Image](https://example.com/update-tool)

**Figure 13. Updating and patching tool to upload new tools versions**

• A service running in the Atos XL-SIEM agent. This tool checks every hour the existence of a new version uploaded through the CIPSEC tool to the Updating and Patching service. If a new version is labelled at the Updating and Patching service as to be deployed, it is automatically installed in the XL-SIEM agent. Figure 14 represents the details of the service, which is running and checking every hour for new versions. When a new version is found it is downloaded, installed and the XL-SIEM agent restarted.

![Image](https://example.com/updating-and-patching-service)

**Figure 14. Updating and patching service running at the Atos XL-SIEM agent**

• The status of the Updating and Patching service can be visualized in the CIPSEC dashboard. Figure 15 shows the Updating and Patching panel, where the version of the Atos XL-SIEM agent is shown. The dashboard shows the history of the different versions available, the date when they were available and
the current version installed. The “deploy” button allows to label that version as to be updated, in order to be installed by the Updating and Patching service.

Figure 15. Updating and patching panel at the CIPSEC dashboard

2.2.1.5 Compatibility with Secocard and HSM

Figure 16 represents the tools that are integrated with the XL-SIEM. The last stage of integration carried out in this period with respect to D2.5 is the integration of the Secocard and HSM components. The exchanged messages’ format between Secocard and HSM and the XL-SIEM agent were defined in D2.5. At this stage, new plugins were developed at the XL-SIEM in order to process and normalize those events. Additionally, the XL-SIEM was also configured to able to process and correlate these new types of events.
- At the XL-SIEM agents: new plugins were required to process Secocard and HSM events:
  - Secocard. The plugin_id 130000 was assigned to Secocard events. Additionally, and following the message format and event taxonomy defined in D2.5 for Secocard events, the following plugin_sids, shown in Figure 17, were assigned to the 12 different types of events that the Secocard device is able to send to the XL-SIEM server. Both plugin_id and plugin_sid will be used by the XL-SIEM server to identify the type of event received (and classify it appropriately).

![Figure 17. Types of Secocard events defined at the XL-SIEM agent](image)

- HSM. The plugin_id 140000 was assigned to HSM events. Same as with Secocard, the events taxonomy defined in D2.5 was used for the creation of the plugin, assigning the following plugin_sids shown in Figure 18, to the types of events that the HSM device is able to send to the XL-SIEM server.

![Figure 18. Types of HSM events defined at the XL-SIEM agent](image)
At the XL-SIEM server the new data types were configured to process Secocard and HSM events. Figure 19 and Figure 20 show the configuration that is required at the XL-SIEM to set the new event types, including the plugin_id and plugin_sid.

In order to correlate the events received from Secocard and HSM devices, new Directives were required. Figure 21 and Figure 22 show the new directives required at the XL-SIEM. For instance, for HSM events, when three hsm_password_failure events are received from the same HSM device in less than 60 seconds a new Brute-force attack against an HSM is triggered.
Finally, Figure 23 shows the reception of Secocard and HSM events to the XL-SIEM, while Figure 24 shows the corresponding alerts triggered by the XL-SIEM when correlating the events received.

2.2.1.6 Final Tool Optimisations

The tool has been optimized to enable full integration of the different information sources used to raise relevant events and alarms. The significance of the information provided by the different solutions has been improved, and this has enabled a better event identification and alarm raising. When necessary, new plugins have been added for the identification of new identified necessary events (mainly with Secocard and HSM).

A main activity has been the integration of the anonymization tool from UPC, in particular to the sharing of information with external entities by means of a MISP server. One task was the connection of the XL-SIEM to a RabbitMQ server to receive the events and alarms asynchronously in a queue. A Python implementation was produced and the needed certificates to make it work were added. Also, several JSON samples were analysed and then we made suggestions on fields that could be anonymized. we researched on how this anonymization could be made in a GDPR-compliant way.

Finally, the XL-SIEM has undergone some internal improvements in order to make it work in a more efficient manner, and the tool has become fully multitenant.
2.2.2 Data anonymization and privacy (UPC)

The data anonymization tool implements a privacy protection mechanism over the logs of the CIPSEC components. Since sensitive data is included in these logs, the tool obfuscates such items to make them undistinguishable and thus more private. This approach goes beyond the mere data encryption that renders the data unusable for those entities not entirely trusted. Thus, the anonymization tool allows other entities to take advantage of the utility of the logs of the CIPSEC components while the privacy of sensitive attributes is preserved.

The anonymization tool requires the interaction of three elements: the dashboard, the XL-SIEM monitoring tool, and the MISP platform. The XL-SIEM, located in the ATOS network, feeds the anonymization tool with JSON-formatted logs in real time. Afterwards, the sensitive attributes of these logs are anonymized by the anonymization backend according to the privacy policies defined by the operator. Such policies can be set through a graphical interface embedded in the dashboard. Both the anonymization backend and the graphical interface lie in the UPC network. Finally, the anonymized logs are stored in a MISP platform where can be safely shared. Figure 25 illustrates the interaction of the components mentioned to carry out the anonymization process.

![Figure 25. Interface of the backend of the anonymization tool with the XL-SIEM and the dashboard](image)

The anonymization backend, which is written in Python, reads a privacy policy file based on the parameters passed by the operator through the graphical interface. These policies implement the strategies of suppression, generalization or pseudonymization over the sensitive attributes of the cybersecurity logs. For the sake of usability, the graphical interface enables the operator just to choose the privacy protection policy and to start or stop the anonymization process. It is worth noting that the scope of the user of the dashboard restricts the logs she is able to anonymize.

2.2.2.1 Final Tool Optimisations

In comparison to the description of the tool provided in Section 3.1.3 of deliverable D2.5, the following changes have been carried out:

Although the initial version of the tool was developed in Java, we switched to Python in May 2018. This decision was taken because Python better supports several functions such as handling with regular expressions and facilitates an easier integration with the CIPSEC framework (especially with the ATOS XL-SIEM component). It also has native support to JSON.
Instead of working with JSON files, the Data Anonymization Tool directly obtain the logs from the ATOS XL-SIEM tool and, after anonymizing them, the output is written into a MISP (Malware Information and threat Sharing Platform) repository. In other words, the input/output of the tool are no longer files, but there is a continuous flow of information generated by the XL-SIEM, anonymized by the anonymization tool and written into the MISP repository.

The selection of the anonymization policy is carried out through the dashboard. Instead of loading a file specifying the selected policies, the user interface integrated into the dashboard allows a selection among three preconfigured anonymizing policies (basic, intermediate and advanced).

Although there are only three preconfigured anonymization policies to be selected by the dashboard, these policies may be different for different operators running different Cis. Therefore, a user-friendly tool for generating policies was also implemented.

The tool has been adapted to be able to work both through the cloud and in-premises mode. The main difference affecting the anonymization tool is that working in-premises mode, it may not be connectivity to the MISP repository. In this case, instead of directly writing to the MISP database, the anonymized logs are temporarily stored in a local file.

2.2.3 Network Intrusion Detection System

The Network Intrusion Detection System has been integrated with the Cyber Agent (as depicted in the figure below) in order to provide the XL-SIEM with a wide range of events following the Event Taxonomy presented in D2.5. Specific plugins have been developed in the Cyber Agent to transform the raw logs coming from the NIDS into events understandable by the XL-SIEM and that can be eventually correlated to obtain security alarms, whether having lower or higher severity. The richness of sources existing in CIPSEC has allowed to make the most of the NIDS to obtain advanced alarms with valuable information.

![Interface of the NIDS sensor and the Cyberagent](image)

2.2.4 Gravity Zone antimalware solution

GravityZone is a business security solution built from ground-up for virtualization and cloud to deliver security services to physical endpoints, mobile devices, virtual machines in private, public cloud and Exchange mail servers.

GravityZone is one product with a unified management console available in the cloud, hosted by Bitdefender, or as one virtual appliance to be installed on company's premises, and it provides a single point for deploying, enforcing and managing security policies for any number of endpoints and of any type, in any location.

---

2 CIPSEC D2.6: CIPSEC Evaluation Plan
GravityZone delivers multiple layers of security for endpoints and for Microsoft Exchange mail servers: antimalware with behavioural monitoring, zero-day threat protection, application control and sandboxing, firewall, device control, content control, anti-phishing and antispam.

The unique architecture of GravityZone allows the solution to scale with ease and secure any number of systems. GravityZone can be configured to use multiple virtual appliances and multiple instances of specific roles (Database, Communication Server, Update Server and Web Console) to ensure reliability and scalability.

Each role instance can be installed on a different appliance. Built-in role balancers ensure that the GravityZone deployment protects even the largest corporate networks without causing slowdowns or bottlenecks. Existing load balancing software or hardware can also be used instead of the built-in balancers, if present in the network.

Delivered in a virtual container, GravityZone can be imported to run on any virtualization platform, including VMware, Citrix, Microsoft Hyper-V, Nutanix Prism, Microsoft Azure.

Integration with VMware vCenter, Citrix XenServer, Microsoft Active Directory, Nutanix Prism Element and Microsoft Azure reduces the effort of deploying protection for physical and for virtual endpoints.

The GravityZone solution includes the following components:

- GravityZone Virtual Appliance with the available roles:
  - Database
  - Update Server
  - Communication Server
  - Web Console (Control Centre)
- Report Builder Virtual Appliance with the available roles:
  - Database
  - Processors
- Security Server
- HVI Supplemental Pack
- Security Agents.

![Figure 27. The Gravity Zone antimalware solution GUI is embedded in the dashboard](image)

---


GravityZone incorporates techniques enabling device hardening and control, detection at both the pre- and on-execution stages, automatic clean-up and rollback action.

Hardening and Control layer to decrease the surface of an attack. This layer includes Patch Management, Full Disk Encryption, Application Whitelisting, Web Filtering, Device control and Firewall.

Detection in pre- and on-execution using static and behavioural analysis with Machine Learning/AI models, Process inspection technologies that are scanning the processes during the entire lifetime providing 0 Trust for all the processes.

GravityZone provides the possibility of sending suspicious files in an isolated sandboxing environment where the files’ behaviour will be analysed and registered.

All these while delivering single-pane-of-glass visibility, alerts and notification, integration and reporting capabilities.

Figure 28. Interface and communications of the backend of the Gravity Zone Antimalware solution

Within the CIPSEC security framework, Bitdefender GravityZone offers protection and security at the endpoint level, providing the best solution in class against internal and external attackers.

The Architecture is composed of two main components, the management console called GravityZone and the endpoint agents called Bitdefender Endpoint Security Tools (BEST).

The BEST agent monitors the systems and reports back to the management console. The management console, receives the events, stores them into the database and forwards the events to the XL-SIEM, CIPSEC Framework.
2.2.4.1 Final Tool Optimisations

Along the implementation of the CIPSEC framework, the Bitdefender GravityZone went through multiple updates and significant additions in terms of unique innovations.

Implemented in January 2017, the Bitdefender Hypervisor Memory Introspection (HVI) is a ground-breaking solution that detects suspicious activities by working directly with raw memory at hypervisor level – a level of insight from which malware cannot hide. The HVI protects virtual machines in data-centres against advanced and sophisticated threats that signature-based engines cannot defeat. HVI enforces strong isolation, ensuring real-time detection of attacks, blocking them as they happen and immediately removing the threats.

Whether the protected machine is a server or a workstation, HVI provides insight at a level that is impossible to achieve from within the guest operating system. By operating at the hypervisor level and leveraging the hypervisor functionalities, HVI overcomes technical challenges of traditional security to reveal malicious activity in datacentres.

By working alongside any endpoint protection (EPP) solution, it provides an unprecedented layer of defence for the most notorious Advanced Persistent Threats (APTs) targeting CIs.

Figure 29. Interface of the Gravity Zone agent with the backend of the Gravity Zone server.

Figure 30. Roadmap of BitDefender’s solutions

Within the last couple of months of CIPSEC implementation, BD GZ integrates layered next-gen endpoint protection and easy-to-use EDR platform to accurately protect enterprises against even the most elusive cyber threats. It offers prevention, automated detection, investigation and response tools so enterprise customers can
protect their digital assets and respond to these threats\(^1\). More, new technologies refinements are conducted each month, these being integrated with the constant updates or additions.

The Bitdefender Network Traffic Security Analytics accurately detects breaches and provides insights into advanced attacks by analysing network traffic. It lets organizations quickly detect and fight sophisticated threats by complementing pre-existing security architecture – network and endpoint – with specialized network-based defence.

It uses AI (Artificial Intelligence) / ML (Machine Learning) and heuristics to analyse network meta-data in real-time and to accurately reveal threat activity and suspicious traffic patterns. With flexible deployment options, Bitdefender Network Traffic Security Analytics is a plug-and-play, out-of-band solution, that focuses on outbound traffic and enables analysis over longer periods of time to accurately detect the most sophisticated malware and APTs with high fidelity.

### 2.2.5 Identity Access – Secocard

Secocard is a generic, programmable, embedded, hardware platform with a focus on security in a small form factor\(^2\). The device is equipped with a powerful, state of the art microcontroller, many communication interfaces and several security features, including, but not limited to, smart card reading capabilities. Since a major goal in Critical Infrastructure Protection is to increase the level of security, it was decided that the Secocard could be used as an identity access device, operating as a smart card reader during the logon process. In particular, the user could logon to the operating system using a two-factor authentication (a smart card and a pin) instead of a password.

In addition to the elevated security, the device was programmed to provide real time logon information through a messaging mechanism. The idea is that the device, through its Wi-Fi communication interface can provide real time information to the XL-SIEM monitoring system, which is at the core of the CIPSEC Framework, regarding a user’s attempt to logon to a Linux or Windows operating system. More specifically Secocard can monitor all intermediate stages of the logon process ranging from the insertion of a smart card to the removal of the device from the host machine. A central monitoring system such as XL-SIEM can therefore decide if a user poses a threat to a Critical Infrastructure or not given the information provided by Secocard. The Secocard must only communicate with the XL-SIEM agent – no other interactions are necessary.

#### 2.2.5.1 Final Tool Optimisations

In order to reach the goals mentioned above the tool has gone through several transformations, both in hardware and in software throughout the project until it reached its final form and could successfully be used as a smart card reader device. In particular, originally the board was equipped with two microcontrollers both dedicated to specific tasks and communicating with each other through a high-speed interface. This hardware architecture caused problems with specific software components and therefore the board was redesigned and the two microcontrollers were replaced by a more generic and powerful one. The smart card reader functionality went over several software revisions to reach the maturity level needed by the project. A graphical user interface was designed to allow the user to enter the pin. Additionally, the Wi-Fi module on the board was reprogrammed to make it possible to support UDP connections to the XL-SIEM monitoring system. A syslog messaging mechanism was implemented on top of that so that the device could communicate with the XL-SIEM and report events. Finally, to enhance the functionality of the device, the software has gone through several low-level and higher level optimizations. The low-level optimizations aim to make the device more stable and include temperature monitoring and watchdog functionality – a components that is necessary in embedded systems. Regarding the higher-level optimizations, these are mostly associated with the responses of the device and in particular with the time between an event and the reporting of this event. Initially the device buffered the events and sent all the messages within fixed intervals. This process has been significantly optimized since the first implementation. At its final stage the device sends the messages within a few seconds after the event has taken place.

---

\(^1\) [https://www.bitdefender.com/business/enterprise-products/ultra-security.html](https://www.bitdefender.com/business/enterprise-products/ultra-security.html)

\(^2\) CIPSEC D5.5: Business Plans Definition
2.2.6 Honeypots

2.2.6.1 Honeypots

FORTH's hybrid honeypot solution is based on low and medium interaction honeypots. Honeypots are valuable tools for detecting and analysing malicious activity on the Internet. Honeypots can serve numerous purposes, they can be configured to emulate services of known services that exist to the Critical Infrastructures' systems, also to detect amplification DDoS attacks, attacks on databases or against Industrial Control Systems/SCADA specific protocols. Currently, FORTH's solution is using the Dionaea\(^1\), Kippo\(^2\), Conpot\(^3\) honeypots as well as a custom DDoS honeypot based on AmpPot\(^4\) and a Cloud IDS solution. More information on the honeypot solutions can be found in past WP2 and WP3 deliverables.

2.2.6.1.1 Honeypots' components

Dionaea is a low interaction malware-capturing honeypot initially developed under The Honeynet Project's 2009 Google Summer of Code (GSoc). Dionaea's main goal is to trap malware exploiting vulnerabilities exposed by services offered over a network, and ultimately obtain a copy of the malware\(^5\). During the validation of the solution COMSEC found out that Dionaea is detectable by the new version of nmap. We explored the issue and performed a patch to Dionaea in order to avoid detection. Moreover, we fixed a number of issues related with the performance of Dionaea and harmonising the logging capabilities of it with the rest of tools in CIPSEC.

Kippo is a medium-interaction SSH honeypot written in Python. Kippo is used to log brute force attacks and the entire shell interaction performed by an attacker. During the lifetime of the project Kippo was fine-tuned to provide alerts to both the honeypot backend and the XL-SIEM. Conpot is a ICS/SCADA specific honeypot, that can emulate a number of well-known industrial server-side control protocols capable to emulate complex industrial infrastructures' control systems. Conpot was altered in order to provide alerts similar to the ones that the rest of

---

1 https://github.com/DinoTools/dionaea
2 https://github.com/desaster/kippo
3 http://conpot.org
4 https://pdfs.semanticscholar.org/81a9/bb0db4573f243ae980447c6eed0275d49d35.pdf
5 https://www.div0.sg/single-post/dionaea-malware-honeypot
the honeypot produced, thus achieving a unified alerting template for FORTH’s solutions. Moreover, the source code was changed in order to report back to both honeypots’ backend and the XL-SIEM.

Our DDoS honeypot is based on the ideas proposed in the “AmpPot: Monitoring and Defending Against Amplification DDoS Attacks” presented in RAID 2015. It can detect amplification DDoS attacks and report the events to the central control system like XL-SIEM. After the first implementations, we fine-tuned it in order to produce alerts that are in line with the format of XL-SIEM. Also, events based on the aforementioned alerts were created and represented in the XL-SIEM during the various validation tests. Finally, a tuning to the parameters was made, after the testing sessions, in order to be more sensitive to DDoS attacks since we are deploying it to critical infrastructures.

![Diagram](image)

**Figure 32. Interfaces and communications of the backend of the honeypot server**

Our solution deploys all the aforementioned tools, to protect the network services of the CIs and generate events. To achieve the event logs integration in the ATOS XL-SIEM, each tool is able to communicate through syslog reporting the events to the XL-SIEM cyber agent over UDP at port 514 (Figure 32). The communication between the honeypot and the backend server occurs over port 5222 using the XMPP protocol (Figure 33) These events are then correlated, ranked and then visualized for the security manager. Moreover, all tools included in our honeypot solution are able to store their findings in separated databases, by communicating to our backend server where they are stored to databases. These databases can be used for statistical and visualization purposes or for further investigation by a security analyst. The honeypots’ backend server, is able to control the honeypots sensors, register new deployed ones, check the status of each sensor and deliver updates and patches to the already deployed sensors and interact with the updating and patching service.
Additionally, we have managed to integrate the Cloud-Based solution into the honeypots ecosystem in our local testbed and is connected to our dashboard. This tool will be able to protect any cloud or VM based tools running in the pilot from attacks within the same cloud or virtual environment. Virtual hosts hosted under the same Hypervisor are able to produce orders of magnitude more network throughput than conventional communication over the internet. This happens as VMs are using the internal CPU BUS to communicate, which can lead to throughput over 30 GB/s, and thus an infected VM could produce massive DoS or other attacks against other co-hosted VMs. The cloud-based solution is capable of detecting possible attacks that take place within a host running many VMs.

The detection system that we have devised, is based on a well-known IDS (SNORT) which is deployed within the host OS of the server hosting the VMs. The system includes a database, a log-processing engine and a web-based interface to visually present the results. The hypervisor of the system is configured to centrally monitor and log all the “malicious” activity related to the VMs of the specific machine and provide results through a web interface or in the form of raw data. Thus, the solution is able to identify intra-VM attacks and Inter-VM attacks, as well as, attacks originating from wherever in the internet and IDS monitoring the uplink of a cloud infrastructure. The solutions need the installation of a special hypervisor and an intrusion detection system on top of it. The solution can be deployed either to the Cloud or locally. Currently, the feasibility of our solution has been tested using the XEN hypervisor which we have used and tested for our current implementation. Additionally, events and alerts generated by our system are reported to our dashboard and to the XL-SIEM through syslog, using a common format as the rest of our solutions and is able to communicate over port 514 for reporting to the XL-SIEM and directly to the honeypots’ back end database server over port 3306 using MySQL queries.

2.2.6.1.2 Honeypots’ GUI Integration

In the Cloud deployment of CIPSEC the honeypot sensors are deployed to the networks we want to protect, in our case into the different pilots’ network. All the information from the sensors is then sent back to the Honeypots’ dashboard that is located in the FORTH’s premises. Using iframe web methodology we are able to embed the Graphical User Interface of the honeypots in the Unified dashboard of CIPSEC. Through the authentication process all the appropriate parameters are passed to the honeypots’ dashboard and thus is able to present the correct data based on the user and the scope.
In order for all individual tools to be integrated into the Unified dashboard, designed by UPC, we decided to use OAuth2 method to authorize users and implement single sign-on functionality. Using OAuth2 it is possible to authorize a user, in the unified dashboard and the same authorisation token to be validated and allow it access Honeypots’ dashboard.

The authentication process is similar for all tools and consists of logging in to the CIPSEC dashboard (1) sending from the client computer a token request to the Authorization Server (2) where the user is authenticated, this request contains the username, the password, a client id and a client secret. The server returns an access token and the scope of the user (2). This data is forwarded to each tool, in our case to the honeypots’ dashboard (3), which uses this validation token and sends a validation request to the Authorization Server (4) to validate the received authorization token. If the validation is successful, the client gets logged in Honeypots’ specific dashboard through CIPSEC unified dashboard, and is able to receive honeypot specific statistics and navigate through the different pages in the dashboard. The authentication process is depicted in Figure 35.

The same process can be replicated and used in an in-premises deployment of the Framework, as far as all the required components are deployed locally. This kind of setup is demonstrated in the Final prototype and described in the respective section of the deliverable.
Below are some screenshots from the unified dashboard of CIPSEC framework for different users demonstrating FORTH’s visualization tool.

Figure 36. Home Page of FORTH’s Honeypots Visualization Tool

The Home page has been enriched with data from all honeypots. Now the user can select and view from which sensor (dionaea, kippo, conpot, ddos) the data will be shown and see them either grouped or stacked. There are three graphs where the user sees the data for the last 24 hours, for the last week per day and for the last month per week. Furthermore, there is a notification message if new attacks occur and a link to a new Summary Page Attack where there are basic graphs from all honeypots for the last hour.
Figure 37. Top IP/Port Statistics Page of FORTH's Honeypots Visualization Tool

Top IP/Port Statistics shows a list with the top Attacker’s IPs, the number of packets, the country and a list with the destination ports of the attacks for the selected time range.
SSH Protocol Statistics is the page, which contains all the information from the Kippo honeypot. Top 10 passwords, Top 10 usernames, Top 10 Username/Password Combinations, Overall Success ratio, Number of connections per Unique IP, Top 5 SSH Clients and a list with Top 10 Inputs for the selected Time Range. Preselected for the pages is the last day.
Our IDS in the cloud stores in a local database all the attacks it detects from outside sources or between the VMs. Here we can see the top type of attacks it discovers scaled on their occurrence as well as their severity. Additionally, we post the top 10 of attackers, the countries where they are originating from and what ports they are targeting.

Figure 39. Intrusion detection statistics of FORTH’s Cloud IDS
2.2.6.2 Final Tool Optimisations

During the final phase of the project the deployment of the complete framework to all the pilots, the complete validation of the framework and the deployment into the two versions of the prototype, a number of optimisations have been incorporated to our solutions.

Namely, during the validation of the solution, COMSEC found out that Dionaea is detectable by the new version of Nmap. We explored the issue and performed a patch to Dionaea in order to avoid detection. Moreover, we fixed a number of issues related with the performance of Dionaea and harmonising the logging capabilities of it with the rest of tools in CIPSEC. Additionally, we have managed to integrate the Cloud-Based solution into the honeypots ecosystem in our local testbed and is connected to our dashboard. This tool will be able to protect any cloud or VM based tools running in the pilot from attacks within the same cloud or virtual environment. The cloud-based solution is capable of detecting possible attacks that take place within a host running many VMs. Concerning the DDoS honeypot solution, we fine-tuned it in order to produce alerts that are in line with the format of XL-SIEM. Also, weighted events based on the aforementioned alerts were created and represented in the XL-SIEM during the various validation tests. Through the testing sessions we tuned all the parameters in order to be more sensitive to DDoS attacks since we are deploying it to critical infrastructures. Finally, the source code of the ICS/SCADA honeypot was changed in order to report back to both honeypots’ backend and the XL-SIEM. Moreover, it was enhanced for providing alerts in the format that the rest of the CIPSEC solutions use, thus achieving a unified alerting template for FORTH’s solutions.

Finally, the visualisation component of the honeypots was completed revamped after the deployment to the pilots’ premises and the prototypes and after the results of the evaluation process. The optimisations include, a live update feature that alerts the user when new events are received by our solution, a new look and feel with more ways to represent the collected data, a completely new mechanism for the productions of all the network traffic graphs, that works correctly in both the offline and the online version of the prototype, achieving better performance with lower overhead. Honeypots’ dashboard was also integrated into the final CIPSEC dashboard, following a harmonious look and feel.

Detailed information about the final versions of the tools can be found in Section 2.2.6.1 and in past deliverables (D2.3, D2.5).

2.2.7 Integrity Management – Hardware Security Module

The Hardware Security module is a physical computing device that manages and safeguards digital keys, as well as offering secure cryptography processing capabilities. Such a module is typically implemented in the form of a plug-in card or an external device that attaches directly to a computer or network server. The University of Patras HSM is such an HSM that is implemented in FPGA SoC technology and constitutes a design that can be migrated to a series of FPGA SoC enabled boards as long as they can support basic communication and storage capabilities (e.g USB to serial communications, flash memory). The UoP HSM provides a secure and trusted environment for executing cryptographic primitive operations as well as several widely used security protocol primitives including message integrity, authenticated encryption certificate generation and validation. It can also support secure storage of keys and credentials for an associated host device. The communication with the host is protected through a password mechanism.

In the final CIPSEC architecture, the UoP HSM is fully integrated with the framework. It has a direct connection with the CIPSEC XL-SIEM agent that is deployed locally or remotely on a Critical Infrastructure System. The UoP HSM presence is not directly visualized in the CIPSEC Dashboard since it operates in a transparent to the user way but it can be indirectly seen in the various log and event entries of the CIPSEC anomaly detection mechanism and the updating and patching service.

2.2.7.1 Final Tool Optimisations

In the final stage of the CIPSEC Project, the FPGA based UoP HSM has been extensively tested, adapted for deployment in the various pilot test sites and has been provided with new capabilities in order to better match the pilot test scenarios as well as to be better fitted in future use cases. For this reason, the HSM has been extended to operate as a KiSA module (a security device that was required by the DB pilot in order to be deployable to the DB interlocking mechanism). We have included in the KiSA design, the HSM along with a Raspberry pi 3
embedded system that acts either as a Critical Infrastructure Host (CI Host) or as a proxy device that relays messages to the HSM from a CI host that is remotely connected to the proxy through WAN or LAN. The goal in all cases of the above setup is to provided end to end encryption and authentication using the HSM as the core security provider. The approach supports the creation of a secure channel between to CI devices in a Critical Infrastructure System. A characteristic test case scenario of the above approach can be seen in the following figure.

Figure 40. Example case of the HSM usage to provide end to end secure channels showcasing also the connection with the XL-SIEM Agent

Apart from the above the UoP HSM and Host Software component undertook several updates/expansions as follows:

- Acting as an end to end token using a proxy host device the functionality of the HSM was extended in order to support Bash scripts through the proxy service. A Linux based wrapper was also created in order to support this mechanism without the need of the proxy.
- The HSM Host software component was extended to support external triggering of logs. The events to the XL-SIEM agent is not only provided from within the Host Software Component but also by the user.
- New cryptography blocks were added in order to provide X509 certificate generation, certificate verification, authenticated encryption, the existing components where optimized with new more secure versions.
- The ECDH key agreement scheme was extended in order to support key agreement not only between Host and HSM but also between two different Hosts (thus complying fully with the original HSM specification).
- Several Bash scripts where designed and tested in order to validate the new HSM/Host functionality.
- Especially, apart from the authenticated encryption mechanism, traditional confidentiality and integrity solutions where designed (e.g. MAC and Encrypt approach)
- Certificate generation and storage mechanism within the HSM was finalized and functionally tested.
- The communication with ATOS XL-SIEM agent was fully realized and tested under various HSM failure scenarios.
- HSM enabled message integrity mechanism between two different CI host devices was implemented and tested in specific Message integrity scenarios under various IP and UDP package types. The mechanism was enhanced in order to support not only message integrity but also message confidentiality.
- The HSM hardware FPGA SoC design and implementation was made migratable to different FPGA platforms.
- The HSM hardware FPGA SoC and its associated embedded system software was packaged in order to be used for the CIPSEC updating and patching service. A similar approach was followed for the HSM Host Software component.
2.2.8 DoSSensing jamming detector

Several changes and improvements have been made to the DoSSensing jammer detector during the final stages of the framework integration and with respect to what was reported in D2.5. These address several aspects, such as message extension, OAuth implementation, dashboard integration, multitenancy and more. All these changes are described in the next section. Also, a simple jamming simulator that allows the user to select between no jamming attack data or the other four types of attack data has been developed (Continuous Wave Jammer, LFM Jammer, Pulsed Jammer and Wide Band Jammer).

In the Cloud deployment of CIPSEC depicted in Figure 42, the DoSSensing sensors are deployed in the different pilots’ network. All the information from the sensors is sent back to the DoSSensing’ dashboard on cloud. Using iframe web methodology, it is also embedded in CIPSEC’s Unified dashboard. To securely integrate the XL-SIEM dashboard with the UPC dashboard, OAuth2 authentication has been implemented with multitenancy functionality. This solution allows the integration of one single cloud user interface for the DoSSensing solution, which securely offers the information related to the sensors of the pilot network.
DoSSensing processes all the data sent from the sensors in its infrastructure on the cloud, which has a time-series database and triggering tasks which analyses the RAW data received from the sensors. The script cleans the RAW data so it can be shown by the dashboard, and makes a consolidated decision on the presence of a jamming attack. If an attack is detected, the script reports it to the Attack logger which generates the syslog message which will be shared with third parties like the XL-SIEM.

DoSSensing backend cloud components are interconnected with the XL-SIEM Agent, which receives the logs of the events generated by the Jammer detector backend. No jammer attack data is transferred between systems, only event data is shared through syslog to the XL-SIEM Agent.
The jamming detection solution is flexible enough to allow a deployment where the jamming sensors are installed on the premises to be protected, whilst the event generator backend can be used as a cloud service. Figure 45 depicts the scenario described. Alternatively, the backend event generator can also be installed on premises of the infrastructure.

2.2.8.1 Final Tool Optimisations

**Message extension:** we have developed new types of messages to extend the functionalities of the Jamming detection tool towards the XL-SIEM. Now we have new messages that are sent to the XL-SIEM syslog. Those are the “Jammer Detector Start” and “Stop” messages, which indicate that the sensor has started or stopped working and the “No Attack Detected” message, which is a periodic keep alive packet. In addition, the format of all JSON messages were adapted to the latest requirements of the XL-SIEM.

Examples:

Jammer Detector Sensor Started at frequency 2.412GHz

Jan 10 16:08:30 dev-sdrjd-demo SDRJD[15760]: INFO:sdrjdsyslog:{"jnr": 0.0, "event_duration": 0, "nodeld": "1", "srclp": "162.13.144.202", "time": "2018-01-10T16:08:30.000", "event": "Sensor Started", "freq": "2412000000", "type": "None"}

No attack detected at frequency 2.412GHz

Jan 10 16:08:31 dev-sdrjd-demo SDRJD[15761]: INFO:sdrjdsyslog:{"jnr": 0.0, "event_duration": 1000, "nodeld": "1", "srclp": "162.13.144.202", "time": "2018-01-10T16:08:31.000", "event": "No Attack", "freq": "2412000000", "type": "None"}

Jammer Detector Sensor Stopped

Jan 10 16:10:30 dev-sdrjd-demo SDRJD[17861]: CRITICAL:sdrjdsyslog:{"jnr": 0.0, "event_duration": 120008, "nodeld": "1", "srclp": "162.13.144.202", "time": "2018-01-10T16:10:30.000", "event": "Sensor Stopped", "freq": "2412000000", "type": "None"}

**OAuth implementation:** the authentication via OAuth was implemented on the Sensor View dashboard and the verification was integrated successfully with the UPC authentication server. Previously, the only way to access the Sensor View of the Jammer detector was through a login screen with a username and password. Now, as OAuth is supported, a token can be generated by the caller interface so that it is provided along with its
corresponding username and scope. These three are validated with the UPC authentication server and only if they are accepted by the server, the Sensor View dashboard can be visualized by the caller.

**Dashboard integration:** several visual adaptations were done on the Sensor View dashboard in order to be consistent with the UPC dashboard. The UPC dashboard requesting the Sensor View is firstly validating its authenticity by using OAuth and then embedding the content of the view on the corresponding iframe.

The following is a screen capture of the final look and feel of the embeddable Sensor View:

![Figure 45. Screen capture of the embeddable sensor view](image)

**Multi-tenancy:** it was a requirement for the pilot deployments that the data generated was only accessed by the corresponding tenant. This feature was adapted on the DoS Sensing software so that each pilot can only visualize their own data separately depending on their username and scope. To achieve this, different organizations were created on the system, one per pilot and a user was assigned to each organization. Having different organizations on the back end allows us to separate data and ensure that only the corresponding user has access to the corresponding data. In addition, each sensor was assigned to its organization on the backend system so that the data is securely inserted in separate databases.

**Precision improvements:** after the pilots were installed, a few adjustments were performed on the jamming attack detection refinement module so that less false positives were produced by the system. The number of detections provided by the sensor (inspection window) was increased in order to have more aggregated information to analyse and obtain a more refined final decision of an attack. In addition, the percentage of positive results in the inspection window was increased so that a higher number of false positive detections is now needed to determine a false attack. The only drawback is that now the detection of an attack takes more time (about 5 seconds) but the advantage is that the precision has increased significantly on these noisy deployments.

**Backend stability improvement:** a new module (watchdog) was developed on the software server in order to ensure better stability and recover the system in case it goes down. Previously, if the main refinement module had an issue, we were unable to automatically recover from that and data was lost. Now, there is a periodical task that ensures that the module is indeed working and it is recovered in case it experiences a failure.

**Jamming attack simulator:** a jamming attack simulator with a web interface that allows a user to select the type of attack that they want to simulate was integrated in order to evaluate the performance of the solution. The connection between the sensor software and the rest of the DoS Sensing system is the same, as well as the interaction with the XL-SIEM and the dashboard.
2.3 Services

2.3.1 Contingency plans

The contingency service was presented in detail in previous deliverables in WP2. Along deliverables D2.1, D2.2, D2.4 and D2.5 the idea was conceived and progressively elaborated. The third year of the project has been dedicated to implement these ideas and shape specific contingency plans in the three pilots, matching their specific needs. In the present document, we are reporting how the service has been implemented. A common approach has been proposed and it has been applied to the end-users DB, HCPB and CSI. In this section, such approach is thoroughly explained and then the details about the experience of the individual pilots during the implementation of the service are provided.

Building on the outcomes of the second year of the project, presented in the abovementioned deliverables, a document template was created, being structured taking into account the kind of questions to be answered so as to implement a sound contingency plan. In the table below the structure is presented, and in the following all the details about each section are presented.

<table>
<thead>
<tr>
<th>1. Introduction</th>
<th>1.1 Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.2 Assumptions</td>
</tr>
<tr>
<td>2. Overview</td>
<td>2.1 Pilot description</td>
</tr>
<tr>
<td></td>
<td>2.2 Main existing critical processes within the scope of the pilot</td>
</tr>
<tr>
<td></td>
<td>2.3 Description of the incident scenarios for the existing critical processes</td>
</tr>
<tr>
<td>3. Response plan in risk scenarios</td>
<td>3.1 First response contacts</td>
</tr>
<tr>
<td></td>
<td>3.2 Response team members and roles</td>
</tr>
<tr>
<td></td>
<td>3.3 Activation of the response procedure and incident notification</td>
</tr>
<tr>
<td></td>
<td>3.4 Description of downtime procedures</td>
</tr>
<tr>
<td></td>
<td>3.5 Media handling</td>
</tr>
<tr>
<td>4. Recovery strategy</td>
<td></td>
</tr>
<tr>
<td>5. Plan testing and maintenance</td>
<td>5.1 Testing the plan</td>
</tr>
<tr>
<td></td>
<td>5.2 Plan update</td>
</tr>
<tr>
<td></td>
<td>5.3 Plan distribution</td>
</tr>
<tr>
<td>6. Raising awareness in the organization</td>
<td></td>
</tr>
<tr>
<td>7. Conclusions</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Structure of the contingency plan template used in CIPSEC
First of all, an introduction is given explaining the purpose of the document. It is specified what the contingency plan serves to. The document must clearly specify the sector to which the contingency plan applies. The goals of the plan are presented. An individual person is appointed to be responsible for the plan. In this very beginning of the document it is explained what needs to be done immediately in case of an incident (for example inform CERT / CSIRT). The implications of the contingency plan, if not applied, are presented, what is linked to the explanation about the importance to apply it. Then, some assumptions are presented before getting into details.

Then, a general overview is presented, presenting as many technical details as available of the devices which are within the scope of the plan. It is necessary to map processes and assets. An estimation of the recovery time if an incident happens is included. The vital records involved are presented, this is, the information both in paper and electronic format which could be damaged if an incident happened. We are focusing on the processes but always from the perspective of the technological part (network / infrastructure). It is explained what happens if a critical process cannot be carried out due to an incident. We focus on incidents provoked by successful cyber-attacks. In such sense, we need to determine the threats to which the organization is exposed, focusing on availability, integrity, confidentiality and access control of the information. We will assess the risk as the product of the likelihood and the impact and then we will obtain a list of priorities. In this sense, the output file of the vulnerability assessment run by COMSEC is needed.

Then it is described the response plan when a cyber-attack is materialized and the processes described in the previous chapter are impacted. The people who must be informed about the incident must be identified, with names and roles. Then, it is necessary to define the criteria to activate the response procedure, how the response procedure will be activated and how the incident will be notified. We will differentiate procedures between working and non-working hours as well as the emergency escalation procedures. A workflow diagram is the most appropriate way to document this chapter. In addition, it must be addressed how the incident is communicated to the authority and how it will be assessed whether the incident could have cross-border implications. It is a good practice to check the National Security Plan, if existing, to know the role to be played by the Government authorities including the CERTs. Regarding downtime procedures, it is needed to describe specific responses for each of the processes and associated assets previously described, in response to different attacks. It is important to differentiate between the first 24 hours and beyond. The resources needed during this downtime phase will be described. As for media handling, the plan must specify what and which information should be protected and what information should be revealed to the media, what the incident was and what the mitigations taken are.

Regarding the recovery strategy, once the incident is under control, the contingency plan will document the process to get back to the normal situation for each of the involved processes, with the steps followed. The plan must say what is priority in the recovery process and what must be done earlier.

The plan must address its own testing and maintenance. It must describe how the testing of the plan will be executed, how often the plan will be updated, considering when the latest update took place and the expected date for a new release. Concerning the distribution of the plan, it must be specified who will hold a copy of the plan.

The plan must describe the steps to follow to present the contingency plan to the technical staff.

Finally, a conclusions chapter with the final remarks will close the document.
Figure 46. Interfaces and communications of the backend of the contingency plans service

Figure 47. Interfaces and communications of the GUI of the contingency plans service
In the CIPSEC dashboard, when clicking on *Contingency Plan* under the *Services* menu the user will be able to check on screen the latest version of the contingency plan. The Figure 48 shows the case when we log in as an admin of DB.

![Figure 48. The contingency plan in the CIPSEC dashboard](image)

Also, an admin user has the chance to download previous versions of the contingency plan or upload a new one in PDF format. This is reflected in the figure below.

![Figure 49. Downloading previous contingency plans and uploading a new one](image)
In the following, the specific work done per pilot with respect to the elaboration of their contingency plan is presented.

2.3.1.1 Contingency plan for the transportation pilot

The purpose of early version (prior working on CIPSEC project) of contingency plan of DB Netz AG was to show, how in case of an emergency, the critical infrastructure can be ensured. Emergency was defined in context of severe wetter, storm, flood, fire, earthquake, terrorist attack and all security incidents were only addressed by safety measures.

That was because, the railway infrastructure consists of many components, which act together to ensure a safe rail operation. Most relevant are the components responsible for signaling, like, for example, interlocking systems (together with object controller and FeAs).

Traditionally in railway there were no data logging and aggregating mechanism – only data that are logged were failures of components (data were stored on HDD and/or printed).

Contingency plan that DB Netz was based and directed only on safety of system and for example, there were no security recovery processes.

In case of an incident, the maintenance personnel replace the affected component, and this component is later checked for the cause of the incident. No forensic analysis was performed.

The same situation applied to stored data - only operational data was kept (timetable data, log of performed interlocking commands, configuration data, etc.).

Process of digitalization in railway domain, and projects like CIPSEC bring DB Netz more benefits and innovation and what most important thing is – more awareness and education, for not only employees but also management.

There is a lot of possible incident scenario that are already described in DB Netz pilot project, and they are now “embedded” in our contingency plan - from Intrusion, MitM attacks, Malware spreading, to Firmware or signal equipment destroying.

Thus, DB Netz had to change contingency plan and update it with a lot of new scenarios obtained from this CIPSEC project:
- Infiltration of networks, transport layer attacks, modify communication,
- Insertion of commands,
- Vulnerabilities in components,
- Bugs in protocols, exploit usable
- Exploit on std. application,
- Malware, Viruses, illegal manipulation of data

Further, list of potential external threats - like terrorist organizations, criminal organizations, competing companies, etc. is expanded with hackers, cyber criminals, hacktivist etc.

The Contingency plan is a dynamic document and DB Netz AG continually working on changing and updating based on technology changed and/or legal regulation change. CIPSEC project was a trigger for updating and changing and evaluating our contingency plan, so not only be aimed on safety but also on security, according framework and pilot test performed two times at DB Netz premises.

Technical support in “new situation” will be gained from SOC, DB Systel CSIRT, intern CSIRT. In critical situation there is a CSIMT Team (Cyber Security Incident Management Team) with full support for dealing with incidents. In critical situation BSI (State Agency for Information Security) need to be contacted. Everything is described in contingency plan of DB Netz.
In contingency plan is elaborated plan and incident handling workflow with a lot of runbooks and use-cases (for most of new situation that we get with new NeuPro system).

Contingency plan of DB Netz AG is labelled TLP:RED and, because of that, is for limited/internal use only. The recipients of contingency plan are CIO, CISO, Persons responsible for BCM and emergency, cyber security/information security department and suppliers and third party provide assistance and support in such of situation.

### 2.3.1.2 Contingency plan for the Health pilot

The participation in the CIPSEC Project has entailed an excellent opportunity for the HCPB to make a leap forward in terms of readiness to respond adequately to high risk scenarios with damaging attacks carried out by cyber criminals. To ensure good outcomes in the process of upgrading their contingency plan, the HCPB has created a multidisciplinary team of experts representing the different stakeholder groups involved. This team has worked on the definition of proper action plans to ensure the correct operation of the centers when potential incidents could occur. The key is to assume and plan for the worst and set up controls in advance to counter cyber breaches, always considering the general regulations of reference. This work is still in progress, a collaboration framework with the competent authorities has been established.

Along with the delivery of D2.7 an annex with the details of the work done at the HCPB is provided. Some parts of the document cannot be disclosed until the authorities give the green light to do so, only those parts that only include generic information of non-confidential data have been included in this document.

Due to the involvement of several stakeholder groups, including the central and regional governments, this contingency plan will not be closed until November 2021, provided that all the iterations and liaisons flow smoothly and no setbacks are found in the process, so this would be the soonest date for the final consolidation of the document.

### 2.3.1.3 Contingency plan for the Environmental Monitoring pilot

Participating in the CIPSEC Project has provided added value to CSI with respect to making a leap forward in the definition of the corporate contingency plan. It has helped further specify the activities and procedures provided for the entire Contingency Plan of the entire infrastructure of CSI Piedmont regarding the ISO 27001 certification. CIPSEC has provided insights mainly on the daily work and knowledge of procedural and operational aspects not previously considered in older versions of the plan.

The CSI contingency plan has historically been created to manage natural events or accidents such as explosions or gas leaks, adapting it over time also to the management of IT risks. This evolution of the plan has allowed us to expand the type of risks and their consequent management by facilitating the drafting and understanding of the contingency plan for CIPSEC.

### 2.3.2 Vulnerability assessment

The vulnerability assessment service was presented in detail in previous deliverables in WP2. This section will summarize the service main characteristics, and present the progress made with regards to dashboard integration.

#### 2.3.2.1 Service Description

In order to address information security challenges in the CIPSEC project, Comsec has crafted an approach that will provide Critical Infrastructures, as well as the project pilots, with a holistic view to identify security vulnerabilities and the ways to address them\(^1\). This will assist Critical Infrastructures in improving and maintaining

---

\(^1\) CIPSEC D2.4 : Services integration on the unified architecture
a consistent level of security across their network. This will be achieved through a security vulnerability assessment.

The Cyber & Information Security Assessment consists of 5 sections outlined below:

1. Mapping the relevant threats, platforms and assets. In this phase, the CI owner will provide Comsec with high level architecture, documentation and design of the tested CI environment.

2. Security Infrastructure. In this phase, a technical security infrastructure assessment is made, that includes reviewing of the network security, firewalls and IPS\IDS, monitoring products, EDR, anti-virus and other infrastructure security controls.

3. Application Security. In this phase, the tester focuses on the applicative interfaces, APIs, CI and proprietary protocols in order to try to execute different applicative related attacks.

4. Deliverables & Reporting. The output of the vulnerability assessment service is a detailed report that includes executive summary, detailed findings with print screens PoCs and demos, road map and recommendations that details how the found vulnerabilities should be fixed. The report will use a predefined Microsoft Excel format, and will be uploaded to the unified CIPSEC dashboard.

2.3.2.2 Approach and Methodology

The general approach of the project will involve several key activities, to be carried out by a team of Comsec’s Security Specialists:

1. Gathering information about the design, architecture and interfaces. This includes a review of technical documentation (design documents, functional descriptions, reports of previous tests performed, development documentation and others), interviews with developers and programmers, interactions with application owner, etc.

2. Initial analysis of the system architecture, data flow, infrastructure and concept.

3. Determining sensitive attack paths (e.g. interfaces, modules) and attack patterns that would be covered during the security review, in both the design and implementation levels.

4. Conducting Security Review and deployment testing of the platform, based on a Top-Down analysis of vulnerable system components. This will include performing manual checks, automated tests and various reviews to uncover security vulnerabilities.

5. Analysis of the gathered data and the results of the various reviews. The analysis includes categorizing the detected vulnerabilities and prioritizing them according to the business and technical context. Comsec methodology includes a systematic Security Risk Analysis & Evaluation process.

6. Report documentation

2.3.2.3 Dashboard Integration

As described above, the assessment report can be imported to the CIPSEC unified dashboard. To allow it, COMSEC and UPC agreed upon an Excel template which contains all the needed content. This report is readable by its own, but to allow the assessed organization to conveniently share the report findings between its employees, it can be uploaded into the unified dashboard.

Uploading a new report is possible via the “Upload New Vulnerability Assessments” section, at the bottom of the page. Each upload completely overwrites the previous content. Previously uploaded reports are also kept and available for downloading by an administrative user, to allow to backups and versioning history.

User guide and screenshots for the vulnerability assessment dashboard page can be found in appendix 7.1.1.
2.3.3 Updating and Patching

Bitdefender’s Updating and Patching service is a custom-made solution for delivering software patches to the tools from the CIPSEC framework: XL-SIEM from Atos, Honeypots from Forth, GravityZone from Bitdefender and Aegis toolkit. Patches are essential to the well-functioning of the systems providing improvements as well as fixes for security vulnerabilities and other bugs. The Updating and Patching service allows for the monitoring and rapid deployment of new updates and patches.

![Updating and Patching](image)

**Figure 50. The general dashboard**

The Updating and Patching service consists of two components: the central management dashboard that is hosted on the Bitdefender network and can be accessed from the Unified dashboard as well as an agent written in python that is installed on the client side and communicates with the server.
Developing the service as a custom-made solution allowed us to easily integrate our management component with the Unified dashboard, designed by UPC, by implementing the Oauth2 access delegation protocol. In the dashboard, an administrator can monitor the status of the tools and deploy new updates when available. When an administrator decides to mark an update as deployed it will be immediately made available to agents that periodically check for new approved updates, using a RESTful API. When an agent receives the information that a new update is available it will proceed to automatically download and install the patch. Upon completion of the update process, it will report back the status of the update process. In case of a failure in the automatic update process the administrator will be notified in the management component and will have to manually update the module.

The Update and Patching components were written with ease of customization in mind. For this purpose, we have used python and mongoDB, which allow for rapid prototyping and improvements to the service while maintaining extensibility. This allowed us to easily integrate with other tools in the framework and provide an easy way of patching a wide range of targets.
The Update and Patching components were written with ease of customization in mind. For this purpose, we have used python and mongoDB, which allow for rapid prototyping and improvements to the service while maintaining extensibility. This allowed us to easily integrate with other tools in the framework and provide an easy way of patching a wide range of targets.
The agent can be configured using an .ini file which will include the URL of the update server and the tool that it needs to monitor for new updates. It will check the current version of the tool using a command line provided by each tool's developer and query the update server for a new version. When an update is available, it downloads the file, checks the file integrity using hash sum and proceeds to install the update. In the case of the Honeypots, the update is done directly from FORTH's SVN servers and the latest version installed is reported to the central management dashboard.
2.3.4 Forensics service

The main goal of the Forensics Service is to improve the speed and accuracy of troubleshooting, monitoring, and recovery, as well as the protection of sensitive data, in the event of a security incident being committed. This is achieved via the Forensics Visualisation Tool which provides the visualisations that help operators achieve the mentioned goals and the Forensic agents and sensors that provide data to the visualisation. In the context of CIPSEC, both components of the service were updated to serve the integration needs of the complete CIPSEC platform. Moreover, optimisation tasks were held to suit the real-life deployments on the three pilots. This process strengthened the offered solution and enabled it with a set of features that will foster its way to marketability.

2.3.4.1 Forensics visualization tool

The Forensics Visualisation Tool has undergone several updates to support the needs of the pilot deployments as well as to optimise the performance and usability of it. The major update involves the implementation of the front-end part of the tool using an adaptable and quick loading visualisation framework based on technologies such as AngularJS, nvd3.js and d3.js. This implementation enabled the tool to handle larger sets of data with higher speed and provide the users with a better-looking interface that provides more accurate control of the visualisations and greater responsiveness to user actions. The following screenshots present some of the main screens of the Tool in its latest version and depict the latest functionalities added, namely Timeline Comparison, Extended Netflow Analysis and Disk Analysis.

![Figure 55. FVT - Timeline comparison](image)

The Timeline Comparison functionality depicted in Figure 55 allows operators of the FVT to compare events that happened in two different time periods. The screen is split in two vertical independent areas which correspond to the selected time periods. A timeline of the events is displayed on top of each ‘column’ and can be adjusted dynamically by the user without affecting the other one. Moreover, all the CIPI graphs displayed underneath get updated according to the selected time period of their respective controlling timeline component. This way, operators can search for identical situations in the past and look for same patterns in the values of the various CIPIs. Undiscovered conditions can be revealed by making these comparisons and provide indications that a repeated situation might hide an abnormal behaviour of the monitored host.
Figure 56 presents some example graphs for a few CIPIs. The graphs have been optimised to handle large datasets by employing techniques like data resampling so as to remain responsive without losing precision of the presented data.
Figure 57. FVT - Netflow Analysis

The Netflow analysis depicted in Figure 57 enables operators to investigate the traffic among the machines of their internal network as well as with external hosts. For the final integrated framework, this functionality was enhanced by adding filters to reduce the flows displayed by their number, number of exchanged packets, IP and date. This way operators have an easier way to filter out unwanted data and focus on these days or host machines that present a chance of unexplained communication activity which might be potentially harmful.
One of the latest additions to the FVT is presented in Figure 58 and shows the disk analysis CIPIs that monitor the number of file activities, the partitions and the partition sizes of the monitored disk. Information of these CIPIs can be used in combination with the previous ones to help the operators during a forensic investigation. Moreover, the FVT has an initial provision of a way to search disk images for specific string literals as presented in the following figure.

![Figure 58. FVT - Disk Inspection](image)

Apart from the previously mentioned updates, the FVT was also updated to cover the integration needs. These updates include:
Dashboard Integration and MultiTenancy: The FVT was included in the integrated dashboard interface via iframe technology so as to allow access to it via the uniform CIPSEC environment. Extra configuration parameters were added so as to allow users from different pilots to access only data that refer to their organisation and therefore preserve privacy.

OAuth 2.0 support: In order to support embedding of the FVT in the dashboard interface in parallel with multitenancy, the update of the authentication mechanism of FVT was imperative. Following the OAuth specification implemented by the Dashboard, the FVT implemented the required client and controlled access to the data according to the logged in user's role and defined scope.

![Diagram of FVT integration](image)

Figure 60. The Forensics Visualization Tool GUI is embedded in the dashboard

Cache mechanism: The retrieval of data from XL-SIEM's database of events has remained the main way to get data into the visualisation of the FVT. Furthermore, to optimise loading speed of the FVT in the integrated platform a caching mechanism was put in place. Data from older dates that have been already retrieved are not subject to any change thus allowing their storage to a cache and eliminate the need for communication with the database when the user asks for them again.
2.3.4.2 Forensics agent and sensors

The Forensic agent includes a set of plugins/tools that can be deployed to a host machine and be properly configured to log information that is relevant to the host. The integrated version of the forensic agent includes the Nagios and Netflow sensors that serve as the source of raw data for the FVT as presented in the figure below.

Figure 62. Interface and communications of the AEGIS forensics agent
Communication with XL-SIEM Agent is performed via the exchange of syslog messages containing the information of captured events. During the finalisation of the Forensic agent, the Netflow sensor was updated to support an effective method of retrieving netflow data and to also support automatic updates. The sensor is a java program that queries CSV files with netflows (generated by nfdump) and produces log entries that are forwarded to XL-SIEM agent, as analytically described in D2.5.

**Netflow Sensor data collection:** The raw data come from the nfdump utility which stores captured network traffic in nfdump files. The Netflow sensor includes a process that queries these dump files and generates CSV files that include only the needed information for a dataflow. Afterwards, another process queries these CSV files and produces the relevant log entries to be sent to the XL-SIEM agent. To optimise these steps, configuration parameters were introduced in the processes to allow the adjustment of the threshold values when generating events. This is particularly useful when deploying the sensor in different environments where network traffic can differ significantly between them.

![Diagram of sensor communications](image)

**Figure 63. Interface and communications of the forensics sensors**

**Updating & patching:** The unified CIPSEC Dashboard includes an updating and patching service which allows operators to easily check for new versions of the tools and apply any pending patch or update from within the dashboard. The mechanism implemented in the Forensic Service supports updating of the Netflow sensor. The sensor was updated to provide version information that is automatically updated on every new build of the code. Using the interface provided by BD to upload new versions of the program, the update mechanism can check if the deployed versions matches the latest one and if not perform the necessary actions to update it.

### 2.3.5 Training Service

The training service is very important within the CIPSEC framework for several reasons. First, it will provide training to the clients and users of the framework. Even if a critical infrastructure relies on technology and services, the main cause of cyberattacks and data breaches is human error (almost 90%) ([chiefexecutive.net/almost-90-cyber-attacks-caused-human-error-behavior](https://chiefexecutive.net/almost-90-cyber-attacks-caused-human-error-behavior)). It is thus important to provide users with the right set of tools to avoid such risks. This is the fundamental reason behind the three-tier approach selected for this training platform, general cyber security, critical infrastructures and CIPSEC overall framework. We aim at providing the best set of information regarding these three aspects of cyber security. Second, this will increase the visibility of the project if the content can be of interest to other projects, universities, security experts and more. The visibility could be further increased by making it available to upload content for external users. Finally, the training service can provide an unfair advantage towards the commercialisation of the...
framework since it provides an added-value to potential clients and we are not aware of any similar platform in potential competitor frameworks.

CIPSEC’s training platform aims at providing a three-tier approach to the groups of courses it will provide. Although the courses will all be specialised, as they address different audiences of the cybersecurity-concerned market employees, it should cater to all profiles, backgrounds and experience levels. The service begins with a generalist introduction to the cybersecurity field, digs a bit deeper into Critical Infrastructures and finally offers specialized training for the CIPSEC security framework. Our 3-tier approach will include multiple courses across various fields of cybersecurity, network protection, antivirus, antimalware etc. and will have a relatively big audience and impact, thanks to its diversity and because it does not only aim at specialists.

![Figure 64. 3-tier approach of the Training Service](image)

**Figure 64. 3-tier approach of the Training Service**

### Cybersecurity

The most general courses constitute the first tier, that with the broadest scope. Its main target is technicians and engineers, whether working in CIs or from the general public, with an interest in cybersecurity. Vulnerability is considered a transversal topic covered in the courses listed below.

**Table 3. General courses for cybersecurity topics**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network Monitoring</strong></td>
<td>Efficient methods and techniques to identify cyber-attacks and combat them at network level.</td>
</tr>
<tr>
<td><strong>Honeypots</strong></td>
<td>Various matters related to honeypots (e.g. deployment use and operation, use for anomaly detection, honeypots for CIs etc.).</td>
</tr>
<tr>
<td><strong>DoS identification</strong></td>
<td>DoS identification methods and mitigation techniques and defences in various levels.</td>
</tr>
<tr>
<td><strong>Antivirus/Antimalware</strong></td>
<td>Antivirus-antimalware and general good practices for cyber-protection.</td>
</tr>
</tbody>
</table>

**Target audience**: technicians and engineers of CIs, employees of CI with previous technical background or interest in cybersecurity, home users.
Critical Infrastructures

The second tier is more specialized, and the courses deal more with Critical Infrastructure protection. They target CI Employers, Technicians, Engineers, Management, etc.

Table 4. Specialized courses on Critical Infrastructures

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System integrity for ICS</td>
<td>Analysing main threats to industrial control systems (e.g. the exploitation of applications, protocols and communication interfaces).</td>
</tr>
<tr>
<td>Network Protection</td>
<td>Protecting the networks against attacks (like Denial-of-Service, network infiltration and communication modification) and ensuring the timelyness of transmitted messages are essential assets for sustaining operation of the CI.</td>
</tr>
<tr>
<td>Hardware security</td>
<td>Vulnerabilities introduced in critical infrastructure system devices by inappropriate hardware or software implementations of security algorithms.</td>
</tr>
<tr>
<td>Privacy</td>
<td>Various privacy issues related to Critical Infrastructures.</td>
</tr>
<tr>
<td>Social engineering</td>
<td>Raise awareness about all social engineering techniques used nowadays, and how human factor is exploited in various combinations to create attacks.</td>
</tr>
<tr>
<td>Best Practices</td>
<td>Promote good security policies and industrial practices to foster common understanding.</td>
</tr>
<tr>
<td>Forensics</td>
<td>General aspects on Forensics in Critical Infrastructure Systems.</td>
</tr>
</tbody>
</table>

Target audience: Technicians, engineers, management and any other employees of CIs.

CIPSEC Platform

Finally, the third tier will include courses based on the CIPSEC platform and will target administrators and users installing and operating the platform. These courses are directed at a technical audience and the contents include the different menus available to configure platform and settings for the data related to security. Instructions about platform installation are also given, together with specific guides of the various specific tools the platform is composed of.

Table 5. Courses focused on the CIPSEC platform

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIPSEC framework</td>
<td>Introduction to CIPSEC framework. Abstract architecture, basic components, features, capabilities etc.</td>
</tr>
</tbody>
</table>
Target audience: CIPSEC platform admins, end users.

Short descriptions of the CIPSEC training courses implemented so far are reported below. A small number of courses were finalised and made available online at the end of the project. Also, it is expected that other projects and users will make use of the platform to upload material even after the end of the CIPSEC project, through collaboration with members of the consortium.

### Honeypots

**CIPSEC partner:** FORTH  
**Target Audience:** IT practitioners  
**Security Areas Covered:** Network traffic anomaly detection  
**Duration:** 45'  
**Required tools (any special hardware software etc.):** Bring your own laptop  

<table>
<thead>
<tr>
<th>Presented:</th>
<th>Yes - 1st CIPSEC Training event in Crete – Greece</th>
<th>Presenter(s):</th>
<th>Online: Yes</th>
<th>Course:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Antonis Krithinakis, Christos Papachristos</td>
<td></td>
<td>Video: <a href="https://www.youtube.com/watch?v=FxBs9Xx-tTY">https://www.youtube.com/watch?v=FxBs9Xx-tTY</a></td>
</tr>
</tbody>
</table>

**Description:** This course aims to help attenders understand the concept of honeypots and their role in the protection of critical infrastructures in the context of the CIPSEC project. In the first part, the course starts with a general presentation on honeypots to inform the audience how this security mechanism operates in order to
detect potential network threats. The first part concludes by summarizing the benefits of using honeypots in the context of the CIPSEC architecture to enhance the security of critical infrastructures.

In the second part, the course demonstrates a honeypot solution which is used in the CIPSEC project. This solution automates the process of honeypot deployment and configuration and provides visualization of results and alerting mechanisms for administrators. This solution is based on the Dionaea low interaction honeypot, and the Kippo medium-interaction honeypot. The first is a general-purpose honeypot which offers multiple emulators for network services, while the second is specialized in the SSH protocol. Through the course, the audience will have the opportunity to see how this infrastructure is operated through a web-based control panel and how a specific attack can be performed and captured in real time.

### DDoS attacks on critical infrastructures

<table>
<thead>
<tr>
<th>CIPSEC partner:</th>
<th>COMSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Audience:</td>
<td>IT practitioners</td>
</tr>
<tr>
<td>Security Areas Covered:</td>
<td>Network and communication security</td>
</tr>
<tr>
<td>Duration:</td>
<td>45'</td>
</tr>
<tr>
<td>Required tools (any special hardware/software etc.):</td>
<td>None for the audience</td>
</tr>
</tbody>
</table>

| Presented: | Yes - 1st CIPSEC Training event in Crete – Greece |
| Presenter(s): | Omri Sagron |
| Online: | Yes |
| Course: | |
| Video: | https://www.youtube.com/watch?v=sJ_ZttX01Jk |

Description: Large-scale distributed denial of service (DDoS) attacks on government sites have been relatively few, but there is evidence to suggest that trend may be changing for the worse. This spate of attacks has raised questions and concerns; if the Internet is vulnerable to DDoS, how reliable is our critical infrastructure, such as utility systems?

DDoS attacks are increasingly fuelled by the millions of devices connected to the Internet of Things (IoT) and it is a threat because the grid is vast and constantly evolving, with some new technologies that are connected to the IoT.

This course aims to help attendees understand the concept of the DoS & DDoS (Distributed Denial of Service) attacks and their implementation in the critical infrastructures area in the context of the CIPSEC project. In the first part, the course starts with a general explanation on DDoS attacks in order to inform the audience how this attack take place as well as trends regarding to DDoS.

The second part dedicated to different DDoS scenarios, divided into 3 main groups:

- Infrastructure Level DDoS
- Application Level DDoS
- Special DDoS Attacks

Finally, the course presents the DDoS service attacks provided by Comsec and how to implement this test technique as part of the CIPSEC framework and services for critical infrastructures.

### A gentle introduction to advanced anonymization of databases
CIPSEC partner: UPC
Target Audience: General audience
Security Areas Covered: Statistical disclosure control
Duration: 45'
Required tools (any special hardware software etc.): None for the audience

Presented: Yes - 1st CIPSEC Training event in Crete – Greece
Presenter(s): Ahmad Mezher
Online: Yes
Course:
Video: https://www.youtube.com/watch?v=GvACLd5ysHk

Description: This course deals with new trends in data privacy, providing a quick glance at some of the privacy challenges and solutions in modern information technologies, with emphasis on the benefits in CIPSEC’s critical infrastructures. The first part of the presentation begins with a general overview of big-data technologies and challenges and concludes with k-anonymous micro aggregation as a privacy mechanism. The second part - discusses the cost effectiveness in advanced privacy technologies. Finally, UPC’s Data Privacy Tool is presented.

Digital Forensics

CIPSEC partner: AEGIS
Target Audience: IT Security, Law Enforcement
Security Areas Covered: Digital Forensics
Duration: 45'
Required tools (any special hardware software etc.): None for the audience

Presented: Yes - 1st CIPSEC Training event in Crete – Greece
Presenter(s): Vassilis Prevelakis
Online: Yes
Course:
Video: https://www.youtube.com/watch?v=LsLI3mFlUXo

Description: When a crime is committed, there are important steps to be taken in order to collect evidence, preserve evidence so that they can be used in a court of law later on, analyse evidence, and draw conclusions which may range from an analysis of the MO of the perpetrators to recommendations to prevent the recurrence of the specific crime. This course will take the audience through all the steps (Collection, Examination, Analysis, Reporting) of an effective forensic investigation using a sample case study as an example. In addition, examples from old cases give to the audience an opportunity to consider the ethical, legal, procedural and financial considerations that may influence the course a forensic investigation.

Securing Critical Infrastructures through hardware means: String points and weaknesses of Hardware security tokens

CIPSEC partner: UOP
Target Audience: Generic and experts

Security Areas Covered: Hardware Security

Duration: 45'

Required tools (any special hardware software etc.): None for the audience

Presented: Yes - 1st CIPSEC Training event in Crete – Greece

Presenters: Apostolos P. Fournaris

Online: Yes

Course:

Video: https://www.youtube.com/watch?v=OK_OCYPjx0c

Description: In this course, an overview of the most important hardware-based technologies that are currently used in order to protect security sensitive devices is made and the most important security gaps that such technologies may have are discussed. The key points of such technologies are primarily focused on security tokens that can be physically associated to a host device to provide security services like smart cards or Trusted Platform Modules (TPMs). The overall architecture of such modules is presented and the design challenges to realize them are highlighted. Also, the cryptography/security services that they can provide is discussed. Furthermore, in this course, possible vulnerabilities that may exist in a security module are presented. More specifically, the course is focused on side channel analysis attacks and fault injection analysis attacks, showing how such attacks can be mounted and how they can compromise the most important assets of a Security token i.e. its cryptography keys. Side channel attacks exploit the information leaking from the physical characteristics of a Hardware or software security/cryptography implementation (power consumption, timing, electromagnetic emission, etc.) and can be easily mounted on a badly implemented Security token. The most widely used applied countermeasures for such attacks are also described in this course. Finally, the hardware security token (the University of Patras Hardware Security Module) design approach followed in the CIPSEC project and the overall security services it provides are presented.

Denial-of-Service Jammer Detector Training course

CIPSEC partner: WOS

Target Audience: Generic and experts

Security Areas Covered: Denial of Service on the Physical layer

Duration: 30'

Required tools (any special hardware software etc.): None for the audience

Presented: Yes – 2nd CIPSEC Training event in Frankfurt – Germany

Presenters: Olmo Rayón

Online: Yes

Course:

Video: Yes (https://youtu.be/RjPC8kXP1Go)

Description: Wireless networking plays an important role in achieving ubiquitous computing where network devices embedded in environments provide continuous connectivity and services, thus improving human’s quality of life. However, due to the exposed nature of wireless links, current wireless networks can be easily attacked by jamming technology. Jamming can cause Denial-of-Service (DoS) problem that may result in several other higher-layer security problems, although these are often not adequately addressed. Jamming makes use of intentional radio interferences to harm wireless communications by keeping communicating...
medium busy, causing a transmitter to back-off whenever it senses busy wireless medium, or corrupted signal received at receivers. Jamming mostly targets attacks at the physical layer but sometimes cross-layer attacks are possible too. DoSSensing provides a device that can detect different types of radio jammer signals at the physical layer and inform the users that a jammer is performing an attack. It will visually provide the jammer type and estimated power over time for a frequency or wireless channel.

The users can then identify the threat and approach the area to find the jammer device. Having several devices on that area will better help to identify and find the threat that will be pointed out by more than one detector.

---

**Bitdefender Anti-Malware Training course**

**CIPSEC partner:** Bitdefender (BD)

**Target Audience:** Generic and experts

**Security Areas Covered:** Various forms of Malware

**Duration:** 30'

**Required tools (any special hardware/software etc.):** None for the audience

<table>
<thead>
<tr>
<th>Presented: No</th>
<th>Presenter(s): N/A</th>
<th>Online: Yes</th>
<th>Course: <a href="http://www.securityaware.me">www.securityaware.me</a></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Video: No</td>
<td></td>
</tr>
</tbody>
</table>

**Description:** Malicious code can potentially be used to manipulate the controls of power grids, financial services, energy providers, defence, healthcare databases and other critical infrastructure, resulting in real-world catastrophic physical damage, such as blackouts or disruptions to an entire city's water supply.

GravityZone is a business security solution built from ground-up for virtualization and cloud to deliver security services to physical endpoints, mobile devices, virtual machines in private, public cloud and Exchange mail servers.

GravityZone is one product with a unified management console available in the cloud, hosted by Bitdefender, or as one virtual appliance to be installed on company's premises, and it provides a single point for deploying, enforcing and managing security policies for any number of endpoints and of any type, in any location.

GravityZone delivers multiple layers of security for endpoints and for Microsoft Exchange mail servers: antimalware with behavioural monitoring, zero-day threat protection, application control and sandboxing, firewall, device control, content control, anti-phishing and antispam.

Users are getting familiar with the Anti-Malware tool, architecture, its main functionalities, how-to install and utilise it.

---

**Training events:**

CIPSEC project validates the proposed training service in different events, during which they are presented to various users and audiences with diverse background and expertise in cybersecurity. Our goal is to create a multi-dimensional service that will be useful even to people with little technical knowledge in cybersecurity.

**1st training event – Crete, Greece**
The 1st CIPSEC training session took place during the 22nd IEEE Symposium on Computers and Communications, in Heraklion, Crete, Greece on the 3rd of July 2017. It was organized by FORTH and lasted for a full day 10am-5pm. More than 60 specialists, academics and students from all over the world participated in the workshop where several tools, technologies, hardware and software mean on how to enhance Critical Infrastructures’ security and more specifically CI in the Health domain was discussed.

2nd training event – Frankfurt, Germany

The second event took place in Frankfurt on the 16th of October 2018. The event targeted the second tier of the audience and was aimed at training employees of Deutsche Bahn. More than 6 employees attended the training session were these courses were given and were trained on the CIPSEC framework.

3rd training event – Barcelona, Spain

The third event took place in Barcelona on the 23rd of January 2019. The event targeted the second tier of the audience and was aimed at training employees of Hospital Clinic (HCPB). More than 8 employees attended the training session were these courses were given and were trained on the CIPSEC framework.

4th training event – Turin, Italy

The fourth event took place in Torino on the 14th of March 2019. The event targeted the second tier of the audience and was aimed at training employees of CSI Piemonte. More than 8 employees attended the training session were these courses were given and were trained on the CIPSEC framework.

All courses for the training service are uploaded in the external training service platform, which shares information directly with the Framework, so the courses can be displayed on the Training Service screen of the unified dashboard located inside UPC’s network, as displayed in Figure 65.

![Figure 65. The training service GUI is embedded in the dashboard](image)
3 CIPSEC Final Prototype

3.1 Overview

The CIPSEC prototype is a dedicated environment of the CIPSEC framework, which also includes critical infrastructure simulators and attacking tools. It is mainly installed on a laptop.

The main objectives of the CIPSEC prototype are as follows:

- Show the feasibility of integrating the products brought to CIPSEC by the partners into a framework.
- Show the added value brought by the products working together, i.e. The value of the integrated framework is higher than that of the sum of the individual products.
- Demonstrate the value of protecting critical infrastructure, using testing scenarios on critical infrastructure simulators.
- Act as a fully operational demo environment for dissemination and PR activities, such as conferences, marketing events and other demonstrations.
- Allow training on a neutral non-production environment.
- Demonstrate different deployment options, such as on premise and cloud deployments.

The CIPSEC prototype contains 3 major elements:

1. **CIPSEC Framework**: The detection and protection platform. Will be described in section 3.3
2. **Critical infrastructure simulators**: The asset to be protected. Will be described in section 3.4 (Prototype Simulators).
3. **Attacking tools**: The attacker in several testing scenarios. Most of the attacked are simulated from a Kali Linux machine, which is an open-sourced Debian-derived Linux distribution designed for digital forensics and penetration testing.

There are two flavours of the prototype, that are installed and operated independently:

1. **On-Premise deployment**: The whole framework is installed locally. No Internet connectivity is needed. Also referred to as “Offline Deployment”.
2. **Cloud deployment**: Most of the framework tools are deployed remotely and operated by the responsible partner. The local components are connected to the Internet and integrated with the remote components.

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 was performed using the following methods:

- Distributed work environment to ease the integration among components and speed up the team work. Remote connections were allowed from the relevant partners to the prototype.
- Several joint working sessions to test the integrated framework and enable collaboration.
- Researching and deploying critical infrastructure simulators, which are either open-source tools or developed in-house.
- Continuous feedback over the life cycle of the project, during general assemblies and advisory boards.

Notice that some of the CIPSEC Framework services are currently not included in the prototype:

- Updating and Patching: A supplementary service to the CIPSEC framework, which is not installed on the on-premise prototype.
- Contingency Plan: Not relevant for laboratory environment.
- Vulnerability Assessment: The local lab environment is too small to properly assess.

When demonstrating the CIPSEC Framework using the prototype, these services can be demonstrated using other methods, such as leveraging a pilot environment.
3.2 Prototype Architecture

3.2.1 Offline Deployment

In the offline deployment, all components are connected directly to each other using a network switch, simulating a common organizational closed LAN network, which is a typical architecture in critical infrastructure networks. No Internet connection is needed, except for occasional updates (such as virus definitions).

The main physical components are as follows:

- A physical laptop machine (A.K.A. “Prototype Laptop”), running VMware Workstation hosting most components as virtual machines in the same network.
- Another laptop, which hosts the XL-SIEM server virtual machine. Separating the XL-SIEM to a different host was necessary due to load distribution.
- Physical CIPSEC components, such as UoP’s HSM devices, EMP’s Secocard and WOS Jamming simulator.

Further information on the above components will be provided in this section.

The following diagram describes the offline deployment environment:

![CIPSEC Prototype Offline Deployment Diagram](image-url)
3.2.1.1 Prototype Host

- The physical machine hosting various virtual guests.
- Technical specifications:
  - Model: DELL Latitude 5480
  - Name: “LAP-CIPSEC”
  - Users:
    - Regular activities: “cipsec”
    - Administration activities by COMSEC: “admin”
  - CPU: Intel Core i7-7600U
  - RAM: 32 GB
  - OS: Windows 10 Enterprise 1709 64-bit
  - Virtualization platform: VMWare Workstation 14.1.2
  - Local IP Address: 192.168.40.200
  - Default Gateway: 192.168.40.2
  - RDP connection address over the Internet: 199.203.120.66:44445
    - Used for installation activities by partners
    - Possible only when located in COMSEC datacenter
    - Allowed from Comsec-WiFi and from partner addresses

3.2.1.2 XL-SIEM Host

A physical laptop that is connected to the local switch containing the XL-SIEM server side, including its dashboard. As presented in previous deliverables, this asset plays the role of Anomaly Detection Reasoner within the CIPSEC Architecture.

- Model: DELL Latitude E5520
- Name: “massif”
- User: “atos”
- CPU: Intel Core i7-2640M
- RAM: 16GB
- OS: Windows 7 64-bit
- Virtualization platform: VirtualBox 5.2.20
- Local IP address: 192.168.40.100

3.2.1.3 Virtual Machines

The following table presents the various virtual machines hosted on the physical prototype laptop:

<table>
<thead>
<tr>
<th>Partner</th>
<th>Description</th>
<th>Machine Name</th>
<th>IP</th>
<th>Remote SSH</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMSEC</td>
<td>Rapid SCADA simulator</td>
<td>comsec-server</td>
<td>192.168.40.133</td>
<td>-</td>
<td>administrator</td>
</tr>
<tr>
<td>COMSEC</td>
<td>PLC simulator</td>
<td>PLC-ModbusSimulator</td>
<td>192.168.40.41</td>
<td>-</td>
<td>administrator</td>
</tr>
</tbody>
</table>
The additional laptop contains the XL-SIEM server, including the XL-SIEM dashboard:

Table 6. Offline prototype virtual machines

<table>
<thead>
<tr>
<th>Partner</th>
<th>Description</th>
<th>Machine Name</th>
<th>IP</th>
<th>Remote SSH</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOS</td>
<td>XL-SIEM CyberAgent &amp; IDS</td>
<td>IDS-Agent_Proto-Atos</td>
<td>192.168.40.134</td>
<td>199.203.120.66:22226</td>
<td>cyberagent</td>
</tr>
<tr>
<td>FORTH</td>
<td>Honeypot sensor</td>
<td>FORTH-Honeypot</td>
<td>192.168.40.132</td>
<td>199.203.120.66:22224</td>
<td>dcsuser</td>
</tr>
<tr>
<td>FORTH</td>
<td>Honeypot visualization</td>
<td>FORTH-Dashboard</td>
<td>192.168.40.136</td>
<td>199.203.120.66:22227</td>
<td>root</td>
</tr>
<tr>
<td>BD</td>
<td>GravityZone</td>
<td>GravityZoneEnterprise-BD</td>
<td>192.168.40.141</td>
<td>199.203.120.66:22221</td>
<td>bdadmin</td>
</tr>
<tr>
<td>AEGIS</td>
<td>Forensics &amp; Dashboard</td>
<td>AegisVM</td>
<td>192.168.40.119</td>
<td>199.203.120.66:22225</td>
<td>aegis</td>
</tr>
<tr>
<td>UPC</td>
<td>Unified Dashboard &amp; Anonymization</td>
<td>dashboard-server</td>
<td>192.168.40.140</td>
<td>199.203.120.66:22223</td>
<td>root</td>
</tr>
<tr>
<td>WOS</td>
<td>DoSSensing</td>
<td>Worldsensing_cips ec_demo_viena2018_ubuntu64-bit</td>
<td>192.168.40.137</td>
<td>199.203.120.66:22228</td>
<td>alampropoul os</td>
</tr>
</tbody>
</table>

Table 7. Offline prototype virtual machines on the additional host

A description of each component is provided in section 2.2.

3.2.1.4 Other Physical Devices

Several physical components of the CIPSEC Framework are also integrated into the offline prototype, and connected through a switch:

Table 8. Other devices connected to the offline prototype

<table>
<thead>
<tr>
<th>Partner</th>
<th>Description</th>
<th>IP</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>UoP</td>
<td>CI Endpoint A</td>
<td>192.168.40.142</td>
<td>Physically connected to a dedicated HSM device.</td>
</tr>
<tr>
<td>UoP</td>
<td>CI Endpoint A</td>
<td>192.168.40.143</td>
<td>Physically connected to a dedicated HSM device.</td>
</tr>
<tr>
<td>EMP</td>
<td>Ubuntu workstation</td>
<td>192.168.40.181</td>
<td>Ubuntu workstation to demonstrate smart card authentication capabilities.</td>
</tr>
<tr>
<td>EMP</td>
<td>Secocard</td>
<td>192.168.40.144</td>
<td>Connected to the workstation with USB. Connected to the network (for XL-SIEM integration) using Wi-Fi through a wireless adapter.</td>
</tr>
<tr>
<td>EMP</td>
<td>Wireless Adapter</td>
<td>192.168.40.180</td>
<td>Physically connected to the switch. Allows Wi-Fi connectivity to the Secocard.</td>
</tr>
<tr>
<td>WOS</td>
<td>Jammer detector</td>
<td>192.168.40.130</td>
<td>Physically connected to the mains, to the network and to an NTP device.</td>
</tr>
</tbody>
</table>
Note that these devices are not essential for every demonstration of the prototype. They can be connected when needed, according to the use-case and specific demonstration requirements.

Additional description of the above components is provided in section 2.2.

### 3.2.2 Cloud Deployment

In the cloud deployment, most of the CIPSEC Framework components (i.e. "products") are hosted remotely and operated by the responsible partner. Only the essential components are installed on the laptop, such as the simulators, honeypot, and various agents which collect information (IDS, Forensics, Anti-Virus agent). This deployment simulates an organizational network with Internet access, where the CIPSEC customer prefers to outsource the operational burden of the tools (such as installation and server hosting).

The main components are as follows:

- A physical laptop machine (A.K.A. “Prototype Laptop”), running VMware Workstation hosting several components as virtual machines in the same network.
- Additional remote CIPSEC components, connected over the Internet.

Physical CIPSEC devices (such as HSM and Secocard) can also be connected, using a switch which will create a LAN.

The following diagram describes the cloud deployment environment:

![CIPSEC Prototype Cloud Deployment](image-url)
3.2.2.1 Prototype Host

- **Name:** “LAP-DEMO"
- **Users:**
  - Regular activities: “cipsec”
  - Administration activities by COMSEC: “comsecadmin”
- **CPU:** Intel Core i7-7600U
- **RAM:** 16 GB
- **OS:** Windows 10 Enterprise
- **Virtualization platform:** VMWare Workstation 11
- **Local IP Address:** 172.16.100.1
- **RDP connection address over the Internet:** 199.203.120.66:44444
  - Used for installation activities by partners
  - Possible only when located in COMSEC datacenter
  - Allowed from Comsec-WiFi and from partner addresses

3.2.2.2 Virtual Machines

The following table presents the various virtual machines hosted on the physical prototype laptop:

<table>
<thead>
<tr>
<th>Partner</th>
<th>Description</th>
<th>Machine Name</th>
<th>IP</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMSEC</td>
<td>Rapid SCADA simulator</td>
<td>comsec-server</td>
<td>192.168.40.133</td>
<td>administrator</td>
</tr>
<tr>
<td>COMSEC</td>
<td>PLC simulator</td>
<td>PLC-ModbusSimulator</td>
<td>192.168.40.41</td>
<td>administrator</td>
</tr>
<tr>
<td>COMSEC</td>
<td>Attacker machine</td>
<td>Kali-Linux</td>
<td>192.168.40.129</td>
<td>root</td>
</tr>
<tr>
<td>FORTH</td>
<td>Honeypot sensor</td>
<td>FORTH-Honeypot</td>
<td>192.168.40.132</td>
<td>dcsuser</td>
</tr>
<tr>
<td>AEGIS</td>
<td>Forensics</td>
<td>AegisVM</td>
<td>192.168.40.119</td>
<td>aegis</td>
</tr>
</tbody>
</table>

Table 9. Online prototype virtual machines

A description of each component is provided in section 2.2.
3.3 CIPSEC Framework Tools of the Prototype

The CIPSEC Framework products described below form the prototype components for both offline and cloud deployments.

3.3.1 ATOS XL-SIEM & IDS

There are several prototype components under the responsibility of ATOS:

- **Suricata Network IDS**: Inspects the network activity between all of the VMs (using promiscuous mode) and send notifications to XL-SIEM.
  
  Installed on "IDS-Agent_Prototype-Atos" virtual machine on the prototype laptop, both in the offline and cloud deployments.

- **XL-SIEM CyberAgent**: Receives logs from other CIPSEC components using the syslog protocol, filters them, aggregates them, and sends to the XL-SIEM server.
  
  Installed on "IDS-Agent_Prototype-Atos" virtual machine on the prototype laptop, both in the offline and cloud deployments.

- **XL-SIEM Server**: Receives information from the XL-SIEM CyberAgent, performs correlations and presents events and alerts in a dashboard. Integrated into the Unified dashboard as well.
  
  - Offline Prototype – Installed on a virtual server\(^1\) on a dedicated laptop host, as described in section 3.2.
  
  - Cloud Prototype – Installed on a remote server\(^2\) and accessed by the XL-SIEM agent over the Internet.

  Both environments can be accessed using the user "proto".

3.3.2 UPC Dashboard

In the offline prototype, the UPC Dashboard is deployed in a virtual machine running Debian with a NodeJS server, named "dashboard-server". It consists of two primary modules, the OAuth2.0 server and the Dashboard.

The OAuth2.0 server is the authentication server for the Dashboard and all its tools and provides the single sign on mechanism. When deploying the dashboard, a users' schematic must be provided in order to set up the user database, after that, a new user may be added at any time. Each tool in the CIPSEC framework must be configured in order to use the deployed OAuth2 server.

The Dashboard is the visualization tool for the CIPSEC framework. It joins the individual dashboard of all the tools into a single interface. When deploying the dashboard, a configuration file must be used in order to set the desired URL address of each tool, as well as the URL address of the OAuth2.0 server that will be used for the single sign on mechanism. When the configuration is established, the dashboard must be recompiled with the new parameters in order to be deployed.


In the cloud prototype deployment, a remote instance of the Unified Dashboard used. Connection is possible over the Internet using the following URL: [https://olympus.epsevg.upc.edu/cipsec](https://olympus.epsevg.upc.edu/cipsec) (user: "prototype").

---

\(^1\) [https://192.168.40.135/xl-siem](https://192.168.40.135/xl-siem)

\(^2\) [https://cipsec.atosresearch.eu/xl-siem](https://cipsec.atosresearch.eu/xl-siem)
3.3.3 Bitdefender GravityZone

An Anti-Malware solution, installed on several endpoints in the prototype (SCADA Server, PLC Simulator and the physical host). The agents on the endpoints are managed by a GravityZone server and sends alerts to the server, which sends them to XL-SIEM.

Events can also be inspected locally by examining the agent status (located in the Windows tray menu).
- Offline Prototype – The server component is installed on the “GravityZoneEnterprise-BD”\(^1\) virtual machine.
- Cloud Prototype – The server component is installed remotely\(^2\) and managed by BitDefender.

3.3.4 FORTH’s Honeypot

The initial version of the prototype was deployed with all the components deployed on the cloud and the communications happening over the Internet. The final version of the prototype that is a ready-to-deploy system, able to operate inside the Critical Infrastructure premises even with no internet connectivity. In the following paragraphs, we are going to be describe both these deployments.

The FORTH’s Honeypot solution consists of three main components: the honeypot sensors, the backend server and the Honeypot dashboard. All these components come in the form of preconfigured VM images and can be deployed either locally or on the cloud based on the configuration. Backend server is used to register new honeypot sensors, manage the existing ones, receives and logging all the alerts from the different honeypot sensors. Moreover, the honeypot sensors can be configured to communicate with XL-SIEM through syslog. All honeypots included in the solution were enhanced in order to be able to send the produced logs and alerts through syslog in a common format that can be parsed and correlated by the XL-SIEM.

3.3.4.1 Cloud Deployment

The Virtual Image for the FORTH’s solution that is used in the cloud prototype is a VM that hosts all the different honeypot sensors. This can include a Low-Interaction Honeypot, a SSH Honeypot, an ICS/SCADA honeypot and a DDoS amplification detection honeypot. Additionally, the backend server and the honeypot dashboard are set up to run on the cloud. The honeypots in the sensors VM are configured to report the events to the FORTH’s backend and to XL-SIEM local agent.

\(^1\) [http://192.168.40.141](http://192.168.40.141)
\(^2\) [https://cipsec.bitdefender.com](https://cipsec.bitdefender.com)
A dedicated VM with the honeypots sensors was deployed in the prototype and was configured to report back to the backend server with dedicated credentials in order to distinguish its traffic for the one coming from the pilots' deployed sensors. A new user account, prototype user, was set up in the backend server and in the dashboard so that we could access all the information and alerts produced during the tests on the prototype. In the final iteration of tests, we updated the honeypot image to be able to avoid detection by known fingerprinting tools and integrated an updating and patching mechanism.

The virtual machine name is “FORTH-Honeypot”.

Sample Configuration for the online deployment:

```bash
// config.ini
[GENERAL]
panel_root = "139.91.XXX.XXX"

[MYSQL_DATABASE]
db_user = "kippo_user"
db_pass = "xxxxxx"
db_host = "139.91.XXX.XXX"

[POSTGRESQL_DATABASE]
db_user = "xmpp_user"
db_pass = "xxxxxx"
db_host = "139.91.XXX.XXX"

[EMAIL_SERVER]
mail_mode = "smtp"
mail_port = "587"
mail_host = "XXX.XXX.com"
mail_authmethod = "login"
mail_from_email = "XXX@XXX.com"
mail_encryption = "tls"
mail_password = ""

[LDAP_SERVER]
port = "389"
```

Sample configuration for Forth’s Honeypot Sensor

```bash
//variables_script.sh
#!/bin/bash

#configure network
ipaddress=xxx.xxx.xxx.xxx
netmask= xxx.xxx.xxx.xxx
getnet= xxx.xxx.xxx.xxx
broadcast= xxx.xxx.xxx.xxx
gateway= xxx.xxx.xxx.xxx

#configure dionaea
xmppserver= xxx.xxx.xxx.xxx
xmppuser=xxx
xmppass=xxx

#configure syslog
syslogip= xxx.xxx.xxx.xxx
```

**Figure 68. FORTH’s Honeypots on cloud Prototype deployment**

A dedicated VM with the honeypots sensors was deployed in the prototype and was configured to report back to the backend server with dedicated credentials in order to distinguish its traffic for the one coming from the pilots’ deployed sensors. A new user account, prototype user, was set up in the backend server and in the dashboard so that we could access all the information and alerts produced during the tests on the prototype. In the final iteration of tests, we updated the honeypot image to be able to avoid detection by known fingerprinting tools and integrated an updating and patching mechanism.

The virtual machine name is “FORTH-Honeypot”.

Sample Configuration for the online deployment:

```bash
// config.ini
[GENERAL]
panel_root = "139.91.XXX.XXX"

[MYSQL_DATABASE]
db_user = "kippo_user"
db_pass = "xxxxxx"
db_host = "139.91.XXX.XXX"

[POSTGRESQL_DATABASE]
db_user = "xmpp_user"
db_pass = "xxxxxx"
db_host = "139.91.XXX.XXX"

[EMAIL_SERVER]
mail_mode = "smtp"
mail_port = "587"
mail_host = "XXX.XXX.com"
mail_authmethod = "login"
mail_from_email = "XXX@XXX.com"
mail_encryption = "tls"
mail_password = ""

[LDAP_SERVER]
port = "389"
```
3.3.4.2 Offline Deployment

For the offline deployment, most of the VMs from the various solutions included in the CIPSEC framework, were deployed in a single machine and another machine was used to host the ATOS XL-SIEM as depicted in Figure 69. The same Virtual Images of FORTH’s solution that were used in the cloud prototype, are also used for the offline deployment. The FORTH’s VM hosts all the different honeypot sensors. This includes a Low-Interaction Honeypot, a SSH Honeypot, a ICS/SCADA honeypot and a DDoS amplification detection honeypot. Additionally, the backend server and the honeypot dashboard were set up to run offline. The honeypots in the sensors VM is configured to report the events to the locally deployed backend and to the local XL-SIEM agent. A dedicated VM with all the honeypots sensors was deployed in the prototype and was configured to report back our local backend with the credentials of the prototype user. A local prototype user was set up in the backend server and in the local dashboard so that we could access all the information and alerts produced during the tests on the offline prototype. In the final iteration of tests, we updated the honeypot image to be able to avoid detection by known fingerprinting tools and integrated an updating and patching mechanism.

The honeypot sensor is installed on the “FORTH-Honeypot” virtual machine, while the backend and dashboard are installed on “FORTH-Dashboard”.

```ini
server = "139.91.XXX.XXX"
client_dn = "USER@ics.forth.gr"
password = ""
base_dn = "XXXXXXXXXXX"

[MYSQL_CONPOT_DATABASE]
db_user = "conpot_user"
db_pass = "xxxxxx"
db_host = "139.91.XXX.XXX"

[CHART_ONLINE]
isOnline=1

[OAUTH_LOGIN]
clientId = "client-testing-01"
clientSecret = "xxxxxx"
redirectUri = "https://prototype-dashboard.ics.forth.gr/index.php"
urlAuthorize = "https://olympus.epsevg.upc.edu/oauth2/validation"
urlAccessToken = "https://olympus.epsevg.upc.edu/oauth2/token"
urlResourceOwnerDetails = "https://olympus.epsevg.upc.edu/oauth2/resource"

syslogport=514
DIONAEA_clean=0
HT_clean=0
NETWORK_set=0
HOSTNAME_set=0

DIONAEA_run=1
KIPPO_run=1
DDOS_run=1
CONPOT_run=1
FARPD_run=0

conpot_sql_enabled=1
conpot_sql_host= xxx.xxx.xxx.xxx
conpot_sql_db=conpot
conpot_sql_user=conpot
conpot_sql_pass=xxx

cidr=" xxx.xxx.xxx.xxx/xx"
cidr=""

//config.ini
```

```bash
//variables.script.sh
```
Sample Configuration for the offline deployment:

```
// offline_config.ini

[GENERAL]
panel_root = "192.168.40.136"

[MYSQL_DATABASE]
db_user = "kippo_user"
db_pass = "xxxxxxxx"
db_host = "192.168.40.136"

[POSTGRESQL_DATABASE]
db_user = "xmpp_user"
db_pass = "xxxxxxxx"
db_host = "192.168.40.136"

[EMAIL_SERVER]
isOnline=1

[LDAP_SERVER]

[MYSQL_CONPOT_DATABASE]
db_user = "conpot_user"
db_pass = "xxxxx"
db_host = "192.168.40.136"

[CHART_ONLINE]

[OAUTH_LOGIN]
clientId = "client-testing-01"
clientSecret = "xxxxxxxx"
urlAuthorize = "http://192.168.40.140:3000/oauth2/validation"

// offline_config.ini
```

### 3.3.5 AEGIS Forensics

The forensics solution by AEGIS, integrated in the CIPSEC prototype, contains several components:

- **Agents**: Installed on local servers (SCADA Server and PLC Simulator) to capture system metrics. For demonstration purposes, it is configured to detect high CPU load.
Forensics Server: Captures traffic in the local network and receives logs from the agents. Installed on the “AegisVM” virtual machine, in both prototype deployments. The logs can be queried locally by running the following command on the Aegis server:

tail -f /usr/local/nagios/var/nagios.log

AEGIS Visualization Toolkit (AVT): A dashboard user interface which present the captured information in various graphs and diagrams. Accessible only through the CIPSEC (UPC) Dashboard. Authentication is allowed using OAuth 2.0.

- Offline Prototype – Installed on the same machine as the forensics component (“AegisVM”).
- Cloud Prototype – Installed on a remote server which is operated by AEGIS, accessed by the forensics component over the Internet.

3.3.6 WOS DoSSensing

DoSSensing is a jamming detector device for wireless denial of service jamming attacks. The solution is integrated in the offline prototype, using a dedicated virtual machine:

Worldsensing_cipsec_demo_vienna2018_ubuntu64-bit.

Basically, the virtual machine is an Ubuntu 16.04 (IP address 192.168.40.137) running the two main blocks of the DoSSensing solution, the sensor and the cloud. If we take a look at the architecture diagram we will see the separation between sensor and cloud.

![DoSSensing Architecture Flow Diagram](image)

The blue blocks correspond to the sensor and the red ones run on the cloud. As we are not connecting to the cloud, the Monitoring Server and Visualization tools should run on the virtual machine. On the other hand, we do not run a real jammer detector (as we do not want to use a real jammer), thus we run the detection software that regularly runs on the sensor (processing board) on the virtual machine too.

Then, instead of having an SDR module, we will use pre-recorded signal (of real jammers) to inject to the detection software and the rest of the flow will be the same as it would be in a real environment scenario. All this software runs locally on the same virtual machine.

The CIPSEC dashboard will integrate the DoSSensing jammer detection dashboard at the following URL: https://192.168.40.137/grafana/d/rRlqASzmk/sdjd-sensor-view?orgid=2&kiosk&refresh=1s&scope=proto&user=prototype1&token=

(token needs to be replaced by the correct token to be validated).

The monitoring software will generate events that it will communicate to the XL-SIEM whenever there is an alert situation going on. In this case, we just configured the corresponding local URL (IP 192.168.40.134, port 514) of the XL-SIEM CyberAgent in order to forward the alert events.

Finally, we also included a web interface to be able to select among the different recorded files to inject to the detection software. This interface can be used to demonstrate the detection and communication of alert events of different types of jammers. It can be accessed at the following URL: http://192.168.40.137:8888/
3.3.7 UPC Data Privacy

A python-based application runs in the background and has two inputs: a privacy policy and the scope provided by the user (the pilot he belongs to). The privacy policy lies within a text file formatted in JSON to define the data to anonymize and the mechanisms to use with such aim.

The application retrieves all the cybersecurity data (logs) from the XL-SIEM to transform the sensitive data by means of different privacy mechanisms (generalization, pseudonymization, suppression). The anonymized logs are then sent to a MISP platform for sharing purposes. The MISP interface where said logs are stored looks as follows.

![Figure 71. Result of the anonymization process performed by the Data Privacy tool. Anonymized cybersecurity data is stored in a MISP sharing platform.](image)

Figure 71 shows the data of a log record stored in the MISP after being anonymized by the Data Privacy Tool. In this example, we can see the effect of pseudonymization (4th row of the table), where instead of the organization name appears a pseudonym (d77d5e503ad1439f585ac494268b351b) which is difficult to be linked with the actual organization name. In the fifth row of the table, we can also see the effect of generalization, where instead of the full source IP address we can only see its network identifier (but not its host identifier). Finally, the three last rows of the table show the effect of suppression, where the complete fields of user data have been erased.

The application is installed on the "dashboard-server" virtual machine of the offline prototype and is integrated with the XL-SIEM server for retrieving cybersecurity logs, as well as with a MISP platform for sharing anonymized data.

The URL to access the tool is the following:
https://olympus.epsevg.upc.edu/cipsec/login

The user could be any of the corresponding three pilots.

3.3.8 Empelor’s Secocard

Secocard is a hardware CIPSEC component for smart card authentication and secure storage purposes.

When integrating the Secocard with the prototype, several components are required:

- **Ubuntu Workstation**: A physical laptop, with Ubuntu distribution of Linux installed. A smart card authentication to the workstation is demonstrated using the Secocard.

- The existing Windows machine can’t be used, as the Secocard supports Linux operating system. The machine cannot be deployed as a virtual machine hosted on the existing Windows host, as smart card authentication can only be performed for physical machines.
- **Secocard Device**: The device itself, connected to the workstation with USB cable and connected to the network using Wi-Fi. Allows smart card authentication to the workstations and sends events to the XL-SIEM agent over the network.

- **Smart Card**: Inserted to the Secocard device to allow authentication.

- **Wireless Network Adapter**: Physically connected to the local switch, allowing Wi-Fi connectivity for the Secocard device.

### 3.3.9 UoP HSM

The UoP HSM is a hardware CIPSEC physical component that is directly connected to a Critical Infrastructure device. It can provide several different services for the Host device, cryptography primitive executions, data/message integrity, and secure storage. The UoP HSM provides a trusted environment that can perform all the above operations. The environment is protected against hardware and software side channel attacks as well as malicious codes. It uses ARM Trustzone technology and dedicated secure and side channel attack resistant hardware core peripherals implemented within an FPGA hardware System-on-Chip to achieve the above goals. The UoP HSM is associated with the XL-SIEM agent (CyberAgent) locally or remotely (depending on the use case scenario).

In the CIPSEC prototype, we use the HSM primarily as a message integrity provision mechanism. The HSM, when connected to a Host device through USB, it provides a secure environment where the Host sends messages and the HSM returns either HMAC based message digests or Asymmetric Key Cryptography digital signatures (ECDSA) thus implementing a message integrity and authentication mechanism (in case of Digital signatures). The HSM is also responsible for validating HMAC digests or ECDSA signatures. In the prototype, when a message integrity failure is noticed then a syslog event is send to the XL-SIEM agent of the prototype.

Due to the need for a host device, in the prototype we have also included two end point devices acting as critical infrastructure endpoints (raspberry pie with embedded Linux installed) and are connected to the prototype local network. Each such device is also directly (through USB) connected with an HSM (one per device). The main functionality of the HSM inside the prototype is related to the services that are provided to the two Critical Infrastructure endpoints. These end points use the HSMs to create a secure communication channel between them that can provide both integrity and confidentiality. To achieve that they use the HSMs to generate public/private key pairs (that are stored only inside the HSMs) and execute a key agreement protocol to generate a session key (AES based key). Then they employ Authenticated encryption (AES CCM) or the MAC and encrypt (HMAC+AES) to transmit messages and communicate with each other through a secured connection. All security related operations for the above scenario is performed inside the HSM secure environment. If an attacker tries to manipulate the communication messages or tries to perform a password attack on the HSM to Host communication, then the HSM Host software component detects it and sends an event to the XL-SIEM agent.
3.4 Prototype Simulators

3.4.1 Rapid SCADA

Rapid SCADA is free, open source, fully featured SCADA software. It is used in the Prototype as the SCADA simulator, in this case the SCADA simulated a cooling system of a server room monitoring the temperature.

The simulator runs on the “comsec-server” virtual machine. It receives temperature information from PLC Modbus Simulator machine and showing the system dashboard as a web server application.

Contains several applications:
- SCADA-Administrator
- SCADA-Communicator
- SCADA-SchemeEditor
- SCADA-Server
- SCADA-TableEditor
- SCADA-Webstation (UI)

All the above applications are accessible through the Start Menu (Start Button > Down arrow for Apps > SCADA).

Direct URL to the user interface: [http://192.168.40.133/scada](http://192.168.40.133/scada) (user: admin)

![Figure 72. Screenshot of the Rapid SCADA server room cooling system dashboard](image)

Full installation notes are documented in COMSEC’s internal knowledge base.

More information on RAPID-SCADA can be found on its public website [1].

3.4.2 PLC Simulator

A Modbus PLC simulator named “Mod_RSsim”, which runs on the “PLC-ModbusSimulator” server.

[1] https://rapidscada.org/
Simulates temperature from several PLC sensors and sends it via Modbus protocol to the Rapid-SCADA server. The server on which this tool runs is hardened and completely closed, mimicking best a behaviour of an HMI sensor receipting temperature and communicating it over a port 502 TCP/IP Modbus protocol. Based on a free TCP/IP Modbus simulator software. More information can be found on its public website\(^1\).

### 3.4.3 Medical Web Site

A Cyber-Clinic simulated application, developed for educational purposes and installed on the “comsec-server” (aka Rapid SCADA server) virtual machine. The application is vulnerable to various web application attacks, such as malware upload, SQL Injection, XSS, credentials brute-force, file upload DoS and parameter tampering.

Can be accessed with a browser using the following address:

http://192.168.40.133

The application state can be initialized by visiting the following address:


### 3.5 Demonstration Scenarios

This section describes how to execute various testing scenarios and how each scenario is detected by the tools. The demonstration scenarios allow to assure the framework works as intended.

Further demonstrations can be performed ad-hoc, according to specific requirement or to demonstrate specific capability.

#### 3.5.1 Network Scan

Intense network scan for reconnaissance purposes.

Detected by [Forth’s Honeypot](http://www.plcsimulator.org) and by [Atos’ IDS](http://www.plcsimulator.org) (Suricata).

1. Open the nmap GUI tool (“Zenmap”) on the Kali machine.
2. Run the default “Intense scan” profile against the virtual subnet.
   a. Fill the target with 192.168.40.0/24.
   b. Use the profile Intense scan.
   c. The command should look as `nmap -T4 -A -v 192.168.40.0/24`.
   d. Click scan. The scan may take a while.
3. After the scan was finished, identify the components and services that have been detected, including:
   a. SCADA Simulator (133)
   b. PLC Simulator (41). Notice that port 502, labelled as `Modbus TCP`, was detected on this host.
4. Verify that the scans were detected by the honeypot and by the IDS, and presented in XL-SIEM, Honeypot Dashboard and the Unified Dashboard.

\(^1\) [http://www.plcsimulator.org](http://www.plcsimulator.org)
Extensions and alternatives:

1. Run the scan only against the honeypot IP address (132) to verify it is detected.
2. Run an in-depth scan on the more interesting hosts (SCADA server, PLC simulator):
   a. Fill the target with 192.168.40.41,133
   b. Use the profile Intense scan, all TCP ports
   c. The command should look as `nmap -p 1-65535 -T4 -A -v 192.168.40.41,133`
   d. Click scan.
3. Run a nmap vulnerability scan on the interesting hosts (comsec-server, PLC simulator):
   a. Enter the following command in the terminal:
      `nmap --script vuln 192.168.40.41,133`
   b. Click scan.
4. Use the CLI nmap tool instead of the GUI tool (Zenmap).
   a. Notice that while the CLI is more powerful, it may not suit well for presentation purposes.
   b. Relevant commands are:
      ```
      nmap -v -A -sV --version-intensity 5 -Pn -oA ~/Desktop/AttackScenarios/NMapOutputs/hosts_scan_output 192.168.40.0/24
      nmap -v -Pn -p 1-65535 -oA ~/Desktop/AttackScenarios/NMapOutputs/depth_scan_output 192.168.40.41,133
      nmap -v -Pn --script vuln -oA ~/Desktop/AttackScenarios/NMapOutputs/vuln_scan_output 192.168.40.41,133
      ```

3.5.2 Vulnerability Exploit

Gain administrative control over the PLC Modbus simulator machine, by using the Metasploit framework to exploit an SMB vulnerability on the machine.

Detected by Atos' IDS (Suricata).

1. Open the Metasploit console by executing the command “msfconsole” on the Kali machine.
2. Use the MS17-010 vulnerability scanner by entering:
   ```
   use auxiliary/scanner/smb/smb_ms17_010
   ```
3. Set the target IP addresses of the SCADA server and the PLC simulator that were identified in the previous scenario (network scan) by entering:

   set rhosts 192.168.40.41,133

4. Execute the scan by entering “exploit”.

   The scan is expected to succeed on host 192.168.40.41 and to fail on host 192.168.40.133, as the comsec-server is patched against this vulnerability.

5. Use the MS17-010 vulnerability exploit by entering:

   use exploit/windows/smb/ms17_010_eternalblue

6. Set the target IP address by entering:

   set rhost 192.168.40.41

   Execute the exploit by entering exploit.

   The exploit is expected to succeed.

   ![Figure 74. Exploiting vulnerability against PLC simulator](image)

7. Run remote commands on the PLC machine.
   a. Gain persistence by creating an administrative user using the following commands:

      net user /add attackeruser Aa123456! # create user “attackeruser” with password “Aa123456!”

      net localgroup administrators attackeruser /add # add user to administrators

      net localgroup administrators # verify that the user was created and added as admin

   b. Stop the PLC Modbus simulator:

      tasklist | find “sim” # find the Modbus simulator process

      taskkill /f /im mod_RSsim.exe # terminate the Modbus simulator process
8. Connect to the ModbusSimulator machine and notice that the simulator was closed.
9. Connect to the SCADA HMI and notice any changes.
   
   http://192.168.40.133/scada (admin)

10. Verify that the attack was detected by the IDS, and presented in XL-SIEM and CIPSEC Unified Dashboard.

Extension and alternatives:
1. Exploit a vulnerability against the Honeypot server. The attempt will be detected by the honeypot, as well as the IDS. A SAMBA attack alarm will be correlated and raised.
2. Use other exploits of services which are hosted by the honeypot (SMB, FTP, MSSQL, MySQL, SIP, WINS).
3. Use other Metasploit modules, such as:
   a. MS10-061 vulnerability exploit: exploit/windows/smb/ms10_061_spoolss
   b. SMB Scanning: auxiliary/scanner/smb/pipe_auditor
   c. FTP Scanning: auxiliary/scanner/ftp/anonymous
4. Use a Metasploit script. Examples are located in COMSEC’s internal knowledge base.
3.5.3 Modbus Man-in-the-Middle

In this scenario, the attacker performs a Man-in-the-Middle network attack between the PLC simulator and the SCADA server. The attacker will sniff the network, detect the temperature data sent from the PLC to the SCADA, and will modify the temperature readings.

Detected by Atos’ IDS (Suricata).

1. Open the Rapid SCADA HMI and inspect the current temperatures. In this example, the initial temperature readings are -40°C. Use the following URL:
http://192.168.40.133/scada (user: admin)

![Rapid SCADA initial temperature readings](image)

2. Open a terminal on the attacker’s Kali machine.

3. Run the script: `~/Desktop/AttackScenarios/EttercapFilters/filter_builder.sh` with the desired temperature as an argument. The script will create an appropriate “Ettercap” filter that is able to inspect Modbus traffic and locate the relevant values. For example, to create a filter which will display the server rooms’ temperature as 85°C on all hallways, run:

```bash
~/Desktop/AttackScenarios/EttercapFilters/filter_builder.sh 85
```

![Creating an ettercap filter](image)

4. Open “Ettercap” on another terminal.

5. Click: Sniff -> Unified sniffing and choose “eth0”, to sniff the main network adapter’s network traffic. Click OK.

6. Click: Hosts -> Hosts list.

   a. Right click 192.168.40.41 -> Add to Target 1 (PLC Simulator).

   b. Right click 192.168.40.133 -> Add to Target 2 (Rapid SCADA).

7. Click Targets -> Current targets. Make sure that under “Target 1” there is 192.168.40.41 and under “Target 2” there is 192.168.40.133.

8. Click Mitm -> ARP poisoning. Enable “Sniff remote connections”. Click OK. This action will use ARP poisoning method to sniff network traffic.
9. Click Filters -> Load a filter. Choose the filter that was created on step 3. It should be at the following location: 
~/Desktop/AttackScenarios/EttercapFilters/<chosen_temperature>.ef

10. Run the attack and wait a for about a minute for the values to propagate.
11. Inspect the Rapid SCADA HMI again and see that all the temperatures were modified to the desired temperature:
12. Verify that the attack was detected by the **IDS** and presented in **XL-SIEM** and **CIPSEC Unified Dashboard**.

### 3.5.4 Malware by USB

Insert a USB external device that contains a malware to the endpoint.

This scenario is likely to happen in critical infrastructures that contains physically unprotected endpoints.

**Detected and prevented by **Bitdefender GravityZone**.**

1. Copy a malware to a USB external device.
   a. An EICAR test file can be downloaded from the Internet, or be created as a *.txt file with the following string:
      \[X5O!P%@AP\{4\}PZX54(P^)7CC)7}{EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H\]
   b. The Cridex Trojan can be copied from the Kali Machine.
2. Plug the USB device to the endpoint.
3. Verify that the attempted attack was detected and prevented by **Bitdefender GravityZone** by examining the agent log (BD tray icon).
4. Verify that the detection is presented in **XL-SIEM** and CIPSEC Unified Dashboard.

### 3.5.5 Downloading Malware

Download an infected file from the demonstration web site.

The victim can be lured into downloading the malware by leveraging a phishing technique.

**Detected and prevented by **Bitdefender GravityZone**.**

1. Make sure that the Apache web server is running on the Kali machine by typing:
   \`
   service apache2 start
   \`
2. Send a phishing email to the CIPSEC demonstrator (the victim) address “cipsecd@gmail.com”. The email should contain a link to “http://192.168.40.129/cridex.exe”.
3. From the comsec-server, open a web browser, access the victim’s email, find the phishing message and access the malicious link. The password for the email account (cipsecd@gmail.com) is documented in COMSEC’s internal knowledge base.
4. Verify that the attempted attack was detected and prevented by **Bitdefender GravityZone**, by examining the BD agent (located on the tray menu).
5. Verify that the detection is presented in **XL-SIEM** and the **Unified Dashboard**.
3.5.6 Uploading Malware

Upload a malicious file to a website.

Detected and contained by Bitdefender GravityZone.

1. Open the Kali machine.
2. Configure Burp as a proxy.
   a. Open the Burp Suite, configure the proxy settings. Disable intercept for now.
   b. Configure the browser with an appropriate proxy (Firefox: Settings, Advanced, Network, Connection Settings, Manual proxy, 127.0.0.1:8080).
3. Enter the following address on the website:

   Figure 82. Cyber-Clinic web application FAQ page

4. Upload an allowed file to the website, using the attachment feature (PDF or picture).
   Verify that the upload was successful.
5. Try to upload a non-allowed file (text or executable).
   For example, use the “demo text file” or the “cridex.exe” located on the desktop.
   Verify that the upload was blocked.
6. Configure Burp to intercept the requests to the web site.
   a. From the web application, upload the infected file. Examples can be located on the Kali’s desktop (EICAR and Cridex).
   b. On Burp’s intercept tab, locate the content type, change it to “image/jpeg” and forward the request.
   c. Try to access the uploaded file from the web site. This is expected to fail as the file was deleted by BitDefender GravityZone.
8. Open the web server VM and verify that the file was detected and deleted.
   a. Review the logs of the Bitdefender agent (BD tray icon).
   b. Ensure that the file was deleted from the following path: C:\inetpub\wwwroot\attachments
9. Verify that the detection is presented in XL-SIEM and CIPSEC Unified Dashboard.

Alternatives:
Upload an allowed file (picture or PDF) and intercept the request with Burp. Change the content to EICAR string and forward the request to the web site.
3.5.7 Denial of Service by CPU Load

Create a major CPU load on a server to simulate denial of service attack.

Will be detected by AEGIS Forensics.

1. Connect to SCADA server and/or to the PLC Modbus Simulator.
2. Run the CPUSTRESS.EXE tool, located at the desktop.
   a. Activate thread 1 and thread 2, with “Activity” set to “Maximum”.
   b. Open the Task Manager to make sure that the CPU is overloaded.
3. Open the SCADA HMI and notice any changes.
4. Check that the logs were captured by the AEGIS server and presented in the XL-SIEM, AEGIS Visualization and Unified Dashboard.

Notice that currently the agent samples the server resources every 5 minutes.

Alternatives:
Perform a remote denial of service attack to simulate a realistic attack. For example, use a vulnerability (scenario 2).

3.5.8 Wireless Jamming Simulation

Once the virtual machine starts running, both the sensor detection software and the cloud monitoring software are launched. We can access the visualization dashboard and see that the simulated sensor is running and there are no current attacks going on.

![Figure 83. DoSSensing Dashboard before initiating attacks](image)

The corresponding events are communicated to the XL-SIEM CyberAgent at 192.168.40.137 on port 514. The following are examples of the messages communicated to the syslog.

```
2018-11-15 08:37:05,003 ; (MainThread) ; sdrjdsyslog ; INFO ; {"user": "prototype1", "jnr": 0.0, "event_duration": 64565000, "nodeId": "3", "srcIp": "192.168.40.137", "time": "2018-11-15T09:37:05.000", "freq": "241200000", "type": "None", "event": "No Attack"}
```
If we connect to the jammer simulator user interface, we can activate/deactivate the different types of jammers to test the detection and communication of alert events to the XL-SIEM.

![Jammer Simulator Options](image)

Each button corresponds to a different type of jammer to be processed by the virtual machine and a different alert should be generated towards the XL-SIEM. The only thing to do is to click on the type of jammer selected and it will be processed by the software. For example, clicking on Pulsed Jammer and going back to the Jamming attacks view, we will see the following.

![DoSSensing Dashboard After initiating attacks](image)

This is going to generate the corresponding logs to the XL-SIEM CyberAgent. The following JSON messages will be received at the syslog.

```
2018-11-15 08:40:28,451 ; (MainThread) ; sdrjdsyslog ; CRITICAL ; {"user": "prototype1", "jnr": 17.391, "event_duration": 0, "nodeId": "3", "srcIp": "192.168.40.137", "time": "2018-11-15T09:40:28.000", "freq": "2412000000", "type": "Pulsed", "event": "Attack Started"}
```

```
2018-11-15 08:40:30,456 ; (MainThread) ; sdrjdsyslog ; INFO ; {"user": "prototype1", "jnr": 20.82, "event_duration": 0, "nodeId": "3", "srcIp": "192.168.40.137", "time": "2018-11-15T09:40:30.000", "freq": "2412000000", "type": "Pulsed", "event": "Attack Running"}
```

### 3.5.9 Logging in with Smart Card

A demonstration scenario for the Secocard is straightforward provided that the device is configured correctly and all the components that are described in 3.3.8 are available. In addition, the XL-SIEM monitoring system must
be properly configured so that communication between the XL-SIEM and Secocard has been established. The connection can be easily tested since the Secocard is sending automatically messages to the XL-SIEM every 50 seconds even if no event has occurred to indicate that the device is operating correctly.

Therefore, assuming that the system has been set up correctly, the operator will insert the card within the device and will try to logon to the operating system. Secocard will display a new screen indicating to the operator that the PIN must be entered on the Secocard. As soon as the operator enters the correct PIN, he will be able to logon to the operating system. In the meantime, the XL-SIEM monitoring tool will have received several messages (card has been presented to the reader, a correct pin has been given, etc.).

3.5.10 Message Integrity Failure Protection Using the HSM

In the demonstration scenario, we assume that we have two critical infrastructure nodes (an IT node or an OT node). One of the nodes has a series of sensors that collect measurements from the environment (e.g., Temperature from weather station or vibrations from train tracks/an industrial machine) and needs to transmit them to the second node that acts as a measurement collection and visualization point. The attack scenario includes a malicious entity that eavesdrops the communication channel and manipulates the transmitted data forcing the measurement collection node to display alarming values that will trigger an emergency alarm (e.g., very high temperature indicating a fire).

The demonstration setup consists of two raspberry pi3 (Rpi) embedded systems that act as the CI nodes. One of them (the Measurement node) has a series of sensors and a small OLED display screen to visualize the collected measurements locally (from the same machine). The other Rpi has a HDMI screen and acts as a CI collection and management centre that graphically visualizes the various remote sensor data, checks for abnormalities and raises alarms. Each one of the two Rpis is connected to a UoP HSM that handle all security operations. All communications are done through ethernet.

The script of the scenario has two stages. In the first stage the attacker eavesdrops the communication channel and gains insight on the current environmental status near the Measurement node. In the second stage of the attack, the attacker collects the transmitted packages and manipulates their payload so that they indicate abnormally high or low values that will trigger a false alarm on the collector node. We use the HSM to create a message integrity protected communication channel, using the HSM message integrity mechanisms to provide message integrity so that when the attacker manipulates packages this is immediately detected and reported to the XL-SIEM Cyber Agent. As an alternative use case we also create a secure channel so that the communication is not eavesdropped, and we also perform a MiTM attack to try to alter transmitted data.

3.5.11 Anonymizing Cyber Security Logs

Cybersecurity logs are aggregated within the XL-SIEM, and when the anonymization process is configured through the CIPSEC dashboard, such logs are then anonymized in real time.

1. In the graphical interface of the Data Privacy Tool, select the privacy policy to be applied: advanced, intermediate, or basic.
2. Start the anonymization process.

![Figure 86. Graphical Interface of the Data Privacy Tool](image-url)
3. The logs extracted from the XL-SIEM are anonymized and sent to the MISP platform, where such information is shared. The privacy-protected data would look as following.

<table>
<thead>
<tr>
<th>Date</th>
<th>Org</th>
<th>Category</th>
<th>Type</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-02-15</td>
<td>Network activity</td>
<td>other</td>
<td>192.168.X.X</td>
<td>Source IP associated to the detected alarm.</td>
<td></td>
</tr>
<tr>
<td>2019-02-15</td>
<td>Targeting data</td>
<td>target-machine</td>
<td>**********</td>
<td>Destination IP associated to the detected alarm.</td>
<td></td>
</tr>
<tr>
<td>2015-02-15</td>
<td>External analysis</td>
<td>other</td>
<td>4</td>
<td>Risk value evaluated by XL-SIEM</td>
<td></td>
</tr>
<tr>
<td>2019-02-15</td>
<td>External analysis</td>
<td>other</td>
<td>5</td>
<td>Priority value evaluated by XL-SIEM</td>
<td></td>
</tr>
<tr>
<td>2019-02-15</td>
<td>Internal reference</td>
<td>other</td>
<td>d7d8e520ad14936580cc494268b051b</td>
<td>Organization where the XL-SIEM Agent has been deployed</td>
<td></td>
</tr>
<tr>
<td>2013-02-15</td>
<td>Person</td>
<td>other</td>
<td>**********</td>
<td>Username associated with the detected event</td>
<td></td>
</tr>
<tr>
<td>2019-02-15</td>
<td>External analysis</td>
<td>other</td>
<td>8</td>
<td>Reliability value evaluated by XL-SIEM</td>
<td></td>
</tr>
<tr>
<td>2019-02-15</td>
<td>Other</td>
<td>other</td>
<td>**********</td>
<td>User data 1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 87. Anonymized data in the MISP platform

4. If necessary, stop the anonymization process using the same graphical interface.
4 CIPSEC Framework Innovations

Although the initial innovation actions of the project were reported in D2.3 CIPSEC products integration on the Unified Architecture on M18 and the final year was mainly dedicated to the evaluation of the framework, some innovative features have been developed by the solution partners in the final eighteen months of the project. Some of the innovation items have been brought forward during the actual integration of the solutions into the complete framework, whilst others are offered by the final framework itself. All these features have been developed in the context of tasks 2.1 and 2.5, respectively CIPSEC Security Framework optimisation and From the prototype to the final CIPSEC Security Framework.

Obviously, the main priority for each solution introduced in the CIPSEC framework is to successfully match the CIPSEC requirements posed by the Critical Infrastructure Security domain and be fully compatible with the overall CIPSEC framework technical and market goals. Nevertheless, all solutions must also be viewed as a commercial stand-alone product and, as such, target individually and through the CIPSEC framework a specific part of the relevant market. For this reason, all the CIPSEC security products/solutions were designed with strong innovation in mind, to better achieve technical and market benefits. The main innovation is described below for each solution:

The CIPSEC anomaly detection reasoner, supported by ATOS XL-SIEM product, can integrate inputs from many heterogeneous observable indicators of cyber-attacks without compromising on reliability. Also, the XL-SIEM system can support even legacy equipment monitoring (typically found in long-lifetime critical infrastructures). XL-SIEM brings intelligence about the traditional correlation ecosystem that exists today, providing information and visibility of the attacks produced on the assets of the organizations in real time. It has a real-time distributed and modular infrastructure that adapts to the specific needs of the organizations.

The Bitdefender antimalware solution can provide proactive detection for previously unseen malwares with an uncharted behaviour. In a way, the antimalware solution is capable of detecting anomalies in the system’s behaviour even if they are unknown to it through the introduction of new technologies like deep packet inspection and machine learning techniques.

Innovative honeypot solutions are integrated and combined in order to capture and analyse a broad range of attacks. They can analyse IT and OT infrastructure traffic and create replicas of real IT and OT services. They also include peripheral security solutions like rootkit hunters and SSH attack detectors.

The CIPSEC framework innovated by introducing, apart from software-based solutions, hardware security solutions. One example of this is that Denial of Service attacks on the physical layer of broad wireless band can be detected in an innovative way by DoSSensing, that operates as an external element sentinel to specifically detect Jamming attacks to any band(s) in which the wireless sensors, industrial IoT elements, and even computers connect to the Critical Infrastructure network.

Another example is Empelor’s innovative programmable, flexible and diverse card reader solution, that can be adapted to any critical infrastructure environment at-hand and that offers multi factor authentication.

The framework also includes a Hardware Security Module solution that is directly connected to CI host devices and acts as a trusted environment for security/cryptography related operations and secure storage. This solution is extremely fast since computation intensive cryptography operations are accelerated by hardware means and thus fits well to the critical, real-time nature of many CI systems.

Another important feature that the CIPSEC framework offers is the ability to visualize forensics events. By implementing and installing in the CI system Critical Infrastructure Performance Indicators (CIPIs), forensics measurements can be collected, analysed and visualized. Providing advanced, intuitive and detailed data visualizations to active (real time) cyber/digital forensics analysis where data from heterogeneous sources are aggregated, combined and presented in an intuitive manner is also an innovative feature.

Finally, the CIPSEC framework can handle private data by including and applying anonymization methodologies through a relevant tool wherever CI system needs it. The tool is based on innovative research on micro-aggregation methods and fast computational responses for anonymizing data.

CIPSEC has also brought innovation in the way education about cyber security applied to critical infrastructures is delivered. First, a three-level approach has been envisioned, which matches well with standard curricular paths that follow a top-down approach. Also, the materials combine slides, documents and videos, as well as physically-

---

1 CIPSEC D2.3: Products integration on the unified architecture
held courses, which have been delivered in the course of the project. To support this, a specific education platform is part of the framework itself and is seamlessly integrated in its architecture.

With respect to the application of contingency services, CIPSEC has provided guidance to the three pilots. In the case of HCPB, it has served as a clear awareness-raising activity that has triggered internal actions to improve their preparedness to undesired events of cyberattacks taking place and even succeeding. For DB and CSI, it has served to revise their plans and update them to be ready against the aggressive landscape of threats and attacks.

The vulnerability assessment service has applied cutting-edge methodologies and tools for early vulnerability detection. It has been applied to the three pilots, delivering confidential reports that have been leveraged by their internal teams to take mitigation actions so as to be ready should unexpected incidents happen.

The updating and patching service allows the user to follow up the status of the different components, ensuring that their running versions are the most recent ones, including all the needed patches. This is done by means of an appealing centralized interface which is part of the unified dashboard and provides at a glance the general picture to the user, enabling decision-making if needed.

Finally, the forensics service offers innovative visualisation tools to study and investigate in detail the facts that lead to determined behaviours within the monitored infrastructure, the motivations of a certain incident taking place, or the course of execution of an attack. The main added value is the way the data are presented and the aggregation algorithms that summarize the information in a friendly and nice-to-digest way, thus providing quick situational awareness to operators and facilitating faster response to similar future incidents.

Apart from innovation from individual components of the CIPSEC Framework, the integration process of those heterogeneous components into a unified, fully functional architecture has introduced several innovation aspects. Such aspects are focused on the acquisition, exchange and management of security related information (events, logs, alerts) existing within the CIPSEC framework. Thus, every component of the framework has introduced some data exchange mechanism or feature to its architecture in order to be compliant, integrated-ready to the overall CIPSEC architecture. In the following figure, all such mechanisms/features are presented and described in brief.

**Figure 88. Overall CIPSEC Innovations due to various solutions integration**
5 CIPSEC final evaluation settings and assessment summary

5.1 Final Evaluation Settings

The CIPSEC evaluation is part of WP4 and is the last in a series of phases that amongst others include design, development, deployment and integration. During the previous work packages, a significant amount of work had taken place to create the unified architecture and a working platform. In WP4, the CIPSEC platform must be evaluated. Therefore, the first deliverable of WP4, namely D4.1, had to set the basis for the evaluation methodology, while the second deliverable, namely D4.2 added additional material and prepared the ground for D4.3.

In particular, D4.1, using a well-known international testing methodology that was slightly modified to be suited to CIPSEC purposes, identified the critical parts of each one of the pilot deployments and devised specific test scenarios that should be performed in each one of the pilots. The testing scenarios where described in detail within the document, including the following:

a) The purpose of the test and the item or items to be tested.
b) The steps and procedures that should be followed to perform the test.
c) The responsible partner that will execute the test.
d) The result of the test

The scenarios were carefully designed for all three pilots to be as generic as possible. All test scenarios were described using a common template called “Test plan and design specification”. A sample can be seen in the table below.

| 1. Test plan & design specification identifier |
| 2. Introduction |
| 3. Test items |
| 4. Features to be tested |
| 5. Features not to be tested |
| 6. Approach |
| 7. Item pass/fail criteria |
| 8. Suspension criteria and resumption requirements |
| 9. Test deliverables |
| 10. Testing tasks |
| 11. Test environment needs |
| 12. Responsibilities |
| 13. Staff and training needs |
| 14. Schedule |
| 15. Risks and contingencies |
| 16. Approvals |
In addition to the test scenarios and their description, D4.1 attempted also an initial estimation of the resources, the people and the equipment necessary to deploy the unified CIPSEC platform. This was achieved by first identifying and quantifying the effort and resources necessary for each one of the tools. After combining the data provided for each tool it was determined that a CI operator/CI Security Administrator will need a minimum of five days to fully deploy the CIPSEC Framework successfully. On top of that, personnel both from FORTH and from AEGIS must be available to deploy the respective tools.

Deliverable 4.2 on the other hands attempts to identify the gaps between D4.1 and D4.3 – the test reports. Within the D4.2, the final list of tests is included along with the final configuration of the pilots. The final test list includes not only the tests that have undergone some modifications in their original specification, but also additional tests that were not finalized or even envisaged during the writing of the D4.1. However, on better examination, they were considered relevant and important. Additionally, in D4.2, many problems that were identified by the operators in practice have been solved and improvements have been proposed to the CIPSEC Framework.

5.2 Framework Assessment Summary

The details of the validation reported below can be found in Deliverable 4.3 “Prototype Demonstration: Field trial results”, where a complete report on the tests performed at each of the test sites (Rail, Hospital, Environment) can be found, as described in D4.1 and later updated and refined in D4.2. Deliverable 4.3 is the main document which reports the tests performed to ensure the technical functionality as defined in the requirements for each test site.

The Test session definition and test session reporting were documented with the same standard (IEEE 829) used in D4.1 and D4.2 to design the test cases, in order to bring homogeneity to the thorough documentation produced regarding the procedures conducted during the test sessions.

Two options were defined for the implementation of the sessions: the first one consists in remote validations to ensure that all solutions were properly deployed at the test site whilst the second one is designed as an on-site test to ensure any specific configuration could be properly addressed by each of the solutions. In the end, DB test site only required two on-site sessions to validate the test cases while HCPB site required seven online sessions and CSI required five.

Two on-site sessions were later performed at the HCPB and CSI sites to complement the results with test cases validation in a physical, on-premise scenario.

The summary of the results obtained for the DB site follows:

<table>
<thead>
<tr>
<th>Test case specification Identifier</th>
<th>Title</th>
<th>Verification means</th>
<th>Test Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB.TPDS.01.TCS.001</td>
<td>KISA module</td>
<td>Test input integrity failures are all identified by the HSM</td>
<td>Pass *</td>
</tr>
<tr>
<td>DB.TPDS.02.TCS.001</td>
<td>Object Controller-Field Devices and Interlocking system and Signalling network</td>
<td>The heartbeat messages are properly displayed in a graphical representation way for the monitored Object Controllers. The graphical representation changes when an object controller stops sending its heartbeat and this irregularity is instantly visible in the relevant visualisation.</td>
<td>Pass</td>
</tr>
<tr>
<td>DB.TPDS.03.TCS.001</td>
<td>Control Room Workstations</td>
<td>Attack widget alerts about an attack at the corresponding frequency with a red colour and the Jammer Attack Started event gets to the XL-SIEM Syslog</td>
<td>Pass</td>
</tr>
<tr>
<td>DB.TPDS.04.TCS.001</td>
<td>Control Room Workstations - Block port scan</td>
<td>Besides blocking the port scan, the attack is reported in the GravityZone Control Center / XL-SIEM and the unified dashboard</td>
<td>Pass</td>
</tr>
<tr>
<td>DB.TPDS.04.TCS.002</td>
<td>Control Room Workstations - Block infected URL</td>
<td>Besides reporting, the HTTP request should also be blocked</td>
<td>Pass</td>
</tr>
<tr>
<td>DB.TPDS.04.TCS.003</td>
<td>Control Room Workstations - Block &amp; delete infected file</td>
<td>Besides reporting, the HTTP request should also be blocked</td>
<td>Pass</td>
</tr>
<tr>
<td>DB.TPDS.04.TCS.004</td>
<td>Control Room Workstations - Block malware</td>
<td>Besides reporting, the file should also be deleted or quarantined</td>
<td>Pass</td>
</tr>
<tr>
<td>DB.TPDS.05.TCS.001</td>
<td>Interlocking system - Availability</td>
<td>The attack is reported (either graphically or through raw text format) in the XL-SIEM / Unified dashboard.</td>
<td>Pass</td>
</tr>
<tr>
<td>DB.TPDS.05.TCS.002</td>
<td>Interlocking system - Robustness</td>
<td>The updating agent is up and running</td>
<td>Not performed</td>
</tr>
<tr>
<td>DB.TPDS.06.TCS.001</td>
<td>Updating and patching service applied to ATOS NIDS sensor</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
No update package available on PC

The agent can detect if the current version is the latest one available.

The updating agent is up and running.

The agent can update the database, get the .deb package in USB.

If the current version is obsolete, the dashboard shows it.

Corrupted file applied to ATOS NIDS sensor – USB - No update package available in USB.

The updating agent is up and running.

The agent can communicate with the database, get the .deb package in USB.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.

Not performed.
The CPU load is properly displayed in a graphical way for the monitored PC. The graphical representation changes when an attack causes the CPU load to increase and this irregularity is instantly visible in the relevant visualisation.

The existence or lack of existence of high or critical severity issues that might compromise the device’s availability or integrity of the device.

The existence or lack of existence of high or critical severity issues that might compromise the device’s availability or integrity.

The existence or lack of existence of high or critical severity issues that might compromise the device’s availability or integrity.

The existence or lack of existence of high or critical severity issues that might compromise the device’s availability or integrity.

If all the existing reports in the database, corresponding to the Pass pilot to which the user is associated, are listed, and all of them can be downloaded to a local machine, then the test is OK. If reports belonging to a pilot other than that of the user are listed, the test fails. If not all the existing reports for the pilot under question are shown, the test fails. If no report is shown, when there is one or more in the database, the test fails. If any report cannot be downloaded, the test fails.

Most of the test cases defined for the HCPB Test site have been validated, only the test cases on the Infusion pump, the Gateways and the Temperature Recorder failed due to technology restrictions on attack detection. These devices do not accept the installation of monitoring agents in their embedded software so the external monitoring of the events possible with CIPSEC framework could not detect the attacks performed due to general restrictions of the technology (different layer at the OSI network model). However, in a real production environment the network configuration, for example using layer 2 switches, would directly prevent man in the middle attacks based on ARP spoofing, which would allow to prevent most of the failed attacks.

Finally, listed below are the results for the CSI site:

### Table 12. Evaluation Results for Environmental pilot

<table>
<thead>
<tr>
<th>Test case specification Identifier</th>
<th>Title</th>
<th>Verification means</th>
<th>Test Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSI.TPDS.01.TCS.001</td>
<td>PC Station - Network attacks protection - Block port scan</td>
<td>Test will be passed once all incidents has been detected by GravityZone and shown at Atos XL-SIEM</td>
<td>Pass</td>
</tr>
<tr>
<td>CSI.TPDS.01.TCS.002</td>
<td>PC Station - Network attacks protection - Block infected URL</td>
<td>Test Pass/Fail</td>
<td>Pass</td>
</tr>
<tr>
<td>CSI.TPDS.01.TCS.003</td>
<td>PC Station - Network attacks protection - Block malicious file download</td>
<td>Test Pass/Fail</td>
<td>Pass</td>
</tr>
<tr>
<td>CSI.TPDS.02.TCS.001</td>
<td>PC Station, OC Server, OC Database</td>
<td>Test will be passed AEGIS is able to detect traffic to extract Pass CIPIs</td>
<td>Pass</td>
</tr>
<tr>
<td>CSI.TPDS.03.TCS.001</td>
<td>OC Database</td>
<td>Detection of the attack at the honeypot and detection at the XL- SIEM</td>
<td>Pass</td>
</tr>
<tr>
<td>CSI.TPDS.04.TCS.001</td>
<td>OC Server</td>
<td>Correct report created by COMSEC</td>
<td>Pass</td>
</tr>
<tr>
<td>CSI.TPDS.05.TCS.001</td>
<td>Update and patching of FVT</td>
<td>The test update is properly working on the FVT after the update process is finished.</td>
<td>Pass</td>
</tr>
</tbody>
</table>

All the test cases defined for the CSI Test site were validated.

Additionally, some test cases required some extra actions at laboratory premises for completeness of the validation of their functionalities, namely:

- KISA Module – HSM message integrity (tested in DB pilot)
- Vulnerability service (pilot-agnostic test)
- Training platform (pilot-agnostic test)
The complete solution assessment of the framework was performed during T4.4 Solution assessment and is reported in deliverable D4.4. The methodology was defined to evaluate the performance and usability of the CIPSEC framework at the pilot domains. The degree of performance of the CIPSEC framework regarding the different security features was measured, to ascertain if there is any kind of underperformance issue. A set of KPIs were defined for the security features and were measured during the course of the project.

Finally, the Total Cost ownership was developed, and the framework successfully characterised in terms of KPI for each of the security features. The methodology for the total cost of ownership adopts a pragmatic approach following a cost structure meaningful enough that permits to identify the most important expense chapters. The approach is flexible enough to cover the cases of both the cloud (hybrid) and on-premise deployment. The benefits of having CIPSEC is directly related to the importance of the assets the company wants to protect. The higher the economic value of an asset, the higher the loss in case of a cyber-attack. We leveraged a methodology used in the WISER project which is an adaptation of ISO31000 and ISO31010 practices in cyber risk assessment. This methodology relies on the identification of the list of assets to be protected and the estimation of the potential impact a successful attack may have on them. The complete methodology is explained in section 3.1 of the D4.4 document.

The Total Cost ownership analysis was performed for all three pilot partners. The finding in those three cases draws similar conclusions. In the hypothetic scenario of a major incident being prevented by CIPSEC, a sole occurrence would result in a positive return-on-investment. For minor incident scenarios, the balance would be positive after the second occurrence.

The table below serves to analyse the affordability of CIPSEC. We compare the yearly expense against the annual exploitation revenues of each pilot that are linked to the exploitation of the IT/OT networks. In this analysis, we include the initial investment in the costs of the first year and we consider that the exploitation revenues have a similar magnitude order each year.

<table>
<thead>
<tr>
<th>USE-CASE</th>
<th>ANNUAL EXPLOITATION REVENUE</th>
<th>EXPENSES Y1 (% of the exploitation revenue)</th>
<th>EXPENSES Y2 (% of the exploitation revenue)</th>
<th>EXPENSES Y3 (% of the exploitation revenue)</th>
<th>EXPENSES Y4 (% of the exploitation revenue)</th>
<th>EXPENSES Y5 (% of the exploitation revenue)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>3238200000 €</td>
<td>6580025 € (0,20%)</td>
<td>1437450 € (0,05%)</td>
<td>1437450 € (0,05%)</td>
<td>1437450 € (0,05%)</td>
<td>1437450 € (0,05%)</td>
</tr>
<tr>
<td>HCPB</td>
<td>24386000 €</td>
<td>336900 € (1,38%)</td>
<td>267875 € (1,10%)</td>
<td>267875 € (1,10%)</td>
<td>267875 € (1,10%)</td>
<td>267875 € (1,10%)</td>
</tr>
<tr>
<td>CSI</td>
<td>55373716 €</td>
<td>297785 € (0,55%)</td>
<td>240927 € (0,43%)</td>
<td>240927 € (0,43%)</td>
<td>240927 € (0,43%)</td>
<td>240927 € (0,43%)</td>
</tr>
</tbody>
</table>

This analysis indicates that the yearly expenses are very minor in comparison to the annual revenues obtained out of the exploitation of the IT/OT network. This means that the budgetary aspect would not pose a major issue to adopt the CIPSEC Framework.
6 Conclusions

This deliverable is the culmination of all the development work that has been conducted throughout the lifetime of CIPSEC project. It includes the description of the final unified Framework, the final CIPSEC prototype description, the evaluation settings and a summary of the evaluation results and the Innovations of CIPSEC.

As demonstrated throughout the document, CIPSEC consortium proposes and delivers a unified security framework that combines various heterogeneous products achieving a high-level of protection in IT and OT system of critical infrastructures. The implemented architecture is modular with many components that can cover a variety of IT/OT systems, expanding its applicability to a number of CI sectors, surpassing the initial three CI pilot sectors that were tested during the project’s lifetime.

The reference architecture of the framework was successfully implemented incorporating different collector components that feed an anomaly detection and visualization system that is able to timely and effectively alert the system administrator of the system’s security status. In addition, a number of services that support the whole framework has been introduced and incorporated in the dashboard of CIPSEC providing added value to the solution raising security awareness of the users of the framework via security training courses, contingency plans, post-event forensics, vulnerability tests and updating/patching services.

To summarize, through this deliverable the ongoing iterative design of the CIPSEC framework is concluded, all the development actions that took place within CIPSEC and how the framework and the prototype has evolved throughout the project is presented including the evaluation results and a comprehensive user guide of the unified dashboard.
7 Appendix

7.1 CIPSEC Unified Dashboard

7.1.1 CIPSEC Dashboard User Guide

The CIPSEC dashboard is a web interface that aggregates all the tools of the CIPSEC framework into one simple user-friendly interface. From this interface, it’s possible to control all the functionalities of the CIPSEC framework, have a quick view of the system status and in case of any threat, the dashboard provides a quick look at all the relevant data related to the detected threat and a deep view and analysis of the data using the various embedded tools of the framework.

- Log in page

  The first page on the dashboard is the login page, here the user must introduce its credentials in order to log in to the framework.

There are two kinds of users, admin users and simple users. Admin users have full access to the dashboard tools while simple users have selective access, their access can be customized depending on the deployment requests, from showing all the tools to showing none of them.

![CIPSEC Dashboard login](image-url)

Figure 89. CIPSEC Dashboard login
• **Dashboard menu**

Once we log in, the main page of the dashboard is displayed, this page contains an overview of the framework status, and it’s a quick look to identify any threats.

On the left, we have the Alarms, starting with a gauge that indicates the Alarm threat level followed by two historic graphics of the past Alarms threat levels, and finally a graphic showing the Alarms classified by type.

On the right, we have the Events, starting with a gauge that indicates the Event threat level followed by two historic graphics of the past Events threat levels, and finally a graphic showing the Events classified by type.

• **Upper navigation bar**

The upper navigation bar contains a drop-down menu that allows for language selection, and another drop-down menu that displays the actual user and performs the log out of the framework.

![Dashboard Overview](https://example.com/dashboard-overview.png)

**Figure 90. Dashboard overview**
• **Overview menu**

The tools in this menu are intended to be an extension to the dashboard main menu, once a threat is detected in the main menu, the admin can go into this overview menus and check the Alarms and Events, in order to detect the threat and then use the tools to view the required information according to the data displayed in these menus.

• **Alarms overview**

This sub menu contains a quick look into all the alarms triggered by the framework along with the most basic data of each alarm. If the admin wants the complete view of the alarms, it must refer to **Tools - Anomaly detection reasoner** section 2.2.1

![Figure 91. Alarms overview](https://www.example.com/alarms_overview.png)
• **Events overview**

This sub menu contains a quick look into all the events triggered by the framework along with the most basic data of each event. If the admin wants the complete view of the events it must refer to **Tools - Anomaly detection reasoner** section, 2.2.1

**Figure 92. Events overview**
- **Tools menu**
  This menu contains all the major tools of the CIPSEC framework.
  Once a threat is detected, according to the information displayed on the dashboard overview, the admin can choose which tool should be checked in order to get all the relevant information on the threat.

- **Anomaly detection reasoner**
  This sub menu contains the Anomaly detection reasoner tool, more information in section, 2.2.1, **Tools - Anomaly detection reasoner**.

![Anomaly detection reasoner in the CIPSEC Dashboard](image)
- **Honeypots**

  This sub menu contains the Honeypot tool, more information in section 2.2.6, *Honeypots*.

![Honeypots in the CIPSEC Dashboard.](image)
- **Anti-malware**

  This sub menu contains the Anti-malware tool, more information in section 2.2.4, **Gravity Zone Antimalware solution.**

  ![Figure 95. Gravity Zone antimalware in the CIPSEC Dashboard.](image-url)
- **Jamming detector**
  This sub menu contains the Jamming detector tool, more information in section 2.2.8, **DoSSensing jamming detector**.

![Figure 96. DoSSensing jamming detector in the CIPSEC Dashboard.](image-url)
- **Anonymization tool**

  This sub menu contains the Anonymization tool, more information in section 2.2.2, Data Anonymization and Privacy.

  ![Anonymization tool interface](image)

  **Figure 97. Data Anonymization and Privacy in the CIPSEC Dashboard.**
• **Services menu**

This menu contains the major services of the CIPSEC framework, the services on this menu are intended to be an extension to the tools of the framework.

• **Forensics analysis**

This sub menu contains the Forensics analysis service, more information in section 2.3.4, *Forensics Service*.

![Figure 98. Forensics Service/Analysis in the CIPSEC Dashboard.](image-url)
• Vulnerability analysis

This sub menu contains the Vulnerability report service, more information in section 2.3.2, **Vulnerability Assessment**.

From the point of view of the user management, when a user is logged in as Admin can see the last vulnerability report, the list of previous files, as well as he/she can upload a new report in the dashboard. Whereas, in the user is logged in as normal user, he/she can only see the last vulnerability report file.

![Figure 99. Vulnerability Assessment in the CIPSEC Dashboard.](image1)

![Figure 100. Managing Vulnerability Assessment reports](image2)
- **Contingency plan.**

This sub menu contains the Contingency plan service, more information in section 2.3.1, **Contingency plans.** From the point of view of the user management, when a user is logged in as Admin can see the last contingency plan, the list of previous files, as well as he/she can upload a new plan in the dashboard. Whereas, in the user is logged in as normal user, he/she can only see the last contingency plan file.

![Contingency plan in the CIPSEC Dashboard.](image)

**Figure 101.** Contingency plan in the CIPSEC Dashboard.
• **Training courses**

This sub menu contains the Training courses service, more information in section 2.3.5, **Training services**.

![Training courses in the CIPSEC Dashboard.](image-url)

**Figure 102.** Training courses in the CIPSEC Dashboard.
**Updating and patching**

This sub menu contains the Updating and patching service, more information in section 2.3.3, *Updating and Patching*.

![Figure 103. Updating and Patching service in the CIPSEC Dashboard.](image-url)
7.1.2 Technical aspects and installation

- Dashboard’s Internal Architecture

In this subsection, we detail the internal architecture of the dashboard. In a high-level view, the dashboard frontend is built with Angular 5 (https://angular.io/) and bootstrap 4 (https://getbootstrap.com/). The backend is built with Node (https://nodejs.org/en/) and Express (https://expressjs.com/).

The frontend leverages an OAuth2 (https://oauth.net/2/) client built in Angular to handle the login strategy. The client is used by the dashboard to login users with a pair username, password, thus avoiding the need for the dashboard to store users and/or passwords. The OAauth2 server is the responsible for that task and also for providing the unified single Sign-On mechanism.

The Express web server is used to serve the frontend on top of the Node server. This configuration of server’s stack has the advantage of providing a high level of scalability when having different simultaneous connections.

- Single Sign-on mechanism

The OAuth2 server will be running on a Node server and all the tools will be required to have the OAuth2 mechanism in their logins to allow external OAuth2 authentications.

Next figure depicts a diagram where the two OAuth2 phases are represented: Login phase and Access Phase.
1) In the login phase, the user provides its username and password to get access to the frontend as well as to the different tools and services. The frontend (OAuth2 client), then, forwards this information to the OAuth2 server responsible for its verification. If the information about username and password is valid the server returns a token to the user.

2) Then, the access phase is performed, allowing the user to access all frontend functionalities. When the user wants to access a tool, the frontend uses the clients token in order to authenticate the client into the tool, the tool later on must check the validity of the token against the OAuth2 server. The tool should verify the token, communicating with the OAuth2 server (in the dashboard) with its own OAuth2 client (in the tool or service). Once this is done, if the token is valid the tool can be displayed in the frontend and the user can interact with no need to provide any additional input for identification.

- **Installation**

The UPC Dashboard must be deployed in a NodeJS server, that will consist of two primary modules, the OAuth2.0 server and the Dashboard server.

The OAuth2.0 server is the authentication server for the Dashboard and all its tools and provides the single sign on mechanism. When deploying the dashboard, a users’ schematic must be provided in order to set up the user database, after that, a new user may be added at any time. Each tool in the CIPSEC framework must be configured in order to use the deployed OAuth2 server.

The Dashboard is the visualization tool for the CIPSEC framework. It joins the dashboard of all the tools into a single interface. When deploying the dashboard, a configuration file must be used in order to set the desired URL address of each tool, as well as the URL address of the OAuth2.0 server that will be used for the single sign on mechanism. When the configuration is established, the dashboard must be recompiled with the new parameters in order to be deployed.

### 7.2 Deployment recommendations

There are three main possibilities for deploying the CIPSEC Framework, namely: public cloud deployment, on-premise deployment and hybrid deployment. The next paragraphs will present some consideration to be taken into account before choosing the right approach for your infrastructure:

- **Public cloud deployment:** In this approach, just the elements of the acquisition layer are installed at the client premises, namely security sensors which gather information that is further processed in upper intelligence layers in the cloud, as presented in previous sections of this document and also in earlier
deliverables during the project. The main advantage of this approach is the potential acceleration of the deployment process, reducing time-to-availability and simplifying the setup process. It allows the chance to outsource cybersecurity tasks, allowing for a managed cybersecurity service. This is especially appealing for small critical infrastructure operators, who do not usually have specific personnel taking care of these tasks or possibility of acquiring the required HW. The main downside has to do with the need to rely on an external cloud and telecom infrastructure to host the components not deployed on-premise. This may raise some privacy/confidentiality considerations that should be addressed (for instance by anonymizing sensitive data) before proceeding with the deployment.

- **On-premise deployment:** This model is the recommended one for critical infrastructures and mandatory (no possibility to choose) for those infrastructures that are isolated from the Internet. In this model, all the CIPSEC components are deployed on-premises. This means that the client needs to have available the corresponding hardware to support all the components. The deployment and setup process are not more cumbersome than in the cloud case but pose some questions to answer. For example, for some component setup or eventual advanced configurations it might be necessary to download some files from an external server. If access to Internet is not envisioned this has to be solved somehow, the most straightforward approach is to bring these files to the premises locally stored. The updating and patching process will have to be managed locally as well. In this approach the client keeps full control of the deployment and the data collected by the system.

- **Hybrid deployment:** In this model, some components are running on premises and other on a public cloud. Same as in the fully cloud deployment, sensors deployed at the acquisition layer are running on premises. However, with hybrid model, additional components can also be running on premises. For example, agents in charge of receiving events from the sensors deployed. The main advantage is the possibility to find a trade-off in terms of costs between paying to a cloud provider and making available hardware resources on premise. Additionally, this model simplifies the interactions between on-premises components and cloud components, reducing the number of connections established between on-premises components and the cloud. For example, tools deployed on premises would report events to an agent deployed also on premises, which would report the normalized events using a single connection to the anomaly detection reasoner. In the cloud deployment, individual tools would need to establish connections to agents in the cloud, increasing the number of connections with the cloud and thus increasing the security concerns and potential risks.

In terms of recommendations, it is important to thoroughly study the needs of the specific critical infrastructure because not all the components of CIPSEC may be necessary. Concerning this, we offer tailored deployments and a customized user interface following the client choice. In this sense, it may be of help the methodology suggested in deliverables D3.1, D3.2, and D3.3. Once it is done, it may be useful to select a part of the critical infrastructure as pilot in order to get some experience in the daily interaction with CIPSEC, before extending the deployment to the whole infrastructure. Regarding the installation of the collectors, when the infrastructure is divided into several LANs it is recommended a study of the needs per LAN, considering the assets present in each one, since perhaps it is not needed to install all collectors in all LANs. The switches must be configured to allow visibility towards the agents that gather the input, adapt the format and send it for processing at the upper layers. It is also important to take into account the bandwidth consumption, as CIPSEC will incorporate new data flows. Therefore, it is recommended to analyse the consumption before deploying CIPSEC, estimate the increase derived from the CIPSEC operation within the network and determine whether (or not) the existing resources are enough to cope with the new demands.

In deliverable D4.1 the deployment process is explained providing details per tool. A relevant recommendation from this documentation is to deploy first and foremost the XL-SIEM and the NIDS, since some other tool depend on this first deployment to be available. For the rest of the tools it is not needed a specific order, and ultimately, they could be deployed in parallel.