D2.3
System and dependency analysis (third iteration) – Pilot scenario specification and self-healing strategies

Grant Agreement number:	740723
Project acronym:	CS-AWARE
Project title:	A cybersecurity situational awareness and information sharing solution for local public administrations based on advanced big data analysis

Principal author:	Thomas Schaberreiter, University of Vienna, thomas.schaberreiter@univie.ac.at
Co-author(s)	Christopher C. Wills, Laurentiu Vasiliu, Alex Papanikolaou
Internal Reviewers	Alex Papanikolaou (InnoSec), Christian Wieser (University of Oulu)

Document version:	2.0

Table of Contents

Revision History

2020-03: Change from Version 1.0 to Version 2.0 to address reviewer comments from the second CS-AWARE review	
Title Page	Added internal quality reviewers
Introduction	Added sentence to introduce Annex 6.
Annexes	Added Annex 6, CS-AWARE requirements collection

Executive Summary

This deliverable for the CS-AWARE project is the third in an iterative series of three deliverables (D2.1 System and dependency analysis (first iteration) – Cybersecurity requirements for local public administrations, D2.2 System and dependency analysis (second iteration) - Pilot scenario definition and D2.3 System and dependency analysis (third iteration) – Pilot scenario specification and self-healing strategies) that are delivered throughout the project run time. The third iteration picks up on the results of the first two iterations, with a focus on providing the final link between the analysis results and the technological part of the CS-AWARE solution. To that end, the deliverable reports on the results of the third iteration of system and dependency analysis workshops with the pilot municipalities, informing the definition of cybersecurity monitoring patterns and self-healing policies, the results of which are reported in this deliverable as well. The third round of system and dependency workshops continues the analysis of the first (assets, dependencies, monitoring points) and second round (business processes and information flows) and adds the dimension of system behaviour to the analysis results. The behaviour of system elements during day-to-day operations according to the identified business processes, and how this reflects in the data sources CS-AWARE collects, is a crucial input for the definition accurate and relevant monitoring patterns. The definition of cybersecurity monitoring patterns constitutes the internal event detection logic of the CS-AWARE technology and is a crucial aspect for providing cybersecurity awareness. The resulting patterns were validated through the consent of CS-AWARE security and data analysis experts as well as the employees of the Municipalities (users, administrators, managers) who ultimately are the ones the cybersecurity awareness is intended for. The CS-AWARE system requirements in this context have been fulfilled. Similarly, self-healing policies have been defined that allow mitigation of events detected by cybersecurity patterns in an automated way. The CS-AWARE system requirements regarding self-healing have been partially fulfilled at this point due to outstanding pilot validation. While the policies have been defined based on the consent of CS-AWARE security experts, a final validation of those policies in the context of the second and third phase of piloting in line with the CS-AWARE project plan has yet to be conducted.

1 Introduction

This CS-AWARE deliverable D2.3 “System and dependency analysis (third iteration) – Pilot scenario specification and self-healing strategies” concludes the analysis started in deliverables D2.1 and D2.2, and concludes the effort to interface the analysis results with the technology part of the CS-AWARE solution. The results of the third round of system and dependency analysis workshops are reported in Section 2, providing the last missing input for creating the cybersecurity monitoring patterns reported in Section 3 and the self-healing policies reported in Section 4. Annex 1 presents a CS-AWARE threat taxonomy derived in this context, Annex 2 and Annex 3 present generic and pilot specific cybersecurity monitoring patterns. Finally, Annex 4 and Annex 5 present generic and pilot specific self-healing policies. Annex 6 provides a summary of the CS-AWARE requirements, which are presented in detail in CS-AWARE deliverables D2.2 and D2.3.

2 System and Dependency Analysis – Third Iteration

2.1 Context of analysis

The context of the third iteration of system and dependency analysis is to generate the final missing link to interface the municipal systems and services to the CS-AWARE technology for awareness monitoring, self-healing and cybersecurity information sharing.
The first iteration of system and dependency analysis (reported in CS-AWARE deliverable D2.1) focused on identifying the assets and dependencies within the Municipality systems, and potential monitoring points that can be utilised by the CS-AWARE system for continuous monitoring (on database, service, network and security appliance level) of the systems.
The second iteration of system and dependency analysis (reported in CS-AWARE deliverable D2.2) focused on the identification of business processes within the municipality systems, and the associated information flows those processes are generating through the asset and dependency graph derived from during the first iteration of analysis. Furthermore, it was identified in which information sources (monitoring points) the interactions of those business processes are captured by the data collected on the four identified monitoring levels. This enabled the definition of the CS-AWARE pilot scenarios, focusing on the business processes of two specific services in Larissa (Human resource management - HRMS and GENESIS, a key municipal service enabling functions like financial services or document archiving), and two specific services in Rome (SUET, a building permission service and IAM, the central identification and authorisation management service). Additionally, a general set of security contexts applicable for awareness and/or self-healing, information sharing was defined. The use cases include vulnerability monitoring, suspicious behaviour monitoring, general security warnings and monitoring of potentially malicious IP and DNS entries.
In this third iteration of this analysis the goal was to define, together with the LPA users and building upon the first two iterations, normal and abnormal behaviour within the identified business processes, and how this behaviour is reflected within the data sources collected from the LPA systems on the database, service, security appliance and network level. This analysis, guided by the input and experience of the LPA users, is the basis for defining automated monitoring and detection patterns as well as dedicated self-healing strategies, discussed in Section 3 and Section 4 respectively and represents the final pieces of information needed to connect the municipality systems to the CS-AWARE technology, and provide context specific awareness and self-healing.
Table 2 of Deliverable D2.2 specifies 7 requirements (I1-I7) for the first two iterations of system and dependency analysis. For the third iteration and based on the above context for analysis, two additional requirements (I8 and I9) are specified in Table 1.

Table 1: LPA specific analysis results for third analysis iteration

#	Requirement	Functional	Non-functional	End user viewpoint
I8	Determine normal and abnormal behaviour related to critical service processes and information flows	X		X
I9	Interface analysis results with CS-AWARE technology solution for continuous monitoring, awareness and self-healing	X		

2.2 Methodology of analysis

The methodology followed for the analysis is based, like the first two analysis iterations, on the soft systems methodology (SSM) by Peter Checkland[1,2]. The first two analysis iterations closely followed Steps 1 and 2 of the methodology (Enter problem situation, Express problem situation), and Steps 3 and 4 (Formulate root definitions, Build conceptual models). This third analysis iteration will diverge

[1] Checkland, P. (1981). Systems Thinking, Systems Practice. Wiley [rev 1999 ed].
[2] Checkland, P. (1990). Soft Systems in Action. Wiley [rev 1999 ed].

more significantly from the final steps 5 to 7 of the soft systems analysis (Compare models with real-world situation, define possible changes, take action to improve situation). The goal of our analysis series is to be able to connect the municipality systems to the CS-AWARE system for continuous monitoring and awareness, whereas the end goal of the soft system methodology is to provide concrete solutions to the identified problems (e.g. strategic and operational changes) – which is not an achievable goal in the cybersecurity context due to the dynamic and constantly changing cybersecurity environment.
While the expected outcome in the CS-AWARE analysis is different to traditional SSM analysis, the methods to achieve the results are the ones used in SSM analysis:

- A workshop environment that includes a stable group of people from different relevant organisational levels (managers, technicians, suppliers, end users, …) and analysts / experts
- Free and open discussion about the environment and potential problems
- Rich pictures as a visual tool to facilitate discussion and recollection/analysis

For the workshops of this analysis iteration it was decided to work directly with the data sources (log files) identified in the previous iterations, and identify together with the workshop participants how those data sources relate to the business processes and information flows identified in previous iterations. For each log file, the analysts presented the log structure (available parameters and their meaning), followed by the identification of the relation of those parameters to the business processes and information flows. In the next step, the workshop participants were asked to define scenarios, events or behaviour in the daily operations that:

1. Are considered disruptive or malicious to the various business processes reflected in the data
2. Are something that the Municipality (represented by the roles of managers, technicians, service users) want to be made aware of
3. Are something that is not part of the current monitoring activities and/or something that cannot easily be monitored within the current set-up

The scenarios provided by the workshop participants were complemented by the discussion of the relevance (and required pilot specific adaptations) of awareness scenarios and generic patterns that were already defined by the CS-AWARE security and data analysis experts prior to the workshops. The results of this analysis provided the required input to derive the pilot specific awareness patterns detailed in Section 3, and the pilot specific self-healing policies detailed in Section 4. The pilot specific procedure for the workshops in Larissa and Rome follow the methodology outlined above and is detailed in Section 2.2.1 for Larissa and Section 2.2.2 for Rome.

2.2.1 Third iteration of system and dependency workshop in the Municipality of Larissa

The third iteration of the system and dependency analysis workshops in Larissa was held in the premises of the municipality from 7.10 – 11.10.2019. The first three days (7.10-9.10) were scheduled to complete the system and dependency analysis according to the methodology outlined above. On 10.10, a pilot and deployment focused workshop session was held, the results of which are not part of this analysis and are discussed in the pilot and evaluation focused CS-AWARE deliverable D5.1. On 11.10, the majority of the workshop participants from the municipality of Larissa was excused, and the workshop was wrapped-up to ensure all information for the later CS-AWARE project phases was captured during the workshop.
The workshop participant list, as listed in Table 3 is very similar to the participant list of the two previous workshop iterations, fulfilling the requirements of a stable analysis team. Therefore, all of the workshop participants were aware of the context and previous results of CS-AWARE system and dependency analysis. A short summary of previous analysis results was given by the CS-AWARE analysts to set the context before shifting the focus to the context of this analysis round, the analysis of data sources and behaviour. In previous workshops it was identified that there are log sources for

the database, the service, the network and the security appliance level available, as described in Table 2.

On day 1 and day 2 the focus of analysis was on the service level and database level, with the CS-AWARE analysts presenting the log structure and available parameters, and determining the meaning of those parameters together with the relevant database and service specialists from Larissa. While for example the audit trail logged by the database follows standardised structure of the database appliance, the service level audit logs are specific to the application and thus less standardised. The input of the local specialists is crucial to understand and capture the full meaning of those log files. Once the group was satisfied that a satisfactory level of understanding of those log files was reached and their relation to the previously identified business processes was understood, the participants were asked to define and present scenarios in the context of the business processes that would be considered suspicious or malicious, and how this would specifically be reflected in the data contained in the previously discussed log files. The scenarios focused on different aspects of data theft and data manipulation for several real-world scenarios (both scenarios that have previously happened and those haven't happened yet but are realistic). Those scenarios could directly be translated into the technological patterns that will be discussed in Section 3.

Day 3 focused on discussions of logs related to the security appliance and network level. Contrary to the more data focused database and service level, with the main context of monitoring for suspicious or malicious behaviour in the data, the main focus on the network and security appliance level is to determine a uniquely identifiable element within the logs that allows to map the event to additional context information from external sources like threat intelligence or social media. For network logs and firewall logs, these unique identifiers are IP addresses and DNS entries. For antivirus appliances, such identifiers would be malware/virus names and descriptions and for IDS/IPS appliances, such identifiers would be event signatures or names/descriptions of the event. The workshop participants agreed on this assessment, and the logs were inspected for such identifiers. On the network and firewall level, the traffic logs contain IP and/or DNS entries for each logged event and the analysis could be concluded quickly. For the antivirus and IPS logs two main observations could be made:

1. In the logs for antivirus and IPS, no single malicious event has been logged since the appliance was activated
2. The unique identifiers for malicious software and network intrusions are specific to the appliance vendor, and potentially proprietary to the vendor. It may be hard to associate additional information from threat intelligence, since no commonly accepted identification for malware or network intrusions exists (in contrast to the widely accepted vulnerability enumeration).

Having in mind the potential difficulties to find unique identifiers in antivirus and IPS logs, the CS-AWARE analysts presented and discussed a set of generic patterns on the network and security appliance level that intend to raise additional awareness in the municipalities based on such events. The participants agreed with the presented patterns, and provided input on how relevant specific patterns are which helped to understand the focus of pattern implementation for CS-AWARE on those levels. It was concluded that on this level the generic patterns presented provide a sufficient level of coverage, and no additional context specific patterns needed to be defined. The resulting patterns on the network and security appliance level are presented in Section 3.

Table 2: Log sources from Larissa systems and services

Logical Level	File name	Description
Database	HRMSGenesis_Database_Authentication_Log.csv	An Audit log derived from the Oracle database that logs session authentication for both the HRMS and the Genesis service. Logging includes the user

		identification, action (login, logoff) and time of the event. CS-AWARE intends to utilise this log to derive behaviour monitoring patterns based on the activities and scenarios defined in the analysis workshop.
Service	HRMSGenesis_Application_Audit_Log.csv	An application audit log that logs all the database operations related to Genesis and HRMS applications. Logging includes the specific operation, the specific database table the change relates to and the origin of the change and the time of the event. CS-AWARE intends to utilise this log to derive behaviour monitoring patterns based on the activities and scenarios defined in the analysis workshop.
	InstalledPackages_MainServer.txt	An automatically collected list of all software and packages installed on operating system level, utilised by the vulnerability use case and general security warnings use case. The context of this log file has already been defined and the log has not been discussed in the context of this workshop.
Network	Cisco_Network_Audit_Log.csv	A firewall log from a hardware firewall appliance. Since the current appliance is in the process of being replaced with a new one from a different manufacturer, the concrete log could not be discussed in detail. However, as a standard network traffic log file is expected, with the main relevant identifier being the IP addresses and DNS entries. The log is intended to be used in the context of IP/DNS monitoring use case described in Deliverable D2.2.
Security Appliance	Symantec_Security_System_Log.csv	Log collected from Larissa main server, containing the general application logging of Symantec Endpoint Protection. It was concluded that this log is not relevant for current CS-AWARE monitoring.
	Symantec_Security_Traffic_Log.csv	Log collected from Larissa main server, containing the results of the Firewall component of Symantec Endpoint Protection. The log contains fairly standard Firewall log entries, with IPs being the main relevant identifiers. The log is intended to be used in the context of IP/DNS monitoring use case described in Deliverable D2.2.
	Symantec_Security_Antivirus_Log.csv	Log collected from Larissa main server, containing the results of the Antivirus component of Symantec Endpoint Protection. Events include scan start/ scan stop entries, as well as alerts if malicious elements

		were detected. In the current logs, no single malicious event was logged so far, due to the strict security policies that isolate the Larissa server environment. In general, CS-AWARE is looking for malicious software/virus identifiers that could be used to identify from external information sources and enrich the awareness events in the CS-AWARE console with this additional information.
	Symantec_Security_IPS_Log.csv	Log collected from Larissa main server, containing the results of the intrusion prevention system (IPS) component of Symantec Endpoint Protection. Events include scan start/ scan stop entries as well as alerts if malicious elements were detected. In the current logs, no single malicious event was logged so far, due to the strict security policies that isolate the Larissa server environment. In general, CS-AWARE is looking for unique identifiers that can be used to identify and associate concrete detected intrusions to more context information from external information sources.

Table 3: Participants of analysis workshop in Larissa

Participant	Organisation	Role
Georgia Kolovou	Municipality of Larissa	HRMS-GENESIS administrator
Heleni Drakou	Municipality of Larissa	System administrator
Thanasis Poultsidis	Municipality of Larissa	System administrator
Christos Topalidis	Municipality of Larissa	IT Dpt Supervisor
Christina Mitroula	Municipality of Larissa	Syetm user
Nikolas Makrigiannis	Municipality of Larissa	System user
Aristotelis Kostoulas	Municipality of Larissa	Manager
Chris Wills	Caris Research	Moderator/ Analyst
Thomas Schaberreiter	University of Vienna	Moderator/ Analyst
Christian Wieser	University of Oulu	Technical Expertise/Security Expertise
Kim Gammelgaard	RheaSoft	Analyst – User Interface
Laurentiu Vasiliu	Peracton	Analyst - Data analysis
Stefania Tola	3rdPlace	Analyst – Data collection and analysis
Nikos Tsiridis	OTS	Technical expertise – Data collection
Jerry Andriessen	Wise&Munro	Moderator/ Analyst

2.2.2 Third iteration of system and dependency workshop in the Municipality of Rome

The third iteration of the system and dependency analysis workshops in Rome was held in the premises of the municipality from 21.10 – 25.10.2019. The first three days (21.10-23.10) were scheduled to complete the system and dependency analysis according to the methodology outlined above. On 24.10, a pilot and deployment focused workshop session was held, the results of which are not part of this analysis and are discussed in the pilot and evaluation focused CS-AWARE deliverable 5.1. On 25.10, the workshop participants from the municipality of Rome were excused, and the

workshop was wrapped-up by the CS-AWARE analysts to ensure all information for the later CS-AWARE project phases was captured during the workshop.

The workshop participant list, as listed in Table 5 is very similar to the participant list of the two previous workshop iterations, fulfilling the requirements of a stable analysis team. Therefore, all workshop participants were aware of the context and previous results of CS-AWARE system and dependency analysis. A short summary of previous analysis results was given by the CS-AWARE analysts to set the context before shifting the focus to the context of this analysis round, the analysis of data sources and behaviour. In previous workshops it was identified that there are log sources for the database, the service, the network and the security appliance level available, as described in Table 4.

The approach and time line followed very closely what was experienced in the Larissa workshop. On day 1 and day 2 the focus of analysis was on the service level and database level, with the CS-AWARE analysts presenting the log structure and available parameters, and determining the meaning of those parameters together with database and service specialists from the municipality of Rome and the relevant service suppliers for the two services under investigation, IAM and SUET. As was the case in Larissa, it was observed that the audit trail logged by the database follows the standardised structure of the database appliance, and that the service level audit logs are specific to the application and thus less standardised. The input of the local specialists is crucial to understand and capture the full meaning of those log files. Once the group was satisfied that a satisfactory level of understanding of those log files was reached and their relation to the previously identified business processes was understood, the participants were asked to define and present scenarios in the context of the business processes that would be considered suspicious or malicious, and how this would be reflected in the data contained in the previously discussed log files. Similar than in Larissa, most of the scenarios focused on different aspects of data theft and data manipulation for several real-world scenarios (both scenarios that have previously happened and those that haven't happened yet but are realistic). Additionally, due to the fact that Rome offers on-line services to its citizens (which is not the case in Larissa), scenarios relating to denial-of-service, and how this behaviour would reflect in the log files, have been defined. Those scenarios could directly be translated into the technological patterns that will be discussed in Section 3.

Day 3 focused on discussions of logs related to the security appliance and network level. Similar to Larissa, it was presented to the participants that contrary to the more data focused database and service level, with the main context of monitoring for suspicious or malicious behaviour in the data, the main focus on the network and security appliance level is to determine a uniquely identifiable element within the logs that allows to map the event to additional context information from external sources like threat intelligence or social media. For network logs and firewall logs, these unique identifiers are IP addresses and DNS entries. For antivirus appliances, such identifiers would be malware/virus names and descriptions, and for IDS/IPS appliances, such identifiers would be event signatures or names/descriptions of the event. In Rome, logs of two different network firewall appliances, managed by two different departments/suppliers, are collected. One firewall is located at the perimeter of the network, and includes logs of IP/DNS facing the Internet. The second Firewall is at the heart of datacentre network traffic, and only traffic from the internal network (no public IP/DNS) is to be expected. Different scenarios and the relation between the two firewall log sources in the context of business processes creating information flows through both appliances were discussed. The generic awareness patterns on the network and security appliance level were discussed with both suppliers, and it was agreed that those patterns cover the awareness requirements for this pilot scenario, and no pilot and context specific patterns need to be applied. It was also discussed that the high-end firewall appliances used in Rome already provide some of the proposed awareness patterns, but a central place like the CS-AWARE interface providing relevant notifications from different appliances in one place and in a uniform format is beneficial.

In Rome, no antivirus appliance is available in the context of the observed services. An IPS appliance is available at the heart of the Rome datacentre network traffic. However, since most potentially malicious traffic is already filtered at the perimeter, very few events are expected to occur in the

relevant log file. The discussion of the IPS log lead to similar conclusions as with the antivirus and IPS logs of Larissa:

1. In the logs of the IPS, only a handful of malicious events has been logged since the appliance was activated.
2. The unique identifiers for malicious software and network intrusions are specific to the appliance vendor, and potentially proprietary to the vendor. It may be hard to associate additional information from threat intelligence, since no commonly accepted identification for malware or network intrusions exists (in contrast to the widely accepted vulnerability enumeration).

Like on the firewall level, the discussion of generic patterns led to the conclusion that all relevant awareness patterns are covered and no pilot specific patterns need to be defined on this level. The resulting patterns on the network and security appliance level are presented in Section 3.

Table 4: Log sources from Rome systems and services

Logical Level	File name	Description
Database	SUET_Database_Audit_Log	An audit log of the database operations by the SUET application, as logged by the Oracle database. The log includes entries for each database operation, including origin (administrative action or SUET application), database table associated to the operation, and time. CS-AWARE intends to utilise this log to derive behaviour monitoring patterns based on the activities and scenarios defined in the analysis workshop.
Service	SUET_Application_Audit_Log	An audit log of database operations, as logged by the SUET application server. In addition to the database log, this log includes the username of the originator of the change. CS-AWARE intends to utilise this log to derive behaviour monitoring patterns based on the activities and scenarios defined in the analysis workshop. The database level audit log and the service level audit log can be utilised to monitor for discrepancies in the data operations.
	IAM_AccessManager_Audit_Log	An audit log for access management relating to the SUET application. Logs user session login/logout including time, as well as access rights and privileges. CS-AWARE intends to utilise this log to derive behaviour monitoring patterns based on the activities and scenarios defined in the analysis workshop.
Network	WAF_Log	A web application firewall log that sits at the perimeter outside the Rome data centre. Logs are filtered to reflect the traffic towards the SUET application, both blocked traffic and regular connections. The logged IP addresses are the main identifier relevant for CS-AWARE.

		The log is intended to be used in the context of IP/DNS monitoring use case described in Deliverable D2.2.
	Internal_Firewall_Log	The internal firewall is a firewall appliance hosted inside the Rome firewall. Since the traffic is translated by a reverse proxy at the perimeter, the traffic expected to pass the internal firewall is solely traffic containing LAN IPs. The logged IP addresses are the main identifier relevant for CS-AWARE. The intended use case for this type of log is to better understand the traffic towards the SUET application and to monitor for potentially unusual patterns.
Security Appliance	Internal_Firewall_IPS_Log	A log of the intrusion prevention component of the internal firewall. Since the traffic that crosses the internal firewall is LAN traffic and it is likely that network intrusions have already been blocked at the perimeter, very few events are expected to be observed in this log. In general, CS-AWARE is looking for unique identifiers that can be used to identify and associate concrete detected intrusions to more context information from external information sources.

Table 5: Participants of analysis workshop in Rome

Participant	Organisation	Role
Arianna Bertollini	Roma Capitale	Project expertise
Omar Parente	Roma Capitale	Project expertise
Massimo Ferrarelli	Roma Capitale	Project expertise
Claudio Guido Ferilli	Roma Capitale	Project expertise
Roberto Massimiliani	Roma Capitale	Data Center
Aniello Marotta	Roma Capitale	Privacy & Data Protection
Andrea Quatrini	Roma Capitale	SUET
Ivano Ottaviani	Roma Capitale	Fleet Mgmt
Walter Duca	Roma Capitale	IAM
Stefania Cogodda	Roma Capitale	Privacy & Data Protection
Valentina Modesti	Roma Capitale	Internal service user
Stefano Vallocchia	Roma Capitale	Internal service user
Annalisa Mannucci	Roma Capitale	SUET
Luca Iezzi	Roma Capitale	Data Center
Ivan Bernabucci	Roma Capitale	Online Services
Cristina Pischedda	Roma Capitale	Administrative support
Marco Liverani	Leonardo NSR	IAM
Raffaele Conforte	Fastweb	Network Supplier
Marco Benucci	Leonardo NSR	IAM
Antonio La Malfa	Accenture	SUET
Valerio Voci	Accenture	SUET
Angelina Marchio	RTI Data Center	SUET DBA
Fabio Nohaman	Fastweb	Network Supplier

Alessandro Ponzo	Accenture	SUET
Chiara Patrizi	Accenture	SUET
Thomas Schaberreiter	University of Vienna	Moderator/ Analyst
Christian Wieser	University of Oulu	Technical Expertise/Security Expertise
Kim Gammelgaard	RheaSoft	Analyst – User Interface
Stefania Tola	3rdPlace	Analyst – Data collection and analysis
Jerry Andriessen	Wise&Munro	Moderator/ Analyst
John Forrester	CeSViTer Consulting	Analyst
Manuel Leiva	CeSViTer Consulting	Analyst
Massimo Della Valentina	CeSViTer Consulting	Analyst

2.3 Discussion of results

A major outcome of the third round of system and dependency analysis workshops was the validation that the required information can be provided by the participants. The third round of CS-AWARE system and dependency analysis required a more technical understanding of the system than the two previous rounds, since it was required to identify how suspicious and abnormal behaviour reflects in the very technical data sources (log files), that the CS-AWARE system collects. Before the workshop the analysts were concerned that the workshop participants might not be able to associate scenarios and behaviour of their daily operations with the technical representation of this behaviour in the logs. The workshops in both Larissa and Rome, however, have shown that this is not the case. Since the general structure of the log files was introduced by the CS-AWARE analysts, and the meaning of the parameters was evaluated together with workshop participants, a deep common understanding of the logs could be achieved quickly, which made the association of business process behaviour as well as normal/abnormal behaviour of those processes to the data reflected in the log files a trivial task.

In both Larissa and Rome the workshop team was able to analyse log files on the database, the service, the network and the security appliance level. As expected, the analysis of the behaviour patterns at database and service level were the most fruitful. As determined in the initial risk analysis of CS-AWARE deliverable D2.1, the data managed by the municipalities are the most valuable asset of local public administrations, and suspicious or malicious operations on the data can be monitored best by observing the databases the data is located in as well as the services that manipulate the data. In the workshops a deep understanding of the data operations generated by the various business processes on a daily bases could be achieved, and reflects the observations of technicians working on the systems, employees operating the systems as well as administrators/ security personal observing the systems. To the best of our knowledge, no security appliance currently on the market can provide this level of customised and use case specific monitoring of data that (1) will help the municipalities to receive alerts and awareness of events that are currently not monitored at all, (2) achieve an awareness system that monitors only behaviour that the municipalities actually care about, and (3) be able to narrow the monitoring parameters to the specific requirements of the municipalities to reduce or eliminate false positives.

On the network and security appliance level the results show that the main goal is not to identify suspicious behaviour, since security appliances like firewalls, IDS/IPS systems and antivirus systems already have those detection capabilities. The main goal on those levels is to provide additional awareness (from e.g. threat intelligence or social media) what those events mean in the current security context, how they relate to their systems, and how to best address those events. The main approach for CS-AWARE is to derive a unique identifier for each event and associate relevant information from external information sources. The workshop results show that with little effort it was possible to identify unique identifiers from network logs and firewall logs, since each event contains IP and/or DNS entries of the event. For IDS/IPS and antivirus appliances the identifiers of the logged events highly depend on the manufacturer/developer of the appliance. It was observed in

both Larissa and Rome that, if a unique identifier is logged by the appliance this is usually not a public identifier but an internal/proprietary identifier of the manufacturer of the appliance. If additional information form external information sources should be provided for those logs it was concluded that, if available, other identifiers like event description could be used to try and find matches in external information sources.

In general, the workshop results show that through **the consent** achieved between the analysts and workshop participants, as well as by including the **(tacit) knowledge** of the participants from the municipalities who work with the systems on a daily basis, the CS-AWARE team is confident that the analysis goals could be reached and that the requirements I8 and I9 set out in Table 1 are fulfilled.

3 Pattern definitions for pilot scenarios

3.1 Context of cybersecurity pattern creation

At a fundamental level, the CS-AWARE cybersecurity patterns are required in order to 'instruct' the decision engine what to look for and what to retrieve from the existing data. The patterns constitute the internal logic of the data analysis engine required for detecting suspicious or malicious events in data, and subsequently associate them with relevant context information from other sources. Theoretically the number and type of possible patterns is limitless, while in practice there is always a finite set of patterns within a certain time frame.
The definition of cybersecurity patterns is the translation of environmental factors as observed by the various information sources of CS-AWARE from outside the organisational context (both static and dynamic external information sources as defined in CS-AWARE deliverable D2.1) and inside the organisational context (the analysis results achieved through workshops in the context structural, business process and dynamic behaviour as detailed in CS-AWARE deliverables D2.1, D2.2 and Section 2 of this document).
For CS-AWARE we have identified two main classes of pattern types relevant for observing the information sources on the database, the service, the security appliance and the network levels of the LPAs:

1. **Behavioural monitoring patterns:** This refers to patterns that observe unusual and/or malicious behaviour in LPA data. This monitoring is closely related to the business processes and associated information flows through the systems. In the context of the CS-AWARE pilot scenarios, behavioural monitoring patterns turned out to be most relevant on the database and service level.
2. **Identifier based monitoring patterns:** This refers to the identification of a unique identifier (like IP addresses, DNS entries, software names/versions, malware identifiers, network intrusion signatures) that can be used to associate relevant contextual information from other information sources to the event for increased awareness. This type of monitoring pattern turned out to be most relevant on both the network and the security appliance level, and for software vulnerability monitoring, which spans across multiple system levels.

This Section presents the main CS-AWARE use cases: vulnerability, suspicious behaviour, general security warnings and potential malicious IP/DNS entries. This was done in the context of the following organisational monitoring levels: database, service, network, security appliance. It should be noted that due to the dynamic nature of cybersecurity, the defined patterns are not to be seen as a finite set, and the CS-AWARE system allows to flexibly adjust patterns or create new patterns if new evidence or user defined scenarios arise.
The creation of relevant monitoring patterns is a key aspect in fulfilling the CS-AWARE system requirements **S1 – Provide cybersecurity awareness** and **S6 – Enable data analysis by setting external and internal data into context**.

3.2 Methodology and process of cybersecurity pattern creation

The Methodology that was followed in pattern creation includes four main steps: (1) the analysis of relevant literature, (2) analysis of common threat taxonomies and derivation of a CS-AWARE threat taxonomy, (3) definition of a generic set of threat patterns on the database, the service, the security appliance and the network level and (4) definition of pilot and use case specific patterns based on scenario and behaviour definition of the third system and dependency workshop in Larissa and Rome.

Step 1: The analysis of literature was the first step in defining the scope for monitoring pattern creation in the context of local public administrations. The detailed results of this analysis are presented in CS-AWARE deliverables D2.1 and D2.2. The most relevant literature that helped to define the scope were those sources that provide regular reports of the threat landscape, as for example the ENISA Threat Landscape Report[3] or the Europol IOCTA[4] report. The initial threat assessment reported in D2.1, Section 2 and confirmed in an update to the assessment in D2.2, Section 3 has shown that the main asset of an LPA is the data that is managed by the LPA and that monitoring in this context will need to focus on the data and the data flows through the LPA systems that day to day operations (business processes) are generating. It was concluded that a classification of common threats to scope the threat landscape to the requirements of CS-AWARE awareness monitoring was required, as defined in Step (2) of this methodology.

Step 2: A major aspect of cybersecurity awareness is the ability to classify an event to a specific attack and/or threat in order to be able to give context information to the administrator/ manager in an organisation that has to deal with this threat. There are several threat taxonomies available that intend to group and categorise different threats in order to provide a comprehensive overview of what risks an organisation can be exposed to (though, most threat taxonomies do not raise the claim to provide an exhaustive list of threats against organizations). An excellent introduction to the most common threat taxonomies, as well as a study of their practical applicability in the organizational context can be found in the 2018 SANS technology report[5] to evaluate the comprehensiveness of IT threat taxonomies. In the report, the four taxonomies that were analysed are the Open Threat Taxonomy[6], the ENISA threat taxonomy[7], the NIST Risk Assessment Threat Exemplary[8] as part of the NIST guide on risk assessment and the Taxonomy of Operational Cyber Security Risks[9]. Those taxonomies also have been the basis for evaluation in the context of the CS-AWARE considerations, with the Open Threat Taxonomy and the ENISA threat taxonomy being the most applicable to our context. In general, all taxonomies try to model the same threat landscape, with slight difference in scope and grouping, and differences in how concrete threats (subgroups) are modelled. For CS-AWARE, with the context of cybersecurity awareness, many of the organizational threat categories listed in above taxonomies (like physical threats, legal threats, …) are not applicable to this context.
Based on the analysis results of external information and the organizational understanding gained in the system and dependency workshops (both reported in D2.1 and D2.2), a CS-AWARE taxonomy was derived from above listed taxonomies that focuses on the threats deemed relevant in the context of cybersecurity monitoring. The resulting taxonomy can be found in Annex 1 of this document. Aside from being a guideline of how detected events in CS-AWARE can be categorized and classified, it was observed that in today's CTI sharing communities, the information about events or attacks that are shared often do not contain general descriptions of the materialized threat associated

[3] https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape
[4] https://www.europol.europa.eu/iocta-report
[5] https://www.sans.org/reading-room/whitepapers/threatintelligence/evaluation-comprehensive-taxonomies-information-technology-threats-38360
[6] https://www.auditscripts.com/resources/open_threat_taxonomy_v1.1a.pdf
[7] https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/threat-taxonomy/view
[8] https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf
[9] https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=91013

to this event. In the context of awareness monitoring, this is however a crucial aspect to define the context of each event and provide high-level human readable descriptions of this context. In this sense, the CS-AWARE threat taxonomy may be extended in future with high-level descriptions of each threat, to be added to the more technical descriptions of events and attacks derived from LPA log sources and threat intelligence sources.

(3) As a next step, and based on the analysis results of the threat landscape described above, the analysis of information sources and the results of the first two rounds of system and dependency workshops (both described in D2.1 and D2.2), a set of generic monitoring patterns was defined driven by the security and data analysis experts in CS-AWARE. For this purpose, multiple dedicated telco meetings were held to discuss those patterns and reach a consent on their feasibility (availability of data, meaningfulness from grouping perspective) and practical relevance in the organizational security context. The resulting patterns can be seen in Annex 2 of this document. The patterns are grouped by the general use cases of CS-AWARE (as defined in D2.2):

- The suspicious behaviour monitoring use case (including the IP/DNS analysis use case, since the two use cases are conceptually the same)
- The vulnerability use case
- The general security warnings use case

The context of the vulnerability use case (match known vulnerabilities with components used within the LPA systems) and the context of the general security warnings use case (match context relevant information for LPA system components from social media or threat intelligence based on user defined keywords) is straightforward and does not require complex patterns to achieve the results. However, for the suspicious behaviour and IP/DNS monitoring use case, the situation is more complex. It is necessary to define observable parameters within the information sources that would indicate a certain threat or attack, and estimate the boundaries within those parameters that differentiate normal from abnormal behaviour. In this work the focus was on identifying the parameters relevant for identifying a specific behaviour that could indicate a threat. The concrete instantiations of those parameters (boundaries between normal and abnormal behaviour) are context specific and need to be defined on a case-to-case basis. The resulting generic patterns for the suspicious behaviour and IP/DNS monitoring use case in Annex 1 are grouped by the monitored system level (database, service, network, security appliance) on which the monitoring data originates.

Step 4: The final step in the CS-AWARE pattern definition methodology is to derive (from the generic patterns) the pilot specific monitoring patterns (Larissa and Rome). The pilot specific patterns were derived in the context of the third round of system and dependency analysis, and the methodology as well as the analysis results are described in Section 2 of this document. The definition of patterns together with the end users is a crucial step in contextualising the generic work on threat landscape analysis and generic patterns with the actual experiences and requirements of the system users. The resulting patterns for both Rome and Larissa can be found in Annex 3 of this document. In the context of this analysis both refinements and concrete instantiations of previously defined generic patterns, as well as new patterns based on concrete scenarios within the LPAs could be defined.

3.3 Discussion of results

The definition of relevant and accurate monitoring patterns for cybersecurity is a non-trivial task that requires a deep understanding of the threat landscape, as well as the organisational and technical systems setup of the systems being monitored in order to provide results that are relevant to the users/administrators of the systems and producing accurate results to minimise false positive alerts. Currently available products like SIEM (security and event management) systems often monitor for generic behaviour that does not take organisational or system specifics into account. This reflects in an extended learning phase that requires the SIEM user to confirm or deny the validity of alerts to filter out false positives, which leads to user frustration and doubts in the usefulness of an otherwise

excellent security tool. One of the core ideas of CS-AWARE is that the results of the system and dependency analysis will provide a deep understanding of the organisational and technical cybersecurity awareness monitoring requirements, enabling the definition of a relevant and realistic set of monitoring patterns. In general, the results of the pattern definition process confirm this idea and the methodology that was followed has shown that it is possible to derive a realistic set of monitoring patterns within a reasonable time frame, that are tailored to the specific monitoring needs of the end users.

The definition of the threat landscape relevant for the CS-AWARE context (steps (1) and (2) of the methodology) has been a straight-forward process. The existing work on assessing the threat landscape (e.g. ENISA threat landscape report, Europol IOCTA report) and threat taxonomies (e.g. open threat taxonomy, ENISA threat taxonomy) provided an excellent starting point for creating a threat landscape/ threat taxonomy relevant for the CS-AWARE context. One point that remains open with the CS-AWARE threat taxonomy, is the requirement for utilising this taxonomy (including high level descriptions of the threats that need to be provided by CS-AWARE) as an information source to provide high-level context and awareness for detected events, since this type of information has not been identified to be provided by any threat intelligence source. At this point it remains unclear if the end users require a high-level context, or if the more technical information provided by analysis and threat intelligence sources is sufficient. This aspect is part of the validations to be performed in the second phase of piloting.

The derivation of generic monitoring patterns by CS-AWARE security and data analysis experts (step (3)) based on the definition of the relevant threat landscape and the analysis of the organisational and system requirements of the municipalities (as described in D2.1 and D2.2) turned out to be a non-trivial task even for experienced security researchers, practitioners and data analysts. While the security and threat landscape is very well understood, the lack of specific context information about how those threats can materialise in the specific user context required a substantial effort and several iterations to define a realistic set of monitoring patterns and achieve consent within the group. The definition of pilot specific patterns (step 3), together with the relevant system users on the other hand was a very smooth process that produced excellent results and consent within a short time frame.

A currently on-going effort that requires the user input from the second phase of piloting is the refinement of monitoring patterns, which includes the refinement of the understanding of how awareness data from the various information sources collected by CS-AWARE is to be included in order to help the end users to be more aware. The end user feedback on the validations planned for the second phase of piloting will allow to create a better understanding of the user requirements in this area. The observation resulting from this exercise is that the top-down approach that derives monitoring patterns from a generic threat landscape is comparatively harder to achieve than the bottom-up approach that derives monitoring patterns based on real-world behaviour and scenarios – which shows the strength of the CS-AWARE system and dependency analysis approach, providing a deep common understanding of the environment and its behaviour. At the same time, it was observed that both the top-down and the bottom-up approach are required as the results are complementing and informing each other.

The validity of the monitoring patterns is confirmed by the consent of the municipality users, the CS-AWARE security experts and the CS-AWARE data analysis experts, fulfilling and validating the system requirement **S6 – Enable data analysis by setting external and internal data into context**. Furthermore, the monitoring patterns are a large part of fulfilling **S1 – Provide cybersecurity awareness**, which is as well partially validated through the consent within this group. It should be noted that the patterns described in Annex 2 and Annex 3 of this document are to be seen as an initial basis for monitoring the defined CS-AWARE pilot scenarios and use cases. Due to the changing needs in the organisational context as well as the changing cybersecurity landscape, monitoring patterns are expected to change or new patterns will be required over time. The CS-AWARE framework was designed with this in mind and allows for flexible adaptation and amendment of monitoring patterns.

4 Self-healing policies for pilot scenarios

4.1 Context of self-healing policies

The definition of self-healing policies represents the link between detected events and automatable mitigation measures to counter those events. For each monitored cybersecurity event, one or more ways to mitigate the event by conducting automated configuration or system set-up modifications may be applicable. The goal in CS-AWARE is to create an initial database of generic self-healing policies to be applied to mitigate a variety or threats. Furthermore, concrete instantiations of self-healing policies will be derived relevant for automated application to the LPA appliances within in the LPA pilot scenarios. However, due to the invasive nature of self-healing to the production environment of the LPAs, it was decided not to interface and apply self-healing directly to the LPA system appliances, but set up a test environment instead that allows to observe and validate the self-healing behaviour without directly affecting LPA system operation. At this stage of reporting, the focus is on providing a list of generic self-healing policies that are able to mitigate threats and concrete attacks associated to the CS-AWARE threat taxonomy and generic/ pilot specific patterns defined in Section 3. Testing of mitigation policies in the CS-AWARE production environments in Larissa and Rome have not yet been conducted at the time of writing of this deliverable, and will be part of the second and third phase of pilot testing.
The creation and application of self-healing policies is a key element in fulfilling the CS-AWARE system requirement **S3 - Enable system self-healing**.

4.2 Methodology and process of self-healing policies

The methodology followed to define self-healing policies includes four steps: (1) Analysis of relevant background and literature, (2) Definition of relevant attack/scenario groups, (3) Derivation of general and pilot specific self-healing policies, (4) Interfacing with relevant appliance for automated application.

Step 1: This step is concerned with an analysis of the relevant environment for self-healing, and includes the analysis of the current cybersecurity environment as well as the analysis of the specific organisational context, as reported in CS-AWARE deliverable D2.1 and D2.2. Resulting from this, the basis for the definition of self-healing scenarios and policies are the CS-AWARE threat taxonomy and generic/ pilot specific cybersecurity monitoring patterns reported in Section 3 of this document.

Step 2: Based on the general threat landscape and the identified organisational requirements, potential attack groups based on a generic understanding or concrete scenarios given by the organisational context. The identified attack/scenario groups are based on well-known information sources in this context, like the Common Attack Pattern Enumeration and Classification (CAPEC)[10] framework as well as the ATT&CK[11] framework. The attack/scenario groups were defined, wherever possible, based on the labelling vocabulary defined by the STIX 2.0 documentation[12].

Step 3: In the next step, relevant mitigation policies are associated to the identified attack or scenario types. Wherever possible, those mitigations are based on the CAPEC framework and the ATT&CK framework. In addition to existing mitigations reported by the cybersecurity community, mitigations are defined by CS-AWARE cybersecurity experts based on the deep understanding of the organisational context of the pilot scenarios as reported in D2.1 and D2.2.

Step 4: The final step of the methodology depicts the concrete instantiation and technical implementation of the self-healing policies interfacing with the relevant appliances within the LPA

[10] https://capec.mitre.org/
[11] https://attack.mitre.org/
[12] https://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part1-stix-core.pdf

systems. This includes the validation of the selected self-healing policies by the LPA personnel and interfacing on the technical level. For CS-AWARE piloting it was decided that instead of interfacing with the production appliances within the LPA system setup, a test environment for validating application of self-healing policies will be conducted. This step is planned for the second and third phase of piloting.

Annex 4 and Annex 5 list the resulting self-healing policies. The list distinguishes between policies that are applicable for self-healing (automated application) as-is, and those policies that are intended as mitigation policies that assist system administrators to apply mitigations manually or semi-automated. Depending on the context and concrete use case, even those mitigation rules may be applicable for fully automated self-healing. It should be noted that the presented list is not to be seen as a final set of mitigation policies. Due to the dynamic nature of cybersecurity as well as the organisational context, self-healing policies are expected to be adapted and amended regularly. The CS-AWARE solution was designed with this in mind.

4.3 Discussion of results

The results of the definition of self-healing policies within the CS-AWARE context are promising. Relying on the deep understanding of the systems and dependencies within an organisation and the current cybersecurity environment that are the basis of the CS-AWARE solution, a realistic set of mitigation and self-healing policies can be derived that are both relevant in the context of the cybersecurity monitoring patterns as well as in the context of the threat taxonomy derived by CS-AWARE. This understanding is based on the consent by the security experts within the CS-AWARE project. A final validation of validity of the approach is still outstanding, since the practical validation of the self-healing scenarios that includes LPA user input and feedback is planned for the second and third iteration of CS-AWARE piloting which is yet to be conducted.
At this point, the CS-AWARE system requirement **S3 - Enable system self-healing** is seen to be partially fulfilled, since consent by CS-AWARE security experts suggests the validity of the approach, but input and validation by LPA personnel in the context of the second and third phase of piloting is still outstanding.

Annex 1 – CS-AWARE threat taxonomy

B1	**Network threats**
c1	Network Reconnaissance
c2	Network traffic manipulation
c3	Information gathering
B2	**Manipulation of information**
c4	Maintaining System Persistence
c5	Memory Manipulation
c6	Repudiation of actions
c7	Address Space hijacking (IP prefixes)
c8	Routing table manipulationAddress Space hijacking (IP prefixes)
c9	Routing table manipulation
c10	DNS poisoning / DNS spoofing / DNS Manipulations
c11	Falsification of record
c12	AS hijacking
c13	AS manipulation
c14	Falsification of configurations
c15	Data Manipulation
B4	**Abuse of authorisations**
c16	Escalation of Privilege
c17	Abuse of System Privileges
c18	Credential Discovery via Open Sources
c19	Credential Discovery via Sniffing
c20	Credential Discovery via Brute Force
c21	Credential Discovery via Cracking
c22	Unauthorised use or administration of devices and systems
c23	Unauthorised use of software
c24	Unauthorised access to the information systems / networks (IMPI Protocol / DNS Register Hijacking)
c25	Network Intrusion
c26	Unauthorised changes of records
c15	Data Manipulation
B5	**Remote activity (remote command execution, botnet activity, ...)**

c27	Remote Command Execution
c28	Remote Access Tool (RAT)
c29	Botnets / Remote activity
B6	**Unauthorised installation of software (malware)**
c30	Web based attacks (Drive-by download / malicious URLs / Browser based attacks)
B7	**Unauthorised activities (in application, in network, ...)**
c31	Application Exploitation via Input Manipulation
c32	Application Exploitation via Parameter Injection
c33	Application Exploitation via Code Injection
c34	Application Exploitation via Command Injection
c35	Unauthorised use or administration of devices and systems
c36	Unauthorised use of software
c37	Unauthorised access to the information systems / networks (IMPI Protocol / DNS Register Hijacking)
c38	Network Intrusion
c39	Unauthorised changes of records
c15	Data Manipulation
B3	**Malfunction (software, hardware)**
c40	Disruption of Electrical Resources
c41	Disruption of Communications Services

Annex 2 – Generic cybersecurity monitoring patterns

Suspicious behaviour monitoring use case and malicious IP/DNS use case

Database level

Pattern name	Pattern Parameters	Description
G1: Suspicious Database Modification Attempt	• **External Parameter** External IP (coming from threat intelligence) • **Frequency of login / 24h**: normal range (0,10), max range 10,000: look for > 10 logins/24h importance level: 3 direction of search: higher better • **Abnormal login time periods** normal range (6am,12pm), max range (0am - 12pm) importance level 5 direction of search: higher better • **Login session duration time**: normal range (1-10min); max range 24h importance level: 6 direction of search higher better • **Abnormal IP addresses**: normal range (A, B), min range, max range: importance level: 8 direction of search: any value • **Not authorised/suspicious database operation** No - access rejected importance level: 10 direction of search: not needed • **Logins from different location –** check on IPs used for logins used by the same account within a limited time period Importance level: 9 direction of search: anything	A complex pattern monitoring database based on login behaviour.
G2: Data modification &	• **Service level: denied requests / 24h** importance level 5 direction of search (TBD)	Monitor database operations for unusually

theft awareness	• **Data entries were copied > X files** importance level 6 direction of search (TBD) • **Other Parameter (TBD)** importance level (TBD) direction of search (TBD)	high frequency of read operations.
G3: Possible brute force/password guessing attack	• **Number of denied login requests for admin root** importance level 8 direction of search (TBD) • **Number of denied login requests for privileged account** importance level 8 direction of search (TBD) • **Show IP address**	Monitor for suspicious unauthorised access behaviour.
G4: Unauthorised access	• **Number of messages for failed attempt to perform a certain operation - unauthorised operations** importance level 5 direction of search: higher better • **Not privileged access that happened** importance level 6 direction of search: higher better	Monitor database for unauthorised access based on login behaviour and failed escalation of privilege attempts
G5: Privilege escalation	• **Suspicious upgrade of privileges** importance level 8 direction of search: not relevant • **Database user is a root** importance level 6 direction of search: not relevant	Monitor privilege behaviour for unauthorised escalation of privilege.
G6: Attempts to modify or delete audit records	• **No Audit' ON** Importance level 9 direction of search: not relevant • **Other Parameters (TBD)** Importance level (TBD) direction of search (TBD)	Monitor for change in audit record behaviour to identify audit record manipulation.
G7: Excessive Privilege Abuse	• **Unusual changes to user objects** (sys, sysdba, system) importance level 10 direction of search: not relevant • **Other Parameters (TBD)**	Simple pattern to monitor privilege abuse.

	importance level (TBD) direction of search (TBD)	

Service level

Pattern name	Pattern Parameters	Description
G8: Service Layer Unauthorised Access	• **Number of messages for failed attempt to perform a certain operation - unauthorised operations** importance level 6 direction of search: higher better • **Not privileged access that happened** importance level 4 direction of search: higher better	Pattern to monitor for unauthorised access attempts on service level.
G9: Bruce Forcing/Authentication Attempt Service Layer Logs	• **Number of denied login requests for admin root** importance level 8 direction of search: higher better • **Number of denied login requests for privileged account** importance level 8 direction of search: higher better • **Show IP address**	Monitor for brute force authentication on service level.
G10: Service Layer Privilege Escalation	• **Suspicious upgrade of privileges** importance level 8 direction of search: not applicable • **Service user is a root** importance level 6 direction of search: not applicable	Monitor for privilege escalation on service level.
G11: Injection of data/Misuse of web service	• **Number of data reads** importance level: 7 direction of search: higher better	Monitor potential misuse of web service based on data operation behaviour and network connection behaviour.

	• **Number of data modifications** importance level: 8 direction of search: higher better • **Number of HTTP requests** importance level: 5 direction of search: higher better	
G12: Service Layer Data theft	• **Service level: denied requests / 24h** importance level 4 direction of search: higher better • **Data entries were copied > X files** importance level 6 direction of search: higher better	Monitor potential data theft based on login and data read behaviour on service level.

Security appliance level

Pattern name	Pattern Parameters	Description
Firewalls		
G13: Top denied source IPs	• **External Parameters** External IP (coming from threat intelligence) • **Denied IP = true** importance level 8 direction of search: not relevant • **Number of hits in 30 days** importance level 6 direction of search: higher better • **Match with external suspicious IP = true** importance level 7 direction of search: not relevant	It is based on the number of hits recorded by the firewall. The aim is to help admins have a clearer view about the origins of the attacks.
G14: Top denied destination IPs or hosts	• **External Parameter** External IP (coming from threat intelligence) • **IP destination** importance level 8	It is based on the number of hits recorded by the firewall. The aim is to help admins have a clearer view about the

	direction of search: not relevant • **Denial time interval** importance level 8 direction of search: not relevant	most attractive targets of their environment.
G15: Top denied source IPs - destination IPs pairs	• **External Parameter (threat intelligence)** **IP** **Threat type** **Attack type** • **IP range** importance level 8 direction of search: any value • **IP destination** importance level 7 direction of search: any value • **Denial time interval** importance level 6 direction of search: any value	The combination of source and destination IPs helps the admins get a better knowledge about who targets what (and the context of potential attacks).
G16: Top denied domains	• **External Parameter** Malicious domain names (threat intelligence) • **IP range** importance level 8 direction of search: any value • **No hits/24h** importance level: 4 direction of search: higher better • **No hits/week** importance level: 5 direction of search: higher better • **No hits/month** importance level: 6 direction of search: higher better	It demonstrates the most active domains, and highlights potentially malicious domains.
G17: Top denied protocols and ports	• **External Parameter** TBD • **Port number** importance level 8 direction of search: not relevant • **Protocol number** importance level 7 direction of search:	Monitor behaviour of protocol/port hits over time for unusual behaviour.

	• **No hits protocol/24h** importance level 7 direction of search: higher better • **No hits protocol/week** importance level 6 direction of search: higher better • **No hits protocol/month** importance level 5 direction of search: higher better • **No hits port/24h** importance level 5 direction of search: higher better • **No hits port/week** importance level 6 direction of search: higher better • **No hits port/month** importance level7 direction of search: higher better	
G18: Malicious traffic unblocked (incoming and outgoing)	• **External Parameter** External IP (coming from threat intelligence) • **Malicious IP** importance level 6 direction of search: not relevant • **Denial interval time** importance level 5 direction of search: not relevant	Incoming traffic originating from IPs or, outgoing traffic to IPs reported as malicious (by external sources), that has not been blocked.
Virus Scanners		
G19: Top virus origins	• **External Parameter** **virus ID** **IP origin** • **IP origin** importance level 6 direction of search: not relevant • **Virus ID** importance level 8 direction of search: not relevant • **Country** string (display name) • **Attack time** importance level 5	Assess malware relevance by origin.

	direction of search: not relevant • **Targets** string (display name)	
G20: Severity of specific malware within specific context	• **External Parameter** • TBD • **Malware signature** string (display name) • **Malware name** string (display name) • **Time** importance level 3 direction of search: not relevant • **Computer** string (display name) • **MAC Address** importance level 4 direction of search: not relevant	Assess severity of malware by including importance of infected elements in organisational context.
Network Intrusion Detection/Prevention System (IDS/IPS)		
G21: Context and Severity of detected Network Intrusions	• **External Parameter** TBD • **Computer** string (display name) • **User** string display name) • **Name** string (display name) • **IDS signature** string (display name) • **Attack pattern** string (display name) • **Invection vector** string (display name) • **Targets** string (display name) • **Address** string (display name)	Assessing severity of intrusion based on importance of assets in organisational context.
G22: Top Network Intrusions	• **External Parameter** **IP source**	Assess top network intrusions.

	• **IP source** importance level 5 direction of search: not relevant • **IDS ID** string (display name) • **Country** string (display name) • **Attack time** importance level TBD drection of search TBD • **Targets** string (display name)	

Network level

Pattern name	Pattern Description	Comments
G21: Incoming traffic in LPA systems originating from malicious IP	**External Parameter** • **IP block list** IP value/string • **Threat type** string • **Behaviour type** string **Internal parameter** • **IP logs** IP value/string	Monitor for malicious incoming traffic.
G22: Outgoing traffic in LPA system destination to malicious IP	**External Parameter** • **IP block list** IP value/string • **Threat type** String • **Behaviour type** String **Internal Parameter** • **IP logs** IP value/string • **Content size uploaded** Size value • **Frequency of upload** Frequency value	Monitor for malicious outgoing traffic.
G23: Unusual traffic patterns	**External Parameter** • **IP block list** IP value/string **Internal Parameter**	Monitor for unusual traffic patterns based on volume, IP Port and time.

Pattern name	Pattern Description	Comments
	• **Traffic volume** Volume /value • **IP addressed to** IP value/string • **Port addressed** Port number/value • **Day/Time** Day/time format	
G24: Administrative actions from external and malicious IPs	**External Parameter** • **IP block list** IP value/string **Internal Parameter** • **Upload** Command / string • **Download** Command / string • **Frequency of upload** Frequency value • **Frequency of download** Frequency value • **Delete** Command / string • **Frequency of delete** Frequency value • **Copy** Command / string • **Frequency of copy** Frequency value	Monitor for unusual administrative activity based on location.

General security warning use case

Pattern name	**Pattern Description**	**Comments**
G27: Most critical discussions on Twitter	• **Keyword** 1,2,...n	Extract most critical events on Twitter based on keywords.

Pattern name	Pattern Description	Comments
	type: filter/string	
G28: Most critical discussions on Reddit	• **Keyword** 1,2,...m type: filter/string	Extract most critical events on Reddit based on keywords.
G29: Most relevant Twitter discussions for specific LPA	• **Keyword** 1,2,...k type: filter/string	Extract most relevant events on Twitter based on keywords.
G20: Most relevant Reddit discussions for specific LPA	• **Keyword** 1,2,...j type: filter/string	Extract most relevant events on Reddit based on keywords.

Vulnerability use case

Pattern name	Pattern Description	Comments
G31: Vulnerability monitoring	• **External parameter** CVE number Software name Software version • **Software name** 1,2,...i type: filter/string • **Software version** 1,2,...h type: filter/string	Match vulnerability to installed software/version.

Annex 3 – Pilot specific cybersecurity monitoring patterns

Larissa specific patterns

Pattern name	Pattern Parameters	Description
L1(a): Suspicious Database Modification Attempt Genesis	• **Frequency of login / 24h**: normal range Genesis (0,300), max range 500: look for > 300 logins/24h importance level: 3 direction of search: higher better • **Filter Genesis** • **Abnormal login time periods**: normal range (6.30am, 7.00pm), max range (6.30am - 11.30pm) • **Number of Users** normal range (0;5) max range (0; 50) no more than 5 users connected after 7.00 pm authentication log if S9211/S92101 are in the log as value if action name column has 'LOGON' importance level 5 direction of search: • **OS_USERNAME (machine name) - display only** • **Login session duration time**: **Logoff time - Logon time** direction of search higher better • **Abnormal IP addresses**: normal ranges 10.128.56.1-10.128.59.254 10.129.40.1-10.129.40.254 10.129.65.1-10.129.65.254 importance level: 8 direction of search • **Not authorised/suspicious database operation**: No - access rejected **>10 rejected logins/day suspicious** importance level: 10 • **Delete operations -** find it in the audit log - 'Kind of change column') filter 'service name' column genesis how many delete operations have been done >30 / day is suspicious	Instantiation of generic database modification attempt pattern for Genesis service database.
	• **Frequency of login / 24h**:	

L1(b): Suspicious Database Modification Attempt Genesis	normal range HRMS (0,30), max range 100: look for > 30 logins/24h importance level: 3 direction of search: higher better • **Filter HRMS** • **Abnormal login time periods**: normal range (6.30am, 7.00pm), max range (6.30am - 11.30pm) • **Number of Users** normal range (0;5) max range (0; 50) no more than 2 users connected after 7.00 pm authentication log if SHR is in the log as value if action name column has 'LOGON' importance level 5 direction of search: higher better • **OS_USERNAME (machine name)** • **Login session duration time**: **Logoff time - Logon time** direction of search higher better • **Abnormal IP addresses**: normal ranges 10.128.56.1-10.128.59.254 10.129.40.1-10.129.40.254 10.129.65.1-10.129.65.254 importance level: 8 direction of search: higher better • **Not authorised/suspicious database operation**: No - access rejected **>10 rejected logins/day suspicious** importance level: 10 • **Delete operations -** find it in the audit log - 'Kind of change column') • **filter 'service name' column genesis** how many delete operations have been done Deletes operations >100 / day is suspicious importance level 7 interval range (100, 10,000) direction of search: higher better	Instantiation of generic database modification attempt pattern for Genesis service database.
L2: Monitor for suspicious delete operations (2 patterns HRMS / Genesis)	• **filter 'kind of change' field / audit log** string type/show value • **filter by HRMS/Genesis** string type/show value HRMS server: >100 number of deletes is day suspicious	Monitor for a suspicious number of delete operations in HRMS/Genesis database.

	• **HRMS deletes/day** importance level 7 normal range: (0: 100) max range: (0; 10,000) direction of search: higher better Genesis >30 deletes /day suspicious • **Genesis deletes/day** importance level 7 normal range (0; 30) max range (0; 1,000) direction of search: higher better	
L3: Suspicious / context specific database modification	• **filter Table name** string type • **filter per multiple columns to monitor** string type • **sub-filter 'update/delete' type of change#** string type	Monitor for data operations in specified tables that should not be modified.
L4: Monitor Tax Manipulation Attempt	• **Locked date for salary record** date value • **Last modification date of the salary** date value • **if Locked date < Change date** boolean: true/false • **Salary record table name** string • **Salary record column name entry** string	A specific scenario-based monitoring pattern that monitors for modification of specific tables after a specified data every month. Modification after that date is suspicious and could represent a malicious manipulation attempt.
L5: Data Theft analysis pattern	• **Log off LRead number** normal range (0; 50) max range (0;100) direction of search: higher better • **Log off PRead number** normal range (0; 50) max range (0;100) direction of search: higher better • **Sum of total Reads** Sum LRead+Pread > XMax	Monitor for potential data theft based on number of data operations.

	boolean: true/false	

Rome specific patterns

Pattern name	Pattern Parameters	Description
R1: Suspicious Database Modification Attempt SUET	• **Abnormal login time periods**: importance level: 5 normal range (7am, 8pm) max range (0am to 12pm) direction of search: • **Show if admin or regular user** string type; display • **Number of Users** normal range (0;2000) max range (0; 5000) direction of search: higher better • **OS_USERNAME (machine name) - display only** os_userid and userhost and host (=IP) • **Login session duration time**: Logoff time - Logon time direction of search higher better • **Not authorised/suspicious database operation**: No - access rejected • **If login rejections number >200 /hour - suspicious** **boolean: true/false** importance level: 10	Instantiation of generic database modification attempt pattern for SUET service database.
R2: Denial of Service Attack	• **Denial of services** **Table "RICHIESTE"** importance level: 8 normal range (0,200) max range (0, 500) • direction of search: higher better • **byte sent file size** importance level: 8 normal range (0;200) MB max range (0; 5,000) MB direction of search: higher better • **number of requests / hour >20** importance level 7 normal range (0; 20) max range (0; 10,000) direction of search: higher better	Denial of service attempt against SUET service database

	sent from same ip in short time (20 requests from same ip in one hour) • **same file size uploaded /hour >10** is suspicious (denial of service)	
R3: Monitor for suspicious update/delete operations	**External Parameter** • **IP block list** IP value/string **Internal Parameter** • **Suspicious Table name** string • **Day/Time of Delete** Time format Importance level 5 • **Deadline Delete/Update** Normal range (1;20) Maximum range (1;300) Direction of search: higher better Importance level 8 • **Frequency of Delete/Update** Frequency value Normal range (0; 2) Maximum range (0; 100) Direction of search: higher better Importance level 6	A pattern to monitor for suspicious delete operations in the SUET database.
R4: Suspicious / context specific database modification	**External Parameter** • **IP block list** IP value/string **Internal Parameter** • **IP logs** IP value/string • **Update type** Command / string • **Frequency of update** importance level: 8 normal range (0,10)	A pattern to monitor for data modification in critical SUET database tables.

	max range (0, 500) direction of search: higher better	
R5: Suspicious IP monitoring	**External Parameter** • **IP block list** IP value/string • **Threat type** string • **Behaviour type** string **Internal parameter** • **IP logs** IP value/string • **User name** string • **WAF IP logs** IP value/string • **Reverse proxy IP** IP value/string • **WAF IP = Reverse proxy IP** True/false Boolean • **Login request** Normal range (0;10) Maximum range (0, 1000) Direction of search: higher better Importance level: 7	Instantiation of generic malicious IP monitoring pattern.
R6: Data theft monitoring	• **Frequency of download** importance level: 8 normal range (0,20) max range (0, 500) direction of search: higher better • **Frequency of daily access** importance level: 7 normal range (0,20) max range (0, 500) direction of search: higher better hours of access importance level normal range (8am-5pm) max range (0am-12pm) • **Frequency of monthly access** importance level: 7	Monitor for data theft based on frequency of data operations.

	normal range (0,20) max range (0, 500) direction of search: higher better • **Specific areas of Rome targeted** Severs string; display	
R7: HTTP status field monitoring	**External Parameter** • **IP block list** IP value/string **Internal Parameter** • **IP from logs** IP value/string • **HTTP status error** True/false - boolean • **HTTP number of errors** Normal range (0;10) Maximum range (0;1000) Direction of search: higher better Importance level 6 • **HTTP error timestamp (errors / minute)** Normal range (0;10) Maximum range (0; 1000) Direction of search: higher better Importance level 5	Monitor for potential malicious activity by observing the http return code frequency for error codes.
R8: Unusual IP address	**External Parameter** • **IP block list** IP value/string **Internal Parameter** • **IP block list = IP from logs** True/false - Boolean • **IP from logs** IP normal range (IP1…IPn) IP max range: any Directions of search: any Importance level 7	Monitor allowed IP range for SUET from network level.

Annex 4 – Generic self-healing policies

Attack/ Scenario	Generic Threat	Healing options	Self-healing	Relevant monitoring pattern	Relevant appliance to be applied to
Adware	c3, c30	Add a firewall rule in order to block the given malicious IP address	X	G13, G16, G19, G23	Firewall
		Add the given malicious domain to the list of Restricted Sites	X		Firewall, antivirus, endpoint protection
		Add the given malicious url to the list of Restricted Sites	X		Firewall, antivirus, endpoint protection
		Block all malicious IP addresses, disconnect your machine from the network and then clean and restore your machine			
Backdoor	c1, c3, c22, c24, c25, c27, c28, c30	Add a firewall rule in order to block the given malicious IP address	X	G13, G15, G16, G19, G23	Firewall
		Add the given malicious domain to the list of Restricted Sites	X		Firewall, antivirus, endpoint protection
		Add the given malicious URL to the list of Restricted Sites	X		Firewall, antivirus, endpoint protection
Bot	c1, c2, c10, c17, c22, c29, c38, c30	Block all malicious IP addresses, disconnect your machine from the network and then clean and restore your machine.		G3, G9, G13, G17, G23	
DDoS	c29	Add proper firewall rules in order to block the malicious IP addresses that perform the DDoS attack	X	G13, G17, G18, G23	Not applicable in current CS-AWARE context.
Ransomware	C15, c17, c30	Add a firewall rule in order to block the given malicious IP address	X	G8, G9, G10, G13, G16, G23, G25, G26	

		Add a filter in your email client in order to block the given malicious email address			Email client
		Add the given malicious domain to the list of Restricted Sites	X		Firewall, antivirus, endpoint protection
		Add the given malicious url to the list of Restricted Sites	X		Firewall, antivirus, endpoint protection
Rootkit	C16, c27, c35	Add a firewall rule in order to block the given malicious IP address	X	G1, G2, G3, G4, G5, G7, G9, G10, G12, G17, G19, G25, G26	Firewall
Virus	B1, B4, B6, B7	Add a firewall rule in order to block the given malicious IP address	X	G1, G2, G3, G6, G7, G8, G9, G10, G12, G13, G16, G19, G23, G25	Firewall
		Add a filter in your email client in order to block the given malicious email address			
		Add the given malicious domain to the list of Restricted Sites	X		Firewall, antivirus, endpoint protection
		Add the given malicious url to the list of Restricted Sites	X		Firewall, antivirus, endpoint protection
Vulnerability	B1, B2, B3, B4, B5, B6, B7	Apply update/patch	X	G1, G2, G4, G5, G6, G8, G10, G12, G13, G16, G19, G23, G25	Software
Malicious IP address	B1, B5	Add a firewall rule in order to block the given malicious IP address	X	G1, G2, G3, G4, G5, G9, G13, G14, G15, G16, G18, G19, G22, G23, G24, G26	Firewall

Malicious domain name / URL	B1, B5, B6, B7	Add the given malicious domain to the list of Restricted Sites	X	G11, G16	Firewall, antivirus, endpoint protection
Generic web threat	B5, B6, B7	Use the three following methods in order to protect your application from WEB threats: Escape user input, Validate user input, Sanitise user input		G11, G13, G16, G23	

Annex 5 – Pilot specific self-healing policies

Larissa monitoring pattern specific self-healing policies

Attack/ Scenario	Generic Threat	Healing options	Self-healing	Relevant monitoring pattern	Relevant appliance to be applied to
Suspicious Database Modification Attempt	c16, c17, c26, B2	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		L1	
		All user-controllable input must be validated and filtered for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Add a firewall rule to block the suspicious IP address that performs the database modification	X		Firewall
Monitor for suspicious delete operations	c17, c26, B2	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		L2	
		All user-controllable input must be validated and filtered for illegal characters as well as			

		SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Block the suspicious user who performs the modification actions			Database
		Add a firewall rule to block the suspicious IP address that performs the database modification	X		Firewall
Suspicious / context specific database modification	c16, c17, c26, B2	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		L3	
		All user-controllable input must be validated and filtered for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside			

		such domains is considered invalid and the query fails			
		Block the suspicious user who performs the modification actions			Database
		Revoke the privileges of the suspicious user on the specific database or table			Database
Monitor Tax Manipulation Attempt	B2, c17	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		L4	
		All user-controllable input must be validated and filtered for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Add a firewall rule to block the suspicious IP address that performs the database modification	X		Firewall
Data Theft	B4, B6	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		L5	
		All user-controllable input must be validated and filtered			

		for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Add a firewall rule to block the suspicious IP address that performs the database modification	X		Firewall
Monitor important parameter tables	B2	All user-controllable input must be validated and filtered for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear		L6	
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			

		Add a firewall rule to block the suspicious IP address that performs the database modification	X		Firewall
		Block the suspicious user who performs the modification actions			Database

Rome monitoring pattern specific self-healing policies

Attack/ Scenario	Generic Threat	Healing options	Self-healing	Relevant monitoring pattern	Relevant appliance to be applied to
Suspicious Database Modification Attempt	B2	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		R1	
		All user-controllable input must be validated and filtered for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Add a firewall rule to block the suspicious IP address that performs the database modification	X		Firewall
Denial of service attack	c29	Configure web server software to limit the waiting period on opened HTTP sessions.		R2	
		Use load balancing mechanisms.			HTTP server
		Add proper firewall rules in order to block the malicious IP			Firewall

		addresses that perform the DoS attack.			
Monitor for suspicious delete operations	B2	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		R3	
		All user-controllable input must be validated and filtered for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Block the suspicious user who performs the modification actions			Database
		Add a firewall rule to block the suspicious IP address that performs the database modification.	X		Firewall
Suspicious / context specific database modification	B2	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		R4	
		All user-controllable input must be validated and filtered for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures -			

		Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Block the suspicious user who performs the UPDATE operations			Database
		Revoke the privileges of the suspicious user on the specific database or table.			Database
Suspicious IP	B1, B5, B6	Add a firewall rule in order to block the given malicious IP address/es:	X	R5	Firewall
Data Theft analysis pattern	B4	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification		R6	
		All user-controllable input must be validated and filtered for illegal characters as well as SQL content. Keywords such as UNION, SELECT or INSERT must be filtered in addition to characters such as a single-quote(') or SQL-comments (--) based on the context in which they appear			
		Use of parameterised queries or stored procedures - Parameterisation causes the input to be restricted to certain domains, such as strings or integers, and any input outside such domains is considered invalid and the query fails			
		Add a firewall rule to block the suspicious IP address that performs the database modification	X		Firewall
Http status field	B6	For applications that leverage remote schemas, use the HTTPS protocol to prevent modification of traffic in transit and to avoid unauthorised modification.		R7	
		Configure the access control correctly.			HTTP server

		Enforce principle of least privilege.			
		Execute programs with constrained privileges, so parent process does not open up further vulnerabilities. Ensure that all directories, temporary directories and files, and memory are executing with limited privileges to protect against remote execution.			
		Input validation. Assume that user inputs are malicious. Utilise strict type, character, and encoding enforcement.			
		Proxy communication to host, so that communications are terminated at the proxy, sanitising the requests before forwarding to server host.			
		Run server interfaces with a non-root account and/or utilise chroot jails or other configuration techniques to constrain privileges even if attacker gains some limited access to commands.			
		Perform input validation for all remote content, including remote and user-generated content.			
		Validate user input by only accepting known good. Ensure all content that is delivered to client is sanitised against an acceptable content specification -- whitelisting approach.			
Unusual IP addresses	B1, B6	Add a firewall rule in order to block the given malicious IP address	X	R8	Firewall

Annex 6: CS-AWARE requirements collection

On reviewer request, a consolidated list of CS-AWARE requirements is added as an Annex to this deliverable. This list represents an updated version of the tables presented in Section 1 of deliverable D2.2 and follows the three categories of requirements defined in D2.2: the requirements for selection of external information sources (Section 1.1 of D2.2), the requirements for LPA specific analysis and information sources (Section 1.2 of D2.2) and the CS-AWARE system requirements (Section 1.3 of D2.2). A detailed description of the context of those requirements as well as the methodology used for requirements selection and the selection procedures can be found in the respective Sections of D2.2. The requirements I8 and I9 have been added at a later stage and are detailed in Section 2.1 of this document.

Requirements for External Information Sources

#	Requirement	Functional	Non-functional	End user viewpoint
E1	Overview of all potential data sources that can be collected to enhance cybersecurity in the context of a dynamic environment that requires constant re-evaluation and integration of new information		X	
E2	Information sources that can be collected dynamically		X	
E3	Information sources that facilitate collaborative and cooperative cybersecurity		X	
E4	Information sources that are community driven (non-commercial operators)		X	
E5	Information sources that follow common cybersecurity exchange standards		X	
E6	Information sources that are especially applicable to LPA cybersecurity context, based on initial risk analysis performed in CS-AWARE deliverable D2.1		X	
E7	Periodisation and selection based on these Requirements	X		

LPA specific analysis requirements

#	Requirement	Functional	Non-functional	End user viewpoint
I1	Identification of the critical assets (socio-technical) in the LPA system	X		X
I2	Identification of the critical dependencies between assets in the LPA system	X		X
I3	Identification of critical LPA services and service processes	X		X

I4	Identification of information flows of critical service processes through assets and dependencies	X		X
I5	Identification of monitoring points able to determine cybersecurity state related to information flows of critical service processes	X		X
I6	Improve the system understanding of LPA personal		X	X
I7	Increase cybersecurity awareness for LPA personal with respect to LPA systems		X	X
I8	Determine normal and abnormal behaviour related to critical service processes and information flows	X		X
I9	Interface analysis results with CS-AWARE technology solution for continuous monitoring, awareness and self-healing	X		

CS-AWARE system requirements

#	Requirement	Functional	Non-functional	End user viewpoint
Technical System Requirements				
S1	**Provide cybersecurity awareness**	X		X
S2	**Allow information sharing**	X		X
S3	**Enable system self-healing**	X		X
S4	Enable data collection from internal LPA and external cyber security information sources	X		
S5	Allow preprocessing to bring data into an unified format	X		
S6	Enable data analysis by setting external and internal data into context	X		
S7	Ensure international usability of the system by providing multiple languages	X		
S8	Identifying relevant internal and external sources	X		
General System Requirements				
S9	Usability (The usability of the CS-AWARE system, as determined by the end users)		X	X
S10	Compliance (Compliance to LPA regulations, policies and procedures)		X	X
S11	Integratability (Integratability of CS-AWARE system into LPA work flows)		X	
S12	Open Source (How much of the CS-AWARE components can be open sourced and how much is kept proprietary)		X	

S13	Internationalization (Integration into different cultural and language contexts)		X	
S14	Cost/Marketability of CS-AWARE solution		X	