UNIVERSITY OF PIRAEUS

Improving the Security Monitoring Process

Vasileios Anastopoulos

Doctoral dissertation

Department of Digital Systems

School of Information and Communication Technologies

December 2019

This doctoral dissertation of Vasileios Anastopoulos, entitled “Improving the Security Monitoring Process” was examined and approved by the following examination committee:

Sokratis Katsikas, Professor
Department of Digital Systems
University of Piraeus

Costas Lambrinoudakis, Professor
Department of Digital Systems
University of Piraeus

Christos Xenakis, Professor
Department of Digital Systems
University of Piraeus

Stefanos Gritzalis, Professor
Department of Digital Systems
University of Piraeus

Christos Kalloniatis, Associate Professor
Department of Cultural Technology and Communication
University of the Aegean

Maria Karyda, Assistant Professor
Department of Information and Communication Systems Engineering
University of the Aegean

Aggeliki Tsohou, Assistant Professor
Department of Informatics
Ionian University

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing, which is the outcome of work done in collaboration, except where specifically indicated in the text.

Vasileios Anastopoulos

December 2019

Acknowledgments

First and foremost, I would like to express my special appreciation and thanks to my supervisor Professor Sokratis Katsikas; he has been a tremendous mentor for me. I would like to thank him for encouraging my research and for allowing me to grow as a research scientist. His advice on both research as well as on my career have been priceless. I would also like to thank my committee members, Professor Konstantinos Lambrinoudakis and Associate Professor Christos Xenakis for their time, interest, and insightful comments.

Vasileios Anastopoulos

December 2019

Abstract

Organizations that maintain information systems are constantly facing new security challenges in cyberspace, as individuals, organized groups, criminal organizations and nation sponsored actors, threaten their infrastructures having different motivation and aims. The need for business continuity and compliance to standards and laws, render their monitoring a necessity, to make the organization aware of its security posture and enable it to respond to security incidents. The problem that was investigated with this dissertation is the need for security monitoring in large-scale infrastructures composed of heterogeneous devices geographically dispersed, aiming to increase and optimize the capabilities of organizations.

This research focused on the design of infrastructures for the management of the data (log data) that security monitoring relies on, aiming to address all aspects of the topic. Research started with literature review and resulted to the proposal of a methodology for the design of a log management infrastructure, addressing all aspects of the topic from the capture of requirements through out the collection of log data to a central location, and can be used as a step-by-step guide from the designer. It continued researching the need to validate the design of log management infrastructure, verifying that its users can actually perform the tasks that result from the requirements, as well as to ensure the optimal usage of its resources and exploitation of the collected log data. It concluded researching the ability to dynamically design a log management infrastructure able to evolve adopting to changing threat landscape it operates in.

The introduction of the field of social network analysis to the design of such infrastructures was an innovation introduced by this research. The application of a this topic, well

established in social sciences, proved to be an agile tool for managing the complexity of designing a large-scale infrastructure. The measurements and analysis techniques available in social network analysis were used to justify and document design decisions and enabled the consideration of additional issues affecting the design. The artifacts of this research were demonstrated and evaluated performing case studies that used the infrastructure of the Greek Research and Technology Network (GRNET S.A), though future work could focus on industrial control systems (ICS), the Internet of Things (IoT) and maritime security.

Περίληψη

Οι οργανισμοί που διατηρούν πληροφοριακά συστήματα αντιμετωπίζουν διαρκώς νέες προκλήσεις ασφαλείας στον κυβερνοχώρο, καθότι μεμονωμένα άτομα, οργανωμένες ομάδες, εγκληματικές οργανώσεις αλλά και χώρες, απειλούν τις υποδομές τους με διαφορετικά κίνητρα και διαφορετικές επιδιώξεις. Η απαίτηση για επιχειρησιακή συνέχεια αλλά και λόγοι συμμόρφωσης στη νομοθεσία και σε πρότυπα, καθιστούν επιτακτική την επιτήρηση αυτών των υποδομών, προκειμένου ο οργανισμός να γνωρίζει της κατάσταση ασφαλείας του (security posture) και να αντιμετωπίζει τα αντίστοιχα περιστατικά. Το πρόβλημα που εξετάστηκε σε αυτήν τη διατριβή είναι η ανάγκη για παρακολούθηση ασφαλείας (security monitoring) σε υποδομές μεγάλης κλίμακας, αποτελούμενες από ετερογενείς και γεωγραφικά διασπαρμένες συσκευές, στοχεύοντας στη βελτίωση και επαύξηση των δυνατοτήτων των οργανισμών.

Η έρευνα εστίασε στη σχεδίαση υποδομών για τη διαχείριση των δεδομένων (αρχείων καταγραφής) στα οποία βασίζεται η παρακολούθηση ασφαλείας, επιδιώκοντας την κάλυψη όλων των πτυχών του θέματος. Η έρευνα ξεκίνησε με ανασκόπηση της διαθέσιμης βιβλιογραφίας καταλήγοντας στην πρόταση μιας μεθοδολογίας για τη σχεδίασης υποδομών διαχείρισης αρχείων καταγραφής η οποία εξετάζει όλες τις πτυχές του θέματος, από την καταγραφή των απαιτήσεων μέχρι τη συλλογή των αρχείων καταγραφής σε ένα κεντρικό σημείο και μπορεί να χρησιμοποιηθεί ως ένας οδηγός βήμα-βήμα από το σχεδιαστή της υποδομής. Συνεχίστηκε με τη διερεύνηση της ανάγκης επικύρωσης της σχεδίασης μιας υποδομής, επιβεβαιώνοντας ότι οι χρήστες της μπορούν να εκτελέσουν τις εργασίες που προκύπτουν από την καταγραφή των απαιτήσεων, καθώς και ότι επιτυγχάνεται η βέλτιστη χρήση των διαθέσιμων πόρων και εκμετάλλευση των αρχείων καταγραφής. Ολοκληρώθηκε

με τη διερεύνηση της δυνατότητας δυναμικής σχεδίασης μιας υποδομή διαχείρισης αρχείων καταγραφής και της δυνατότητας εξέλιξής της, προσαρμοζόμενη στην εξέλιξη των απειλών και του περιβάλλοντος λειτουργίας της συνολικά.

Καινοτομία αποτέλεσε η εισαγωγή του πεδίου της ανάλυσης κοινωνικών δικτύων (social network analysis) στη σχεδίασή των υποδομών διαχείρισης αρχείων καταγραφής. Η εφαρμογή ενός ερευνητικού πεδίου από τις κοινωνικές επιστήμες, αποδείχθηκε ευέλικτο εργαλείο διαχείρισης της πολυπλοκότητας της σχεδίασης υποδομών διαχείρισης αρχείων καταγραφής σε υποδομές μεγάλης κλίμακας. Οι μετρήσεις και οι τεχνικές ανάλυσης που είναι διαθέσιμες,χρησιμοποιήθηκαν για τη λήψη και τεκμηρίωση των σχεδιαστικών αποφάσεων, ενώ κατέστησαν εφικτή τη συμπερίληψη στη σχεδίαση επιπλέον θεμάτων που την επηρεάζουν. Τα αποτελέσματα της έρευνας επιδείχθηκαν και αξιολογήθηκαν χρησιμοποιώντας την υποδομή του Εθνικού Δικτύου Υποδομών Τεχνολογίας και Έρευνας (ΕΔΥΤΕ Α.Ε.), ενώ θα μπορούσε να συνεχιστεί διερευνώντας τομείς όπως τα βιομηχανικών συστημάτων ελέγχου (Industrial Control Systems), το διαδίκτυο των πραγμάτων (Internet of Things) και της θαλάσσιας ασφάλειας (maritime security).

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Organizations need to monitor their infrastructures either for security, or operational, reasons. The log records generated by the devices comprising their infrastructure provide information on the events occurring on the infrastructure. The collection and analysis of log data allows the organization to compile its security posture as well to investigate security incidents or operational problems. The multitude and variety of the log generating devices as well as the volume of the generated log data make their collection and analysis a demanding task, creating the need for log management.

Log management is the process of generating, transmitting, storing, analyzing and disposing the security log data and it is necessary for an organization to perform security monitoring of its infrastructure. Apart from securing the organization’s operation, log management is mandatory for specific categories of organizations through legislation and the need for compliance to relevant standards and regulations. Log management does not come at no cost; organizations need to invest in infrastructures, to define requirements, policies and procedures, as well as to dedicate specialized personnel to relevant roles. When planning for log management, the log data volume, the network bandwidth, the data storage and the specialized personnel are indicative factors to be considered. The resulting costs cannot be neglected, posing the need to balance the limited log management resources with the continuous flow and collection of log data.

1.2 Research Questions

Log management is a common practice among organizations wishing to monitor their infrastructure and maintain awareness of their security posture. Reviewing the available literature prior performing this research, resulted in that the design of such infrastructure is commonly based on vendors' proposed practices and in some cases on the designer's intuition.

With this study we try to address the need to perform real-time security monitoring on Wide Area Networks (WAN) consisting of heterogeneous and geographically dispersed devices, aiming to advance the cyber situational awareness of organizations. This problem was analyzed to the following research topics, investigated in this dissertation:

- How to design a log management infrastructure considering both high-level and low-level aspects of log management and justifying design decisions that are commonly based on experience and intuition.

- How to validate the design and configuration of this large scale log management infrastructure, allowing the measurement of its performance and the application of corrective actions.

- How to make the design of the infrastructure dynamic, enabling it to adapt to the evolving threat landscape or log management requirements.

1.3 Scope and Objectives

The objective of this work is to improve the security monitoring capabilities of organizations that maintain large scale infrastructures, such as WANs and Industrial Control Systems (ICS), enabling timely and accurate security awareness in the cyber domain. It researches both high-level and low-level aspects of log management, from the definition of the log management requirements, to the management of the log data and measurement of performance. Log analysis and visualization are out of scope of this research and are not addressed. Though the focus is on security the outcomes of this research are also applicable for administrative reasons, too.

The main objective of timely and accurate security monitoring was analyzed to the following objectives:

- The definition of a step-by-step methodology for the design and implementation of a large scale log management infrastructure, where the design decision would be formally documented and justified. This methodology had to be vendor-independent, though incorporate industry best practices for both high-level and low-level aspects of log management.

- The definition of a methodology that would validate the design of a log management infrastructure and would enable corrective actions to optimize its design. The design of an infrastructure should enable the analysts to perform their tasks and fulfill the requirements for which it was built.

- Considering the evolution of threats and business needs, the design of a log management infrastructure should be dynamic, enabling it to adapt to the evolving threat landscape in an optimal way.

1.4 Research Methodology

This dissertation research seeks an innovative solution to the real-world problem of performing securing monitoring on large scale infrastructures; it is a problem that affects modern business operations and is a challenge that most organizations have to deal with. Surveys and reports from organization and institutions, as well as academic research, helped to formulate the problem, as organizations are reluctant in releasing information of their deployments and operations.

Research on security monitoring started reviewing the available literature, industry and vendors' practices and solutions, as well as recent academic research. Log management, log analysis and log data visualization were identified as candidate topics of research, choosing to focus on log management and specifically on large scale infrastructures, due the challenges that derive from their scale, geographic dispersion and heterogeneity of their components, and the lack of literature addressing them.

The proposed solution was a methodology that would address all aspects of the design and implementation of a log management. The proposed solution was conceptualized approaching large scale infrastructures as complex systems. The concepts and techniques of social network analysis (SNA) were considered applicable, as large scale infrastructures could be modeled as social network, where prior research was available. Graph analysis, a subset of social network analysis, had already be applied on power grids, transportation networks, watering grids, etc.

The research methodology selected for this research work is the Design Sciences Research Methodology (DSRM) [1], which relies on the creation of “knowledge and understanding of a design problem, and its solution is acquired in the building and application of an artifact” [2]. It also involves the “analysis of the use and performance of designed artifacts to understand, explain and very frequently to improve the behavior of aspects of information systems” [3]. In [1] the authors presented a DSRM framework consisting of five activities, Figure 1. Studying the artifacts designed and developed in the area of information technology and information systems [4], they were classified into eight categories: System Design (A description of an IT-related system), Method (Define the activities to create or interact with a system), Language/Notation (A -generally formalized- system to formulate statements that represents parts of reality), Algorithm (An executable description of behavior of a system), Guideline (Provide a generalized suggestion about system development), Requirements (Statements about a system), Pattern (Definition of reusable elements of design with its benefits and application context), and Metric (A mathematical model that is able to measure aspects of systems or methods).

The main results of this research work are three methodologies: for building a log management infrastructure; for validating the design of log management infrastructure, and; for evolving the design of a log management infrastructure. These methodologies encompass and contribute various design science artifacts which, applying the categorization proposed in [4], can be grouped as follows: systems design (architectural design of log management infrastructures; systems configuration for log management tasks), methods (for building a log management infrastructure; for the validation of the design of a log management infrastructure; for the evolution of the design of a log management infrastructure), guidelines (for the execution of log management tasks; for making design decisions; for capturing requirements), requirements (for the design of a log management infrastructure), metrics (measurements program for the building of a log management infrastructure; measurements of the importance of various components of an infrastructure; measurements for the documentation of design decisions).

The problem of performing real-time security monitoring on large-scale infrastructures, was explicated as a log management problem and specifically, as the need: to properly design a log management infrastructure; to validate the design of the log management infrastructure; and the need to evolve the design of the log management infrastructure. Focusing of these three needs was decided after reviewing the available research literature, vendors' documentation and survey results (conducted by third-party institutions and organizations).

Moving to the definition of requirements, the absence of a methodologies that could be used as step-by-step guides by designers was identified, especially in the case of large-scale infrastructures, where the complexity of the design demands for a structured approach. The methodologies should address both high-level and low-level aspects of log managements, from the definition of the log management requirements through to the measurement of the performance of the implemented infrastructure, aiming to enable timely and accurate security awareness of organizations in the cyber domain.

The three main artifacts (methodologies) were designed and developed, using creative methods: the methodology for designing a log management infrastructure, was the result of modeling an infrastructure as a social network and analyzing its structural properties. A topic present in social sciences, namely the SNA, was introduced to justify and document the design decisions. Having reviewed the available literature all the aspects of log management that should be considered by a designer, were identified, adjusted and integrated to the resulting methodologies. At the best of our knowledge, no such methodology existed, as the available ones addressed only specific issues of log management. The definition of the eleven steps of the methodology was the result of reviewing the available literature, the challenges stakeholders face as well as the fulfillment of the requirements set for the artifact development. For the second artifact (methodology for the validation of the design of an infrastructure), the novel approach of considering an infrastructure as a complex organization was employed, thus enabling the application of concepts and techniques available in Meta-Network Analysis (MNA), a topic used to analyze the performance of large organizations. For the design and development of the third artifact (the methodology for evolving the design of a log management infrastructure) the concepts of affiliation analysis, a subset of SNA used to analyze human relations, was introduced to correlate the design of an infrastructure with the risk it faces.

All three methodologies, and their design science artifacts, where evaluated, applying the demonstration, simulation and metrics, evaluation patterns [3], Table 1. Applying the demonstration evaluation pattern, the solution was constructed and it was demonstrated that it is realizable, feasible and acceptable. Employing the simulation pattern, a model of the problem and its solution was developed, the entities and their interactions were captured and the validity of the solution to the problem was tested. Simulating the solution of a complex problem overcome the problems of cost and restrictions in data availability, while the working of the artifact were evaluated on a real-life organization. Finally, with metrics evaluation pattern, metrics already established in literature were used to evaluate the proposed solutions and prove the correctness of the hypotheses regarding the solutions. The metrics of SNA, used to analyze social networks of actors (humans), were properly applied to validate the performance of the proposed solutions, as well as to prove the correctness of the hypotheses.

In order to demonstrate the solution, for each research question, we conducted the respective case studies. To define the research question with accuracy, we used surveys conducted by organizations and institutions. In all case studies we demonstrated the proposed solution applying it on the infrastructure of a real organization using real data that were publicly available. Having collected the necessary data, we constructed the models and applied the selected measurements and techniques, on which the proposed solutions were based.

Table 1: Evaluation of design science artifacts [3]

Artifact	**Evaluation Methodology**	**Context and Applicability**
Methodology for building a log management infrastructure	Demonstration; Simulation; Using metrics	The evaluation and validation of the solution in the real-life setting is costly. Demonstrate that the solution is realizable. Established metrics exist in the literature.
Methodology for validating the design of a log management infrastructure	Demonstration; Simulation; Using metrics	The problem and its solution can be accurately modeled on a computer. Demonstrate that the solution is realizable.

Artifact	Evaluation Methodology	Context and Applicability
		Metrics are available to solve a similar problem.
Methodology for the dynamic design of adoptive log management infrastructure	Demonstration; Simulation; Using metrics	The problem and its solution can be accurately modeled on a computer. Demonstrate that the solution is realizable. Metrics are available to solve a similar problem.

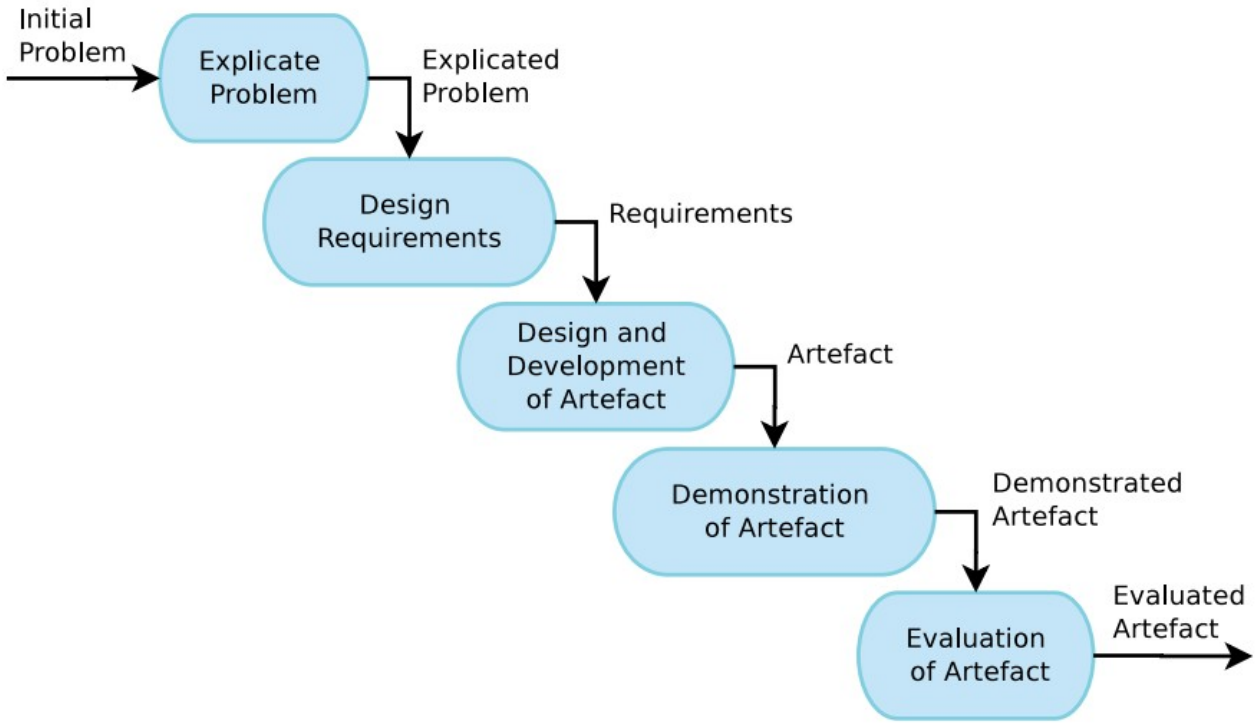

Figure 1: Design Science Research Methodology activities

1.5 Thesis Overview

The remainder of this thesis in organized in four chapters. In Chapter 2, we discuss the design of a log management infrastructure. The related work in building such infrastructures is reviewed and a new methodology is proposed. The new methodology integrates available processes and industry practices. It addresses all issues of log management from the generator of the log records, through to their collection and storage to a central point. The discipline of social network analysis is introduced to provide the means for performance and the justification of design decisions. The proposed methodology is applied on real infrastructure, to demonstrate its workings and contribution. It is a complete guide, for the design and implementation of log a management infrastructure, that ensures that all aspects have been properly considered.

Having resulted to the design of a log management infrastructure, Chapter 3 aims to verify this design. The implementation of an infrastructure aims to satisfy the corresponding requirements, that can derive from regulatory reason, compliance to standards, support to the business process, etc. The resulting infrastructure needs to validated against those requirements, which in this research is refined to the ability to verify that on each log collection point, the security personnel is able to perform the necessary tasks, using the log data collected on that specific point. An extension of SNA, Meta-Network Analysis (MNA), is employed to model a log management infrastructure as an organization, and leverage its concepts, techniques and measurements to verify our hypothesis. The resulting methodology is demonstrated on a real infrastructure compiling a detailed case study.

In Chapter 4, the need for dynamic log management is researched. The previous chapters researched the design of large scale log management infrastructures ensuring that all issues affecting them have been considered to the design process and that they properly configured for the desired tasks. Despite this optimization, the security needs, the threat landscape and the business needs are in constant change, and may render a log management infrastructure to be ineffective. The results of the research presented in Chapter 4, enable the evolution of a log management infrastructure to properly adapt for the threats it actually faces, which can greatly differ from those that lead to its implementation at the first place. Risk is introduced in the design process, dynamically correlating it with the log data and the log management tasks. Concepts and techniques from SNA and MNA enable this correlation and result to a novel methodology for the dynamic design of log management infrastructures.

In the concluding Chapter 5, the finding and contribution of this research are summarized, its limitations are discussed and directions for further research are identified.

1.6 Summary of Contribution

The contribution of this research is focused on the design of log management infrastructures. Despite the significance of log management for security, compliance and other reasons, no step-by-step methodology existed for the design of such infrastructures. Existing methods addressed specific issues of log management, or were vendor specific. There were no mentions, in bibliography and related research, on validating that an implemented infrastructure actually fulfilled the requirements that drove its implementation, or on dynamic design; all methods resulted in static configurations without any provision for adjustment to new business needs. The output and contribution of this research is summarized as follows:

- A step-by-step methodology for the design of log management infrastructure that addresses both high-level and log-level issues of log management.

- A methodology for the validation of the log management infrastructure design, ensuring both the achievement of the log management tasks and the optimal usage of log management resources.

- A methodology that enables the dynamic design of log management infrastructures enabling their optimal configuration and evolution according to the threat landscape they operate in.

- The introduction of the discipline of social network analysis (SNA) to log management, enabling the justification and documentation of design decisions as well as the consideration of factors such as risk to the design process.

The contribution of this research from the resulting publications, is summarized in Table 2.

Table 2: Research contribution

Published paper	Research objective	Key results
[5]	To define formal methods for the making of decisions for the design and implementation of log management infrastructures in heterogeneous and geographically dispersed infrastructures.	A methodology for the design and implementation of a vendor-independent log management infrastructure on large scale and dispersed infrastructures.
[6]	To provide the means to validate the design of a large scale log management infrastructure.	A method for validating that on each log collector the required analysis tasks can be performed, and a measure of each log collector's criticality.
[7]	How to design of a log management infrastructure in large scale infrastructures, ensuring all aspects of log management and properly justifying design decisions.	An extension of [5] key results, defining a methodology that addresses both high-level and low-level aspects of log management, and can be used as a step-by-step guide for the implementation of a log management infrastructure, from the log source through to the log data collection and storage in a central point.
[8]	To extend the design methodology proposed in [7] correlating risk to the design process.	A methodology for the identification of high-risk assets and prioritization of incident response.
[9]	To provide the ability to dynamically design log management infrastructures, adaptable to the evolving threat landscape.	An extension of [8] contributing a methodology that enables the evolution of a log management infrastructure. It enables it to adapt to the threat landscape, and be optimally designed for the log management requirements and risk affecting it.

2 Designing a log management infrastructure

The collection of log data is a common task for organizations for security related or other reasons. The need to secure the business processes as well as the need to comply to legislation and standards has increased the demand for log management and the corresponding requirements. Security incidents, ranging from a corporate policy violation to the Advance Persistent Threats (APTs), rely on the availability of specific log data extended both in time and detail in order to be detected and handled by the security personnel. Log management is a demanding task for large organizations, since the log data from various heterogeneous and geographically dispersed devices, needs to be collected and properly managed, to leverage their value. Surveys conducted among organizations [10] result in that organizations collect log data, identifying their prepossessing and correlation among the difficult tasks to perform.

2.1 Background

A framework for the design and implementation of a log management infrastructure is presented in [11]. It provides a high-level viewpoint of log management technologies and aims to assist organizations to understand the need for security log management. It provides a guide for the development, implementation and maintenance of log management practice throughout the organization. It covers several topics such as the establishment of log management infrastructures, and the development of log management processes, though it is not a step-by-step guide for the implementation or usage of the corresponding technologies. The publication is composed of four sections. The first section is an introduction to computer security log management, documenting the need for log management, the usefulness of logs and the corresponding challenges. Architectural decisions and log management software are discussed in section two, namely syslog based infrastructures and SIEM systems. A log data collection framework is proposed in [12], composed of four steps. It starts with the identification of the threats affecting an organization, prioritizing

them based on the resulting risk. It continues with the identification of the data sources that are necessary to technically address those threats, and the detailed analysis of the selected ones. Having defined the threats posed to the organization a detailed understanding of the infrastructure is aimed at, through the examination of the underlying technologies and the engagement of the personnel involved it the business processes. These threats are prioritized following a risk assessment process that estimates the impact and the probability of a threat affecting the confidentiality, integrity and availability of critical systems previously identified. Following, issues concerning the data sources are discussed, such as the storage, processing and administration overhead, the provisioning of adequate hardware and data retention requirements. A cost-benefit analysis of the introduced costs and the value added, result to the selection of the necessary data sources. A high-level guide for building a log analysis system is also available in [13], composed of five phases. The planning phase, where the system requirements are documented, the software selection phase, discussing the selection of the adequate software tools, the policy definition phase, where routines and procedures are defined and result to the system architecture, and the final phase, addressing scaling issues of the log management infrastructure. Compliance needs, use cases and other requirements are identified during the planning phase, roles and responsibilities are assigned to the personnel; goals are set, the data sources are identified and the necessary resources are estimated. The software selection phase resolves on the selection among open source software, commercial software or managed services, for the implementation of a log analysis solution, proposing evaluation criteria. The policy definition phase, deals with the establishment of procedures for log aggregation, protection, retention and review, among others. Four log analysis deployment models are presented in the architecture phase, and factors affecting the ability of the log analysis to scale are detailed in the final phase. In [14] a module for the acquisition and transmission of data is presented focusing on networks security monitoring. A distributed and multi-protocol supported network security monitoring system is proposed, describing its architecture. The authors focus on the system acquisition layer of the proposed system, designing two methods for the collection of monitoring data, namely the syslog-based collection and the real-time traffic-based collection. Log management is discussed in conjunction with commercial SIEM systems in [15], where the authors propose a log management architecture with common functions that are used by vendors. Log collection, normalization, categorization, queue prioritization and events storage by sensor, are identified as the main functions, and a suitable architecture that can send a normative, synchronized and prioritized log in an efficient way, is proposed. In [16] the authors present a comprehensive list of systems and techniques for log analysis. The authors study over 200,000 log analysis queries

from the Splunk [17] data analytics platform, quantitatively describing log analysis behavior to inform the design of analysis tools. Their work includes machine based descriptions of typical log analysis pipelines, cluster analysis of the most common transformation types, and survey data about Splunk user roles, skill sets and use cases. They result that log analysis mainly involves reformatting, filtering and data summarization, and that non-technical users decision making increasingly relies on data from logs. In [18] the authors propose a method that can guarantee the completeness of logs when transferred over untrusted networks. Their aim is to provide log forensically sound and adequate as evidence in a law court. They design a log management system composed of the following layers: the collection and storage of raw logs layer, the database liaison layer, and the log analysis and reporting layer. For the collection and storage of the log files they use software implementing the syslog [19] protocol and a hierarchy of directories is created for their storage. The integrity of the transferred log file is assured using hash algorithms to verify its integrity prior and after being transferred, while they leverage the Virtual Large-Scale Disk (VLSD) toolkit ([20] as cited in [18]) to construct a large-scale storage. They also propose a new log format that can be searched transversely, and discuss functions related to log collection and storage, though their focus is on the forensic soundness of the transferred log data. The delegation of log management to cloud is presented in [21] and [22] discussing the achievement of data properties such as availability, confidentiality and privacy. In [21] the authors propose a cloud based log management system, using adequate methods and protocols to strengthen the execution of queries on a logging database in the cloud. An homomorphic encryption scheme is employed to enable the generation of a query, that can be executed by a third party, without revealing the underlying algorithm or the processed log data, thus securing the log data in the case of a cloud based log management system. In [22] a system composed of three modules is proposed, for the delegation of log records to the cloud. It handles the log file preparation for secure storage, aggregating and encrypting the log files, the management of encryption keys proposing a secret sharing algorithm, and upload, retrieval and deletion of the log data is performed using tag info. A distributed intrusion detection system is presented in [23] aiming to the providers of cloud services. It is a hybrid and hierarchical approach for event correlation in intrusion detection in cloud computing. It uses distributed security probes to collect diverse information at various cloud architectural levels, in order to detect symptoms of intrusion. It uses a complex events processing engine to analyze event, while a knowledge-base represented by an ontology is used to identify the cause and target of an intrusion. A prototype of the proposed intrusion detection solution is also presented.

Social network analysis is based on the assumption of the importance of relationships among interacting units. The social network perspective encompasses modes, theories and applications, that are expressed in terms of relational concepts or processes; a fundamental component of network theories are relations, as defined by linkages among units. Actors and their actions are not considered as autonomous units rather as interdependent, the linkages among them as flows of material or non-material resources, and social, economic and other kinds of structure conceptualized as patterns of relations among actors.

In [24] the authors provide various measurements and techniques for the analysis of social networks. The structural and locational properties of nodes is performed employing the measures of centrality, while the cohesiveness of groups of nodes is measured using concepts such as *cliques*, *clans*, *clubs*, *k-plexes* and *k-cores*. In [25] the authors focus on the exploratory analysis of social networks and provide various techniques for studying groups' cohesion (cohesive subgroups, affiliations, sentiments and friendship), brokerage (center and periphery, brokers, bridges and diffusion), as well as node ranking. The inefficiency of the centrality measures is discussed in [26], [27] with the authors proposing more complex methods for the identification of the key player in a social network. In the former the identification of the key player problem is separated to two sub-problems; first, given a social network a set of nodes that if removed would maximally disrupt communication among the remaining nodes; second, given a social network, a set of nodes that is maximally connected to other nodes is aimed to be identified. In the latter, a procedure for the identification of the key players is presented, based on the assumption that the optimal selection depends on the what the nodes are needed for. Two cases are discussed; first, the diffusion of something in the social network where the key players are used as seeds; second, the disruption or fragmentation of the network removing the key player nodes.

The development and implementation of a performance measurement program is presented in detail in [28]. It is a guide for the development, selection and implementation of information system-level and program-level measures, to indicate the efficiency/effectiveness and impact of security-related activities. It can assist an organization to specify the adequacy of security policies and procedures, as well as in investment decisions. It details the measurement development and implementation process, and how those measures can support risk-based decisions and justify security related investments. In [29] the definition and application of security metrics is presented in detail, discussing the definition of criteria for the evaluation of metrics and the business needs driving their adoption, as well. They are used both to diagnose technical problems, such as vulnerability

management, password quality and patch latency, and for the measurement of high-level security activities. Analysis and visualization techniques are also discussed, along with the automation of the collection of the necessary data and the performance of the calculations, concluding with the compilation of scorecards, to provide holistic view of the organizational security effectiveness.

2.2 Methodology for designing a log management infrastructure in WANs

The proposed methodology consists of eleven steps. It starts with the capture of log requirements, the creation of an assets inventory and of the network topology diagram. Following these, what needs to be logged is defined and implementation architectures are examined. The log management infrastructure is divided into two tiers. The first is the log generation and issues concerning the log sources are addressed. The second is the log collection and storage tier which breaks down into the log collection and the log storage sub-tiers, which address the placement of the logging equipment and the log data life-cycle management respectively. The methodology continues with addressing time synchronization, preprocessing and scalability issues. Security considerations are addressed were applicable, with the aim of embedding security to the design of the log management infrastructure. Finally, a performance measurement program is used to measure aspects of performance and to decide upon corrective actions, if needed. Figure 2 summarizes the steps of the proposed methodology and the sequence of their execution.

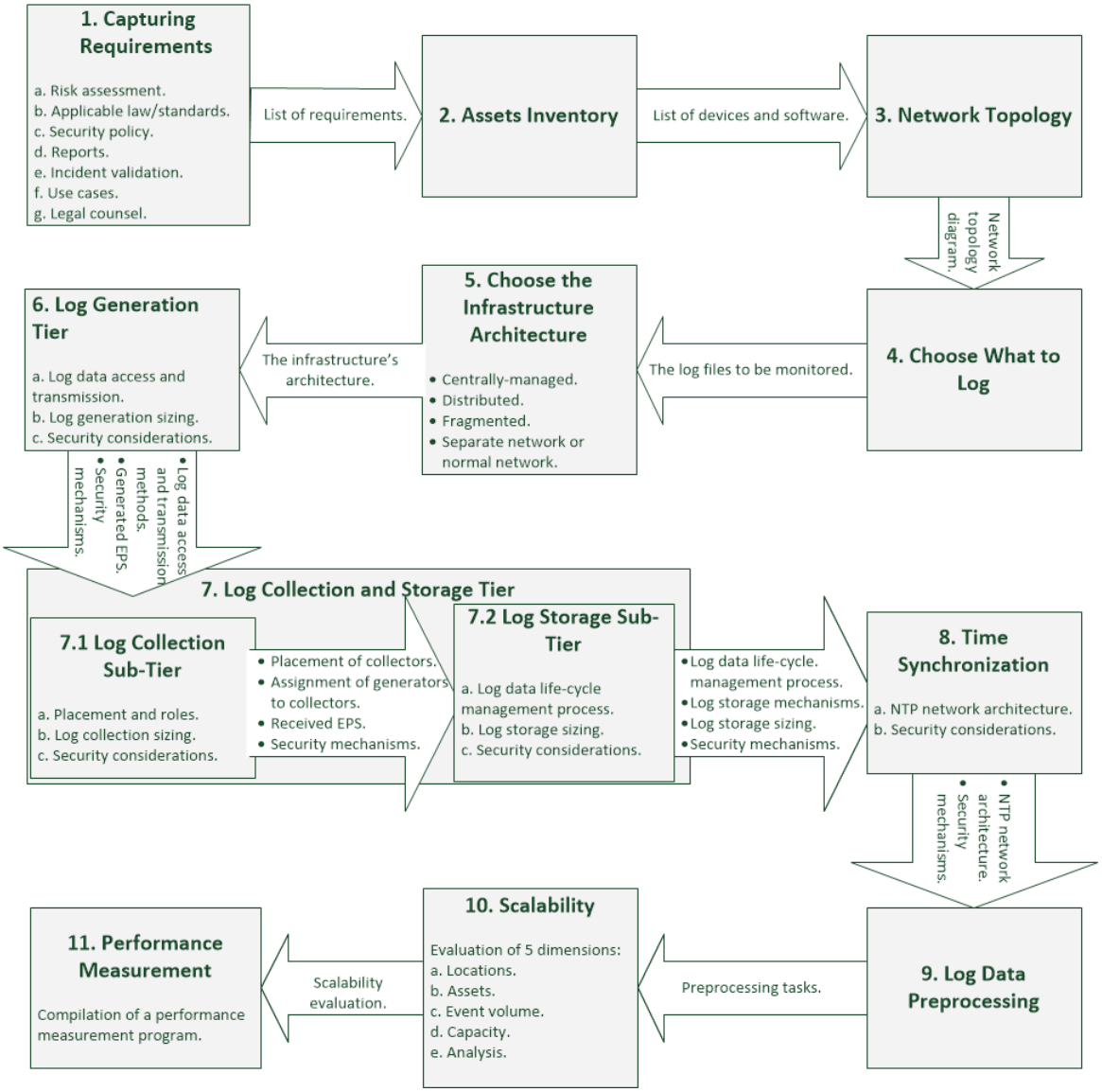

Figure 2: Proposed methodology block diagram

2.3 Capturing Requirements

The methodology starts with the capture of the log management requirements. Although these depend on many factors, including of course the aims and the type of the organization, the following are applicable in most cases:

1. *Applicable law and standards*: The applicable law and standards depend on the country and the type of the organization. The Sarbanes-Oxley (SOX), Health Insurance Portability and Accountability Act (HIPAA), Gramm-Leach-Bliley Act (GLBA), Federal Information Security Management Act (FISMA) and Payment Card Industry (PCI) Data Security Standard

(DSS) are encountered often in the United States [11]. An organization may have to be compliant to more than one of the above depending on its activities and on the services it provides.

2. *Incident validation*: A security incident may not be detected for a long period of time. The log records are the only evidence of its existence and they have to be adequate to track it both at host and network level [30].

3. *Reports*: They are a useful means of reviewing large amounts of log files, gaining insight into what is “normal” and detecting the “abnormal”. The specific reporting needs of the organization will indicate what has to be logged and in what detail.

4. *Security policy*: The security policy of the organization sets the barriers of acceptable personnel behavior and resources usage. The generated log files have to provide the necessary information to monitor users' compliance and hold them accountable in case of violation.

5. *Use cases*: The security events that are anticipated or have affected other organizations can be used for the compilation of use cases and the extraction of log requirements.

6. *Risk assessment*: Estimating the risk of each asset [31] is valuable for the determination of the systems’ criticality and, consequently, for the derivation of logging requirements.

7. *Legal counsel*: In some cases the logging activities may include the collection of personal or sensitive data. In this case legal advice is required to ensure that no privacy or security issues will arise.

Depending on the organization it may not be necessary to include all of the above sources of requirements. Only the applicable ones can be selected, enriched or otherwise adjusted to meet the specific needs of the organization. The output of this step is a list of requirements that the log management infrastructure must satisfy.

2.3.1 Assets Inventory

The full spectrum of the devices used by the organization is recorded including network devices, security devices, servers, desktop computers, mobile devices as well as network enabled printers

and scanners. It is crucial to include standalone devices too; even though these are not connected to the network, files may be transferred using removable storage media. The type of each device, the operating systems, the services and the installed applications are details of interest and should be recorded, as each of them is a potential log generator. The output of this step is a detailed list of the different types of devices and their multitude, along with the installed software.

2.3.2 Network Topology

The position of each device and the connections among them are recorded to create the network topology diagram. Standalone devices or devices that connect occasionally are also included, since their log files can be transferred on demand. The output of this step is the network topology diagram enriched with the standalone devices.

2.3.3 Choose What to Log

The selection of the devices that should be logged is not a trivial task, taking into account the verbosity of the log files. Ideally, everything should be logged as the needs of a future forensic or other type of investigation cannot be safely predicted and the absence of the necessary data could impede them, but the multitude of the log generators and the amount of log files they generate can lead to huge log files that are very demanding in computational and storage resources. Thus, logging everything is considered a non-realistic approach as it largely increases the storage, processing and network requirements [11]. In order to determine what needs to be logged, the proposed methodology modifies the process of selecting a SIEM system described in [32], abbreviated as Top-Down Bottom-Up Middle-Out (TDBUMO).

Top down: The information available from the assets inventory is organized in a tree structure. The root node is empty, even though it could be a log analysis or SIEM product. The system types (network devices, desktops, etc) are grouped in the next level of the tree and are further detailed at the following tree level, specifying the versions of these systems (Windows 8, RHEL 7, etc). At the last tree level, each leave is the type of the log files that the parent node (e.g. operating system) can generate. This provides an understanding of the different types of log sources and the data flow

inside the log management infrastructure. The output of this step is a tree structure where the leaves contain all specific types of logs that can be generated by the available devices.

Bottom Up: Starting from the leaves of the tree each specific log file is documented in detail. The logging levels (verbosity), the format of the records and the ways of accessing or retrieving these log files (agent, agent-less) are recorded. The record format of each log file is crucial, as the contained fields will indicate whether the necessary information is available and whether these fields can serve as the common element for their correlation with the log files of other devices. The output of this step is a detailed document with the specifics of each log file (file name, log levels, log format, record fields, method of access, etc).

Middle Out: A matrix is composed, that has the log requirements as rows and the specific types of logs as columns. Each log file is mapped to a log requirement adding the required log format and the desired log level (verbosity) to the corresponding cell. Of course, more information may be added to the cells, according to the needs of the analysis.

The output of the TDBUMO process is the list of log files and their specifics that are necessary for the fulfillment of the log requirements. These files are of interest to the log management infrastructure and have to be monitored.

2.3.4 Choose the Infrastructure Architecture

In this step of the proposed methodology the log management infrastructure architecture is selected, based on the findings of the previous sections, the budget constraints, the goals of the organization and its future plans and needs. A log management infrastructure is composed of hardware, software, networks and media that generate, transmit, secure, store, analyze and dispose log data [11], [12], [13]. It consists of the following basic tiers:

- *Log generation*: This tier includes the devices that generate data, the log generators. These data are made available to the log servers of the following tier running an application (agent), a service (e.g. syslog) or allowing servers to remotely access the log files (agentless).

- *Log collection and storage*: This tier includes the log servers that collect the log data that are sent from the log generators. These servers are called collectors or aggregators. The transmission of the data can be performed either in real-time or in batch mode. The collected log data can be stored in the collectors or in separate storage systems.

- *Log monitoring*: This tier is composed by the user interface that is used to review the log files, to perform analysis and generate reports, as well as to administer the log generators and the collectors. This tier is out of the scope of the proposed methodology, but it is included in the design process to facilitate the integration of the resulting log management infrastructure with log analysis or SIEM products.

Concerning the architecture two major design decisions have to be made. The first is whether the log management infrastructure will be centrally-managed, distributed or fragmented. With a centrally-managed infrastructure the organization can review all the available log data; this is considered to be the only realistic approach in order to achieve accuracy and security. A distributed infrastructure uses separate infrastructures for each scope of the monitored infrastructure. The scopes can be defined based on the types of systems, the types of logs or the physical location of the equipment. These separate infrastructures may operate independently or inter-operate through a central management point. This architecture can flexibly adapt to organizational changes and is proposed for large organizations [11]. Following a fragmented architecture, each department, network, system or even log generator, implements its own log management solution and only the corresponding administrators have access to the log data. The application of policies and processes is not feasible and there is no efficient way for the organization to compose an overall picture of its network.

The second decision that has to be made is whether the transmission of the log data will be performed over a separate and dedicated network or over the normal network of the organization [11]. Using a separate network (physically or logically) has performance and security advantages. Handling a security incident is more efficient over a fully functional log network than over a performance degraded one, due to malware infection for example. Since the cost of implementation is a drawback, the separate network can be used for the connection of the critical devices, operating the non-critical ones over the normal network [11]. Using only the normal network does not have any advantage, apart from the lower cost, the ease of configuration and maintenance. The above approaches can be combined based on the specific needs of the organization.

The output of this step of the proposed methodology is a decision on the architecture of the log management infrastructure and the type of network to be used.

2.3.5 Log Generation Tier

The log generation tier includes the log generators that were selected during the TDBUMU process and the issues that are addressed are the log data access and transmission, the log generation sizing as well as security considerations.

Log data access and transmission: One approach for accessing log files is the installation of an agent on the log generator. The agent parses and transmits the data to the collector without intervention of the hosting system. Disadvantages of this approach are the need to update its parsing features, to support new applications or log formats, as well as the increased administrative overhead. Advantages are the automation of the transmission and normalization of the log data. Accordingly, this option is proposed for servers and workstations [33].

An agent-less approach has the advantage of not requiring the installation of any additional software, thus having small impact on the systems and the administrative personnel. The server usually authenticates to each host and the log data are then pulled. A disadvantage of this approach is that no log filtering or normalization is performed on the individual host, increasing the amount of data that is transferred and processed on the receiving server. This can prove to be system-intensive and slow across WAN connections. In addition, it introduces the need for a credentials management scheme, since the receiving server has to authenticate itself to each of the log sources [11].

A popular means of accessing and transmitting log data is the syslog protocol [19], which is pre-installed in most Linux distributions and is supported by network devices. Some popular implementations of the syslog protocol are dsyslog, rsyslog, syslog-ng and Kiwi server. Rsyslog is currently the default implementation in Debian and RedHat based Linux distributions. It supports the transmission of messages over UDP, TCP, Reliable Event Logging Protocol (RELP) and Transport Layer Security (TLS), as well as storage of the log data in relational databases. Additional advantages are its filtering, relaying and caching features. The UDP protocol is not recommended, as it does not provide reliable delivery. The TCP protocol provides reliable delivery but with some

caveats due to its model of operation [34]. On a log collector, when TCP receives and buffers the messages, it reports success to the client. In case of a system crash the syslog messages stored in the buffer are lost, although the client considers them sent. The RELP protocol is designed for reliable log delivery [35], but it is not mature or widely tested yet. Logs generated from routers and switches can also be accessed via SNMP traps, but the usage of syslog is encouraged due to its verbosity and its ability to identify more exceptions and degradation warnings in a network, as compared to SNMP [36].

Log generation sizing: In this step the volume of the log data, which will be sent from the generators, is estimated. A common and consistent metric used for this task is the Events Per Second (EPS) metric, which is defined as the number of events a device can generate or receive in a second [33]. The EPS is estimated for each category of log generators, as they were grouped in the TDBUMU Top-Down part of the methodology. Depending on the available time or other type of constraints, the log files can be collected from all the participating devices or from a single device for each category; in the latter case the estimate is accepted as valid for the whole category. The average and the maximum EPS are calculated. Depending on the amount of the available historical data, these can be mined for patterns, e.g. working hours versus non-working hours or working days versus weekends etc. For large data sets the application of data mining algorithms and statistical analysis provides valuable results. The following formula is applied twice, once for calculating the average and once for the maximum number of events:

EPS = number of events / time period in seconds

Depending on the transmission method, e.g. syslog protocol, multiplying the EPS with the maximum message length (2048 octets for syslog [19]), the average and maximum bandwidth is estimated using the following formula:

Bandwidth= EPS x (size of event)

The event size can also be estimated from the log files already present on the log generator. The maximum record length, found in the log files, and the average one can provide a good estimation of what is usually generated by the specific log source. This process can only result into estimates, as log records vary in length and the log generation rate usually oscillates. In addition, the underlying protocols (Layer 4,3,2) add overhead to the transmission and additional protocol

specific functions, such as IP fragmentation, add overhead and increase the amount of data that is finally transmitted over the network and form the value of the used network bandwidth.

Security considerations: Physical access has to be restricted. Privileges of system users must not allow the deletion or the modification of the log entries, limiting them to addition. In case log files are stored locally on the log source, their integrity and availability must be protected. The log files should be regularly backed up to ensure their availability and recovery.

The output of this step is a decision on the method that will be used to access and transmit the log data, the estimates of the EPS that will be sent and the bandwidth that will be used, as well as the means of securing the log generators.

2.3.6 Log Collection and Storage Tier

2.3.6.1 Log Collection Sub-Tier

In this tier two approaches can be followed, namely the syslog-based and the vendor specific. The vendor specific approach depends on the product and its architecture. They are usually composed of agents installed on the log sources and the log data is sent to a central collection point. The central collection point is divided into an analysis engine, a storage mechanism, the agents' manager and the user interface. A hybrid approach is also feasible, since most vendors support the syslog protocol in their products.

Following the syslog based approach, the second tier is composed of the log servers that receive the log data. A log server can be assigned the role of originator, cache/relay or collector/aggregator [11], [19]. An originator generates log files concerning its function, which can be locally stored or transmitted to one or more destinations. A cache/relay log server collects data from log sources and forwards them to other log servers. They can be used to protect the collectors from traffic peaks or alleviate the problem of network bottle-necks. A collector/aggregator receives and stores log data either locally or on separate storage media and is usually mapped to a group of originators and/or cache servers.

This is a flexible tier; the addition of more sub tiers is feasible, by combining the aforementioned server roles as follows [11], [33], [37]:

1. Multiple log servers, each performing a specific task. One server may be used for log analysis, another one for live/production data and one for archived data

2. Multiple log servers performing analysis and/or storage for a category or specific log generators. One server could analyze and/or store router log files, while another could perform the same tasks for server systems. Adding such levels benefits the infrastructure in terms of redundancy, as a log generator can switch to a backup server in case of system or communication failure

3. Two or more levels of distributed log collectors that preprocess or simply forward the log data to a next tier of more centralized collectors. A tier of caching servers can be included to protect the infrastructure in case of a traffic peak or an attack, as well as to alleviate the problem of low network performance. This approach adds flexibility, scalability, and redundancy to the log management infrastructure.

Placement and Roles: The placement of the various components of the log management infrastructure is determined. As the components of each commercial product depend on the vendor, the proposed methodology elaborates in syslog implementations. Nonetheless, an agent can be considered as an originator and the component that collects the events can be considered as a collector, thus fitting the architecture of the specific product to the proposed methodology.

The architecture of the log management infrastructure has already been decided upon in a previous step, imposing restrictions and placement criteria. The placement criteria that are considered by the proposed methodology are the following [36]:

- *Geographic location*: For a WAN that extends across multiple locations the placement of a collection point to each region is proposed.

- *Collectors close to their originators*: The collectors have to be placed close to their originators. This is desired as a network problem or the spread of a malware infection could disrupt the log collection process.

- *Hub-and-spoke architecture*: Many originators forward the log data to a collector and many collectors forward them to a central point or in the case of multiple layers to a central collection point at the following layer.

- *Hierarchical*: The placement of the components has to follow a hierarchical fashion starting from the originators and ending to the central collection point, where all the log data are collected and processed by the organization.

To fully and effectively address the above criteria one needs to analyze both the structural properties of the network and the patterns of interaction among its actors; an appropriate technique that allows both is SNA [24], [25]. In SNA, a *node* (or actor) is a social entity. It can be a discrete social unit (an individual) or a collective social unit (a group of people, a corporate department, etc). Though termed actors, it is not implied that they have the ability to act. *Links* (or social ties) connect actors establishing a tie between a pair of actors. A *relation* is the collection of a specific kind of ties formed among the actors of a specific set of actors. Social networks are composed of nodes and links, that can be directed or not. A node can have one or more attributes and a link can be binary or valued. Using graph theory notation, $G=(V,E)$ is a social network G with $|V|$ nodes and $|E|$ links among them. It is represented by a $|V| \times |V|$ adjacency matrix, where the existence of a link between node $v_i \in V$ and node $v_j \in V$, is indicated by a value in the $e_{ij} \in E$ cell.

The social network is constructed based on the network topology diagram. The log sources are the actors/nodes and the network connections among them are the social ties/links of the social network. One relation is constructed resulting in the respective adjacency matrix, where the presence or the absence of a link among two nodes is denoted by a binary value in the corresponding cell (1 for present and 0 for absent).

The analysis of the social network aims to identify the important nodes of the network based on the above defined placement criteria and the SNA theory. Though SNA can analyze many aspects of a network, in this step of the methodology only the identification of the important nodes is required. A node is identified as important when it is close to other nodes (geographic location and closeness to the originators) and when it connects many other nodes (hub-and-spoke fashion). The aim is to place the collectors at specific locations where they will be close to and easy to connect to as many originators as possible. For the purposes of the analysis, the total degree, the closeness, the eigenvector and the betweenness centrality [38] are selected among the available centrality measures.

Total degree centrality: The degree centrality is used to identify the nodes that actively participate in the social network. It is the number of links a node has. It is distinguished into in and out degree, when the links are directed to or from the node, respectively. The Total degree centrality

of a node is its normalized in plus out degree. Let $G = (V,E)$ be the graph representation of a square network and a node v. The Total degree centrality of node v = deg $/ 2 * (|V| - 1)$, where deg = card $\{u \in V | (v,u) \in E \text{ V } (u,v) \in E\}$ ([24] as cited in [38]). A node with high degree centrality is a well-connected node and can potentially directly influence many other nodes [26].

Closeness centrality: The closeness centrality highlights the nodes that are close to other nodes and can thus easily and quickly interact with them. It is the average geodesic distance of a node from all other nodes in the network. The geodesic distance is the length of the shortest path between two nodes. Let $G = (V,E)$ be the graph representation of a square network, then the closeness centrality of a node $v \in V$ is $v = (|V| - 1) /$ dist, if every node is reachable from v and $v = |V|$ if some node is not, where dist = $\Sigma d G(v, i), i \in V$ ([39], as cited in [38]). The closest a node is to others, the fastest its access to information and greater its influence to other nodes [40].

Eigenvector centrality: It is a measure of the node's connections with other highly connected nodes. It calculates the eigenvector of the largest positive eigenvalue of the adjacency matrix representation of the square network. To compute the eigenvalues and vectors, a Jacobi method is used ([41], as cited in [38]).

Betweenness centrality: The betweenness centrality is used to identify the nodes that hold a critical position and can consequently affect the social network if removed. A node with high betweenness is important because it connects many nodes and a possible removal would affect the network. Betweeness centrality is defined, for a node v, as the percentage of the shortest paths, between node pairs, that pass through v. Let $G = (V, E)$ be the graph representation of a symmetric network. Let $n = |V|$ and a node $v \in V$. For $(u, w) \in V \times V$, let $n_G (u, w)$ be the number of geodesics in G from u to w. If $(u, w) \in E$, then set $n_G (u, w) = 1$. Now, let $S = (u, w) \in V \times V | d_G (u, w) = d_G(u, v) + d_G(v, w)$ and let $between = \sum(n_G (u, v) \times n_G (v, w))/n_G (u, w), (u, w \in S)$, then the betweenness centrality of node $v = between/((n - 1)(n - 2)/2)$ ([39], as cited in [38]).

It should be noted that adding to or modifying the placement criteria could result to additional measurements and analysis; nevertheless, the above metrics are an efficient measure of the nodes' importance [26].

Having performed the centrality measurements, the nodes are sorted based on their total degree centrality in descending order. When a node is highly ranked it means that it has many connections with other nodes with distance one, i.e. one hop away. Placing a collector at this node's location is

adequate, as it is close and directly connected to many log sources. A cache server could also intermediate at this node between the log sources and the collector. The fact that many log sources are adjacent to this node indicates that high traffic and high EPS rate is expected. The analyst then identifies where the total degree centrality decreases suddenly. This sudden drop can be used to separate the nodes into layers. The low-ranked nodes could form a layer that would forward its log data to a higher-ranked node, in a hub-and-spoke mode. The same process is repeated for the closeness centrality. When a node is highly ranked it means that it can be reached from other nodes with a few hops. A highly ranked node location is appropriate for a log collector, since log sources are close to it and a device failure, that would make some network paths unavailable, is less likely to affect it. On the contrary, placing a log collector to a node location distant from the log sources would increase the risk of failing to deliver the log data. The administrative overhead is also less, avoiding the modification of firewalls or other devices that may otherwise need to be reconfigured to allow the log data to flow through the network. The process is repeated once more for the eigenvector centrality. A highly ranked node is a node that has a lot of links with nodes that are well connected too. A node with low eigenvector centrality is connected with nodes that have few connections. The location of a node with high eigenvector centrality is a suitable location for the placement of a collector or for the central collection point.

At this point the locations where the collectors, or the cache servers, could be placed have been identified. The analysis continues with the identification of the nodes that if removed would increase the number of the social network components (maximal connected sub networks), the “boundary spanners” (or gatekeepers) [38]. These nodes hold a critical position in a social network as their removal would result into sub-networks that do not link to each other. The value of each node is calculated as the ratio of the betweenness centrality to the total degree centrality of the node. Nodes with high betweenness centrality and low total degree centrality are identified as boundary spanners. If a router is a boundary spanner and for some reason fails to route the traffic, the result will be the partitioning of the log sources to sub-networks unable to communicate outside their subnet. As a result, a collector placed in the same location would fail to communicate with its originators. Placing the equipment of the log management infrastructure on such nodes should be avoided. Although the SNA measurements do not mandate the placement of the components of the log management infrastructure, they provide a means of identifying the important nodes and the respective locations in the WAN. The measurements can be analyzed independently or be combined to form new metrics.

The methodology continues with the assignment of originators to collectors or cache servers. The originators have to be divided into groups, were each group will forward its log data to the same collector or cache server (more than one destination can be assigned). To achieve this, the Newman algorithm is employed [42]. It is an agglomerative hierarchical clustering algorithm for detecting community structure in large networks. At the starting state of the algorithm each node is the only member of a community. At each step, the communities are repeatedly joined into pairs choosing the join that results in the greatest increase in modularity. *Modularity* (*Q*) is a network property proposing a specific division of that network into communities. If the division is good many links will exist among the community nodes and only a few links will exist between the communities. In [42] the modularity is defined as

$$Q=(1/2m)\,\Sigma_{vw}[A_{vw} - k_v k_w/2m]\,\delta(c_v,c_w)$$

where $k_v k_w/2m$ is the probability of an edge existing between nodes v and w if the links are formed at random, but respecting the nodes' degrees. $\mathbf{A_{vw}}$ is the adjacency matrix of the network, with $\mathbf{A_{vw}}$=1 if nodes v and w are linked and $\mathbf{A_{vw}}$=0 otherwise. The δ-function $\delta(i,j)$ is 1 if $i=j$ and 0 otherwise and c_v denotes that node v belongs to community c_v. If the fraction of the links inside the community is the same as for the randomized network, then Q=0. High values in modularity indicate good division of the network into communities. The progress of the clustering can be represented as a tree that shows the ordering of the joins.

The grouping that the Newman algorithm outputs assists the analyst to assign the originators to collectors or cache servers, as well as to validate the placement, by checking whether a collector has been placed close to each group. On the other hand, having placed many collectors into a small group may indicate an error in the placement.

Log Collection Sizing: The amount of log data that each originator transmits is already available as the output of the log generation tier, thus the estimated EPS that each collector will receive and/or transmit can be calculated. A collector that has been assigned five log originators will receive the sum of their corresponding EPS. The bandwidth calculation has been addressed in a preceding step of the methodology and the necessary volatile memory can be calculated as follows:

Volatile memory= EPS x event size

where event size is the size of each event that is transmitted or received by the collector. In the case of a syslog implementation this can oscillate from the minimum to the maximum syslog packet size. Depending on the log management solution and the configuration options, more parameters can be added to the formula. With rsyslog, for example, queues can be configured and volatile memory can be allocated to them. Based on this estimate the necessary hardware can also be specified.

Security Considerations: When the syslog protocol was designed, security was not a concern or a high priority. The transmission over UDP has the disadvantage of unreliable delivery. The syslog messages are transmitted in clear text over the network and neither the sender nor the receiver is authenticated.

The unreliable delivery can be alleviated by transmitting over TCP instead of UDP. The confidentially, integrity and authenticity of the messages can be protected by employing the TLS protocol. Each sender and receiver has to install its own certificate to verify its identity and perform the required encryption functions. Nevertheless, even in this way the traffic remains exposed to traffic analysis and adds the administrative overhead of key management and devices' configuration and maintenance. The implementation of VPNs is a feasible solution, while some implementations provide the feature of sending the syslog messages through Secure Shell (SSH) tunnels. The use of TLS or SSH increases the bandwidth consumption and the time needed for the transfer of the data. The availability is also protected using TLS or SSH, since an illegitimate user can not send data to a receiver. In addition, configuring the receiver to have a rate limit or protecting it with a firewall, can pose restrictions to the receiving traffic and protect against some DoS attacks.

The output of the log collection sub-tier step is the placement of the log servers, the role that each one will be assigned, the assignment of originators to collectors and the log collection sizing (the EPS each server is estimated to receive), the required bandwidth and the volatile memory that is required. The necessary security measures for the protection of the log data are also included in the output.

2.3.6.2 Log Storage Sub-Tier

Log storage is a critical component of the log management infrastructure and the proposed methodology approaches it with the design of log data life-cycle management process. For such a process the data life-cycle stages, the storage mechanisms, the amount of log data and the functions that will be performed on the data need to be considered.

Log data life-cycle management process: The proposed life-cycle includes the following stages:

- *Production/live data*: Data are used for real-time analysis and on-going review.
- *Back up data*: Copy of the production data, intended to be used in case the former become unavailable.
- *Archive data*: Data that are kept in long-term storage for regulatory or forensic reasons or for the benefit of the organization, such as carrying out data mining or statistical analysis.
- *Disposed data*: Data that are no longer necessary for the organization and they are disposed. Depending on the sensitivity of the data and on the security policy, they can be simply deleted or securely removed to avoid their recovery and a possible data leakage.

Depending on the data access requirements, various storage mechanisms can be employed [13]:

- *On-line storage*: The data are stored in high-performance systems, such as NAS or SAN, where the access time is a few milliseconds. The data are available to a large number of users.
- *Near-line storage*: In these storage systems the access time is measured in seconds. The data are available for infrequent use to a small number of users.
- *Off-line storage*: The data are stored on external media. They cannot be accessed unless mounted to the system.

The above storage mechanisms can be implemented using databases or raw files:

- *Raw files*: They have the advantage of speed both in write and read operations and they can store the data in their original format. This is important to maintain the data in forensically sound condition. On the other hand, processing them is more difficult and a parsing process is usually necessary.
- *Databases*: They pose the advantage of facilitating the processing of the data. Each log record is divided into separate fields, each of which is stored into the respective database table columns. They are usually the bottleneck of the log management infrastructure, as the

database cannot insert messages at the same speed that the log server can collect them. Some of the syslog implementations include features that enable the configuration of message queues on the syslog server, where the log data are temporarily stored until the database becomes available again [43]. In addition, most databases offer different storage engines, some of which are suited more for writing than for reading transactions (e.g. the MySQL MyISAM storage engine [44]). Configuring clusters of databases is also a solution that alleviates the problem of high transaction rate as well as scalability problems.

Functions that can be performed on the log data are [11]:

- *Log rotation:* It is the function of closing a log file and opening a new one based on its size or time parameters, to keep the log files in manageable size.

- *Log retention:* Logs are archived on regular basis, usually as part of the standard procedures.

- *Log preservation:* Logs are archived due to special interest, like forensic or incidence handling reasons.

- *Log compression:* The log data are compressed to save storage space. Care must be taken to use lossless algorithms.

- *Log encryption:* The data are encrypted to protect their confidentiality. It is a recommended function when data are stored in an external device or are transferred by any means. Proprietary encryption algorithms should be avoided in favor of publicly known and tested ones (e.g. AES).

- *Log reduction:* The log records that are of no interest are removed to reduce the occupied storage space.

- *Log conversion:* The format of the log records is modified, e.g. converting relational databases data to XML files.

- *Log normalization:* The representation of the fields of the log records is altered to facilitate analysis and reporting.

- *Log file integrity checking:* The message digests of the log files are calculated to ensure the apprehension of an integrity violation.

Based on the organization's requirements and intended use of the log data, the data stages, the storage mechanisms and functions are combined to define the log data life-cycle management process.

Log storage sizing: Having defined the life-cycle management process the log storage sizing is estimated when required. The placement and the sizing of the log collectors is available from previous steps of the methodology, thus facilitating the estimation of the log storage sizing. Assuming a syslog collector is expected to receive an average rate of 1000 EPS of estimated event size 1024 bytes, this equals to 1000 * 1024 = 1024000 bytes per second. A requirement for 15 months of retention would result into 1024000bytes * 38 880 000 sec = 36.21 Terabytes of required storage.

Security considerations: The storage sub-tier needs to be adequately protected to ensure the confidentiality, integrity and availability of the log data. An attacker may gain physical access to the log storage and affect the data or remotely exploit a vulnerability of the storage system. Modification or destruction of the log data is feasible if access is achieved with specific user privileges. To mitigate this risk, the physical protection of the storage system has to be addressed and the operating systems and/or database servers have to be properly configured for access control. Encryption mechanisms such as hash functions, symmetric ciphers and digital signatures are effective in protecting the confidentiality and the integrity of the data. Their availability can be protected through a back up process and hardware redundancy. Systems security is not further addressed in the proposed methodology, as it should be part of the organization's overall security policy and mechanisms.

The output of the log storage sub-tier section is the log data life-cycle management process, the log storage sizing, the storage mechanisms and the necessary security measures.

2.3.7 Time Synchronization

A critical issue in the implementation of a log management infrastructure is the synchronization of the logging equipment as log correlation requires accurate and uniform timing to combine the log

data from the various sources and identify the events of interest. Processing the log data after their generation is not a realistic approach, due to their volume, the multitude and the variety of log generators. The recommended solution is the employment of time synchronization technologies such as the Network Time Protocol (NTP) [45] or the Precision Time Protocol (PTP) [46]. The PTP protocol is used for the precise synchronization of clocks in measurement and control systems that communicate using packet networks and are implemented using technologies like network communication, local computing and distributed objects. It supports accuracy in the range of sub-microsecond and requires minimal network and local clock computing resources. The devices are organized in a master-member hierarchy where all the members are synchronized with the master clock.

Both protocols are suitable for a log management infrastructure, although the NTP is preferred in the proposed methodology due to its wide and long usage on the Internet and the familiarity of the administrative personnel with it. The NTP is a widely used protocol to synchronize computer clocks among distributed time servers and clients and has potential accuracy of tens of microseconds. An NTP network usually gets the time from an authoritative time source such as an atomic clock. The NTP server with the attached authoritative time source forms stratum 1. It can be a public server or a private one and distributes the time to the following stratum. The concept of stratum defines how many hops away is a device from the authoritative time source. In WANs NTP usually achieves synchronization at 10 milliseconds and 1 millisecond at Local Area Networks (LAN). It avoids synchronization with possibly inaccurate machines by not synchronizing to a machine that is not synchronized itself and by comparing the time reported from more than one machines [45].

In [45] three modes of operation are defined for the NTP protocols:

- *Client/server Mode*: A client sends a request to usually more than one server and expects the answers.

- *Symmetric Mode*: In this mode the server both obtains and supplies time providing mutual back up; it is recommended for redundant time servers connected through diverse network paths.

- *Broadcast and/or Multicast Mode*: An NTP server can broadcast time in a subnet. This broadcast is restricted in the specific subnet, as it is not routed by the routers. Accuracy and

reliability are degraded, but administration is eased when a large number of clients are synchronized to a few time servers.

Combining these modes of operation, different NTP architectures can be formed [47]:

- *Flat peer architecture*: All the NTP servers peer each other, while some of them, geographically dispersed, synchronize to external systems.

- *Hierarchical structure*: The network routing hierarchy is copied and used for the NTP hierarchy. The top nodes (core servers) of the hierarchy synchronize to external systems and each stratum has a client/server relationship with its lower stratum. This is the recommended architecture to achieve scalability, stability and consistency.

- *Star structure*: All the devices have a client/server relationship with a few time servers in the core.

The proposed methodology uses again the already available SNA measurements to derive the NTP servers' locations. The closeness and the betweenness centrality are combined to derive the NTP strata. In the case of a flat peer architecture, the highest ranked nodes are connected to each other and some of them access a reference clock, while in the star architecture the highest ranked node or a few highly ranked ones will be accessed for time data by all others. If a hierarchical architecture is chosen, then the nodes need to be divided into groups to form the corresponding strata.

The nodes are sorted in descending order, based on their closeness and betweenness centrality. Those that are highly ranked in both metrics form stratum 1; selecting at least three nodes for this stratum is advised as best practice. Then the measurements are observed for sudden decreases in their values. These patterns are used to separate the nodes into strata. The results of this analysis are a means of assisting the analyst to map the nodes (log devices) into strata, even though the final decision is left to the analyst, since additional factors may need to be considered.

Security Considerations: The NTP security model treats the time values as public; the aim is not to hide the data but to authenticate the sources. Security in time servers involves the client and server authentication and the integrity of the time values. An analysis of possible attacks is available in [48]. NTP supports the implementation of general purpose Access Control Lists (ACL), the use of a symmetric key authentication scheme (from version 3) and the Autokey public key authentication

scheme (from version 4) [45], [49]. Employing symmetric or asymmetric encryption adds protection to the messages' authenticity; the confidentiality and the availability of the time information remain vulnerable.

The output of this step is the NTP network architecture; the locations and the devices where the NTP servers will be installed; their separation into strata; the modes of operation; and the security settings.

2.3.8 Log Data Preprocessing

The log data are generated from various types of devices and applications and various formats are followed by vendors and developers. Preprocessing tasks on the log data include data transformation, data filtering based on the facility or the priority, data aggregation of frequently appearing records, as well as data reduction when not all of the available data are necessary for the analysis tasks. The log data can be processed on the log source, on transit or while in storage. When a vendor specific approach is followed and an agent is installed on the logging devices, the log records are usually normalized prior to their transmission, thus performing the necessary preprocessing. An obvious drawback of this approach is the need for the agents to support all the necessary parsers, as well as the need to update those parsers every time the log originator alters its logging format.

The log files that will be managed by the log management infrastructure and their specific formats have already been defined in the TDBUMO process, thus facilitating the definition of the preprocessing requirements. Depending on the log generation, collection and storage sizing and the estimated overhead, the components of the infrastructure that will perform the preprocessing tasks can be selected. This can be performed on the log generators, on the log collectors or in the storage mechanism.

The output of this step is the preprocessing tasks to be performed and the corresponding devices.

2.3.9 Scalability

The scalability of the log management infrastructure needs to be evaluated since the organization may expand, modify its security policy or a security incident may drive the need for more accurate and voluminous log data collection. Scalability is evaluated against five dimensions:

1. *Event volume*: What is evaluated is the ability of the infrastructure to handle a large increase in the number of events. This can be a long term change, due to an expansion or change in the requirements or a temporary peak of the traffic, after a security or other unexpected event. The former could require the acquisition of additional equipment and the expansion of the log management infrastructure, while the latter could be accommodated with the reconfiguration of the present equipment.

2. *Assets*: Along this dimension the mechanism for compiling and updating the assets inventory is evaluated. The assets' modeling and the available attributes per asset type, such as known vulnerabilities and patch history, are considered.

3. *Locations*: Supporting logging in new geographic locations is a demanding task. The log collection, transfer and storage are obvious implications, as well as the administration and maintenance of these remote devices. Agent installation, agent-less access, configuration and software maintenance issues are considered.

4. *Capacity*: An aspect of the infrastructure is the storage capacity. The storage mechanisms are evaluated in terms of ease of expansion both geographically and in capacity. This is desired to be achievable at low economic cost and low administrative overhead. The entire life-cycle management process is included in the evaluation.

5. *Analysis*: The data volume and the correlation rules that can be processed in real-time are evaluated. An increase in the volume of log data or the correlation rules that need to be examined may result in inefficient analysis and may cause the infrastructure to miss its targets.

2.3.10 Performance Measurement

The methodology concludes with the development of performance measures to monitor certain activities and to apply corrective actions if needed. The use of measures benefits the decision making and facilitates the achievement of the defined goals and objectives. The type of measures depends on the maturity of the program; a long running program is accompanied by refined processed and procedures, by documentation and historical data, as well as measurements/metrics collection mechanisms. The methods for collecting the measurement data should not be intrusive and due to their sensitivity they should be properly managed. In [28] the measures are categorized into implementation, effectiveness/efficiency and impact measures.

1. *Implementation measures*: They are used to monitor the progress of a task usually as a percentage.

2. *Effectiveness/Efficiency*: They are used to measure if an action is implemented correctly and if it results in the desired outcome.

3. *Impact measures*: They measure the impact to the organization by a security control, policy or other related task.

In order to develop the measures that will be used to measure the performance of the log management infrastructure the following process is proposed [29]:

1. *Definition of goals and objectives*: The goal of the measurement program is stated and it is broken down into objectives that guide to the achievement of the goal. The people that will review the measurements/metrics and will handle the decision making participate in the goal and objectives definition.

2. *Measures/metrics development and selection*: Through the development process the measures that apply to an individual action can be defined, as well as the ones that apply to the whole program. In the former case the measures are mapped to the specific action, while in the latter they are mapped to the goal or the objectives. To develop the measures, a top-down or a bottom-up approach can be used. Following the top-down approach for each program objective the measures/metrics are generated; when choosing the bottom-up approach, the

measurements/metrics that are already available or easy to generate from the monitored tasks are examined to assess whether they are adequate for the program objectives.

3. *Targets and benchmarking*: The success of an action or objective is indicated by the achievement of the related measurement/metric targets. An approach is to set the target for a metric/measurement according to a baseline and an alternative is not to set the target until the measurement/metric is run at least once. Benchmarking can provide comparative and meaningful results through the comparison of the organization's performance to best practices and other organizations' practices.

4. *Data source and collection:* The data that will be used for the measurements/metrics are selected and their sources and access methods are defined.

5. *Program Review / Refinement*: The metrics / measurement program is reviewed to determine the accuracy of the measurements/metrics, the effort they require and the value they add to the organization.

2.4 Case study

The methodology was applied to a real network, the Greek Research & Education Network (GRNET.SA network) [50], as it was on November 2015. The log requirements were assumed, and included, among others, the logging of the execution of privileged actions, invalid access attempts and security system events, as well as the protection of the log data, specific retention periods and the need for scalability. The method for compiling the assets inventory and the network topology diagram is left undefined in the proposed methodology. For this case study the relevant data were publically available on the organization’s web site. The assets inventory was composed of 17 routers and 51 switches from various vendors [51], [52], [53] connected with 75 links. The network connects all major cities in Greece and was composed of partial MANs. The TDBUMO process was applied. This proved to be a time consuming process, since details of all log files that can be generated by the participating devices had to be recorded and associated with the requirements; the advantage is that the process specifies effectively the necessary log files, the verbosity level and the log format.

The geographic dispersion of the GRNET.SA network and MANs that were identified from the network topology diagram, led to the architectural decision to combine distributed collection points to a centrally managed one, where all the log data would be available for analysis. Due to the extent of the network and the (assumed) limited budget, only the critical equipment was decided to be connected over a separate logical network, while the rest of the log data would be transmitted over the normal network.

Moving to the log generation tier, it was decided that the log files were to be transmitted to the log collectors using the rsyslog [43] implementation of the syslog protocol, as supported by the devices. The log generation sizing was estimated based on [36] and resulted to 10 EPS per router and 5 EPS per switch. It was assumed that the equipment was adequately protected against unauthorized access.

With regards to the log collection sub-tier, the SNA measurements were performed using CASOS ORA version 2.3.6, a software tool by Carnegie Mellon University [54]. The GRNET.SA network was modeled as $G=(V,E)$, were $|V|=68$ (68 nodes) and $|E|=75$ (75 links among them).

The creation of the social network and the calculation of the centrality measurements are easy and fast using ORA. For each centrality measurement the top ranking nodes were identified as important and the sudden decreases in their values were used to separate them into layers. For example, for the total degree centrality, Table 3, the top four nodes would form Layer 1 of the collectors; nodes from the fifth and below would form Layer 2. The sudden decrease of the values from the fourth to the fifth node is used as the basis for this decision. This decision is subjective depending on the analyst.

In Figure 3 the social network is laid out according to two centrality measurements. The x-axis represents the closeness centrality and the y-axis the total degree centrality of the nodes. The mapping of nodes to layers was performed by moving from top to bottom and from right to left, from the highly valued ones, thus important, to the less important ones. The circular nodes are the Layer 1 collectors, the square ones the Layer 2 collectors and the triangle nodes are originators forming Layer 3. The log data will be generated on the originators and collected on the Layer 2 syslog servers, which will forward them to the Layer 1 servers.

Table 3: Sample total degree centrality

Rank	Agent	Value	Unscaled
1	KOL1	0.269	18.000
2	EIE2	0.194	13.000
3	THES2	0.134	9.000
4	PATR2	0.119	8.000
5	XAN2	0.075	5.000
6	IOAN2	0.075	5.000
7	SYR	0.075	5.000
8	HER2	0.075	5.000
9	EIESW1	0.075	5.000
10	LAR2	0.060	4.000

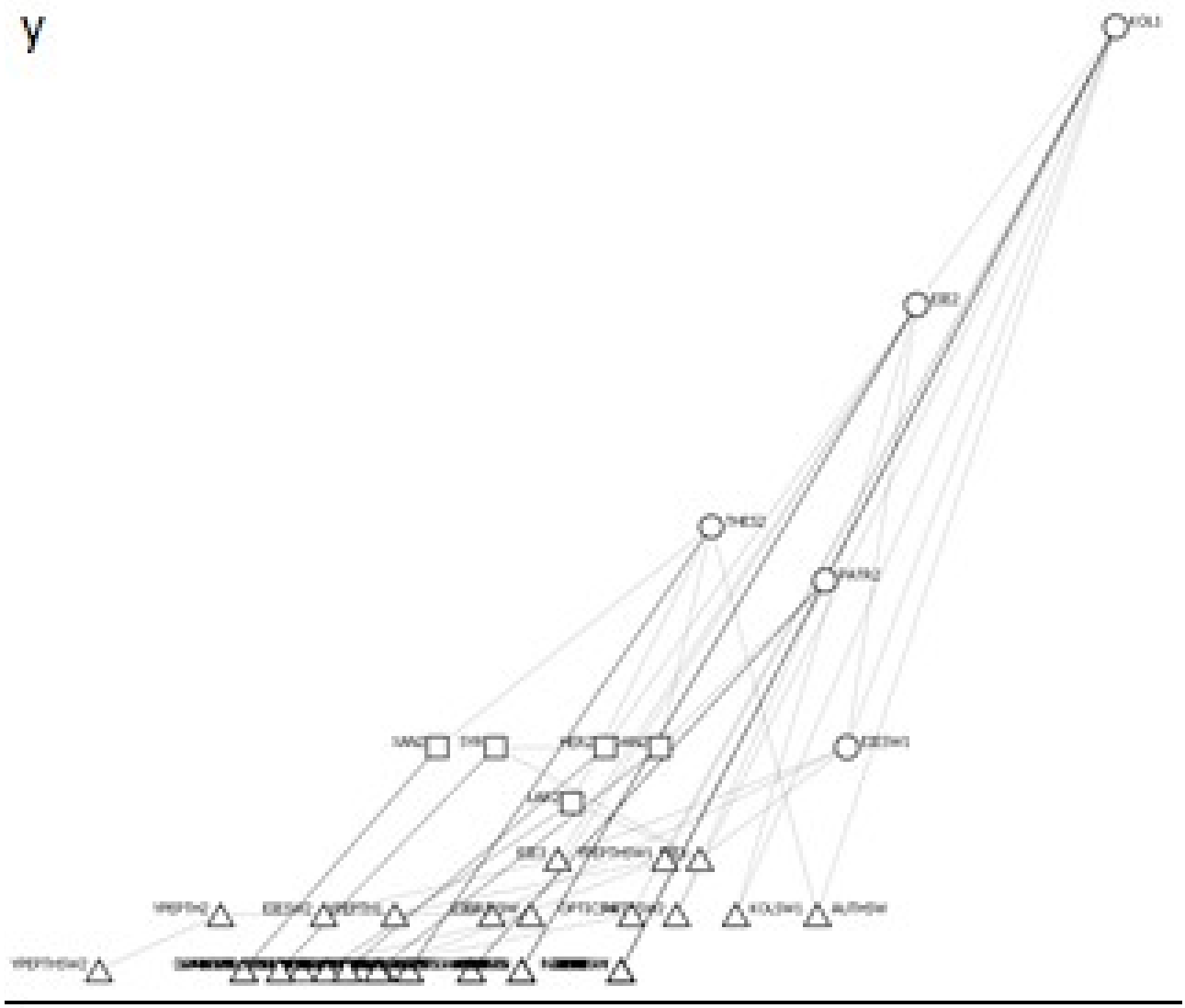

Figure 3: Closeness to total degree centrality layout

By applying the Newman grouping algorithm, the originators were assigned to the collectors. The algorithm separated the social network into seven node groups. In Figure 4 each node group is represented by a different shape. With the application of the Newman grouping algorithm the assignment of the originator to collectors was performed quickly and its output was close to what intuitively would have been decided based on the network topology.

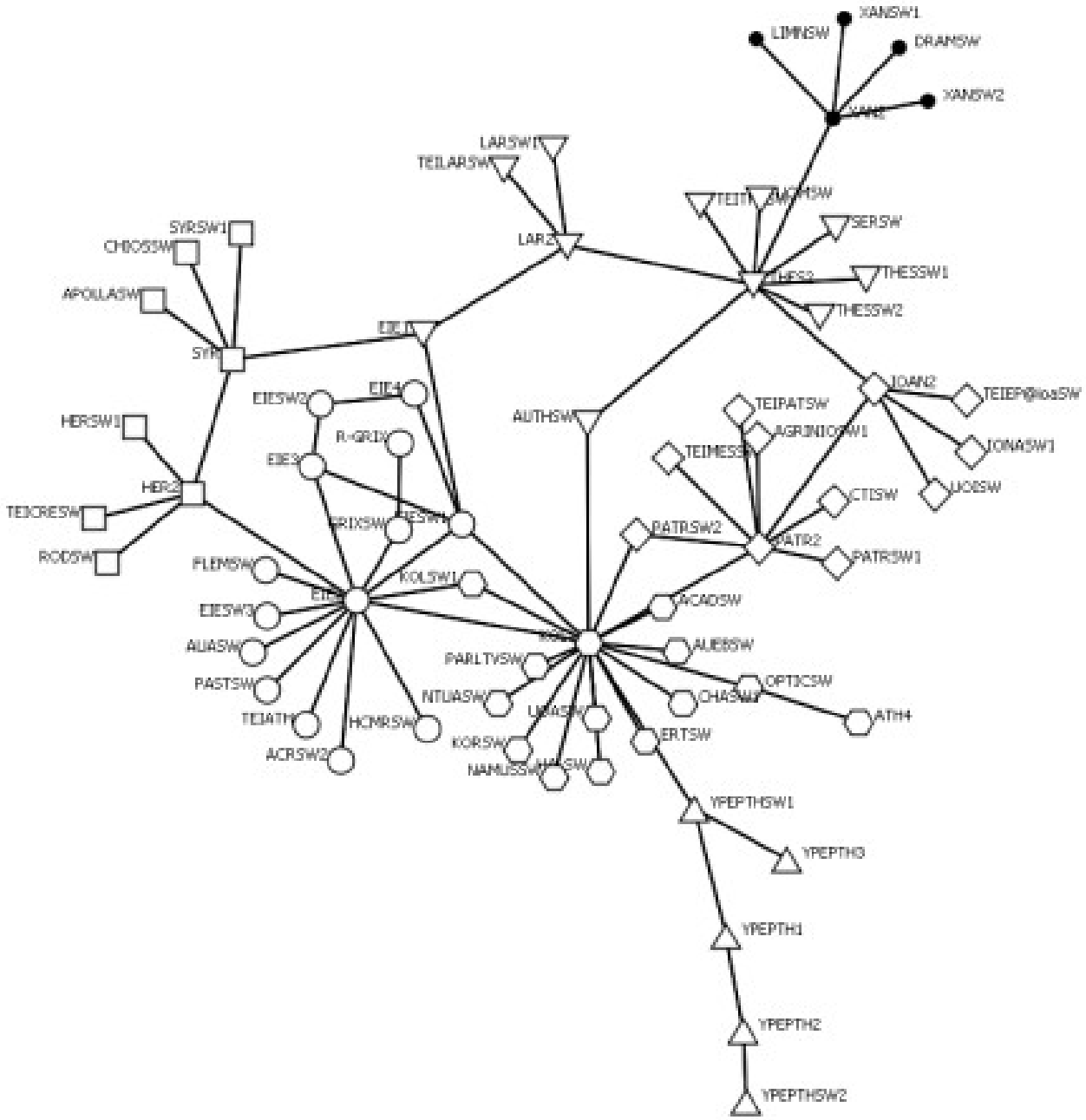

Figure 4: Groups of nodes - Newman algorithm.

Having determined the placement of the collectors and the assignment of the generators, the log collection sizing was estimated. For each collector the EPS that the collector was expected to receive was the sum of the EPS that the assigned originators would generate. The estimates of the EPS, the network bandwidth and the volatile memory for Layer 2 are listed in Table 4. Security concerns were addressed with the implementation of a separate logical network that would be used

to connect the connectors in Layer 2 with those in Layer 1. This was in accordance with the architectural decisions previously made and was achieved by configuring the use of TLS on the syslog servers.

Table 4: Layer 2 log collection sizing

Layer	**Collector**	**Server Requirements**	**Layer**	**Collector**
		Total EPS	**Total Bandwidth (Mbps)**	**Memory-RAM (bytes)**
Layer 2	HER2	125	0.4149	30360
	SYR	125	0.4149	30360
	LAR2	130	0.4558	30360
	XAN2	130	0.4559	30360
	IOAN2	125	0.4149	30360
	YPEPTHSW1	140	0.5377	30360

For the compilation of a log data life-cycle management process four data stages were identified (live, back up, archive and disposed data); the functions to be performed on them were log rotation, retention, compression and integrity check. Based on the requirements for log retention and the log collection sizing, the necessary storage, for each data stage and collector, was estimated resulting into Table 5. Defining a process that would meet the requirements was facilitated by the availability of the log collection sizing.

Table 5: Log storage sizing

Layer	Collector	Storage Stages (Gbytes)		
		Live data (5 days)	**Back up data (5 days)**	**Archived data (1 year)**
Layer 1	KOL1	61.5084171295	61.5084171295	4490.1144504547
	EIE2	30.7542085648	30.7542085648	2245.0572252274
	THES2	59.3036413193	59.3036413193	4329.1658163071
	PATR2	38.8491153717	38.8491153717	2835.9854221344
	EIESW1	51.0638952255	51.0638952255	3727.6643514633

The synchronization of the logging equipment was approached through the implementation of a hierarchy of NTP servers. Using ORA and the centrality measurements, the social network was laid out placing the nodes on the x-axis based on their betweenness centrality and on the y-axis based on their closeness centrality. The high valued nodes were placed to the right and top. Moving from the top-right to the bottom-left, the nodes were separated into the strata that would form the NTP hierarchy. The inversed triangle nodes formed Stratum 1, the square ones Stratum 2 and the dotted nodes Stratum 3. Performing this process was efficient, as calculating the centrality measurements is trivial using software and the patterns of the nodes' distribution are easily identified using visualizations. The placement of the NTP strata is shown in Figure 5 and it is close to what intuitively might have been chosen**.**

The resulting log management infrastructure was evaluated against the five scalability dimensions. An increase in the volume of the generated log data could be alleviated changing the configuration of the syslog servers (for a temporary increase) or adding more collectors (for a long term increase) and with the addition of storage space. Due to the distributed architecture of the infrastructure, expanding to a new geographic location may require the addition of collectors for the new originators. Depending on its impact on the network topology, the separation of the nodes into layers may have to be performed again. Tracking the available assets is effective with the use of software tools and is not expected to affect the infrastructure's scalability.

A performance measurement program was compiled, stating the goal of the log management infrastructure to “provide real-time security monitoring of the GRNET.SA network”. This goal was broken down into objectives and resulted into measures/metrics.

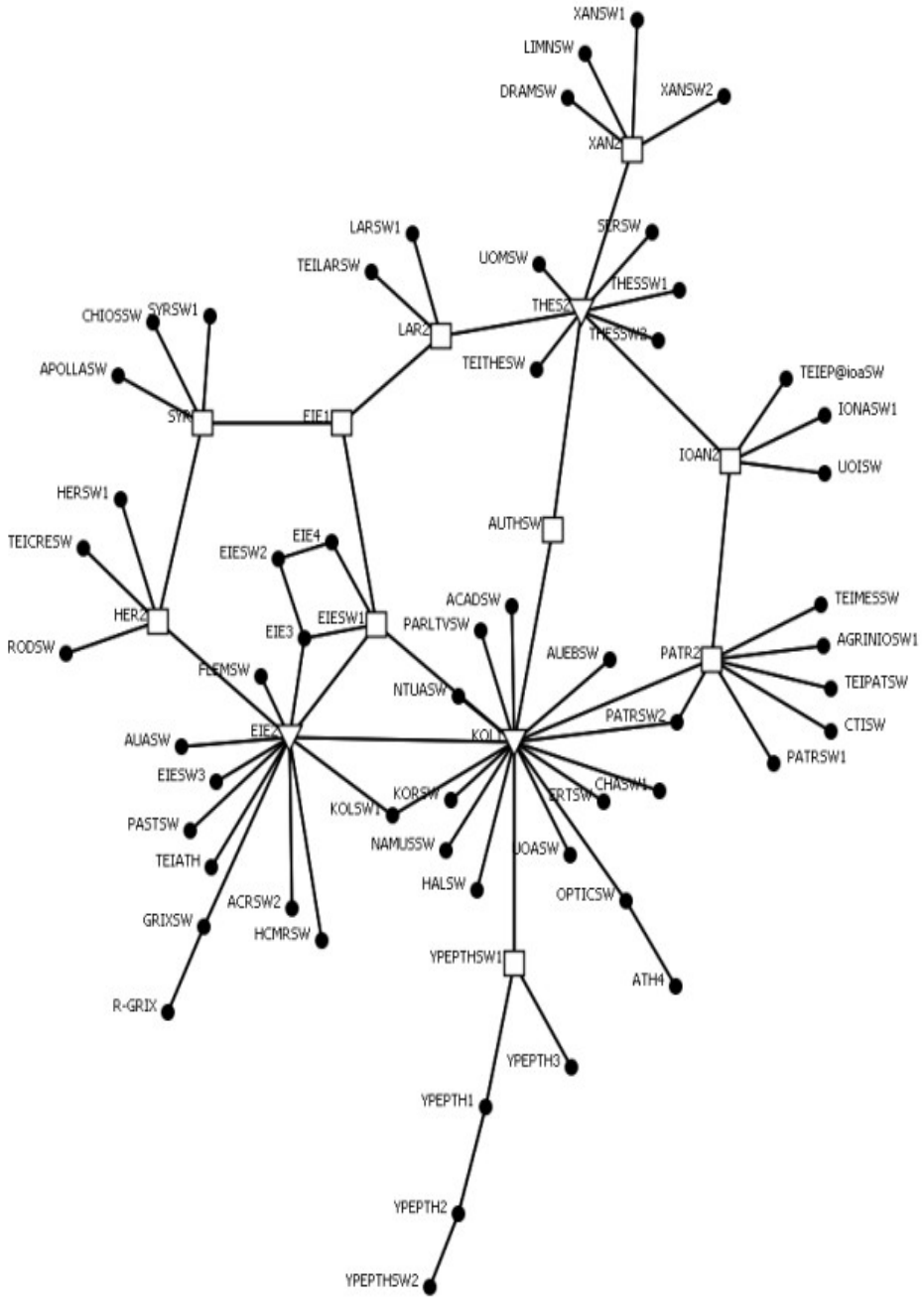

Figure 5: NTP strata

2.5 Results and Discussion

Designing a log management infrastructure is demanding and various issues need to be considered. Leveraging knowledge from log data is certainly beneficial for an organization, and in addition legislation and compliance needs has rendered log management a necessity. Existing literature does not address all aspects of the design process, focusing on specific issues of log management each time. The need for a step-by-step methodology that can guide the design and implementation of a log management infrastructure remains, and results from this research. Existing methods and industry proposed practices are considered and integrated, were applicable, leveraging the field of social network analysis to justify and document design decisions. The target audience of the proposed is large and dispersed infrastructures (e.g. WANs), though applicable to smaller infrastructures both partially and as a whole. Issues affecting its adaptability to new requirements and business needs are also considered, such as scalability and measurement of performance. The proposed methodology was applied on a the infrastructure of a real installation, the GRNET S.A., demonstrating its workings, using publicly available data as provided by the organization. This work researched log management issues from the log source through to the collection of the log data to a central point for storage. Additional aspects that could be discussed are the assignment of roles to the security personnel, as well as the establishment of standard operating procedures. Project management of such implementations would be also of interest.

In the following sections, the log management infrastructure resulting from the proposed methodology, is evaluated, to validate that the set business requirements are actually fulfilled and the business processes can be adequately supported.

3 Validating the design of a log management infrastructure

In the previous chapter we used SNA to model the log management infrastructure of an organization and analyze it to reach an optimal design. The placement of the log collectors, among other issues affecting the design of such an infrastructure, where justified and documented using SNA concepts and techniques. Continuing the research with the validation of the derived design, emerges the need to model and analyze the *design structure* of the log management infrastructure. The *design structure* of an infrastructure, is the relationship among different types of nodes, collectors, log management tasks and log data. Whereas SNA can be used to model the log management infrastructure and identify its key nodes (log collectors/generators), it does not lend itself to modeling and analyzing their design structure. In order to analyze a network composed of multiple types of nodes and complex cross-connected networks an extension of SNA is needed, namely the MNA.

3.1 Background

Social network analysis is based on an assumption of the importance of relationships among interacting units. The social network perspective encompasses theories, models, and applications that are expressed in terms of relational concepts or processes. In social network analysis the identification of the key nodes is a task that is commonly achieved using the measurements of centrality. In [24] various measurements are defined along with their possible interpretation and meaning, depending on the context. In [25] various methods of analyzing social networks are presented. One of them is the separation of the social network into a core and a periphery part based on the centrality of the nodes. More complex methods are proposed in [26], [27], where the author addresses the inefficiency of the centrality measures in identifying important and key nodes. SNA usually handles networks with one type of nodes, such as agent networks or task networks. The SNA methods do not lend themselves well to treating complicated data structures such as those

encountered in multi-mode networks, where three or more modes may coexist [24]. Therefore, whereas SNA can be used to model and analyze the placement of log collectors, it does not lend itself to modeling and analyzing their design structure, which is the relationship among different types of nodes; collectors, log management tasks and log data. To achieve this, it is essential to extend SNA to multiple types of nodes and to more complex cross-connected networks; this is accomplished by MNA.

Meta-networks were first described in [55] and [56] where the authors propose that the structure or architecture of organizations can be better understood, analyzed and even managed, understanding the structure of interdependencies that exist within its boundaries. They propose three domains in organizations, that are considered universal, as follows: organizations are composed of individuals; individuals are assigned tasks to accomplish, as members of the organization; each task requires specific number of resources to be accomplished. Each organization, team or group has an organizational architecture. In order to understand it, it is necessary to first understand how various elements of these domains map onto one another. Their approach is stated to be more comprehensive than other network based approaches, as most of them mainly focus on one domain (personnel or task) and only one type of relationship, while the grammar they defined covers multiple domains and multiple relations. In [57] MNA is used to understand the complex interactions in the organizational network of a project. The authors generate a three-dimensional meta-network model that includes the connections among a project's organization, tasks and knowledge. They use six network measures to investigate the efficiency of task assignment in a project organization and the results are validated through a case study, which identified key agents and tasks that significantly effected task completion and project performance. In [58], the authors developed and analyzed a dynamic supernetwork framework that consists of a social network and a supply chain network, that allows for electronic and physical transactions. They modeled the behavior of decision-makers (manufacturers, retailers and consumers related with the demand markets), involving multi-criteria decision-making (profit functions, risk and value relationships). Manufacturers are assumed to sell their products through physical or electronic links to retailers, while retailers can sell the product through physical links to consumers. Increasing the levels of relationships is assumed, in the framework, to reduce transaction costs and risk, though adding some costs that have to be considered by the decision-makers in the multi-leveled supernetwork. The decision-makers are placed in at distinct tiers in the supernetwork aiming to optimize their objective functions while facing multiple criteria. Their work contributed to the integration of social networks with other complex networks and identified the levels of relationship and the flows of

product transactions. The authors in [59]propose a dynamic supernetwork theory for the integration of financial and social networks.. The consider markets that demand various financial products and decision-makers that poses sources of funds. They model the multi-criteria decision-making behavior of of the various decision-makers constructing a multilevel supernetwork framework, composed of the financial network and the social network, including the maximization of net return and relationship values, as well as the minimization of risk. In the context of the proposed framework it is assumed that an increase in the levels of relationship reduces the transaction costs and risk, and has additional value for the decision-makers. They explore the dynamic evolution of the financial flows, the associated product prices and the relationship levels on the supernetwork, until an equilibrium pattern is achieved. They propose a discrete-time algorithm to track the evolution, over time, of the relationship levels, financial flows and prices, while providing qualitative properties of the dynamic trajectories. The structure of the financial and social networks emerges as a byproduct of the equilibrium pattern, as it identifies both the pairs of nodes that have flows and the size of the flows. The foundation for a new theory for the management of knowledge intensive systems or organizations, termed *knowledge supernetworks*, is set with [60], proposing a conceptual and theoretical framework for the study of knowledge organizations. The proposed framework allows for the abstraction of decision-making that involve knowledge organizations. The authors developed fixed demand and an elastic demand version of the knowledge supernetwork, and demonstrated how such organization can be formalized as a supernetwork. They derived the optimal, or equilibrium, conditions and demonstrated that they all can be uniformly formulated as variational inequality problems. Qualitative properties of the flow patterns on the knowledge networks are provided, as well as computational schemes, implementing both the fixed and elastic demand versions, to analyze several numerical examples on knowledge supernetworks.

Validating the design of a log management infrastructure

3.1.1 Modeling a log management infrastructure as a social network

A log management infrastructure is composed of the following entities: the *log management tasks* (resulting from the log management requirements), the *log collectors*, the *log files* and the relationships among these entities. In order to perform a log management task an analyst needs the data contained into specific log files, while each log file is sent to one or more log collectors [5].

Each log collector is “assigned” specific log management tasks, meaning that an analyst has to be able to perform the defined subset of tasks, using the log data collected at that collector. This results into relationships among the aforementioned entities as shown in the UML entity relationships diagram, Figure 6, where a log file, a log collector and a log management task are linked with many-to-many relationships.

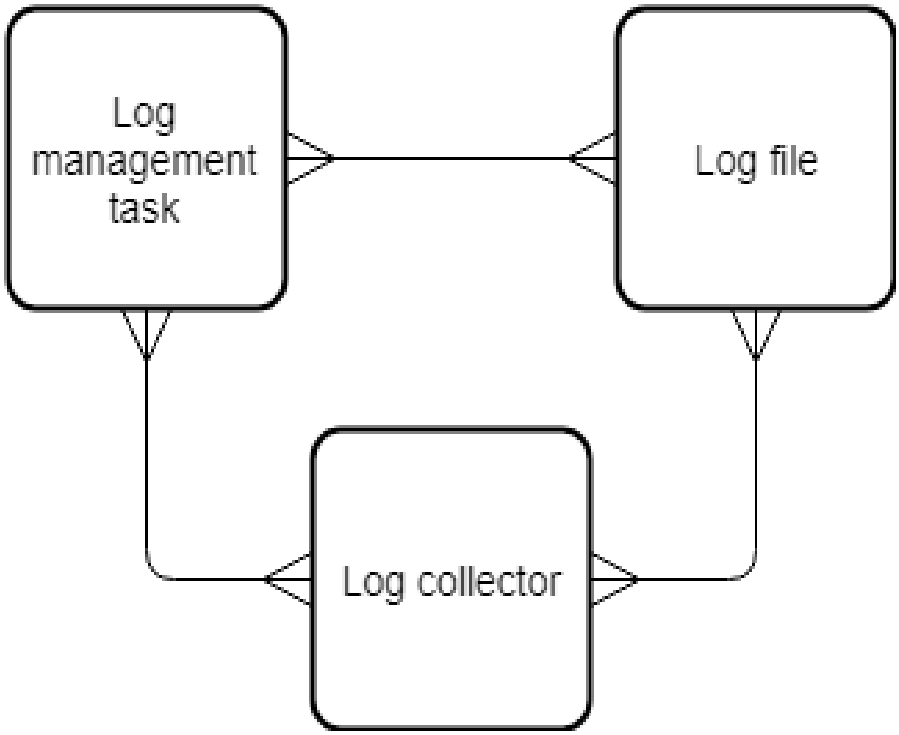

Figure 6: Relationships among log management infrastructure entities

A *node* (or *actor*) is a social entity in SNA and it can be a discrete social unit (an individual) or a collective social unit (a group of people, a corporate department, etc.). Though termed “actor” it does not mean it has the ability to act. *Links* (or *social ties*) connect actors, establishing connections between pairs of actors. A *relation* is the collection of a specific kind of links formed among the actors within a specific set of actors. *Social networks* are composed of nodes that relate to other nodes through their links. When the links have a direction the network is directed and the link from node A to B is different from node B to A. When a link direction is not specified, the network is undirected and the link from A to B is not different from the one from B to A. A node can have one or more attributes describing it and a link can be binary or have a value. Using graph theory notation $G = (V, E)$ is a social network G, with |V| nodes and |E| links among them and it is represented by a |V| × |V| adjacency matrix, where the existence of a link between node $v_i \in V$ and

node $v_j \in V$, is indicated by a value in the $e_{ij} \in E$ cell. This is a *one-mode* network [61], as the links are formed between the nodes of the same set of nodes. The term *mode* refers to the distinct set of entities on which structural variables are measured. A *two-mode* network is formed between two distinct sets of nodes, N and M, and is represented by a $|N| \times |M|$ incidence matrix. For the proposed methodology, the entities of the log management infrastructure and the relationships among them, used to construct a three-mode social network, are identified as follows:

- $T = \{t_1,t_2,\ldots,t_r\}$, the log management tasks.
- $C = \{c_1,c_2,\ldots,c_c\}$, the log collectors.
- $F = \{f_1,f_2,\ldots,f_f\}$, the log files.

and the links (relationships) among them are represented by the following incidence matrices:

- $|F| \times |T|$, the log files necessary to perform each log management task.
- $|F| \times |C|$, the collector to which each log file is sent.
- $|T| \times |C|$, the log management tasks “assigned” to each log collector.

3.1.2 Modeling the log management infrastructure as a meta-network

The log management infrastructure is then modeled as a meta-network. In the context of MNA the design structure of a log management infrastructure is composed of *agents*, an entity that can process information; *tasks*, a set of actions that lead to the accomplishment of an assignment; *knowledge*, the information that is available [62]. Each entity corresponds to a *node class* and the collection of nodes belonging to a class forms the corresponding *nodeset*. Links can be established between the nodes of the same node class or among the nodes of different node classes. Both nodes and links may have attributes further describing the nodes and provide context to their relationship. Each of the resulting networks, multiple networks can be created, represents a specific type of connection between the nodes. When the network is formed between nodes of the same node set it

is called a one-mode network, while when it is formed among N node sets it is an N-mode network. The collection of these networks form the meta-network.

In the context of this methodology, the log management infrastructure is considered as an organization. An organization composed of log collectors (agents), log files (knowledge) and log management tasks (tasks). In Table 6, the constructed meta-network is described, where following the notation of [62], AT is the *agent x task* matrix, KT´ is the transposed matrix *knowledge x task*, and AK is the *agent x knowledge* matrix.

Table 6: Constructed meta-network

Meta-network (organization):		**Log management infrastructure**
Node class	**Node set**	**Interpretation**
Agent (A)	Log collectors (C)	The systems where the log files are collected.
Knowledge (K)	Log files (F)	The log files sent by the log sources.
Task (T)	Log management tasks (T)	The log management tasks (components of the requirements).
Network	**2-mode network**	**Interpretation**
Agent × Task (AT)	|T| × |C|	The log management tasks expected to be accomplished on each log collector.
Knowledge × Task (KT')	|F| × |T|	The log files that are necessary to perform each log management task.
Agent × Knowledge (AK)	|F| × |C|	The log files actually collected on each log collector.

Having constructed the infrastructure's meta-network, the methodology continues with its analysis, which is performed computing the measurements of *Agent Knowledge Needs Congruence*, *Agent Knowledge Waste Congruence*, and *Knowledge Potential Workload* [62], [63].

- *Agent Knowledge Needs Congruence*, is the amount of knowledge that an agent lacks to complete its assigned tasks, expressed as a fraction of the total knowledge required for the assigned tasks. The measure compares the knowledge needs of the agent to do its assigned tasks with the actual knowledge of the agent. The measure value for an agent increases when it has need of knowledge to which it is not assigned. Let NK = AT*KT' be the knowledge needed by agents to do their assigned tasks; then the output value for agent *i* is:

  sum(NK(i,:) .* *AK(i,:))/sum(NK(i,:)).

- *Agent Knowledge Waste Congruence*, is the amount of knowledge that an agent has that is not needed by any of its tasks expressed as a fraction of the total knowledge of the agent. The formula compares the knowledge of the agent with the knowledge it actually needs to do its tasks. Any unused knowledge is considered wasted. Let NK = AT*KT' be the knowledge needed by agents to do their assigned tasks, then the output value for agent *i* equals to:

  sum(*NK(i,:) .* AK(i,:))/sum(NK(i,:)).

- *Knowledge Potential Workload*, is the maximum amount of knowledge an agent could use to do tasks if it were assigned to all tasks. If an agent is assigned all the tasks this measure will compute a value expressing its potential to carry out all the tasks based on his connections to the knowledge needed for the tasks [54]. The value for agent *i* equals to:

  sum((AK*KT(i,:))/sum(KT).

  The higher the value of this measurement, the more tasks can be completed using the knowledge of this node, thus the more critical this node is.

3.1.3 Validating and improving the design structure

Calculating the agent knowledge needs congruence, we identify the log collectors that do not collect the log data that are necessary to perform the log management tasks that are assigned to them. An analyst that has access to such a collector, cannot perform its full set of tasks. Log sources can be reconfigured either redirecting log data to more log collectors, or changing the log's verbosity. The measurement is expressed as a fraction, providing the means to prioritize the reconfiguration actions.

The measurement of the agent knowledge waste congruence, allows for the identification of the log collectors that collect log data needless for the assigned task. Collecting unnecessary log data, results in waste of resources (e.g. log storage, network bandwidth), as well as lessen responsiveness for an analyst processing those data. These measurement is also expressed as a fraction, facilitating the prioritization of log collectors. A measure of a log collector's importance is established measuring the knowledge potential workload; a log collector whose data can be used to perform a large set of log management tasks, is identified as critical for the infrastructure. Disrupting its operation would result in inability to perform a large set of log management tasks, impacting the log management infrastructure's efficiency.

3.2 Case study

In order to demonstrate the workings of the proposed method, we assume a log management infrastructure where 25 log files from various devices are sent to 5 log collectors. The collected log data were used to perform a set of 10 analysis tasks and on each log collector a subset of analysis tasks was desired to be accomplished. The number of nodes and links was selected to be small in this example in order to ensure the readability of the visualizations. Each log analysis task needs specific log files in order to be accomplished. We note, however, that the available MNA tools can easily handle thousands of nodes and links, posing in practice no limit to the scalability of the proposed approach. The log collector for each log file was configured during the implementation of the infrastructure, as well as the subset of analysis tasks for each collector and the log files that are required for each task. Each log collector is placed on a different physical location and it is configured to collect log data from a specific category of devices or operating systems [11]. Four log collectors, one for Linux generated logs; one for Windows generated logs; one for logs

generated from network devices; one for logs generated from security devices, form a layer of collectors, Layer-2. The fifth collector, Layer-1, receives all the logs from Layer-2 as well as the logs generated by specific services/applications. The aim of the analysis is to validate whether or not the tasks assigned to each collector can be actually performed with the specific log data they collect.

The infrastructure is modeled as a meta-network composed of three node classes and three 2-mode networks. The node classes are agent (A), knowledge (K) and task (T), having the corresponding node sets of log collectors |C| = 5, log files |F| = 25 and log management tasks |T| = 10. The meta-network is summarized in Table 7, where the three 2-mode networks are the log files collected on each log collector |F| × |C|, the log files that are required for each analysis task |F| × |T| and the analysis tasks “assigned” to each collector |T| × |C|. Tables 8, 9 and 10, list sample data of the matrices representing the aforementioned networks, respectively.

Table 7: Summary of meta-network data

Node class	**Node set**	**Node name**
Agent (A)	|C|=5	Layer-1-Central (c1) Layer-2-Windows (c2) Layer2-Linux (c3) Layer-2-Network Devices (c4) Layer-2-Security Devices (c5)
Knowledge (K)	|F|=25	Windows-Security log (f1) Linux-secure/auth log (f5) VPN Server log (f10) Firewall log (f12) Antivirus log (f19)
Task (T)	|T|=10	Authentication failures and successes (t1) Execution of scheduled tasks (t3) Outbound connections from internal and DMZ systems (t6) Critical errors (t8) Malware (t9)

Table 8: Sample log_file x collector matrix

Collector	Windows-security log (f1)	Windows-scheduled tasks log (f2)	Windows-system log (f3)	Linux-secure/ auth log (f5)	Firewall log (f12)
c1	1	1	1	1	1
c2	1	1	1	0	0
c3	0	0	0	1	0
c4	0	0	0	0	0
c5	0	0	0	0	1

Table 9: Sample log_file x analysis_task matrix

Analysis task	Windows-security log (f1)	Windows-scheduled tasks log (f2)	Windows-system log (f3)	Linux-secure/ auth log (f5)	Firewall log (f12)
t1	1	0	0	1	0
t3	0	1	0	0	0
t6	0	0	0	0	1
t8	0	0	1	0	0
t9	1	0	0	1	0

Table 10: analysis_task x collector matrix

Collector	Authentication failures and successes (t1)	Multiple login failures followed by success (t2)	Execution of scheduled tasks (t3)	System and service restarts and shutdowns (t4)	Application install and updates (t5)	Outbound connections from internal and DMZ systems (t6)	File transfers (t7)	Critical errors (t8)	Malware (t9)	Database users executing CREATE,GRANT (t10)
c1	0	0	0	0	0	0	0	0	0	0
c2	1	0	1	1	1	0	1	1	0	0
c3	1	0	1	1	1	0	1	1	0	0
c4	1	0	0	1	0	0	0	1	0	0
c5	1	1	1	1	1	1	1	0	1	1

For the needs of this case study we assume three teams of personnel in the organization; the security team, the system administration team and the network administration team. Each team is located in a different physical location and has access to log collectors as listed below:

- Security team: Layer-2-Security Devices (c5)
- Systems' administration team: Layer-2-Windows (c2), Layer-2-Linux (c3)
- Network administration team: Layer-2-Network Devices (c4)

Each team needs to perform a subset of analysis tasks as shown in Table 10. Combining these two-mode networks results to the three-mode network depicted in Figure 7. In this figure, the circles represent the log collectors, the hexagons the analysis tasks and the squares the log files.

The construction of the multi-mode social network, the visualizations and the measurements were performed using the CASOS ORA-NetScenes 3.0.9.9.29 tool [54], developed by Carnegie Mellon University. It is a network analysis tool used to detect risks or vulnerabilities on the design structure of organizations, analyzing their structural properties. The calculated measures for this case study are shown in Table 11. The Agent Knowledge Waste Congruence of c1 is one, meaning that the log

data collected on this collector is not needed for the “assigned” analysis tasks. This was expected, as this collector is the central point where every log file is stored, though in Table 10 we observe that no analysis task is “assigned” to it. Collector c4 has a value of 0.500 as it receives both syslog and NetFlow protocol data, though it only needs the syslog data for the “assigned” tasks. Concerning the Agent Knowledge Needs Congruence, we observe that no collector has the required log data. The value for c4 is the highest, as the syslog data it receives from the network devices are not enough to track the system and service restarts, the critical error, etc. throughout the infrastructure.

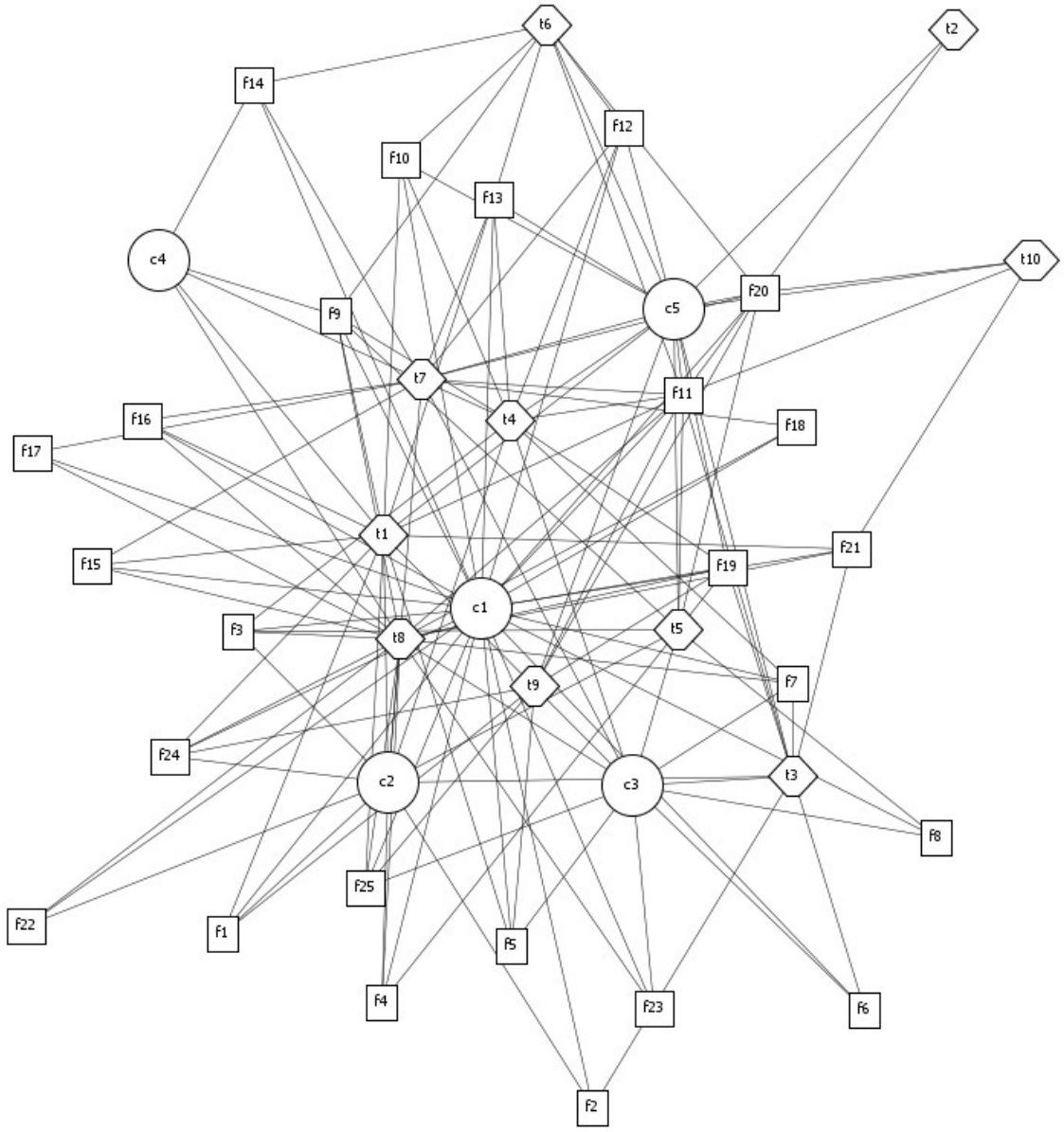

Figure 7: Generated three-mode social network

Table 11: MNA measurements

Collector	Agent Knowledge Waste Congruence	Agent Knowledge Needs Congruence	Knowledge Potential Workload
c1	1.000	0.000	1.000
c2	0.000	0.821	0.162
c3	0.000	0.839	0.149
c4	0.500	0.912	0.095
c5	0.000	0.525	0.405

Applying MNA on this simulated infrastructure we were able to quickly identify that log collector c4 not only lacks the necessary log data to perform its tasks, but it receives log data irrelevant to the "assigned" tasks as well; this results in waste of resources. None of the remaining collectors has at its disposal the necessary data, but they avoid receiving irrelevant log data, too. Figure 8 visualizes the c4 node and its links. The dotted lines represent the log files it actually receives and the solid lines represent the log files it should be receiving to accomplish its tasks. The social network's layout has been adjusted for readability and the positions of the nodes have no special meaning. As a result of the analysis, a collector that both lacked necessary and received unnecessary information was identified, and proper adjustments can be easily performed assisted by the visualizations. This analysis should be repeated for collectors c2, c3 and c5, as well as the process of adjustments and analysis until the desired measurements are achieved.

Moving to the identification of nodes' criticality, ignoring the central collector (c1), the most important collectors for the performance of all the log management analysis tasks are, in decreasing importance, c5, c2, c3 and c4, based on the measurement of the Knowledge Potential Workload. Following this analysis the files that are missing or are in surplus on each collector can be easily identified, and corrective actions may be applied, by reconfiguring the log files that are sent to each log collector. Apart from having predefined subsets of analysis tasks "assigned" to each collector, a different use case could be that of an analyst seeking the optimal collectors in order to perform specific analysis tasks as part of their investigation, be it related to security or not.

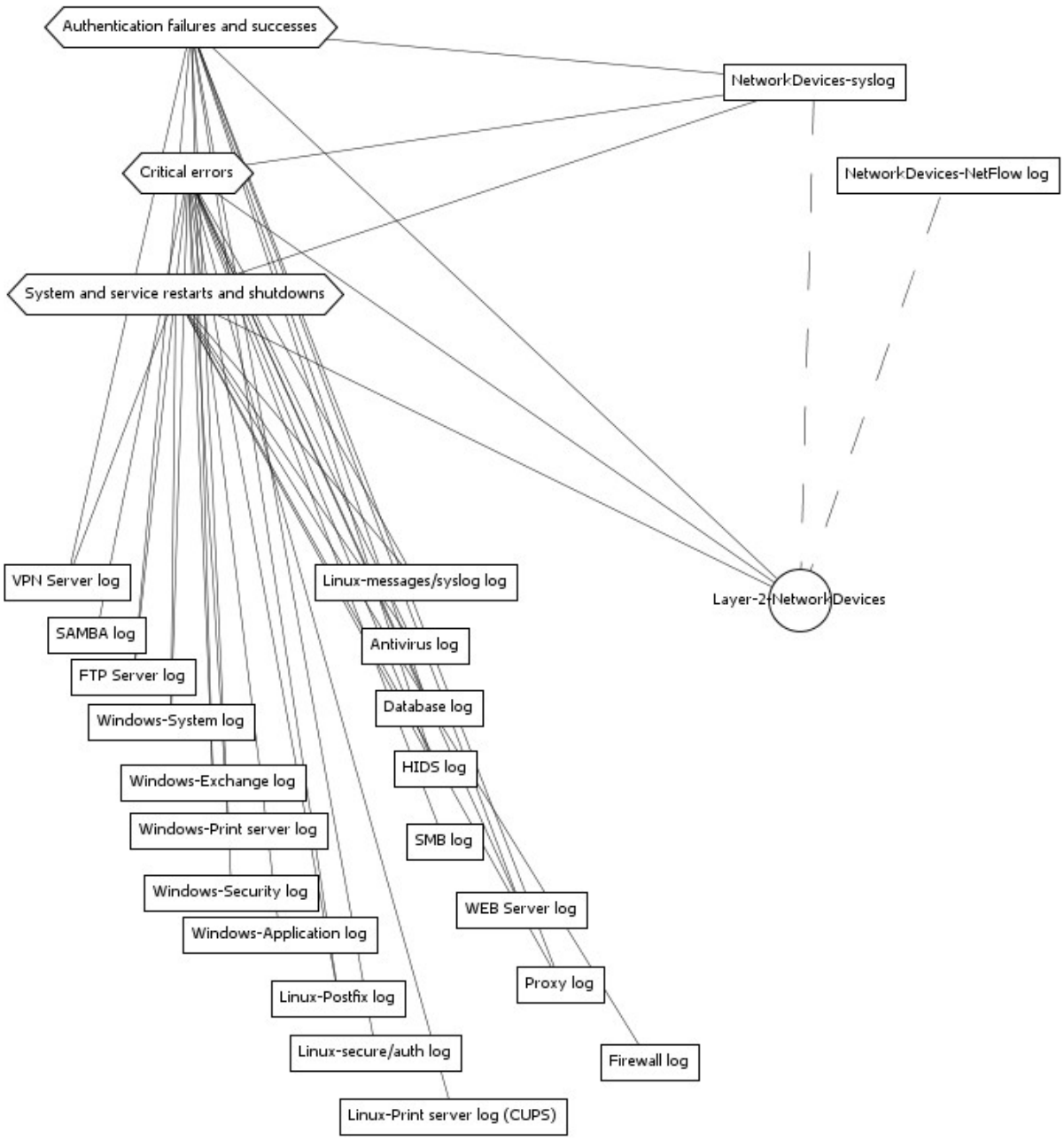

Figure 8: Links of the c4 node

3.3 Results and Discussion

Designing a large-scale log management infrastructure is a demanding task that needs to consider various technical criteria, in order to collect the log data to the desired locations and log collectors for storage and analysis. Apart from validating the log data flow, emerges the need to validate that the infrastructure is optimally designed for the requirements that drove its implementation. Collecting unnecessary log data results to waste of resources, while the absence of necessary log data hinders the efficiency of the analysts. Modeling a large-scale log management infrastructure as a complex organization, allowed us to apply concepts and techniques provided by MNA. Applying the MNA measurements we were able to identify log collectors that waste resources collecting log data that are required for the analysis tasks, as well as log collectors that lack log data that are necessary. The proposed methodology can be easily applied while the use of SNA software enables the processing of models composed of thousands of nodes and links. Whenever a log source, a log collector, or a log management requirement changes, the proposed methodology can be performed to verify the alignment of the log managements infrastructure with the log management requirements that are expected to support the business processes.

In the following section, we extend the research of optimally designing an infrastructure for the organizations requirements, to optimally design the infrastructure for the evolving threat landscape it operates in.

4 Evolving the design of a log management infrastructure

The collection of log data is necessity for most organizations for various reasons such as security, routine maintenance and troubleshooting, as well as compliance to legislation and standards. Despite the fact that automation in log collection and analysis has advanced, still the security personnel has to perform lacking necessary data. Technical challenges, policy restrictions as well as the increasing variety of log sources [64] contribute to the lack of data and require the proper adjustment of log management methodologies [65]. The security controls once implemented by an organization may result ineffective, as the threat landscape is in constant change. While organizations' mission evolves and new technologies emerge, new threats also emerge, resulting to the need to dynamically design a log management infrastructure, a large-scale infrastructure, that is able to adjust to the evolving threat landscape it operates in. In the following sections we employ SNA to research the ability for fast analysis of a log management infrastructure design in correlation with the risks it faces, to enable its optimal adjustment for the current threat landscape.

4.1 Background

Social network analysis is based on an assumption of the importance of relationships among interacting units. The social network perspective encompasses theories, models, and applications that are expressed in terms of relational concepts or processes. In social network analysis the identification of the key nodes is a task that is commonly achieved using measurements of centrality. Methods and measurements for the analysis of social networks are detailed in [24] and [25] with the later focusing on exploratory analysis. In [7] a guide for implementing a log management infrastructure in WANs was presented, where SNA was used to justify design decisions that were formerly based on intuition or experience.

In [66] the authors present a theoretical model for performing risk analysis based on SNA, aiming to evaluate risks of complex systems in real-time. They characterize risks as failure rates of network nodes and links and the level of their importance to the network operation; they also recognize and calculate the network nodes as consolidated representations of smaller networks. A sequence of five-steps is followed in order to model a system: 1) reduce the structure under analysis only to its essential, 2) build a sociomatrix that represents the structure (nodes and connections), 3) consider that each node contains a subsystem that must be modeled, 4) collect data in real-time automating the structure under analysis, and 5) continuously analyze the complex system and subsystem. The authors in [67] employ SNA to build a model for the recognition of the key risk elements in cooperative technological innovation. They follow a three-step analysis process (factor analysis, relations analysis, matrix analysis) to identify the key risk elements. During the first step, factor analysis, they develop a cooperative technological innovation risk elements system, selecting five sorts of elements (collaborative risk, fund risk, performance risk, environment risk and market risk). In the second step, relations analysis, they build an adjacency matrix to analyze the relations between elements, using data collected from analysis archives, observations, questionnaires, etc. In the final step, matrix analysis, they use the centrality measures of in-degree, out-degree and closeness, to identify the key elements; these elements are often easy to directly impact on other elements, but not easy to be interfered by other elements. In [68] the inherent risks within the urban infrastructure system are researched. The risk is considered to be a product of a complex set of network processes, and this complexity is assessed by studying the risk nature, interrelationship dynamic and impact propagation pattern. Various tools and assessment models are used to study risk nature, though they fail to capture the interconnections that shape risk based on community participation. The authors propose a model to draw and simulate the risk impact propagation pattern and interrelationships, using a one-mode network, and following the concept of social risk amplification, which is a process of developing risk network map based on community perspective. They apply the proposed model on a water supply infrastructure system, revealing that participatory networked approach to risk interrelationship analysis are better suited to capturing the intricate that process that shape infrastructure risk. The resilience of the banking system to a contagion (failure of an institution and spill over to the whole financial system), and the channels of contagion are analyzed in [69]leveraging SNA. An apparently robust system may in fact be fragile, due the fact that a high number of interconnections among institutions will amplify, rather absorb, a relevant shock. Additionally, in the financial network formed by institutions, there are both players with many connections and players with few connections. The highly connected players will greatly

impact such a network in case they are disrupted. There work highlights how the dynamics of contagion, as well as direct and indirect interlinkages among financial institutions, can be influenced by three network characteristics: 1) the degree of connectivity, 2) the degree of concentration, and 3) the size of exposures. They result in that SNA can give insights into the various amplification mechanisms in the global web of financial connections, as well as provide the means for simulating the effect of credit and funding shocks on banking and financial stability. Factors affecting social stability risk are researched in [70] studying the complex interrelationship among stakeholders of large hydraulic engineering projects in China. The authors investigate the risk factors and their interrelationships performing a literature review and interview of stakeholders. They obtained a risk list of 45 factors, and they used SNA to identify the key risk factors, the ones that significantly impact on other factors. There findings demonstrated that the government and project developers were the most important stakeholders, and the main factors for social stability risk are the project funding issues, the interest compensation issues, the group events impacting issues, and the project limit time issues. Using SNA they combined stakeholder management with risk management providing reference for the social stability risk management of large hydraulic engineering projects. They also propose an interest compensation mechanism, a mechanism for the prevention of future events, a multi-channel project financial system and a project schedule control system, in order to mitigate the social stability risks. The application of SNA for risk assessment is also demonstrated in [71] in assessing risk of entrance of an exotic disease into a geographical area with naive hosts. They integrate SNA with an existing risk assessment model for with-country animal movements, resulting in a combined tool for the estimation of spatial probability of the introduction of at least on affected animal. Their research aims to develop a spatially explicit risk assessment model to the estimate the spatial probability of the introduction of at least on affected animal by province within Italy per month. To demonstrate the working of their model, it is applied using Italy as a case study.

4.2 Enabling the adoptive design of a log management infrastructure

For the needs of the proposed methodology the various devices and equipment composing a WAN, is referred to as assets. It can be a network device, an operating system, an application, a computer room, physical security mechanisms etc., and it is exposed to various threats that may exploit its

vulnerabilities. When two assets face the same risk, either because they share a vulnerability and/or a common threat, an implicit connection is formed among them.

In SNA, a *node* (or *actor*) is a social entity that can be a discrete social unit (an individual) or a collective social unit (a group of people, a corporate department, etc.). Despite they are termed actors, it is not implied that they have the ability to act. *Links* (or *social ties*) connect actors establishing a tie between a pair of actors, while a *relation* is the collection of a specific kind of ties formed among the actors of a specific set of actors. Social networks are composed of nodes and links, that can be directed or not; the link from node A to B is different from node B to A. A node can have one or more attributes and a link can be binary of valued. Using graph theory notation, G = (V, E) is a social network G with |V| nodes and |E| links among them. It is represented by a |V|x|V| adjacency matrix, where the existence of a link between node $v_i \in V$ and node $v_j \in V$, is indicated by a value in the $e_{ij} \in E$ cell. This is a one-mode network [61], as the links are formed among the nodes of the same set, where the term mode refers to the distinct set of entities on which structural variables are measured. A two-mode network is formed between two distinct sets of nodes, N and M, and is represented by a |N| x |M| incidence matrix. The two-mode network can then be folded to create two one-mode networks, one for each dimension. Network folding is achieved by transposing it to the desired dimension and then multiplying it with the initial incidence matrix, resulting in the adjacency matrix. Folding the |N| x |M| incidence matrix will result in the |N| x |N| and |M| x |M| arrays, where each cell contains the weight between n_i and n_j, m_i and m_j, respectively.

4.2.1 Analysis of the infrastructure

The infrastructure is modeled as an undirected one-mode social network, where the assets are the actors (nodes) and the communication links, or physical connections, among them are the social ties (links) of the social network. The physical connections among the assets of the infrastructure are modeled by the |I| x |I| adjacency matrix, where a link among asset *j* and asset *m* is indicated by a value of 1 in the *jm* cell. A value of zero indicates that the two assets are not physically connected. The analysis starts with the identification of the *boundary spanners* [62] (or *articulation points* [72]). Let G = (V, E) be a connected, undirected graph. An *articulation point* of G is a node which if removed G is disconnected. A value of 1 identifies an asset as a boundary spanner, whereas a value of zero does not [62]. An asset with the structural property of the boundary spanner plays a critical role in the infrastructure as, if it became unavailable, a part of the network would be disconnected.

Security-wise, the availability of this asset should not be compromised, as it would disconnect part of the WAN.

4.2.2 Analysis of the affiliation network

In continuance, the relation among assets and risks is modeled constructing an undirected two-mode social network. A = {a_1, a_2,..., a_n} is the node set of the assets of the log management infrastructure, R = {r_1, r_2,..., r_m} is the node set of the risks that the log management infrastructure faces, and the links among them are represented by the |A| x |R| incidence matrix, where an asset a_i threatened by risk r_j , has a value of 1 in the *ij* cell. Concepts and techniques used in affiliation networks are applied for the analysis of the |A| x |R| undirected social network.

Affiliation networks are two-mode networks where the linkages among members of one of the modes are based on the linkages established through the second mode [24]. The first mode is a set of actors and the second mode is a set of events. An event does not necessarily consist of face-to-face interaction; it can rather be a wide range of occasions: participation to a club, a party, a committee or a board of directors etc. An actor belonging to a club is affiliated to that event; when two actors belong to the same club they are affiliated (linked) by the same event. An affiliation network is represented by an affiliation matrix |A| x |E|, a two-mode matrix where a value of 1 in the *ij* cell affiliates row actor *i* to column event *j* [24]. Folding this two-mode matrix results to one array for each dimension. The array of linkages among actors through their participation to events |A| x |A|, where a value in cell *ij* indicates the events two actors share. The array of linkages among events through the participation of actors |E| x |E|, where a value in cell *ij* indicates the actors that two events share. For the proposed methodology, an asset is represented by an actor and a risk is represented by an event. Two assets are linked when they share the same risk and two risks are linked when they affect the same asset. The |A| x |R| incidence matrix is folded, resulting to the |A| x |A| and |R| x |R| arrays. Our analysis will focus on the undirected one-mode network of assets that link through their common risks, represented by the |A| x |A| matrix.

Total degree centrality is used to identify the nodes that actively participate in the social network. It is distinguished into *in* and *out degree*, when the links are directed to or from the node, respectively. The total degree centrality of a node is equal to its normalized in degree, plus its out degree. Let G = (V, E) be the graph representation of a square network and a node *v*. The Total degree centrality of

node v = deg / 2 * (|V| - 1), where deg = card {u∈V|(v,u) ∈ E ∨ (u,v) ∈ E}([24] as cited in [38]). A node with high degree centrality is a well-connected node and can potentially directly influence many other nodes [26]. The total degree centrality of each asset is measured, the nodes are sorted in descending order based on their total degree centrality and the high valued ones are identified. These assets share the same risks with many other assets, thus an attacker can easily pivot from one asset to another, exploiting their common vulnerabilities, or a threat might exploit multiple assets, by exploiting the same vulnerability present on these assets.

The analysis of the |A |x |A| matrix continues with the identification of the *m-slices*. An m-slice is a maximal sub network containing the links with multiplicity equal or greater than m and the nodes incident with these links [25]. The 0-slice assets are “isolated” as they share no risks with the rest of the assets. A 4-slice link, for example, is a pair of assets that have four common risks. The links are sorted in descending order, based on their value, and the high valued pairs are identified. This results in the identification of sub groups of assets that share many common risks; hence, they pose an increased attack surface and enable multiple threat actors to exploit multiple assets. For example, a malware would spread easily through assets that share multiple common vulnerabilities, as the attack surface of each system is increased compared to the rest of the systems.

4.2.3 Prioritization of assets

Using the SNA measurement the assets of the infrastructure are prioritized. The prioritized assets are considered for incident response actions that lead to evolution of the log management infrastructure design. Adopting [73] incident response actions resolve into preparation (system hardening), detection (focus the efforts of the security personnel), containment (minimize the impact of an incident), eradication (mitigation of vulnerabilities) and recovery (restore normal operation) .

The measurement of boundary spanners has identified the assets that can greatly impact the infrastructure’s availability, total degree centrality has identified the assets that can facilitate the spread of malware in the infrastructure, and m-slices has identified the part of the infrastructure that poses increased attack surface. The order by which the measurements are considered depends on the needs of the analyst and the specifics of the case study. If, for example, protecting the availability of the infrastructure is a priority, then the boundary spanners should be considered first; if the

avoidance of the spread of a malware that could exploit a known vulnerability that is present on many systems is of interest, then total degree centrality should be considered first; if the analysis aims to reduce the attack surface of the infrastructure then the m-slices can be considered first.

4.2.4 Methodology validation

In order to validate the proposed methodology, the *immediate impact analysis* [62] was performed on the undirected one-mode social network modeling the log management infrastructure, |I| x |I|. During this analysis, one or more selected nodes are removed from the infrastructure to determine the effect on specific measures, enabling the performance of what-if analysis about the impact of the absence of these specific nodes. In the proposed methodology the nodes that are removed are the previously prioritized ones (removing a node corresponds to its preparation, containment or eradication [73]) and the measurement that is evaluated is *diffusion*. Diffusion is a network-level measurement that computes how easily something could spread throughout the social network and it is based on the length of the shortest path between nodes. When the nodes are close to each other, the value is high and when they are apart, it is low. Let A be a one-mode network with N nodes, let D(i, j)=shortest path distance from node i to node j (and N if no path exists), let:

TotalDistance = sum(D) , and let

AverageDistance = TotalDistance / (N*(N-1)), the average distance between nodes. Then

Diffusion=N/(N-1)*[1-AverageDistance/N] [62].

The effect of these removals on the measure of diffusion is evaluated, measuring the diffusion on the |I| x |I| undirected one-mode social network before the nodes' removal and after the nodes' removal. A decrease in its value indicates reduced capability of spreading through the network while an increase indicates the opposite.

4.3 Case study

The methodology was applied on the network infrastructure of the Greek Research and Education Network (GRNET network) as it was on March 29, 2018 [50]. The GRNET is a WAN composed of

78 devices connected with 85 links, and its network topology was used for the needs of this case study. The real risks affecting the organization are not publicly available, thus a set of risk was assumed to demonstrate the working of the methodology. The SNA measurements were performed using CASOS ORA version 2.3.6, a software tool developed by Carnegie Mellon University [74].

4.3.1 Analysis of the infrastructure

The GRNET.SA network infrastructure was modeled as an undirected one-mode social network represented by the |I| x |I| adjacency matrix. It is composed of 78 device (nodes) and 85 physical connections (links) among them, Figure 9; the circles represent the nodes of the social network (WAN devices) and the triangles represent the boundary spanner nodes. Removing the “Larisa” node, for example, will disconnect the “Karditsa”, “Trikala”, “Volos” and “Lamia” parts of the WAN (nodes enclosed in the rectangular). In the context of the proposed methodology the node removal refers to a device that has become unavailable either as a result of a security incident or as part of an incident response process (moved to a VLAN, disconnected to stop data exfiltration, vulnerability patching, etc.). Hence these nodes should be prioritized for security monitoring, preparation, containment, eradication and recovery [73].

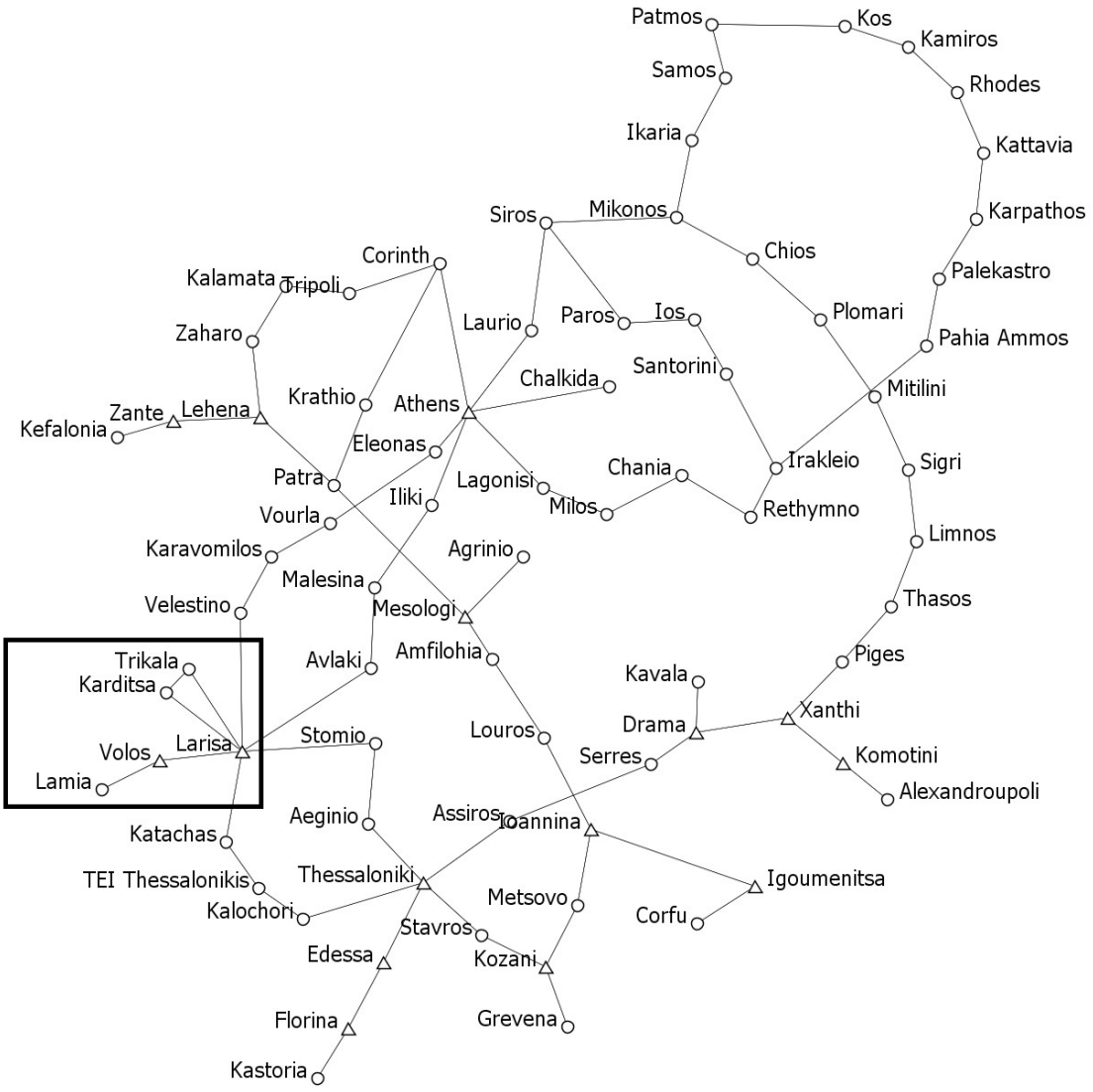

Figure 9: |I|x|I| WAN infrastructure social network

4.3.2 Analysis of the affiliation network

For the needs of this case study a node set |R| of 20 hypothetical risks was assumed as well as their links with the assets |A| node set, i.e. the assets these risks affect. The multitude of risks was deliberately kept low to ensure readability of the visualization, though the SNA software can easily manipulate thousands of nodes. Each asset of the |I| x |I| social network is linked to the corresponding risks, generating the |A| x |R| undirected two-mode affiliation network, Figure 10, where a circle represents an asset of the WAN and a diamond represents a risk (the nodes have been rearranged for visibility). Assets “Rhodes”, “Volos”, “Ios” and “Katachas” face the same risk,

namely Risk-12 (nodes enclosed in the rectangular). Nodes with no link, face no risk and were excluded from analysis.

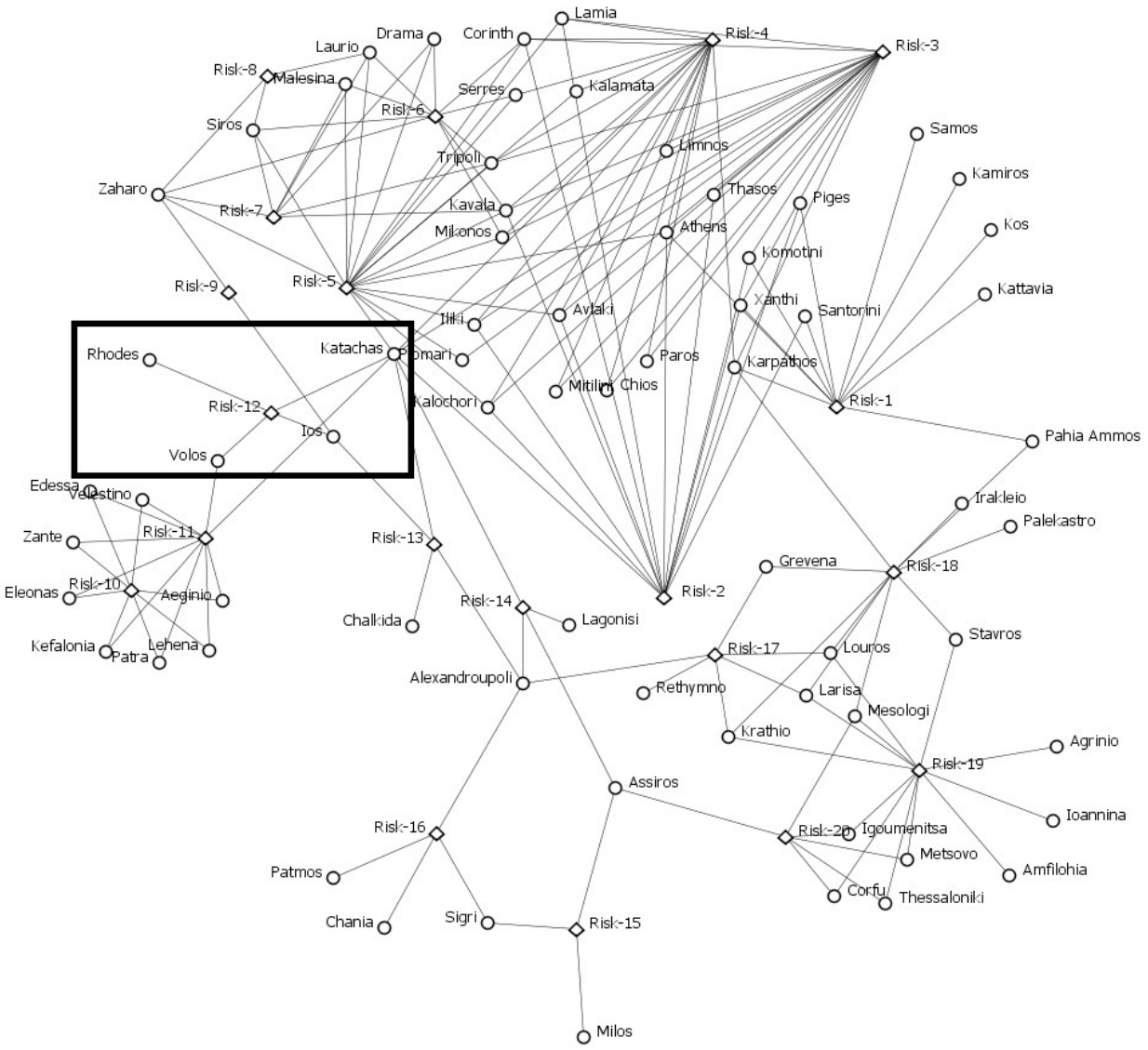

Figure 10: Asset x Risk two-mode affiliation network

The |A| x |R| undirected two-mode affiliation network is folded resulting in the |A| x |A| undirected one-mode network of Figure 11, where a link is formed between two nodes when they link to the same threat. Nodes “Ios” and “Rhodes”, for example, form a link as they are both affected by Risk-12. The link value indicates the multitude of common risks, which is 1 for this pair of nodes. On the folded one-mode social network we measure the total degree centrality of the nodes. The highest ranking nodes are listed in Table 12. Node “Athens”, for example, has common risks with 32 assets (WAN devices). High valued nodes face common risks with many other nodes (not necessarily

physically neighboring ones), thus an attacker that has compromised one of them can easily pivot through these assets by exploiting their common vulnerabilities. As a result, these nodes need to be closely monitored for security incidents and be quickly contained and/or eradicated to prevent the spread throughout the network. In case of a human actor, appropriate action should be taken restricting her access or raising her awareness, for example.

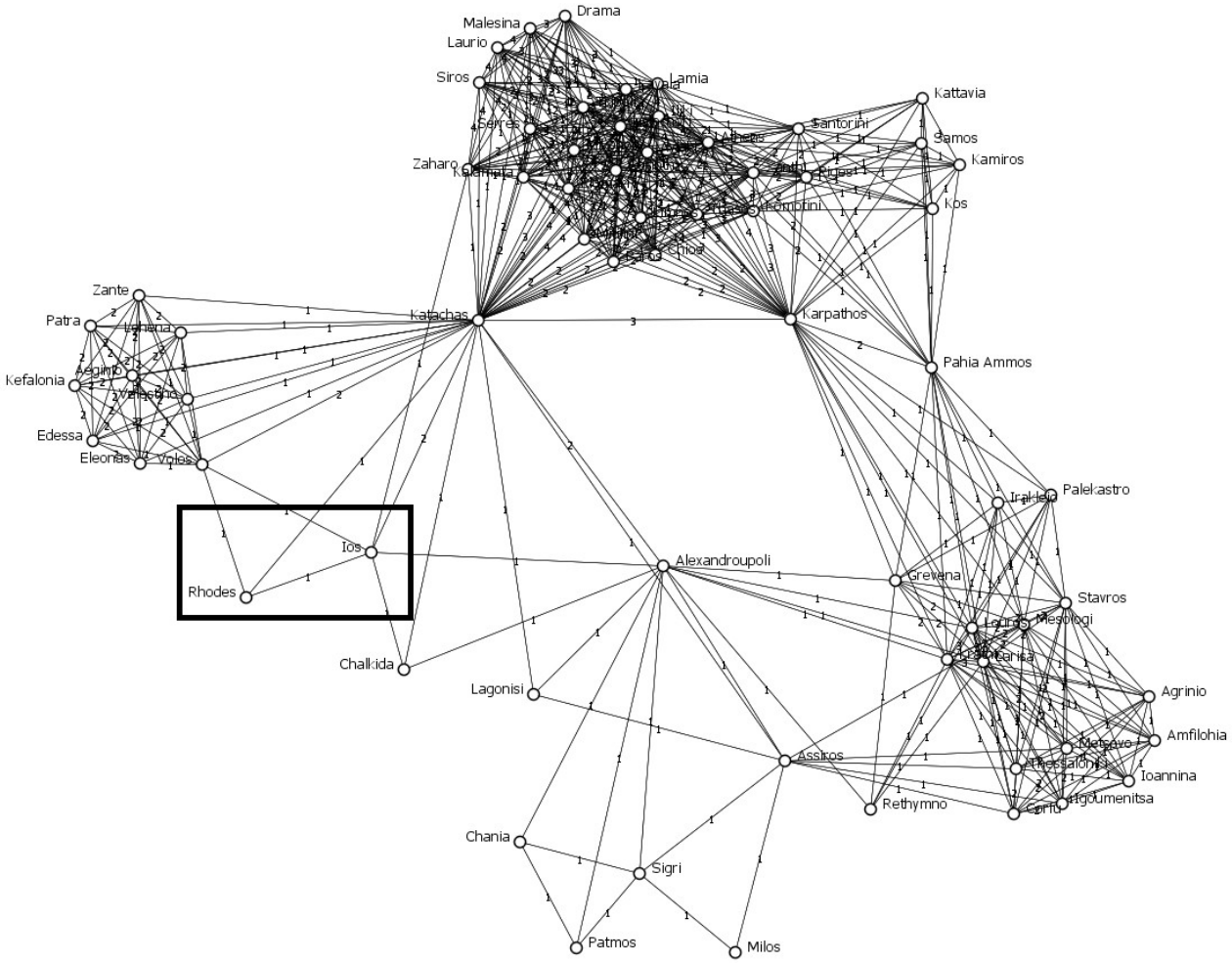

Figure 11: The one-mode |A| x |A|folded social network

Having identified the nodes that face common risks with many other nodes, the proposed methodology continues with the identification of the pairs on nodes that face many common risks. In Figure 12, the three circled nodes at the top of the visualization are connected with a link value of 5, thus they form a 5-slice. The nodes below them form a 4-slice and so on. The remaining 3-slices and 2-slices are not depicted in the interest of readability. An attacker that has compromised node “Kavala” can easily pivot to “Tripoli” and “Corinth”, as both nodes share much vulnerability with

node “Kavala”. These nodes need to be prioritized both for security monitoring and incident response.

Table 12: Sample total degree centrality high ranking nodes

Rank	Node Name	Total Degree Centrality
1	Katachas	42
2	Karpathos	35
3	Athens	32
4	Avlaki	27
5	Corinth	27

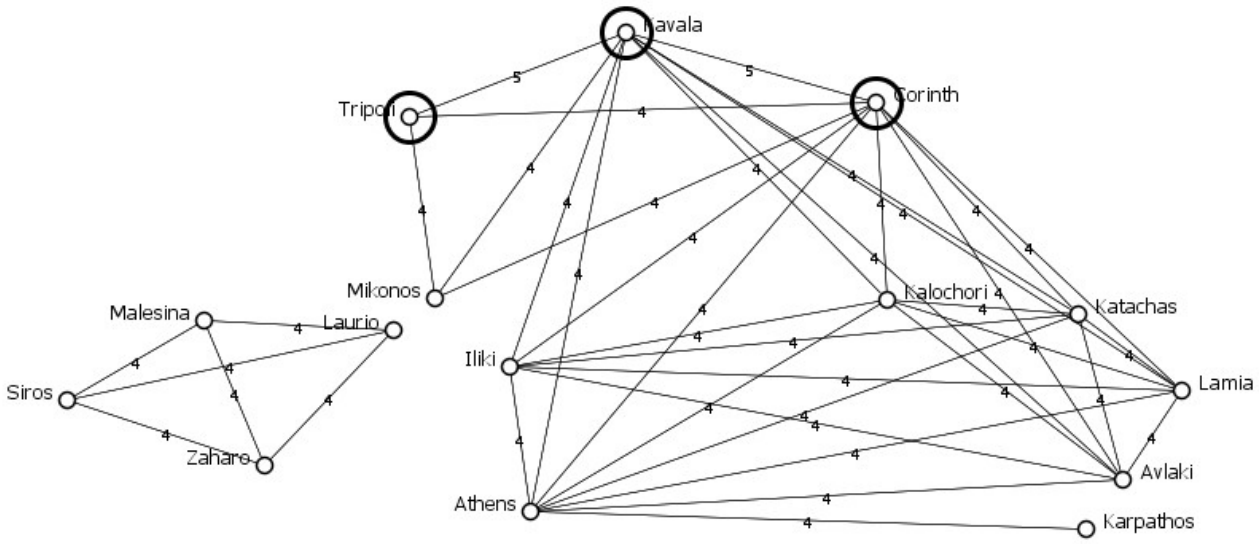

Figure 12: 4-slices of the one-mode folded social network

4.3.3 Prioritization of assets

For the needs of this case study we assume that the aim of the analysis is to protect the availability of the infrastructure, to avoid the spread of malware and to reduce the attack surface of the infrastructure. In Figure 13, each small circle is a node, the large circled nodes are the top ranking nodes in total degree centrality and the nodes with a ring are the boundary spanners; the visible links are the ones valued with 4 or higher, i.e. the 5-slices and the 4-slices. Lower valued m-slices are not depicted, to ensure the readability of the graph.

Node “Athens” is a boundary spanner, member of the 4-slice and highly valued in total degree centrality. It is identified as the top priority the asset, since its removal would disconnect part of the infrastructure (boundary spanner), it faces the same risks with many assets (total degree centrality) and it has multiple common risks with many assets too (member of the 4-slice). Nodes “Kavala” and “Corinth” are highly ranked in total degree centrality and part of a 5-slice, thus selected for the second level of priority. The prioritization continues with the third level of priority that includes the remaining high ranking nodes and node spanners. The use, or not, of a categorization of nodes based on their priority depends on the needs of the analyst. The 4-slice formed by nodes “Laurio”, “Malesina”, “Siros”,” Zaharo” is also of interest, as these nodes face many common threats, though physically distant and located across the country. Exploiting one of them would enable the attacker to affect many users that are geographically distant to each other.

4.3.4 Methodology validation

In order to validate the results of the proposed methodology, we perform impact analysis using using CASOS ORA software. The assets previously identified as critical are removed and the diffusion of the undirected one-mode social network (modeling the infrastructure) is calculated. First the Level-1 nodes (“Athens”) are removed, then the Level-2 nodes (“Kavala”, “Corinth”) and so on, progressively measuring the diffusion of the social network. In the context of the proposed methodology the removal of a node represents making a system immune to exploitation from threats, or a system where the threat was mitigated. The results of impact analysis are listed in Table 13. Each of the first three rows lists the impact of removing the identified nodes by measurement, and the fourth row lists the impact of removing all the prioritized nodes.

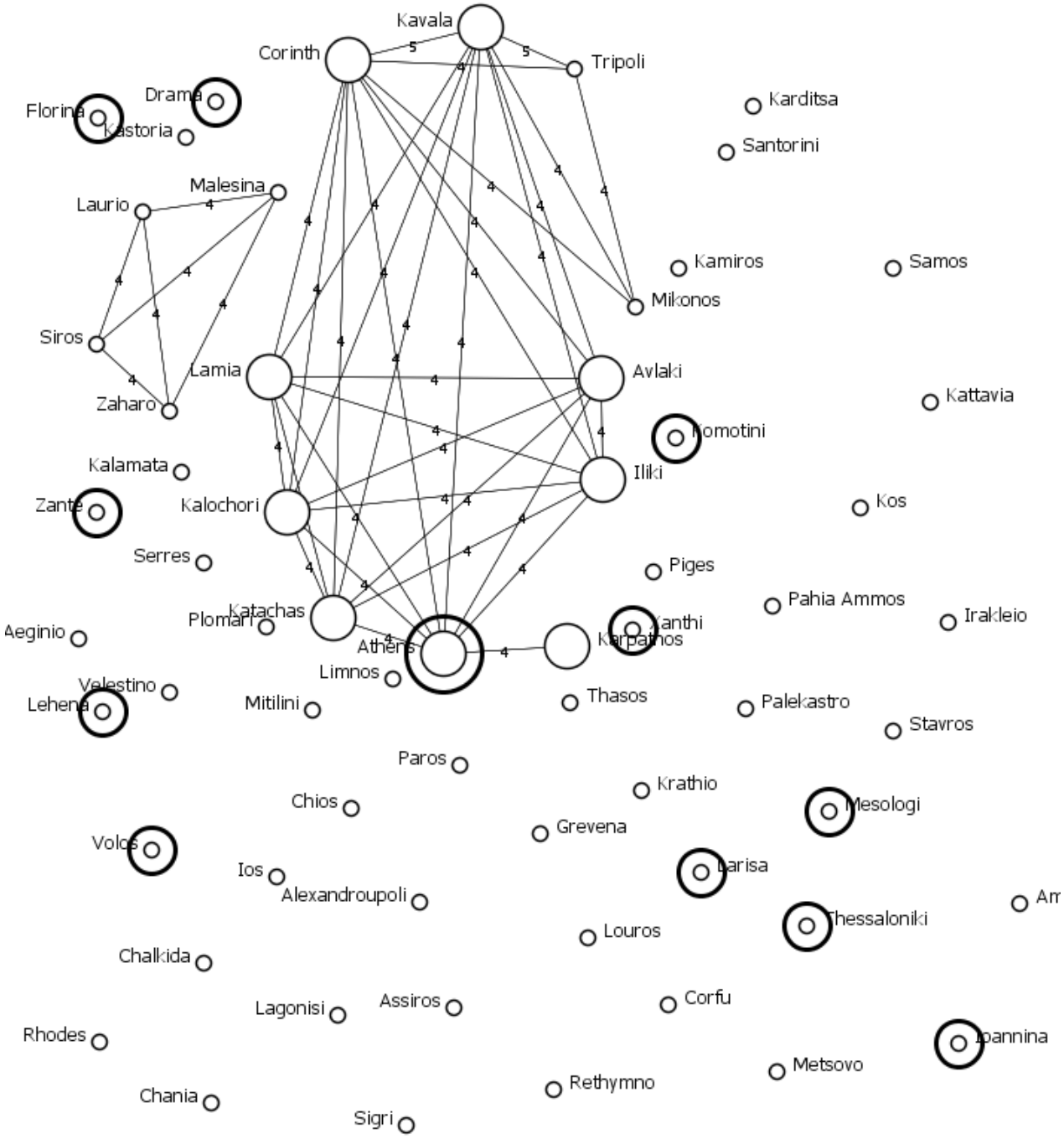

Figure 13: Measurements visualization

Table 13: Impact analysis results

Nodes Removed	Diffusion		
	Before	**After**	**Change (%)**
Boundary spanners (15 nodes)	0.908	0.193	-78.76
High ranking total degree centrality (9 nodes)	0.908	0.756	-16.68
m-slices (7 nodes)	0.908	0.729	-19.72
All prioritized nodes (27 nodes)	0.908	0.118	-86.96

The results of the analysis show that removing the prioritized nodes the diffusion of the social network (model of the log management infrastructure) is highly impacted. The decrease of diffusion is interpreted as a decrease in the capability of a malware to spread throughout the infrastructure, of an attacker to pivot using vulnerable systems, as well as a measurement of the effectiveness of a threat mitigation plan.

4.4 Results and Discussion

The threat landscape is constantly evolving with the emergence of new threats that target known or new technologies. Security mechanisms try to strengthen the defenses of infrastructure while the adversaries advance their capabilities to overcome them. Most log management infrastructures are currently being designed based on technical criteria that do not consider the actual threats the organization faces or is about to face. On large scale infrastructures the problem is more intense, due to the multitude of assets, the design complexity, as well as the difficulty in identifying the risks the organization faces. The proposed methodology provides the means for the fast and accurate analysis of an organization’s risks and for making design decisions based on the network’s structural properties and measurements. It correlates the risk assessment (result of an adequate methodology) with the design of the log management infrastructure, leveraging the concepts and

techniques of SNA, allowing for the adjustment of the log management infrastructure to better address the current risk. Affiliation network analysis provides the means for such correlation and analysis, while assisted with SNA software tools the measurements can be easily performed and validated. During this research the focus was on assets affiliated through common risks; researching on risks affiliated through common or similar assets would also be interesting, aiming to identify the key risks and properly prioritize their mitigation.

5 Conclusions

5.1 Summary of findings and contributions

Designing a log management infrastructure in large-scale infrastructure is a demanding task due to its complexity. Such an infrastructure is usually extended and distributed to various locations across a country or even across the globe. Any device generating log records is of interest for the organization owning the infrastructure and the need to secure it is of great significance. Surveys conducted by relevant organizations and institutions inform us that log management is a common practice, for most organizations. Information technology personnel spends valuable time examining logs aiming to detect and track suspicious behavior, support routine operations and forensics analysis, among other Despite their efforts and assignment of resources, achieving visibility is still a challenging task. The emergence of new technologies and business models, results to the emergence of new threats and risks for organizations. New technologies, such as “the cloud”, have raised new issues for security professionals as well as new challenges for log management. In order to support the business process, an organization needs to constantly monitor its security state, thus its infrastructure, in a timely manner that needs to be as close as possible to real-time.

This work was motivated by the problem of performing real-time security monitoring on a large-infrastructure that consists of heterogeneous and geographically dispersed devices. The objective was to improve the security monitoring capabilities of such an organization, enabling its accurate and timely security awareness in the cyber domain. Initiating this research three possible topics of research where identified: log management; log analysis and; log visualization. The topic of log management was selected among them due to the challenges it reportedly poses to organizations. The amount of logs generated is constantly increasing, on the other hand, the log management resources remain limited. This poses the challenge of balancing the increasing amount of data with the limited resources, aiming to achieve the optimal results and performance. The research aimed to

address the research questions of: how to design a log management infrastructure; how to validate the design of a log management infrastructure and; how to evolve the design of the log management infrastructure, to adopt the changing threat landscape and requirements of the organization. The focus of the research was on large-scale infrastructures, though the resulting artifacts are applicable to infrastructures of any size. The scope included both high-level and low-level aspects of log management, from the definition of the requirements through to the measurement of the infrastructure's performance, aiming to provide the designers/analysts with step-by-step guidance on designing such infrastructures. On the other hand, log analysis and visualization where not addressed excluding them from the scope of this work.

The research methodology that was employed is the DSRM, which is composed of the five following steps: explicate problem; design requirements; design and development of artifacts; demonstration of artifact and; evaluation of artifact. The resulting artifacts can be grouped to the artifact categories of systems design, methods, guidelines, requirements and, metrics, while their evaluation and validation was performed using the patterns of simulation, metrics and demonstration.

This work started reviewing the available literature and industry common practices, and resulted to the proposal of a methodology for the design of a log management infrastructure that could be used as a step-by-step guide in a large-scale infrastructure. The resulting methodology addresses both technical and managerial issues of log management, from the log source through the collection of the log data to a central location. This was achieved introducing the topic of SNA to log management, whose concepts and techniques were used to justify the design decisions, which were formerly based on intuition and experience. SNA is a scientific area focused on the study of relations, and since its metrics and techniques are based on graph theory, they are applicable regardless of the type of nodes in the network or the reason for the connections. The methodology is composed of eleven steps, that guide the implementation of an infrastructure and properly justify the design decisions about the placement of the log collectors, the assignment of log generators to log collectors, as well as the design of the time synchronization infrastructure. The SNA measurements of centrality (total degree, closeness, betweenness, eigenvector) are used to address design criteria commonly found in literature, facilitating the analysis of large and complex infrastructures. These measurements allow for the justified and documented design decisions, that formerly where based on intuition on vendors' practices, and to demonstrate its wokrings the methodology in applied on the infrastructure of a real organization.

Having designed a log management infrastructure is not enough. The implementation of such as infrastructure is financed in order to support the business process the organization's operation in total. It is supposed to fulfill a set of requirements and this ability has to be validates every time that something changes through out the organization (requirements, equipment, personnel, topology, etc). Leveraging the topic of MNA (an extension of SNA) personnel, tasks and log data, where correlated and analyzed to validate that under a specific design, the log management infrastructure provides its personnel the means to perform their tasks, and that its resources are optimally used. MNA was first used to analyze the performance of organizations, based on the assumption that each person is assigned tasks, while to perform a task it needs access to certain knowledge or resources. Approaching a log management infrastructure as a complex organization, a log collector was mapped to a person, a log file was mapped to knowledge, and a log management task was mapped to a task. Performing MNA measurements the log collectors that did not collect the necessary logs, or collected unnecessary logs, where identified, allowing for the application of the proper corrective actions on the design of the log management infrastructure. The proposed methodology is flexible allowing to both validate the current design, or perform what-if analysis to whichever combination of infrastructure design and log management requirements.

Motivated by the constant evolution of the threat landscape, there was identified the need, for the design of a log management infrastructure, to be adoptive to the evolving business and risks environment. The continues evolution of business models and related technologies often render security mechanisms to be ineffective, as they were designed and implemented to protect from different threats. Addressing the problem, this research resulted in another methodology for the design of log management infrastructure that allows for a log management infrastructure to be optimally designed for the actual threat landscape it operates in. The concepts and techniques used to analyze affiliation networks in SNA were employed to correlate the risks that affect the assets of a log management infrastructure with its design. In human affiliation networks, when two actors participate to the same event a link is created among them. Correspondingly, the assets of the infrastructure where considered to the the actors and the risks where to considered to be the events. As such, whenever two assets faced a common risk, an implicit link was created among them. Leveraging the SNA measurements used for the analysis of affiliation networks, it was made feasible to identify the assets (nodes) of the infrastructure that were critical due the risks they faced, properly adapting the infrastructure's design and prioritizing incident response. This enables the designer/analyst to validate that the infrastructure is optimally designed to address the current threat landscape, as well as to perform what-if analysis whenever the threat landscape and/or log

management infrastructure design change.The workings of the methodology were demonstrated on the infrastructure of a real organization that operates a large-scale infrastructure. Through out this research the performance of the tasks was facilitated by the use of SNA software that enables the analysis of large social networks, consisting of thousands of nodes and links, with greate speed and visual assistance.

5.2 Limitations

This research reached its aims though it poses some limitations. The work performed during this research included the evaluation of the proposed methodologies using the infrastructure of the GRNET S.A. and the data publicly available on its website. In some cases when the data necessary for research were not available proper assumptions were made in order to demonstrate the workings of the methodologies. The research depends largely on SNA, and its extensions, to perform the various analysis. Thus the designers of a log management infrastructure needs an understanding of the relevant concepts and techniques in order to apply the methodologies and properly use the software tools. The research is limited to the design of log management infrastructures, it does not address issues related to log analysis, such as intrusion detection, log correlation or visualization.

5.3 Future work

Having identified the limitations of this research, this work could continue focusing on automation. The methodologies proposed and evaluated herein, require for the collection of various data about the organization’s infrastructure in order to generate the required inputs. The automation of the collection process would greatly benefit the methodologies as one the aims to perform security monitoring in as real-time as possible. The performance of various measurements and execution of algorithms could also be automated contributing too. The methodologies as presented through the cases studies in the preceding sections, rely on the presence of a human analyst, that evaluates analysis results. The role of the analyst could also be automated, to a certain extent, compiling various playbooks that could guide the cases where prior knowledge exists. The research could also continue focusing on industrial control systems or the Internet of Things.

Appendix: Publications

V. Anastopoulos and S. Katsikas, “A Methodology for Building a Log Management Infrastructure,” presented at the 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Noida,India, 2014, pp. 000301–000306.

V. Anastopoulos and S. Katsikas, “Design of a Log Management Infrastructure Using Meta-Network Analysis,” in *Katsikas S., Lambrinoudakis C., Furnell S. (eds) Trust, Privacy and Security in Digital Business*, Porto, Portugal, 2016, vol. 9830, pp. 97–110.

V. Anastopoulos and S. Katsikas, “A structured methodology for deploying log management in WANs,” *JISA*, vol. 34(2), pp. 120–132, Jun. 2017.

V. Anastopoulos and S. Katsikas, “Design of a Dynamic Log Management Infrastructure Using Risk and Affiliation Network Analysis,” in *Proceedings of the 22nd Pan-Hellenic Conference on Informatics*, Athens, 2018, pp. 52–57.

V. Anastopoulos and S. Katsikas, “A Methodology for the Dynamic Design of Adaptive Log Management Infrastructures,” *EAI*, Jan. 2019.

Bibliography

[1] P. Johannesson and E. Perjons, *An Introduction to Design Science*. Springer] International Publishing, 2014.
[2] S.-O. Olusola, D. Shimabukuro, S. Chatterjee, and M. Muthui, “Meta-analysis of Design Science Research within the IS Community: Trends, Patterns, and Outcomes,” 2010, pp. 124–138, doi: 10.1007/978-3-642-13335-0_9.
[3] Vijay K. Vaishnavi and William Kuechler, *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*, 2nd ed. CRC Press, Inc. Boca Raton, FL, USA, 2015.
[4] Philipp Offermann, Sören Blom, Marten Schönherr, and Udo Bub, “Artifact Types in Information Systems Design Science – A Literature Review,” presented at the Global Perspectives on Design Science Research, 2010, vol. 6105, doi: https://doi.org/10.1007/978-3-642-13335-0_6.
[5] V. Anastopoulos and S. Katsikas, “A Methodology for Building a Log Management Infrastructure,” presented at the 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Noida,India, 2014, pp. 000301–000306.
[6] V. Anastopoulos and S. Katsikas, “Design of a Log Management Infrastructure Using Meta-Network Analysis,” in *Katsikas S., Lambrinoudakis C., Furnell S. (eds) Trust, Privacy and Security in Digital Business*, Porto, Portugal, 2016, vol. 9830, pp. 97–110, doi: https://doi.org/10.1007/978-3-319-44341-6_7.
[7] V. Anastopoulos and S. Katsikas, “A structured methodology for deploying log management in WANs,” *Journal of Information Security and Applications*, vol. 34(2), pp. 120–132, Jun. 2017, doi: 10.1016/j.jisa.2017.02.004.
[8] V. Anastopoulos and S. Katsikas, “Design of a Dynamic Log Management Infrastructure Using Risk and Affiliation Network Analysis,” in *Proceedings of the 22nd Pan-Hellenic Conference on Informatics*, Athens, 2018, pp. 52–57, doi: 10.1145/3291533.3291570.
[9] V. Anastopoulos and S. Katsikas, “A Methodology for the Dynamic Design of Adaptive Log Management Infrastructures,” *European Alliance for Innovation Endorsed Transactions on Security and Safety*, Jan. 2019, doi: 10.4108/eai.25-1-2019.159347.
[10] J. Shenk, “SANS Eighth Annual 2012 Log and Event Management Survey Results: Sorting Through the Noise,” SANS, May 2012.
[11] K. Kent and M. Souppaya, “NIST SP 800-92, Guide to Computer Security Log Management - SP800-92.pdf.” 2006.
[12] Chris Sanders and Jason Smith, *Applied Network Security Monitoring*, 1st ed. Syngress, 2014.
[13] Anton A. Chuvakin, Kevin J. Schmidt, and Christopher Phillips, *Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management*, 1st ed. USA: Syngress, 2013.

[14] G. Kunlun, L. Jianming, G. Jian, and A. Rui, “Study on data acquisition solution of network security monitoring system,” presented at the 2010 IEEE International Conference on Information Theory and Information Security, 2010, doi: 10.1109/ICITIS.2010.5689487.
[15] M. Afsaneh, R. Saed, and G. Hossein, “Log management comprehensive architecture in Security Operation Center (SOC),” presented at the 2011 International Conference on Computational Aspects of Social Networks (CASoN), 2011, doi: 10.1109/CASON.2011.6085959.
[16] S. Alspaugh, “Analyzing Log analysis: an empirical study of user log mining 28th large installation system administration conference,” presented at the LISA 14, Seattle, 2014.
[17] “SIEM, AIOps, Application Management, Log Management, Machine Learning, and Compliance,” *Splunk*. [Online]. Available: https://www.splunk.com. [Accessed: 05-Apr-2019].
[18] A. Tomono, M. Uehara, and Y. Shimada, “Trusted log management system,” presented at the Trustworthy ubiquitous computing, 2012, pp. 79–98.
[19] R. Gerhards <rgerhards@adiscon.com>, “RFC5424 The Syslog Protocol.” [Online]. Available: https://tools.ietf.org/html/rfc5424. [Accessed: 08-Apr-2019].
[20] M. Uehara, “A Toolkit for Virtual Large-Scale Storage in a Learning Environment,” presented at the 21th International Conference on Advanced Information Networking and Applications Workshops/Symposia 2007, 2007, vol. 1, pp. 888–893.
[21] A. Murugan and T. Kumar Kala, “An Effective Secured Cloud Based Log Management System Using Homomorphic Encryption,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2268–2271, 2014.
[22] P. Anil and R. Sagar, “Development of Highly Secured Cloud Rendered Log Management System,” *International Journal of Computer Applications*, vol. 108, no. 16, 2014.
[23] M. Ficco, “Security event correlation approach for cloud computing.,” *International Journal of High Performance Computing and Networking*, vol. 7, no. 3, pp. 173–158, 2013.
[24] K. Faust and S. Wasserman, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
[25] Wouter De Nooy, Andrej Mrvar, and Vladimir Batagelj, *Exploratory Network Analysis with Pajek*, 2nd ed. Cambridge University Press, 2011.
[26] Borgatti SP, “The key player problem,” presented at the Dynamic social network modeling and analysis: workshop summary and papers, 2003.
[27] S. Borgatti, “Identifying sets of key players in a social network,” *Computational & Mathematical Organization Theory*, vol. 12, no. 1, pp. 21–34, 2006.
[28] “NIST SP 800-55 rev.1, Performance measurement guide for information security.” National Institute of Standards and Technology, 2008.
[29] A. Jaquith, *Security metrics: replacing fear, uncertainty, and doubt*. Addison-Wesley, 2007.
[30] “CIS Controls™,” *CIS*. [Online]. Available: https://www.cisecurity.org/controls/. [Accessed: 11-May-2019].
[31] Joint Task Force Transformation Initiative Interagency Working Group, “NIST SP 800-30, Rev.1 Guide for Conducting Risk AssessmeNnts.” Sep-2012.
[32] “Effective Use Case Modeling for Security Information & Event Management,” p. 19, 2010.
[33] Harris S, Harper A, VanDyke S, Black C, and Miller D, *Security Information and Event Management (SIEM) Implementation*. McGraw-Hill, 2011.
[34] J. Postel, “RFC793 Transmission Control Protocol.” [Online]. Available: https://tools.ietf.org/html/rfc793. [Accessed: 11-May-2019].
[35] “RFC3195 Reliable Delivery for syslog.” [Online]. Available: https://www.ietf.org/rfc/rfc3195.txt. [Accessed: 11-May-2019].
[36] D. Clayton, “Building Scalable Syslog Management Solutions.” CISCO, 2011.

[37] D. Harrington, B. Wijnen, and R. Presuhn, “RFC3411 An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks.” [Online]. Available: https://tools.ietf.org/html/rfc3411. [Accessed: 11-May-2019].
[38] K. M. Carley and J. Reminga, “ORA: Organization Risk Analyzer*,” Carnegie Mellon University School of Computer Science, Institute for Software Research International, CASOS Technical Report CMU-ISRI-04-106, Jul. 2004.
[39] Linton C. Freeman, “Centrality in Social Networks Conceptual Clarification,” 1979, pp. 215–239.
[40] T. Frantz, “Annual tools/computational approaches/methods conference,” Carnegie Mellon University, 2008.
[41] Phillip Bonacich, “Power and Centrality: A Family of Measures,” *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.
[42] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Phys. Rev. E*, vol. 70, no. 6, p. 066111, Dec. 2004, doi: 10.1103/PhysRevE.70.066111.
[43] “The rocket-fast Syslog Server - rsyslog.” [Online]. Available: https://www.rsyslog.com/. [Accessed: 11-May-2019].
[44] “MySQL.” [Online]. Available: https://www.mysql.com/. [Accessed: 11-May-2019].
[45] Jack Burbank, David Mills, and William Kasch, “RFC5905 Network Time Protocol Version 4: Protocol and Algorithms Specification.” [Online]. Available: https://tools.ietf.org/html/rfc5905. [Accessed: 11-May-2019].
[46] “IEEE 1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.” [Online]. Available: https://standards.ieee.org/standard/1588-2008.html. [Accessed: 11-May-2019].
[47] “Network Time Protocol: Best Practices White Paper,” *Cisco*. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/availability/high-availability/19643-ntpm.html. [Accessed: 11-May-2019].
[48] T. Mizrahi <talmi@marvell.com>, “RFC7384 Security Requirements of Time Protocols in Packet Switched Networks.” [Online]. Available: https://tools.ietf.org/html/rfc7384. [Accessed: 11-May-2019].
[49] D. L. Mills and B. Haberman, “RFC5906 Network Time Protocol Version 4: Autokey Specification.” [Online]. Available: https://tools.ietf.org/html/rfc5906. [Accessed: 11-May-2019].
[50] “GRNET Topology,” 21-Jan-2019. [Online]. Available: https://grnet.gr/infrastructure/network-and-topology/. [Accessed: 21-Jan-2019].
[51] “Cisco - Global Home Page,” *Cisco*. [Online]. Available: https://www.cisco.com/c/en/us/index.html. [Accessed: 11-May-2019].
[52] “Juniper Networks - Networking & Cybersecurity Solutions.” [Online]. Available: https://www.juniper.net/us/en/. [Accessed: 11-May-2019].
[53] “Customer-Driven Enterprise Networking Solutions - Extreme Networks.” [Online]. Available: https://www.extremenetworks.com/. [Accessed: 11-May-2019].
[54] “Homepage | CASOS.” [Online]. Available: http://www.casos.cs.cmu.edu/. [Accessed: 11-May-2019].
[55] D. Krackhardt and K. Carley, *PCANS model of structure in organizations*. 1998.
[56] K. Carley, “Computational organizational science and organizational engineering,” *Simulation Modelling Practice and Theory*, vol. 10, no. 5, pp. 253–269, 2002.
[57] L. Yongkui, L. Yujie, L. Dongyu, and M. Liang, “Metanetwork Analysis for Project Task Assignment,” *Journal of Construction Engineering and Management*, vol. 141, no. 12, 2015.
[58] T. Wakolbinger and A. Nagurney, “Dynamic supernetworks for the integration of social networks and supply chains with electronic commerce: modeling and analysis of buyer–seller

relationships with computations," *NETNOMICS: Economic Research and Electronic Networking*, vol. 6, no. 2, pp. 153–185, 2004, doi: https://doi.org/10.1007/s11066-004-4339-x.

[59] A. Nagurney, T. Nagurney, and L. Zhao, "The Evolution and Emergence of Integrated Social and Financial Networks with Electronic Transactions: A Dynamic Supernetwork Theory for the Modeling, Analysis, and Computation of Financial Flows and Relationship Levels," *Computational Economics*, vol. 27, no. 2–3, pp. 353–393, 2006, doi: https://doi.org/10.1007/s10614-006-9031-9.

[60] A. Nagurney and J. Dong, "Management of knowledge intensive systems as supernetworks: Modeling, analysis, computations, and applications," *Mathematical and Computer Modelling*, vol. 42, no. 3–4, pp. 397–417, 2005, doi: https://doi.org/10.1016/j.mcm.2004.01.015.

[61] L. R. Tucker, "Implications of factor analysis of thee-way matrices for measurement of change," in *Problems in Measuring Change*, Chester William Harris., University of Wisconsin Press, 1963, pp. 122–137.

[62] K. M. Carley, J. Pfeffer, J. Reminga, J. Storrick, and D. Columbus, "ORA User's Guide 2013," Institute for Software Research School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213, CMU-ISR-13-108, Jun. 2013.

[63] J.-S. Lee and K. M. Carley, "OrgAhead: A Computational Model of Organizational Learning and Decision Making," Carnegie Mellon University, School of Computer Science, Institute for Software Research International, Pittsburgh, Technical Report CMU-ISRI-04-117, 2004.

[64] Christopher Crowley, "Future SOC: SANS 2017 Security Operations Center Survey," May-2017.

[65] Jerry Shenk, "Ninth Log Management Survey Report," Oct. 2014.

[66] Mauro Faccioni Filho, "Complex Systems: Risk Model Based on Social Network Analysis," presented at the 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE), Santa Clara, CA, USA, 2016, doi: 10.1109/ISIE.2016.7744859.

[67] Xiaoyan Ge, "Key Element Identification in Cooperative Technological Innovation Risk on Social Network Analysis," presented at the 2014 Seventh International Joint Conference on Computational Sciences and Optimization (CSO), Beijing, China, 2014, doi: 10.1109/CSO.2014.66.

[68] Citra S. Ongkowijoyo and Hemanta Doloi, "Understanding of Impact and Propagation of Risk based on Social Network Analysis," in *Science Direct, Procedia Engineering*, Bangkok, Thailand, 2017, vol. 212, pp. 1123–1130.

[69] "Recent Advances in Modelling Systemic Risk Using Network Analysis," Jan-2010.

[70] Zhengqi He, Dechun Huang, Changzheng Zhang, and Junmin Fang, "Toward a Stakeholder Perspective on Social Stability Risk of Large Hydraulic Engineering Projects in China: A Social Network Analysis," *Project Management and Sustainable Development*, vol. 10, no. 4, 2018, doi: https://doi.org/10.3390/su10041223.

[71] CristianaMaurella, Gianluca Mastrantonio, and Silvia Bertolini, "Social network analysis and risk assessment: An example of introducing an exotic animal disease in Italy," in *Microbial Risk Analysis*, 2019, doi: https://doi.org/10.1016/j.mran.2019.04.001.

[72] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 2nd ed. Massachusetts London, England: The MIT Press Cambridge, 2001.

[73] Paul Cichonski, Tom Millar, Tim Grance, and Karen Scarfone, "NIST SP 800-61, Computer Security Incident Handling Guide, Rev.2-SP800-61.pdf." Aug-2012.

[74] "Carnegie Mellon University," *Carnegie Mellon University*, 2018.