EAI Endorsed Transactions
on Security and Safety

Research Article EAI.EU

A Methodology for the Dynamic Design of Adaptive Log Management Infrastructures

V. Anastopoulos[1,*] and S. Katsikas[2,3]

[1] Department of Digital Systems, University of Piraeus, Piraeus, Greece
[2] Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Gjøvik N-2802, Norway
[3] Faculty of Pure and Applied Sciences, Open University of Cyprus, Nicosia, Cyprus

Abstract

Organizations collect log data for various reasons, including security related ones. The multitude and diversity of the devices that generate log records increases, resulting to dispersed networks and large volumes of data. The design of a log management infrastructure is usually led by decisions that are commonly based on industry best practices and experience, but fail to adapt to the evolving threat landscape. In this work a novel methodology for the design of a dynamic log management infrastructure is proposed. The proposed methodology leverages social network analysis to relate the infrastructure with the threat landscape, thus enabling it to evolve as threats evolve. The workings of the methodology are demonstrated by means of its application for the design of the log management infrastructure of a real organization.

Keywords: log management, social network analysis, organizational risk analyzer, risk.

Received on 01 December 2018, accepted on 12 January 2019, published on 29 January 2019

doi: 10.4108/eai.25-1-2019.159347

*Corresponding author. Email: Vasanasto@gmail.com

1. Introduction

Most organizations, in order to comply with standards and legislation, or for ensuring efficient maintenance, troubleshooting and security, need to maintain data logs. Because current technology allows it, at low cost, the types of log sources continuously increase; this raises the challenge of managing and analyzing large amounts of data, as well as modifying log management methodologies [1]. Automation facilitates the performance of log management tasks; yet, due to security policy restrictions, geographic dispersion and operational costs, security personnel have to carry out their tasks without having all of the necessary data at their disposal [2].

Recent reports e.g. [3] describe an ever evolving and dynamic threat landscape, where attack and defense is an arms race. Combined with emerging business models and new technologies, new threats also emerge, posing new challenges, that may render current security controls ineffective [4]. Even though log management exhibits dynamic characteristics, organizations implement log management infrastructures by making design decisions based solely on technical criteria, neglecting the threat landscape they will operate in.

The design of a log management infrastructure for a Wide Area Network (WAN) is a demanding task, particularly when the network is dispersed and heterogeneous. As stated in [5], "a log management infrastructure consists of the hardware, software, networks, and media used to generate, transmit, store, analyze, and dispose of log data", and it typically comprises three tiers: the *Log Generation* tier that contains the hosts that generate the log data, the *Log Analysis and Storage* tier, which is composed of one or more log servers, often called *collectors* or *aggregators*, that receive log data or copies of log data from the hosts in the first tier; and the *Log Monitoring* tier, that contains

consoles that may be used to monitor and review log data and the results of automated analysis [5],[6].

The placement of the log collectors is a design problem that is usually solved by following vendors' guidelines and recommended good practices. For example, in [6] a hierarchical placement of the log management components is proposed, placing the collectors close to the log originators, in a hub-and-spoke architecture, considering also the geographic dispersion of the WAN, with the aim of collecting the log data to a central point for analysis. In [5] the complexity and variety of the log analysis and storage tier is handled by deploying multiple log servers, each performing specific analysis, or storage functions, for specific log generators. Thus, each log collector of a large scale infrastructure can store or process only part of the generated log data, either because of the design criteria or because of lack of resources; the volume of the generated log data may prohibit its storage or transmission to a single central location, due to the incurred economic cost. In this context, security staff working in different locations and performing varying duties, need access to specific log data sources to accomplish their assigned log management tasks in order to meet the requirements. Adjusting the design of the log management infrastructure, though necessary, impacts the availability of the log data (collectors may be added or removed, log generators may be redirected, the retention policy may change, etc.) possibly hindering the accomplishment of the log management tasks.

The problem addressed in this work is how to design a dynamic log management infrastructure for a WAN, able to adapt its design as the threat landscape evolves, while continuing to meet the set of operational log management requirements.

The work presented in this paper is motivated by the need for a methodology that does not limit itself to the evaluation and documentation of technical design decisions, but also takes account of the dynamic threat landscape into the design decision making process, and further validates the alignment of the log management infrastructure to the log management requirements. Validation herein means ensuring that an analyst accessing a log collector can perform the required log management tasks using the log data actually collected on it; if not, a measure of the lack of data, or excess of data, occurs.

In this work we adopt the risk definition of [4]: "a measure of the extent to which an entity is threatened by a potential circumstance or event". This definition includes all aspects of risk that could possibly affect a log management infrastructure (physical security, business processes, external relationships, human factor, etc) and is not limited to software related risks. The relationships among log data, log collectors and log management tasks, comprise the *design structure* of the log management infrastructure, which can be modeled as a collection of interlocked networks, called a *Meta-Network*. A Meta-Network can be represented using the Meta-Matrix conceptual framework [8], enabling the extension of the Social Network Analysis (SNA) techniques and concepts [9] to those of Meta-Network Analysis (MNA) [10] that are used to analyze the structural properties of real-world organizations. A log management infrastructure is viewed as a complex organization, enabling the application of MNA concepts and measures.

The proposed methodology leverages the techniques of Social Network Analysis (SNA) to analyze the design of the log management infrastructure in relation to the risks that its assets face and results in a new design that mitigates those risks. It then applies concepts and measures provided by Meta-Network Analysis, to analyze the design structure of the resulting infrastructure to confirm that the ability to accomplish the required log management tasks is maintained. Combining the SNA and MNA results in an optimal design of the log management infrastructure, customized to the risks it faces.

The contribution of this work is a novel methodology that

- allows the evolution of a log management infrastructure whenever the threat landscape, the log management requirements or the logged infrastructure changes;
- takes into account the security risk when making design decisions on the log management infrastructure;
- leads to a design optimized for the threat landscape in which the infrastructure operates;
- validates the fulfillment of the design requirements;
- provides the security personnel with measurements of the effectiveness of the design, that guide and document the design decisions.

The remainder of this paper is organized as follows: In Section 2 related work is discussed, in Section 3 the proposed methodology is presented, and in Section 4 its application to a log management infrastructure of a real organization is demonstrated. In Section 5, the conclusions are summarized and directions for future work are presented.

2. Related work

In [11] the authors present a high level guide for building a log analysis system, where the organizational risk is considered for defining the log retention policy and the log storage requirements. The four-step process for the collection of log data proposed in [12] starts with the definition of the threats that an organization faces and continues with prioritizing them, based on the risk they induce for the organization. It then identifies the data feeds that are required to address each threat and concludes by further analyzing the selected sources. A high level perspective of the design process is also provided in [5], where an organization prioritizes its goals and log management requirements according to the perceived reduction of risk and the required resources. Additionally, [13] argues that, in order to allow for sensible security monitoring that avoids gaps and

excessive controls, the level of monitoring of information processing facilities should result from a risk assessment exercise.

Functions related to log collection and storage are discussed in [14], where a method that ensures the forensic soundness of log data transferred over untrusted networks is proposed. Log collection and storage functions are also addressed in [15], that proposes a log management architecture in conjunction with commercial SIEM products. In [16] the authors propose a framework for log management in distributed systems, aiming to address log-based anomaly detection and problem identification tasks. Though [16] aims to provide an end-to-end log management framework, no validation of the resulting design is offered, and its application is not demonstrated. In [17] the log management task of preprocessing is addressed. The current state of the art of log parsers is evaluated in terms of efficacy, using real-world datasets composed of millions of log messages. The authors designed and implemented a parallel log parser, to address the deficiencies of current log parsers when logs grow to a large scale. The system's performance in log mining tasks, in terms of accuracy, efficiency and effectiveness was evaluated. In [18] the authors present an integrated system that aims to perform data analytics on system logs from complex networks and high-throughput web-based applications. It utilizes data mining techniques to assist the analysts in conducting log knowledge discovery, system failure diagnosis and system status investigation. A web-based framework for the analysis of log files provided by the user is proposed in [19]; it aims to correlating events in huge log files. The authors developed a prototype that parses log files, based on text phrases selected by the user and visualized the results of the analysis. Indicative works of delegating log management to the cloud are [20] and [21]. Block chain technology is leveraged in [22], where the authors propose a secure log storage platform that uses block chain on the cloud to achieve the integrity of the log data and the log process, as well as scalability for longer retention and deep analysis.

Social network analysis is based on the assumption that relationships among interacting units are of importance. The network perspective encompasses theories, models, and applications that are expressed in terms of relational concepts or processes. In [9] and [23] methods and measurements for the analysis of social networks are documented along with their interpretation; the latter focuses on exploratory analysis, whilst in both works the identification of the key nodes is performed using measures of centrality. The inefficiency of these measures in identifying important and key nodes, is addressed in [24] and [25], that introduce new methods of analysis.

SNA is used in [26] to build a model for the recognition of the key risk elements in cooperative technological innovation. The authors follow a three-step analysis process (factor analysis, relations analysis, matrix analysis) to identify the key risk elements. Our work is different, as it analyzes the relations among the assets as the result of facing common risks rather than their direct linkage to risks. The work presented in [27] employs SNA to explore the dynamics of risk causality and interaction patterns, using both participatory and computerized techniques. A risk analysis model is proposed in [27], that allows to capture, model and simulate the capacity of risk interactions in respect to the network structure; SNA visualization tools are leveraged to gain visibility into risk characteristics. In contrast, our work leverages SNA to integrate risk complexity into the process of designing a log management infrastructure, instead of only modeling it. The authors in [28] propose a theoretical model for risk analysis of complex systems. The model is based on SNA and characterizes risk as failure rates on network links and nodes, aiming to evaluate the risk of the whole complex system in real-time; however, the applicability of the model is not demonstrated. 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 studied in [29], by applying SNA measurements. A model for capturing, drawing and simulating the risk impact propagation patterns and interrelationships is proposed in [27]. The proposed model is applied on a water supply infrastructure system, revealing the value of participatory networked approaches in capturing the intricate processes that shape infrastructure risk. SNA is also used to study risk in large hydraulic engineering projects in [30]. The authors combined stakeholder management with risk management, to provide a reference for the social stability risk management. Additional applications of SNA in studying risk are presented in [31] and [32] where the risk of spreading a disease is discussed.

SNA usually handles networks composed of nodes of the same type, such as agent networks or task networks. Its 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 [9]. Therefore, 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*, i.e. the relationships among different types of nodes: log collectors, log files and log management tasks. MNA, on the other hand, extends SNA and enables its application to complex cross-connected networks composed of multiple types of nodes. Meta-networks were first described by means of the precedence, commitment of resources, assignment, networks, and skills (PCANS) model [33]. They involve key entities that influence organizational design, such as tasks, resources, knowledge, and agents, as well as their relations [34] and have been applied to diverse fields [35],[36],[37],[10].

A guide for implementing a log management infrastructure in WANs by applying SNA to justify design decisions that where formerly made based on intuition or experience is presented in [7]. It encompasses both high-level and low-level aspects of log management, and guides the design, implementation and evaluation of such

infrastructure, following an eleven-step method. However, the method in [7] does not alow the design of **dynamic** log management infrastructures. An extension to the work in [7] was presented in [38] where MNA was used to dynamically design log management infrastructures.

The work herein further consolidates that in [38] and [7] to propose a complete methodology for the design and validation of risk-adaptive log management infrastructures and to demonstrate its workings.

3. Adaptive log management infrastructure design methodology

In SNA a *node* (or actor) is a social entity that can be a discrete social unit (e.g. a person) or a collective social unit (e.g. a corporate department); the term *actor* does not imply that it has the ability to act. The actors are connected by establishing *links* (or social ties) and the collection of social ties formed among a specific set of actors is a *relation*. A *social network* is composed of nodes and links and can be either directed, i.e. the link from node A to node B is different from the link from node B to node A, or undirected. A node can have attributes and a link can be valued or binary. In graph theory notation, G = (V, E) is a social network G with |V| nodes and |E| links and it is represented by a |V| x |V| adjacency matrix. 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. When the links are formed among the nodes of the same set, the network is a one-mode social network [39]; the term *mode* refers to a distinct set of entities on which the structural variables are measured. A two-mode network is formed between two distinct sets of nodes, N and M, and is represented by the |N| x |M| incidence matrix. Folding the two-mode network results in two one-mode networks, one for each dimension. A network is folded when it is first transposed to the desired dimension and then multiplied 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.

The infrastructure of a WAN comprises a variety of devices and equipment that in the context of this work are referred to as *assets*. Assets are not limited to network equipment; they also include operating systems and applications, physical security mechanisms etc. When two assets face the same risk (they have a common vulnerability and there exists a threat that can exploit it), they are implicitly connected.

An overview of the proposed methodology is depicted in Figure 1, while its components are discussed in detail in the subsequent sections.

3.1 Adjustment of the dynamic log management infrastructure design

Affiliation networks are two-mode networks where the first mode is a set of actors and the second mode is a set of events. The linkages among members of one of the modes are based on the linkages established through the second mode [9]. An event can be a wide range of occasions, such as participation to a club, a party or a committee and it does not necessarily correspond to a face-to-face interaction. An actor belonging to a club is affiliated with that event. When two actors belong to the same club they are affiliated (linked) by the same event. A two-mode matrix, |A| x |E|, is used to represent an affiliation network, where a value of 1 in the *ij* cell affiliates row actor *i* to column event *j* [9]. Folding this two-mode matrix results in an array of linkages among actors through their participation to events, |A| x |A|, and an array of linkages among events through the participation of actors to these events, |E| x |E|. In the proposed methodology an asset corresponds to an actor and a risk corresponds to an event. Two assets are linked when they share the same risk and two risks are linked when they pertain to the same asset.

The relation among the assets of the log management

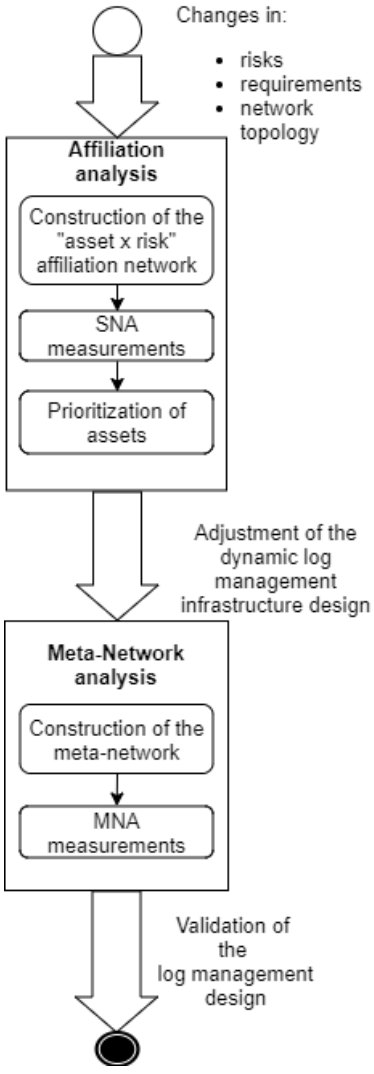

Figure 1. Overview of the proposed methodology

infrastructure and the risks they are exposed to is modeled as a two-mode social network. Let A = {a_1, a_2, …, a_n} be the node set of the assets of the log management infrastructure and let R = {r_1, r_2,…, r_m} be the node set of the risks that these assets face. When an asset a_i is threatened by risk r_j, the entry in the *ij* cell of the |A| x |R| incidence matrix is equal to 1. The |A| x |R| incidence matrix is folded, resulting to the |A| x |A| and |R| x |R| arrays.

Total degree centrality is the number of links a node has and it is **used** to identify the nodes that participate actively in the social network. It is distinguished into *in* and *out* degree centrality, where the links are directed to and 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}([9] as cited in [8]). A node with high degree centrality is a well-connected node and can potentially influence directly many other nodes [24].

The total degree centrality is measured on the |A| x |A| one-mode social network and the nodes are sorted in descending order to identify the high-valued ones; these nodes (assets) face the same risks with many other nodes. A threat could pivot among them by exploiting their common vulnerabilities, or it could compromise multiple assets by exploiting a 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 [23]. The 0-slice assets are “isolated” as they share no risks with the rest of the assets, while a 4-slice, for example, is a pair of assets that have four common risks.

The m-slices are sorted in descending order, identifying the high-valued pairs. These sub groups of assets share many common risks, that result in increased - compared to the rest of the assets- attack surface for the WAN. A threat could compromise the whole sub group of assets by exploiting their multiple shared vulnerabilities.

The SNA measurements (total degree centrality and m-slices) are used to identify the high-risk assets, and to prioritize them for log management. For example, these assets could be prioritized for system hardening (preparation), deployment of sensors and log analysis (detection), minimization of the impact of a possible incident (containment), mitigation of vulnerabilities (eradication) and restore of normal operation (recovery) [40].

Based on the prioritization of the assets and the incident response measures that the organization [4] chooses to implement, the log management infrastructure is adjusted by modifying the log generation, analysis, storage or monitoring tiers, as defined in [5]. The output of this step of the proposed methodology is the design of the log management infrastructure optimized to address the risks that it currently faces.

3.2 Validation of the log management infrastructure design

The dynamic design of the log management infrastructure needs to be validated to confirm that after its adjustment it still fulfills the log management requirements for which it was designed.

Social network model construction

A log management infrastructure comprises of *log files*, *log collectors* and log *management tasks* (component of the requirements). These three entities are related as each log file is sent to one or more log collectors, each log management task needs the data contained into specific log files, and with the log data stored on a specific log collector a subset or the full set of log management tasks is expected to be carried out by the analyst. The relationship among these entities is depicted in the entity relationship diagram of Figure 2, where a log file, a log collector and a log management task are linked with many-to-many relationships.

These three entities are used to construct a three-mode

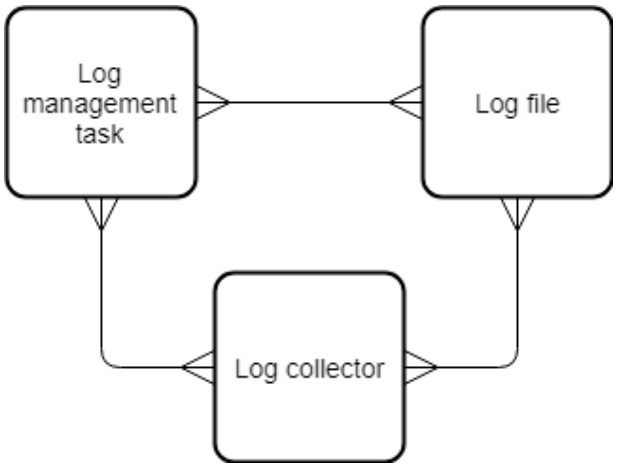

Figure 2. Entity relationships diagram

social network with the following modes:

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

The links formed among these three modes are represented by the following incidence matrices:

|F| x |T|, the log files necessary to perform each log management task.
|F| x |C|, the log collectors to which each log file is sent.

|T| x |C|, the log management tasks expected to be accomplished with the log data of each log collector.

Meta-network model construction

In MNA the design structure is composed of the following entities: *agents*, an entity that processes information; *tasks*, a set or subset of actions that accomplish an assignment; and *knowledge*, the available information [41]. Each of these entities corresponds to a *node class*, and the nodes belonging to a node class form a *node set*. The nodes can be connected with links, while both may have attributes that describe them and provide context to their relationship. When the links are formed among the nodes of the same node set, the result is a one-mode network; when they are formed among N node sets, the result is an N-mode network. Different networks may describe specific relationships among the nodes; the collection of these networks is referred to as a *meta-network*.

In the context of the proposed methodology, a log management infrastructure corresponds to an organization with agents (log collectors), knowledge (log files) and tasks (log management tasks). This analogy is summarized in Table 1, where adopting the notation of [41], AT is the *agent x task* matrix, AK is the *agent x knowledge* matrix and KT' is the *knowledge x task* transposed matrix. The constructed meta-network is analyzed by applying knowledge measurements to the organization (meta-network of the log management infrastructure), as documented in [41] and [42].

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 completing the assigned tasks. This metric measures the difference between the knowledge that the agent needs to do its assigned tasks and the agent's actual knowledge. The value of the metric is increased when the agent needs knowledge that has not been assigned to it. 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, but is not needed for any of its tasks; it is 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 perform 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,:)).

The adjustment of the log management infrastructure design, i.e. the output of the analysis of the affiliation network, may have impacted the ability to perform the log management tasks on the log collectors. This is validated using the first two metrics, which identify the log collectors that need more data for the accomplishment of their tasks, and the collectors that receive more log data than they actually need, respectively. Specific log files or entire log generators can be reassigned to log collectors, in order to verify that the latter have the necessary log data for the accomplishment of the log management tasks at their disposal.

4. Case study

The proposed methodology was applied to the infrastructure of the Greek Research and Education Network (GRNET network) [43]. The WAN of GRNET extends to most parts of Greece and provides connectivity services to academic institutions; it is composed of 78 devices with 85 physical connections among them, forming the topology depicted in [43]. The SNA measurements were performed using CASOS ORA version 3.0.9.9.81, a statistical analysis package by Carnegie Mellon University [44] used for the analysis of complex systems.

Table 1. Construction of the meta-network

Organization (Meta-network)	Log management infrastructure	
Node class	Node set	Interpretation
Agent (A)	Log collectors (C)	Log files collection/storage.
Knowledge (K)	Log files (F)	The generated log files.
Task (T)	Log management tasks (T)	The log management tasks (part of the requirements).
Network	2-mode network	Interpretation
Agent x task (AT)	|T| x |C|	The tasks to be performed on each log collector.
Knowledge x task (KT')	|F| x |T|	The log files required for the accomplishment of each log management task.
Agent x knowledge (AK)	|F| x |C|	The log files currently collected/stored on each log collector.

For the needs of this case study we assume that a log management infrastructure is already implemented for security and administration reasons. Following [5], the organization has deployed log collectors in three different

physical locations, one for each category of log records, as follows:

- Athens: logs generated by the network devices.
- Thessaloniki: logs generated by the security devices.
- Siros: logs generated by the systems' operating systems.

The organization has formed three teams: 1) the network administrators, 2) the security analysts and 3) the system administrators, and has granted them access to the log collectors in 'Athens', 'Thessaloniki' and 'Siros' respectively.

4.1 Adjustment of the dynamic log management infrastructure design

Evolution of risks

To demonstrate the workings of the proposed methodology, we assume that the organization becomes aware of [45], a vendor's report on the current attack landscape, concluding that the organization is expected to be affected by the following risks:

R-1: NTP and DNS amplification attacks.
R-2: Ransomware and cryptomining software.
R-3: Email spam.
R-4: Software vulnerabilities.
R-5: Use of legit online services for malicious acts.
R-6: Insiders.

The WAN is composed of 78 assets, hence |A|=78, and is affected by 6 risks, hence |R|=6. When an asset is affected by a risk a link is created among them, resulting in the two-mode affiliation network represented by the |A| x |R| incidence matrix. The real risks each asset faces are not publicly available, thus they are assumed for the needs of this case study, resulting in the affiliation network depicted in Figure 3, where a circle represents an asset, a triangle represents a risk and the nodes highlighted with a ring are log collectors.

Folding the |A| x |R| two-mode network results to the |A| x |A| one-mode network shown in Figure 4, where two nodes are linked when they face a common risk. The assets are sorted based on their total degree centrality to identify the high-valued ones, as in Table 2. Nodes 'Xanthi', 'Kalamata', 'Santorini' and 'Athens' are high-valued; a high-valued node is a high-risk asset, as it faces common risks with many other assets.

The methodology continues with the identification of the m-slices, that aims to identify groups of assets that face common risks, hence increased attack surface. The identified m-slices are depicted in Figure 5, where only the 4-slices and the 3-slices are depicted, to ensure readability. Assets 'Santorini' and 'Kalamata' form a 4-slice, as they face four common risks. Assets 'Thessaloniki' and 'Athens' are log collectors and face three common risks forming a 3-slice; they are critical for the infrastructure and pose an increased attack surface, thus they should be prioritized for risk response [4].

The affiliation analysis resulted in the identification of assets 'Kalamata', 'Santorini', 'Xanthi', 'Thessaloniki' and 'Athens' being the high-risk assets, for the current threat landscape, as perceived by the organization conducting the analysis. Assets 'Santorini' and 'Kalamata' are both part of a 4-slice and high-ranked in total degree centrality; the log collectors 'Thessaloniki' and 'Athens' form a 3-slice while 'Athens' is also high-ranked in total degree centrality.

Evolution of risks and network topology

For this second scenario we assume that additional to the changes on the risks the organization faces, the network topology has also changed. The organization, hypothetically, deployed ten more systems, five in Kos and five in Ios, as shown in Figure 6, where each new system is represented by a black circled node. Each of these systems is affected by risks which may be common to the ones affecting the already deployed ones. This causes the affiliation network to change as new assets (nodes) are introduced to the network model, and the common risks create new links among them, resulting in the affiliation network depicted in Figure 7. In both figures the nodes have been rearranged to assure the readability of the visualizations.

The affiliation network resulting from the new deployments is folded and the SNA measurements are repeated (its visualization is omitted due to its poor readability). The ranking of the nodes has changed, resulting in the nodes listed in Table 3, were the nodes are sorted in descending order based on the value of the total degree centrality. Comparing Tables 2 and 3, we observe that the value of the measurement has increased for all the listed nodes; this was expected, as the new deployed systems face common risks with the high-ranked ones.

A new asset is identified in the list of the high-ranking assets, "Ios-1", as it is affected by a risk that affects many other systems (R-2: "Ransomware and cryptomining software"), and asset "Irakleio" surpassed "Athens". On the other hand, the formation of 3-slices and 4-slices was not affected, remaining as depicted in Figure 5 (the low valued m-slices have changed, but only the high valued ones are considered by the proposed methodology).

The output of the affiliation analysis is then used for the establishment of the organization's risk response [4]. In the context of the proposed methodology, adjusting the design of the log management infrastructure is considered to be part of the risk mitigation process [4]. Based on the findings, the organization redesigned its log management infrastructure by deploying more log collectors, by enabling more log generators, and by defining new log management requirements. Thus, the log management infrastructure evolved to adapt to the needs of the new threat landscape.

Table 2. Sample total degree centrality

Rank	Asset	Total Degree Centrality
1	Xanthi	39
2	Kalamata	38
3	Santorini	38
4	Athens	34
5	Irakleio	34
6	Malesina	32
7	Rhodes	32
8	Alexandroupoli	31

Table 3. Sample total degree centrality with new deployments

Rank	Asset	Total Degree Centrality
1	Xanthi	48
2	Kalamata	47
3	Santorini	47
4	Irakleio	43
5	Athens	41
6	Ios-1	41
7	Malesina	41
8	Rhodes	41

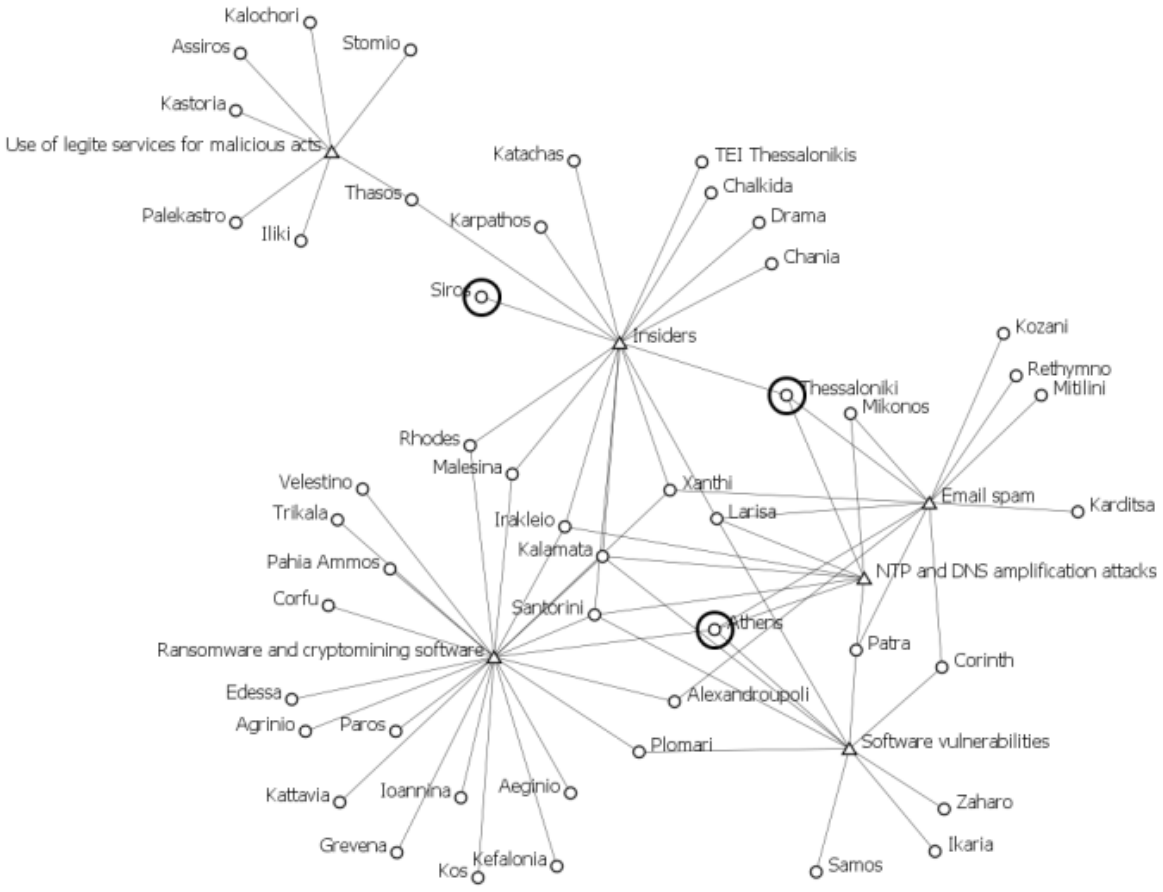

Figure 3. |A| x |R| affiliation network

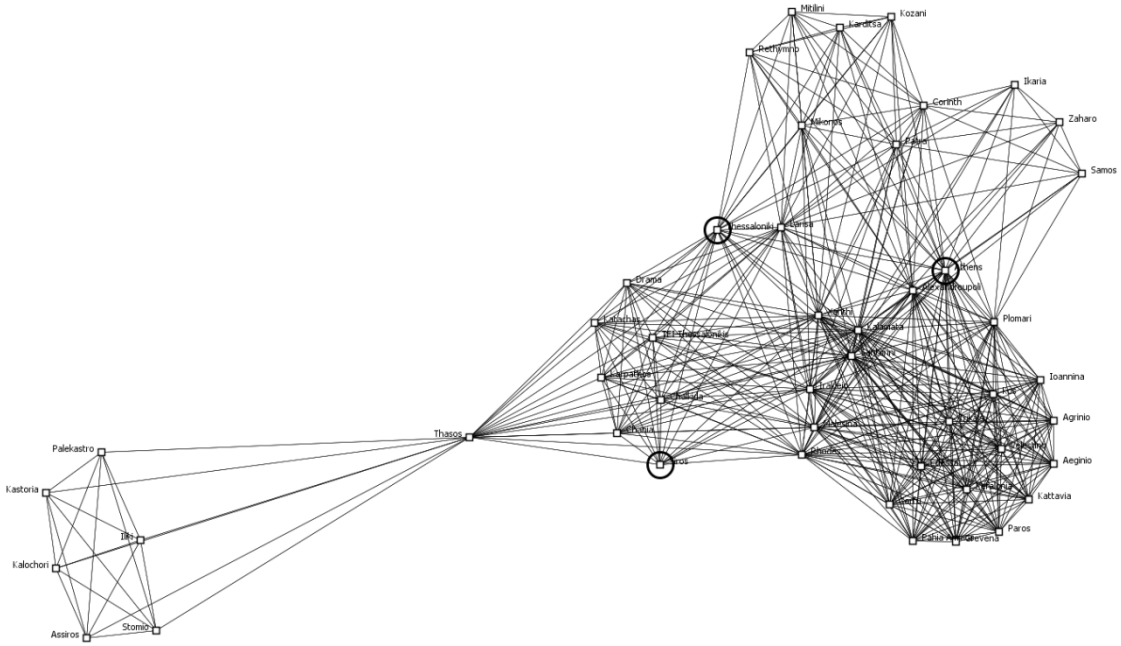

Figure 4. Folded |A | x |A| social network

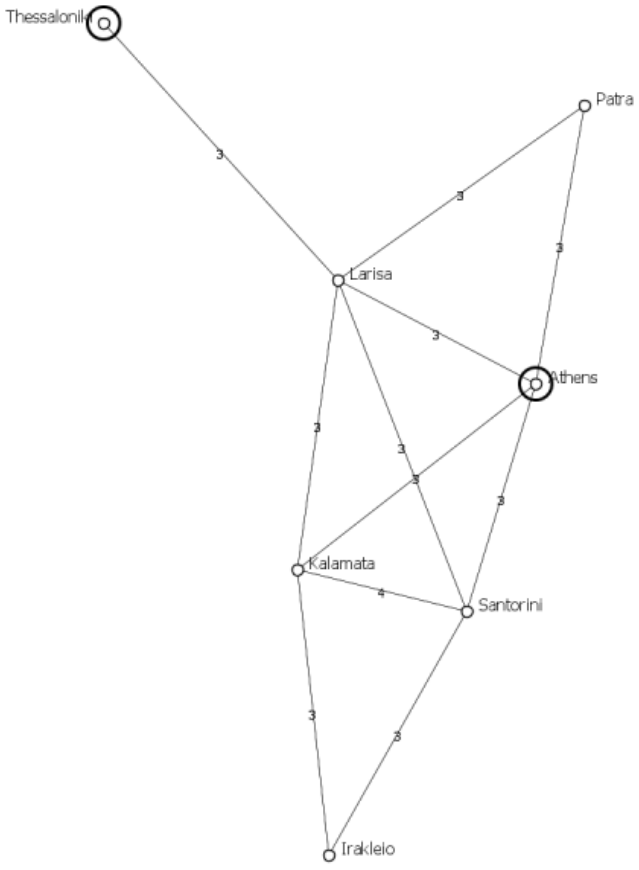

Figure 5. m-slices of the |A| x |A| social network

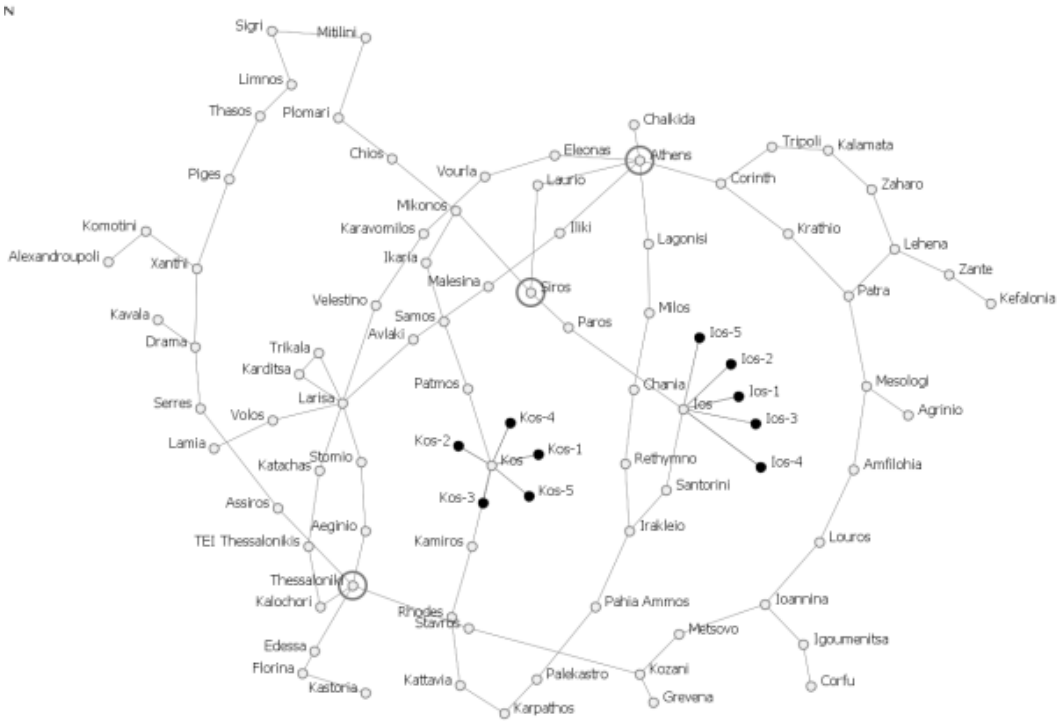

Figure 6. GRNET network topology with new deployments

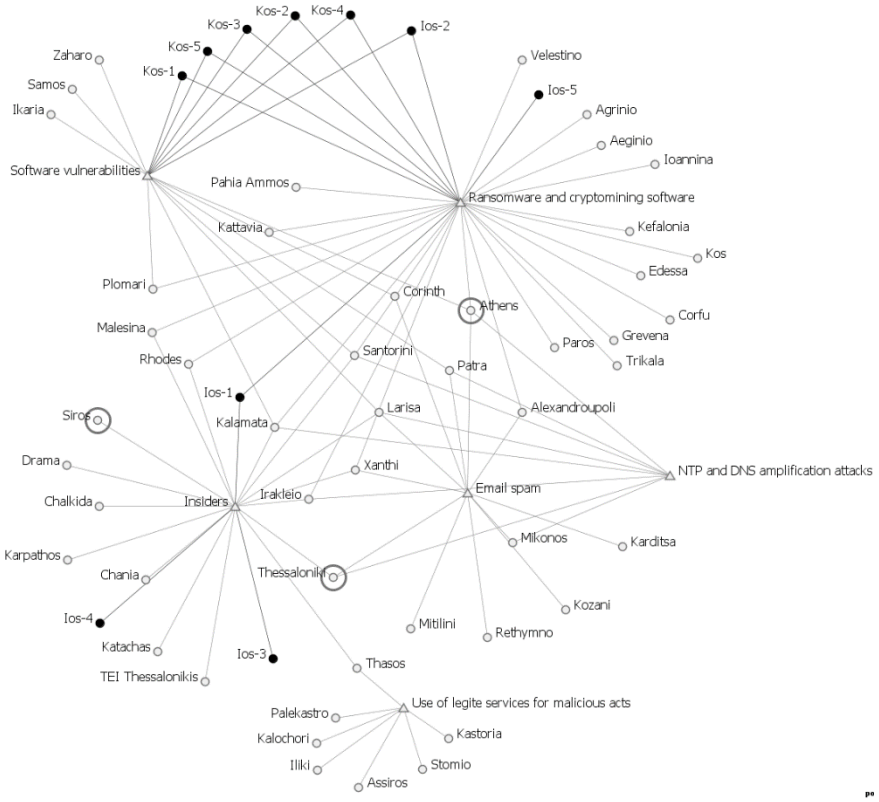

Figure 7. Affiliation network with new deployments

4.2 Validation of log management infrastructure design

The methodology continues with validating that the log management infrastructure design is aligned with the new set of log management requirements (part of the organization's risk response), and that it still enables the three teams (security analysts, network administrators, system administrators) to perform their assigned analysis tasks. Enabling log generators without properly directing them to log collectors may result in the collection of unnecessary log data, or in the absence of required log data. Adding a new log management requirement may need directing a log generator to additional log collectors, or the modification of its verbosity e.g. a Web server that does not log the user agent string, even though this is needed for the analysis tasks.

The design structure of the log management infrastructure is modeled as a meta-network composed of three node classes, as shown in Table 4, and as the following two-mode networks:

- |C| x |F|: the log files collected on each log collector.
- |C| x |T|: the log management tasks required to be performed on each log collector.
- |F| x |T|: the log files required to perform each log management task.

The combination of these two-mode networks results in the meta-network model of the organizations' structural properties, shown in Figure 8, which will be used for the SNA measurements. In the depicted meta-network each log collector (circle) is linked with the log files (rectangle) it receives and the tasks (polygon) that should be performed on it; each task is linked with the log files that are required to perform it. The software tool that was used is CASOS ORA version 3.0.9.9.81 [44], a tool for the analysis of the structural properties of organizations and the detection of risks/vulnerabilities in their design structure.

In order to validate that each log collector actually collects the log files that are required to perform the log management tasks expected to be performed by the analysts on each collector, the values of *agent knowledge needs congruence* and *agent knowledge waste congruence* were calculated, as listed in Table 5. The value of the *agent knowledge needs congruence* of collector 'Siros", for example, is 0.909, that is the portion of the log files the collector lacks in order to perform its assigned tasks. Figure 9 depicts the tasks (polygon) assigned to 'Siros' (circle), the log files (rectangle) that are needed, whilst only one of them ("antivirus alerts log files") is currently collected on 'Siros'. The value of the *agent knowledge waste congruence* of collector 'Siros' is 0.500, which is the portion of log files that are collected on that collector, though not needed for its assigned log management tasks ('vulnerability scanning results').

The same process can be repeated for the remaining log collectors in order to assure that only the log files that are necessary for the analysis tasks are collected. Corrective actions on the design of the log management infrastructure may include modification of the log generators' verbosity, of the assignment of log generators to log collectors, as well as the reassignment of log management tasks to log collectors. The entities considered in the MNA model of the log management infrastructure design structure are the collectors, the log files and the log management tasks; adjusting each of these can result to the desired values of the MNA measurements (agent knowledge needs congruence and agent knowledge waste congruence).

Table 4. GRNET meta-network

Node class	Node set	Node name
Agent (A)	|C| = 5	Athens (c-1)
		Thessaloniki (c-2)
		Siros (c-3)
		Ioannina (c-4)
		Patra (c-5)
Knowledge (K)	|F| = 10	Windows log files (f-1)
		Linux log files (f-2)
		Firewall log files (f-3)
		VPN log files (f-4)
		Antivirus alerts log files (f-5)
		Vulnerability scanning results (f-6)
		DNS log files (f-7)
		NTP log files (f-8)
		Router log data (f-9)
		Cloud services log data (f-10)
Task (T)	|T| = 8	Track authentication and authorization (t-1)
		Detect systems' configuration integrity changes (t-2)
		Track outbound network connections (t-3)
		Detect amplification attacks (DNS, NTP) (t-4)
		Detect malware activity (t-5)
		Monitor usage of cloud services (t- 6)
		Monitor systems performance (t-7)
		Monitor network utilization (t-8)

Table 5. MNA measurements

Collector	Agent Knowledge Needs Congruence	Agent Knowledge Waste Congruence
Athens	0	1
Patra	0	1
Ioannina	0	1
Thessaloniki	0.591	0
Siros	0.909	0.500

5. Conclusions and future work

In this work, concepts models, tools and techniques of SNA and MNA were leveraged to design a dynamic log management infrastructure optimized for the risks an organization is expected to face in the threat landscape it operates in.

A novel methodology was proposed, consisting of two main steps; 1) the adjustment of the log management infrastructure's design, by leveraging SNA for the identification of the high-risk assets, 2) the validation of the alignment of the resulting design with the defined log management requirements, by leveraging MNA concepts and measures. The workings of the proposed methodology have been demonstrated on the WAN of the GRNET network using real data when available, and assumed data where real data are not publicly available. The use of SNA software facilitates the computation of the necessary values continuously, thus enabling an organization to decide on corrective actions and to assess the effectiveness of the resulting design structure. Future work will focus on the creation of a software tool that will support the use of the proposed methodology, and of its evaluation in a real working environment. Additional SNA measurements and analysis techniques, involving more entities in the design structure of large scale infrastructures, will also be considered.

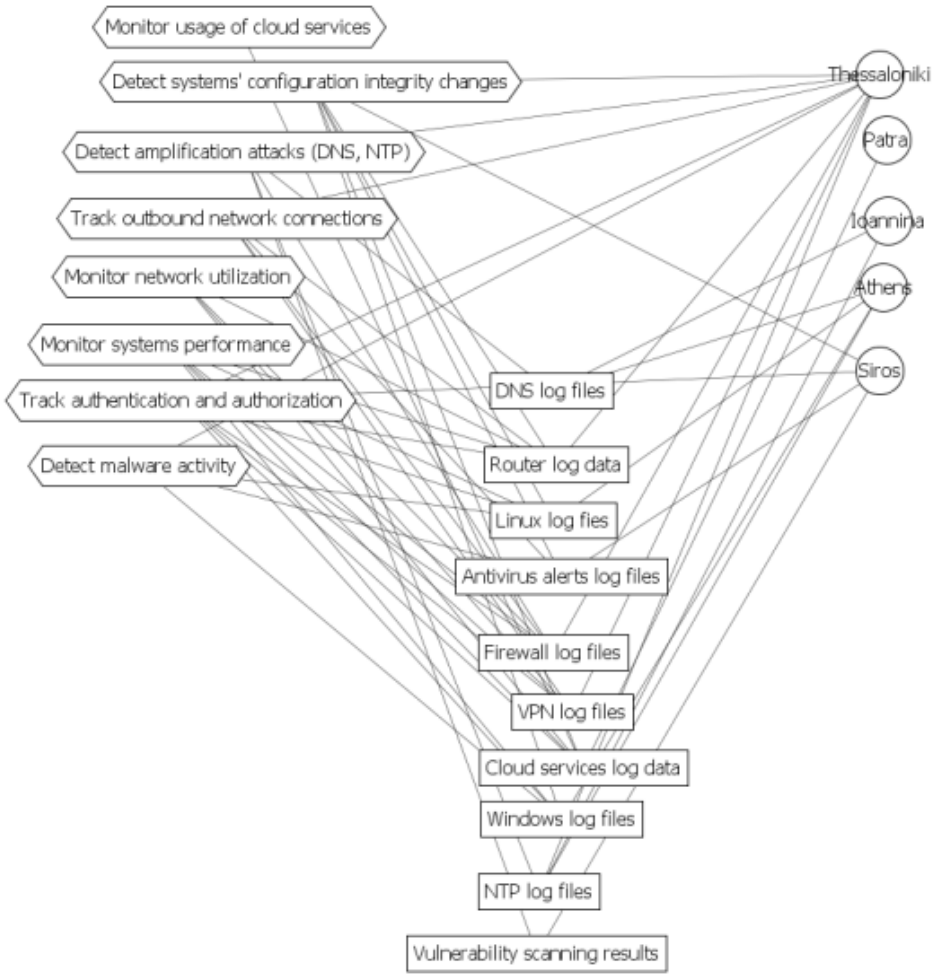

Figure 8. GRNET meta-network visualization

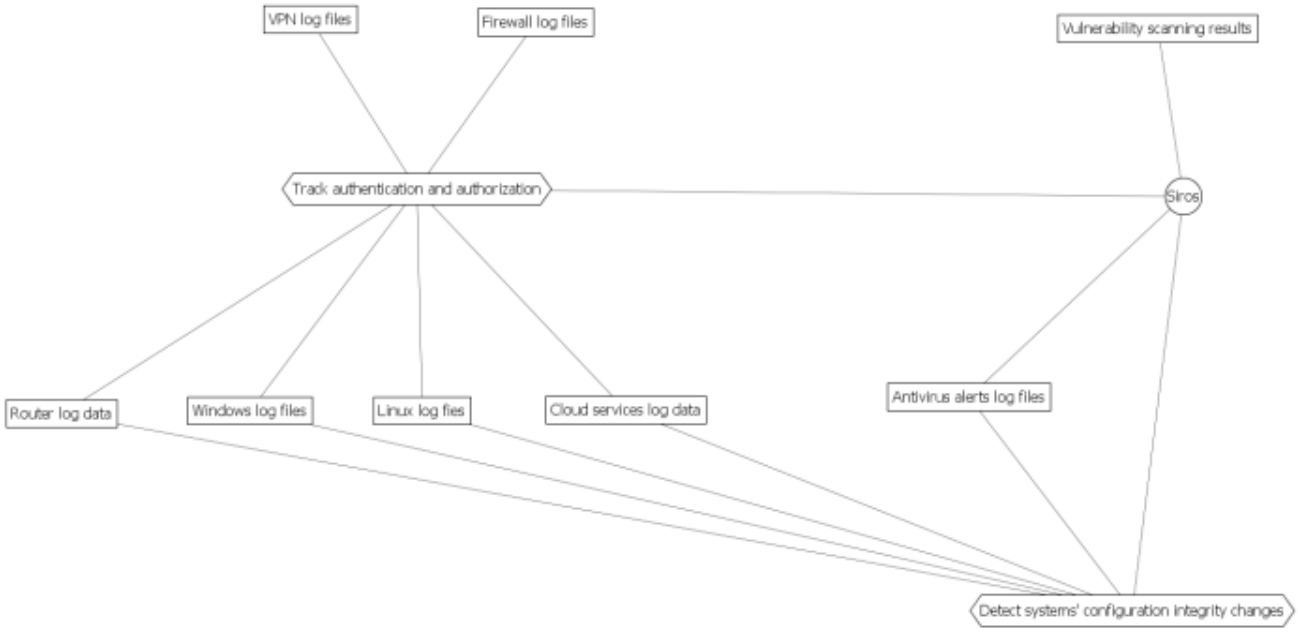

Figure 9. 'Siros' log collector meta-network

References

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

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

[3] "ENISA Threat Landscape Report 2018 15 Top Cyberthreats and Trends," ENISA, Jan. 2019.

[4] Joint Task Force Transformation Initiative Interagency Working Group, "NIST SP 800-30, Rev.1 Guide for Conducting Risk AssessmeNnts." Sep-2012.

[5] K. Kent and M. Souppaya, "NIST SP 800-92, Guide to Computer Security Log Management - SP800-92.pdf." 2006.

[6] D. Clayton, "Building Scalable Syslog Management Solutions." CISCO, 2011.

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

[8] 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.

[9] K. Faust and S. Wasserman, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[10] 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.

[11] 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.

[12] Chris Sanders and Jason Smith, *Applied Network Security Monitoring*, 1st ed. Syngress, 2014.

[13] "ISO/IEC 27002," 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.

[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.

[16] P. He, "An End-To-End Log Management Framework for Distributed Systems," in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, Hong Kong, China, 2017.

[17] P. He, J. Zhu, S. He, J. Li, and M. R. Lyu, "Towards Automated Log Parsing for Large-Scale Log Data Analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 931–944, Oct. 2017.

[18] T. Li *et al.*, "FLAP: An End-to-End Event Log Analysis Platform for System Management," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, NS, Canada, 2017, pp. 1547–1556.

[19] R. Rastogi, A. S, Shobha G, Poonam G, Pratiba D, and Ankit Singh, "Design and development of generic web based framework for log analysis," in *2016 IEEE Region 10 Conference (TENCON)*, Singapore, 2016.

[20] 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.

[21] P. Anil and R. Sagar, "Development of Highly Secured Cloud Rendered Log Management System," *International Journal of Computer Applications*, vol. 108, no. 16, 2014.

[22] M. Kumar, A. Kumar Singh, and T. V. Suresh Kumar, "Secure Log Storage Using Blockchain and Cloud Infrastructure," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Bangalore, India, 2018.

[23] Wouter De Nooy, Andrej Mrvar, and Vladimir Batagelj, *Exploratory Network Analysis with Pajek*, 2nd ed. Cambridge University Press, 2011.

[24] Borgatti SP, "The key player problem," presented at the Dynamic social network modeling and analysis: workshop summary and papers, 2003.

[25] S. Borgatti, "Identifying sets of key players in a social network," *Computational & Mathematical Organization Theory*, vol. 12, no. 1, pp. 21–34, 2006.

[26] 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.

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

[28] 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.

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

[30] 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.

[31] 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.

[32] Cathy J. Reback, Kirsty Clark, Jesse B. Fletcher, and Ian W. Holloway, "A Multilevel Analysis of Social Network Characteristics and Technology Use on HIV Risk and Protective Behaviors Among Transgender Women," *AIDS and Behavior*, Jan. 2019.

[33] D. Krackhardt and K. Carley, *PCANS model of structure in organizations*. 1998.

[34] K. Carley, "Computational organizational science and organizational engineering," *Simulation Modelling Practice and Theory*, vol. 10, no. 5, pp. 253–269, 2002.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] "GRNET Topology," 21-Jan-2019. [Online]. Available: https://grnet.gr/infrastructure/network-and-topology/. [Accessed: 21-Jan-2019].

[44] "Homepage | CASOS." [Online]. Available: http://www.casos.cs.cmu.edu/index.php. [Accessed: 22-Jan-2019].

[45] "Annual Cybersecurity Report," *Cisco Systems, Inc.*, 2018. [Online]. Available: https://www.cisco.com/c/dam/m/digital/elq-cmcglobal/witb/acr2018/acr2018final.pdf?dtid=odicdc000016&ccid=cc000160&oid=anrsc005679&ecid=8196&elqTrackId=686210143d34494fa27ff73da9690a5b&elqaid=9452&elqat=2. [Accessed: 11-Jan-2019].