Machine-assisted Cyber Threat Analysis using Conceptual Knowledge Discovery

Martín Barrère, Gustavo Betarte, Victor Codocedo, Marcelo Rodríguez, Hernán Astudillo, Marcelo Aliquintuy, Javier Baliosian, Rémi Badonnel, Olivier Festor, Carlos Raniery Paula dos Santos, et al.

Machine-assisted Cyber Threat Analysis using Conceptual Knowledge Discovery

– Position Paper –

Martín Barrère*[1,5], Gustavo Betarte[1], Victor Codocedo[2], Marcelo Rodríguez[1], Hernán Astudillo[3], Marcelo Aliquintuy[3], Javier Baliosian[1], Rémi Badonnel[2], Olivier Festor[2], Carlos Raniery Paula dos Santos[4], Jéferson Campos Nobre[4], Lisandro Zambenedetti Granville[4], and Amedeo Napoli[2]

[1] InCo, Facultad de Ingeniería, Universidad de la República, Uruguay
[2] LORIA/INRIA/CNRS - Nancy, France
[3] Universidad Técnica Federico Santa María, Valparaíso, Chile
[4] Institute of Informatics, Federal University of Rio Grande do Sul, Brazil
[5] Imperial College London, UK

Abstract. Over the last years, computer networks have evolved into highly dynamic and interconnected environments, involving multiple heterogeneous devices and providing a myriad of services on top of them. This complex landscape has made it extremely difficult for security administrators to keep accurate and be effective in protecting their systems against cyber threats. In this paper, we describe our vision and scientific posture on how artificial intelligence techniques and a smart use of security knowledge may assist system administrators in better defending their networks. To that end, we put forward a research roadmap involving three complimentary axes, namely, (I) the use of FCA-based mechanisms for managing configuration vulnerabilities, (II) the exploitation of knowledge representation techniques for automated security reasoning, and (III) the design of a cyber threat intelligence mechanism as a CKDD process. Then, we describe a machine-assisted process for cyber threat analysis which provides a holistic perspective of how these three research axes are integrated together.

1 Introduction

The goal of this paper is to introduce some novel applications of formal concept analysis [13], knowledge discovery in databases and, in a broader sense, artificial intelligence techniques to support security analysis of computer networks and systems. Computer networks are very dynamic environments composed by diverse entities which, on a daily basis, hold thousands of virtual activities. Additionally, they often require configuration changes to satisfy existing or new operational requirements (e.g. new services, upgrading existing versions, replacing faulty hardware). Such dynamicity highly increases the complexity of security management. Even if automated tools help to simplify security tasks there is a

* mbarrere@fing.edu.uy, m.barrere@imperial.ac.uk

need for advanced and flexible solutions able to assist security analysts in better understanding what is happening inside their networks.

The research work we put forward is being developed in the context of the AKD (Autonomic Knowledge Discovery) project [7], a research collaboration effort involving five teams with different expertises. We have identified several key aspects in which the use of artificial intelligence techniques, and particularly formal concept analysis (FCA), can quickly improve on the current state of affairs for processes and tasks in the field of computer and network security. We describe how we envision an adaptation of the conceptual knowledge discovery on databases (CKDD) machinery to provide support in developing scientifically grounded techniques for the domain of cyber threat intelligence. In particular, we are concerned with vulnerability management and cyber threat analysis. We also motivate the benefits of using ontology engineering methods and tools to improve the state of the art of security-oriented automated reasoning.

The remainder of this paper is organized as follows: Section 2 points out the scientific challenges of the research that is being developed in the context of the AKD project. Section 3 motivates three different research fields in which artificial intelligence techniques can be used to provide machine-assisted support to the domain of cyber security. Section 4 describes a cyber threat analysis process aimed at detecting and recognizing security threats within computer systems and points out how and where the techniques previously discussed apply. Finally, Section 5 concludes and summarizes research perspectives.

2 Scientific challenges

Vulnerabilities, understood as program flaws or configurations errors, are used by attackers to bypass the security policies of computer systems. Therefore, vulnerability management mechanisms constitute an essential component of any system intended to be protected. During the last decades, strong research efforts as well as dozens of security tools have been proposed for dealing with security vulnerabilities [5]. However, current security solutions still seem to work under certain boundaries that prevent them to act intelligently and flexibly, i.e. strictly sticked to the available security information in order to analyze, report and eventually remediate found problems.

In addition to this inflexibility, remediating vulnerabilities is already a complex problem and despite the great advances made in this area, remediation tasks are reactive by nature and they can be hard to perform due to costly activities and performance degradation issues. They may also generate consistency conflicts with other system policies. Therefore, our scientific posture in this context is that instead of detecting vulnerable states and then applying several corrective actions, it would be better to anticipate and avoid these vulnerable states in the first place. This objective constitutes a challenging problem. Firstly, mechanisms for understanding the behavior and dynamics of the system are needed. Secondly, sometimes vulnerabilities are not known, so techniques for analyzing

the available knowledge and extracting measures that might allow the system to make decisions are essential.

The aforementioned security challenge gets more complex when considered in dynamic networked scenarios. The accelerated growth of highly heterogeneous and interconnected computer networks has severely increased the complexity of network management. This phenomenon has naturally affected network security where traditional solutions seem unable to cope with this evolving and changing landscape. The main problem is that even when current security techniques may enable high levels of automation, they might fail to achieve their purpose when certain aspects of a managed environment slightly change. We need to provide systems with mechanisms to understand, reason about, and anticipate the surrounding environment. In light of this, we firmly believe that an advanced, flexible, and clever management of security knowledge constitutes one of the key factors to take security solutions to the next level. Our vision is that, independently of the nature of an automated solution (automatically assisting an administrator or automatically making security decisions), the ability to intelligently manage knowledge is essential.

In the broad sense of knowledge management, several scientific areas within the artificial intelligence domain can contribute to achieve our vision. In this work, we identify domains such as formal concept analysis (FCA), ontological engineering, information retrieval (IR), case-based reasoning (CBR), and conceptual knowledge discovery on databases (CKDD), as sound scientific areas that may support a new level of smart cyber security solutions. Fig. 1 illustrates our research strategy for the short, medium and long term.

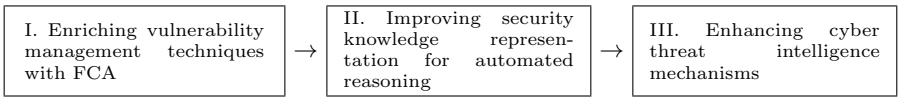

Fig. 1: Research strategy for the short, medium and long term

In the short term (I), our objective is to understand to what extent FCA can enrich and advance the state of the art of vulnerability management techniques. Vulnerability management can be usually seen as the cyclical process of assessing and remediating vulnerabilities. Anticipation techniques are not considered in the classical definition, although the concept of foreseeing future vulnerabilities perfectly fits the vision of flexible and adaptive systems. Therefore, the idea is to begin solving basic problems within the sub-area of vulnerability assessment and progress towards FCA-based mechanisms for anticipating and remediating security vulnerabilities. We understand that a clever use of available knowledge requires a formal and robust underlying machinery that allows systems to process, reason, extract, and extrapolate information and knowledge among other features. In the medium term (II), we aim at investigating the link between current security standard efforts such as the STIX language [3] and knowledge representation methods such as security ontologies. The results of this research

activity may provide a robust support to intelligently deal with security issues. In the long term (III), the objective is to integrate the results and experience obtained in (I) and (II) to develop novel approaches to deal with cyber security threats supported by KDD-based techniques. In the following section, we explain in detail each one of these stages, their impact and importance, and how we envision their development.

3 Research roadmap

3.1 Enriching vulnerability management techniques with FCA

One of the main objectives of our research is the study of vulnerability anticipation mechanisms from the perspective of FCA. Usually, a vulnerability is considered as a combination of conditions that if observed on a target system, the security problem described by such vulnerability is present on that system [5]. Each condition in turn is understood as the state that should be observed on a specific object. When the object under analysis exhibits the specified state, the condition is said to be true on that system. In this context, a vulnerability is a logical combination of conditions and therefore, identifying known vulnerabilities implies the evaluation of logical predicates over computer system states. In brief, we characterize vulnerabilities and system states by the properties they present. From a technical perspective, the OVAL language [2] maintained by MITRE [1], is a standard XML-based security language which permits the treatment and exchange of this type of vulnerability descriptions in a machine-readable manner.

V_1: $c_1 \wedge c_2$
V_2: $c_1 \wedge (c_2 \vee c_3)$
V_3: $\neg c_2 \vee c_3 \vee c_4$
V_4: $\neg c_3$

Table 1: Vulnerabilities as logical formulæ

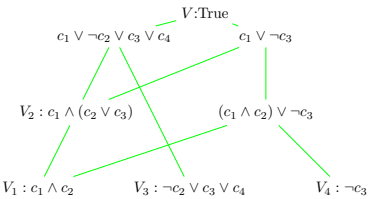

Table 2: Semi-lattice representation of the vulnerability set

As an example, let us consider Table 1 depicting four vulnerabilities $V = \{V_1, V_2, V_3, V_4\}$ as logical formulæ, where $\wedge, \vee, \neg$ represent the logical connectors AND, OR, NOT respectively, and $C = \{c_1, c_2, c_3, c_4\}$ are four system conditions (e.g. "port 80 is open", "httpd server is up", "firewall is off", etc.). A system state s is defined as a set of conditions $c_i \in C$ such that c_i is true on s. Therefore, the process of vulnerability assessment over a system state s can be defined as follows:

$$f(s) = \begin{cases} vulnerable & \exists V_i \in V, s.t. V_i(s) = true \\ safe & otherwise \end{cases}$$

A system state s is considered *vulnerable* if there exists at least one vulnerability that evaluates to true when taking the values from the system for the involved conditions, and *safe* otherwise. For example, considering $s = \{c_1, c_3\}$, it can be observed that $f(s) = vulnerable$ since $V_2(s) = V_3(s) = true$.

From the perspective of FCA [13] and particularly, using the formalization of Logical Concept Analysis (LCA) [12], this can be formalized as follows. Let V be a set of vulnerability labels associated to formulæ in the logic $\mathcal{L}$ with $\wedge, \vee, \neg$ denoting the logical operators and atoms $\mathcal{A}$ containing a set of system conditions $c_i \in C$. A vulnerability label $v \in V$ is associated to a formula in $\mathcal{L}$ through the mapping function $\delta(v) \in \mathcal{L}$.

Let us define the logical context $\mathbb{K} = (V, (\mathcal{L}, \vDash), \delta)$ with the following derivation operators for a subset of vulnerabilities $A \subseteq V$ and a formula $d \in \mathcal{L}$:

$$A^{\square} = \bigvee_{v \in A} \delta(v) \qquad d^{\square} = \{v \in V \mid \delta(v) \vDash d\}$$

For any two vulnerabilities labels $v_1, v_2 \in V$, we have that $v_1 \vDash v_2 \iff v_1 \vee v_2 = v_2$ denotes that v_1 *is a model of* v_2. A pair (A, d) is a formal concept if and only if $A^{\square} = d$ and $d^{\square} = A$. It can be shown that the derivation operators generate a Galois connection between the power set $\wp(V)$ of vulnerability labels and the set of formulæ $\mathcal{L}$ and thus, a concept lattice can be obtained from the logical context $\mathbb{K}$. Within our approach, such a concept lattice generates the search space for vulnerability assessment and correction.

Analogously to the Boolean model of *Information Retrieval* [15], we can use the concept lattice to *classify* the system state s and search for exact or partial answers, i.e. vulnerabilities which affect or *may* affect the system. For instance, the semi-lattice illustrated in Table 2 can be used to understand that if a system is affected by vulnerabilities $V2$ and $V3$, then *it may* be also affected by vulnerability $V1$. In particular, the formula labeled by v satisfies a formula d in some context $\mathcal{K}$ if and only if the concept labeled with v is below the concept labeled with d in the concept lattice of $\mathcal{K}$ [11]. Additionally, using the classification algorithm inspired in case-based reasoning presented in [9], it is easy to show that the assessment process becomes a search in the hierarchy generated by the semi-lattice, i.e. the assessment has a sub-linear complexity.

Vulnerability remediation on the other hand consists in changing the right properties of a system ($c_i \in C$) to bring it into a safe state. This is an explosive combinatorial problem [4]. However, we believe that a concept lattice can be useful to guide the search for corrective actions that do not lead to new vulnerable states. Furthermore, there might be no solution in some cases, so an interesting approach would be to approximate safe solutions by weighting the impact of vulnerabilities using scoring languages such as CVSS (Common Vulnerability Scoring System) [10]. Lastly, our final goal is to understand to what extent FCA can contribute to the process of anticipating vulnerabilities, which basically consists in predicting potential vulnerable states due to changes in the system. Considering known vulnerabilities, a concept lattice can be used as an

approximation map to avoid unsafe configuration changes. Extrapolation and pattern detection mechanisms are also worth to be explored though ontological engineering and data mining techniques might better suit such objectives as discussed in the following section.

3.2 Improving security knowledge representation for automated reasoning

Several vocabularies have been proposed in the context of cyber security. Some of the most important ones are: Structured Threat Information eXpression (STIX), Common Attack Pattern Enumeration and Clasification (CAPEC), Common Vulnerability and Exposures (CVE), Cyber Observables eXpression (CybOX), Malware Attribute Enumeration and Characterization (MAEC) and Common Weakness Enumeration (CWE) [24]. Most of these vocabularies were defined by particular organizations, like MITRE and NIST, to facilitate the exchange of information regarding vulnerabilities, security issues and attack descriptions.

The benefits of introducing vocabularies are plenty and well-known. They establish a common language that can be used by different organizations to describe the same concepts and provide a framework for documentation allowing the structured and systematized creation of a body of knowledge. Vocabularies have proven not only be relevant for humans, but for autonomous agents in several applications as well. At the syntactic level, they enable different systems to communicate in a common pre-defined structured manner. At the semantic level, vocabularies have played a major role in the last decade allowing autonomous agents to *reason* about the information within a dataset. For example, let us consider a security analyst looking through different databases for a *malware* that could affect a given system. A malware is a very generic term used to identify a piece of software specially designed to violate the security integrity of a computer system. Thus, the search task can be very difficult given that there are several types of malware, namely trojan horses, spywares, backdoors, worms, among others. Instead, a vocabulary could easily integrate these descriptions by stating that trojan horses, spywares, backdoors and worms are *types of* malware. An autonomous agent can profit from the vocabulary by automatically inferring that an object catalogued as a "trojan horse" is relevant for the search of "malware".

In the semantic web, vocabularies are usually supported by ontologies, a meta-model to provide a structured description of the concepts in a given domain [21]. Ontologies can provide different levels of description, namely at the entity level, at the relational level and at the instance level. The entity level describes the concepts that compose a given domain (Malware, Trojan Horse, Spyware) and their attributes (Malware *has_name*, Trojan Horse *has_target_os*, etc.). The relational level describes relations among concepts (Trojan *is_a_type_of* Malware, Trojan Horse *has_target_operating_system* Windows, etc.) and their attributes (*is_a_type_of* is a non-symmetric, transitive relation). Finally, the instance level describes the relations between instances, their types (trojan1 *is_a*

Trojan), and their attributes (trojan1 *has_name* "Zeus"). Furthermore, ontologies support a similar level of inference as first-order logic through its logical formalism called description logics.

Several research communities have undertaken the task of formalizing their domain knowledge with vocabularies. and many of them have moved forward towards describing their vocabularies through ontology definitions. For example, in [8] an ontology learning approach is proposed for the astronomical domain. In [14] the authors propose an ontology to document software architecture decisions providing an automated annotation process over software design documents. In [22], the authors propose a knowledge discovery process to build and populate an ontology for the cultural heritage domain using a relational database schema. Extensive reviews on ontology learning and construction using formal concept analysis can be found in [18, 20, 23].

As mentioned before, the domain of cyber security has already acknowledged the benefits of defining common vocabularies. Furthermore, initial steps have been taken towards building a comprehensive ontology definition which integrates the different vocabularies within the domain. In [24], the authors describe the process through which they manually crafted a domain ontology with the goal of supporting security analysts in the task of detecting cyber threats. This work is indeed a big step forward, however we are confident that the use of state of the art ontology learning techniques, particularly formal concept analysis, can greatly improve the quality of an ontology for cyber security. For instance, techniques like ontology alignment [23] can overcome overlapping issues in current vocabularies for cyber security, a fact that is oversought in [24]. The great potential for automatically building description logic knowledge bases using FCA [8, 20] would allow to further extend the support provided to security analysts in a more dynamic environment, a major drawback in manual approaches for ontology building. Finally, the definition of a domain ontology for cyber security is a necessary condition to support more advanced data mining techniques. In our project, this represents a milestone that would enable us to provide security analysts with advanced features for threat detection such as integrated search from multiple repositories [16], partial matching based on case-based reasoning [9], or document annotation [14].

3.3 Enhancing cyber threat intelligence mechanisms

The traditional approaches for cyber security, which have mainly focused on understanding and addressing vulnerabilities in computer systems, are still necessary but not longer sufficient enough. Effective defense against current and future threats requires a deep understanding of the behavior, capability and intent of the adversary. Threat environments have evolved from widespread disruptive activity to more targeted, lower-profile multi-stage attacks aiming at achieving specific tactical objectives and establishing a persistent foothold into the threatened organization. This is what is called an Advanced Persistent Threat (APT). The nature of APTs requires for more proactive defense strategies in contrast to the traditional reactive cyber security approach. To be proactive, defenders need

to move beyond traditional incident response methodologies and techniques. It is necessary to stop the adversary before he can exploit the security weaknesses of the system. In the cyber domain, cyber intelligence is the understanding of the adversary capabilities, actions and intent. According to [19]: *Cyber intelligence seeks to understand and characterize things like: what sort of attack actions have occurred and are likely to occur; how can these actions be detected and recognized; how can they be mitigated; who are the relevant threat actors; what are they trying to achieve; what are their capabilities, in the form of tactics, techniques and procedures (TTP) they have leveraged over time and are likely to leverage in the future; what sort of vulnerabilities, misconfigurations or weaknesses are likely to target; what actions have they taken in the past; etc.*

One important objective of our research is to develop techniques and tools for providing assistance to accomplish different cyber threat intelligence procedures. In particular, we are focused on processes aiming at leveraging capacities for threat environment identification (type of attack, from where, how) and early detection of vulnerability exploitation attempts. We also aim at the generation and enrichment of (semantically structured) knowledge repositories, preferably in a way that is decoupled from the specifics of a particular technology for conducting threat analysis and correlation.

For a threat analysis tool to be useful in practice, two features are crucial: i) the model used in the analysis must be able to automatically integrate formal vulnerability specifications from the bug-reporting community and formal attack scenarios from the cyber security concerned community; ii) it is desirable for the analysis to be able to scale to complex networks involving numerous machines and devices. As a more ambitious goal, we aim at developing a prototype of an engine, in the spirit of MulVAL [17], able to consume low-level alerts (e.g. taken from OVAL scanning activities) and produce high-level attack predictions based on the scenario under analysis.

4 A machine-assisted approach for cyber threat analysis

In this section we put forward a cyber threat analysis process aimed at detecting and/or recognizing (potential) security attacks. We explain the most relevant procedures involved in the analysis and point out how and where automated support can be provided using the techniques discussed in sections 3.1, 3.2 and 3.3. The cyber threat analysis process, depicted in Fig. 2, embodies procedures that give support to the key phases of the search of compromise: derivation of threat indicators, collection of evidence, evaluation of the results and decision. In what follows we explain the process in further detail.

1. The process begins at step 1 with a security analyst providing information about some identified threat or anomaly, and characteristics of the target system. This information constitutes the *initial seed* for the cyber threat analysis, and might specify for instance, a compromise involving a suspicious file found on a Linux system. The involved information shall be represented using the STIX language, in particular using the notion of *indicator of com-*

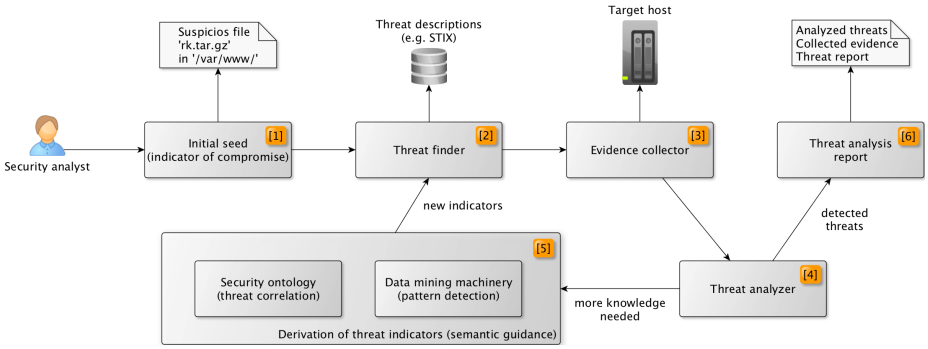

Fig. 2: Cyber threat analysis overview

promise. One such indicator allows to specify the different types of objects that can be found on a computing system/network such as ports, processes, threads, files, etc. Additionally, an indicator may capture metadata for the involved objects as well as logical relations between them thus providing further information to security analysts.

2. Once the seed has been provided, a *search of compromise* is performed at step 2. To that end, the threat finder component queries a database containing machine-readable descriptions of known threats specified in a formal language such as STIX. Only those cyber threats which are found to be related with the provided information are considered for subsequent analysis.
3. The retrieved threat descriptions are then used at step 3 by the evidence collector component to gather all the relevant information from the target system in order to decide whether the latter is compromised by at least one of the identified related cyber threats. The process of information gathering involves, for instance, collecting the list of open ports or running processes in the system. Standard languages such as OVAL provide great support for evidence specification and automated collection procedures [6].
4. The collected evidence is then evaluated by the threat analyzer component at step 4 in order to determine the level of compromise of the system. A target system may be considered compromised by a specific cyber threat if it presents a combination of objects (*threat indicator*) which are commonly found on infected systems. The threat analyzer decides whether the collected evidence is sufficient enough to indicate that the target system has been compromised or conversely whether more knowledge is needed to diagnose its status. In the first case, the process moves to step 6 where the information about the detected cyber threats is provided to the security analyst. Otherwise, the process continues at step 5 where a semantic machinery is used to derive new indicators that may lead to cyber threats not previously evaluated.

5. In the case that none of the spotted cyber threats are found on the system, a derivation process is triggered at step 5 in order to select new cyber threats that were not analyzed before. This new selection is performed by deriving threats related to the relevant evidence found on the system while gathering information in the previous stage. Derivation mechanisms may vary according to the available information and context, and they constitute a key objective within this research work. The FCA-based technique described in Section 3.1 may provide a map for finding vulnerable configurations close to the current system state. Additionally, two sub-components may semantically guide the search for new related threats. As discussed in Section 3.2, a security ontology may relax strict descriptions making context awareness procedures more flexible, i.e. security information that is not explicitly encoded a priori can be derived by considering semantic associations. Data mining techniques on the other hand may provide the ability to extrapolate information and extract security patterns thus increasing detection capabilities even more. The process of derivation (step 5), threat identification (step 2), collection (step 3) and analysis (step 4) shall be repeated until a conclusion or a stop condition is reached.
6. The outcome of a finished search process may be either that the system appears to be compromised or not enough evidence has been found to determine its compromise status. In any case, the process informs about the tested cyber threats as well as the evidence found on the system at step 6 in order to assist the security analyst to proceed with the analysis.

Open discussion. The selection of information and techniques for inferring and discovering new knowledge might be assisted by a human being, the security analyst in this case, thus following a methodology closer to CKDD. However, interesting research questions arise from this scenario. One of them is to what extent can we automate the whole process and let a security solution to make decisions for us? Going one step further we pose the question of autonomic solutions where self-adaptive and self-governed approaches come into scene. Our vision is that to achieve any of these objectives, a clever knowledge management is essential. In that context, we believe that FCA and CKDD may highly contribute to accomplish such goal.

5 Conclusions and perspectives

In this paper we have motivated and explained how different artificial intelligence techniques, in particular FCA and CKDD, can be used to enhance the state of the art of machine-assisted cyber security analysis. In addition to the objectives depicted in our research roadmap, we also target the construction of an experimental testbed for emulating hostile and unsafe environments. This can provide the ability to deploy implementation prototypes and anticipation solutions in order to evaluate the feasibility, scalability and accuracy of our approach. We have already experimented with a preliminary version of a tool that provides

mechanical support for conducting the cyber threat analysis process described in section 4. We are convinced that the extension of the tool with mechanisms that make use of conceptual knowledge discovery techniques will greatly improve the accuracy and efficiency of the process.

References

1. MITRE Corporation. `http://www.mitre.org/`. Last visited on May 17, 2015.
2. OVAL Language. `http://oval.mitre.org/`. Last visited on May 17, 2015.
3. Structured Threat Information expression. `http://stix.mitre.org/`. Last visited on May 17, 2015.
4. M. Barrère, R. Badonnel, and O. Festor. A SAT-based Autonomous Strategy for Security Vulnerability Management. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'14)*, May 2014.
5. M. Barrère, R. Badonnel, and O. Festor. Vulnerability Assessment in Autonomic Networks and Services: A Survey. *IEEE Communications Surveys & Tutorials*, 16(2):988–1004, 2014.
6. M. Barrère, G. Betarte, and M. Rodríguez. Towards Machine-assisted Formal Procedures for the Collection of Digital Evidence. In *Proceedings of the 9th Annual International Conference on Privacy, Security and Trust (PST'11)*, pages 32 –35, July 2011.
7. M. Barrère et al. Autonomic Knowledge Discovery for Security Vulnerability Prevention in Self-governing Systems. `http://www.sticamsud.org/`. Last visited on May 17, 2015.
8. R. Bendaoud, Y. Toussaint, and A. Napoli. Pactole: A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In *Proceedings of the 16th international conference on Conceptual Structures: Knowledge Visualization and Reasoning*, pages 203–216, 2008.
9. V. Codocedo, I. Lykourentzou, and A. Napoli. A semantic approach to concept lattice-based information retrieval. *Annals of Mathematics and Artificial Intelligence*, pages 1–27, 2014.
10. CVSS, Common Vulnerability Scoring System. `http://www.first.org/cvss/`. Last visited on April 12, 2015.
11. S. Ferré and R. D. King. A dichotomic search algorithm for mining and learning in domain-specific logics. *Fundam. Inform.*, 66(1-2):1–32, 2005.
12. S. Ferré and O. Ridoux. A Logical Generalization of Formal Concept Analysis. In B. Ganter and G. W. Mineau, editors, *ICCS*, volume 1867 of *LNCS*, pages 357–370, 2000.
13. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Dec. 1999.
14. C. López, V. Codocedo, H. Astudillo, and L. M. Cysneiros. Bridging the gap between software architecture rationale formalisms and actual architecture documents: An ontology-driven approach. *Sci. Comput. Program.*, 77(1):66–80, 2012.
15. C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. July 2008.
16. N. Messai, M.-D. Devignes, A. Napoli, and M. Smaïl-Tabbone. BR-Explorer: A sound and complete FCA-based retrieval algorithm (Poster). In *ICFCA*, Dresden/Germany, 2006.

17. X. Ou, S. Govindavajhala, and A. W. Appel. Mulval: A logic-based network security analyzer. In *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, SSYM'05, pages 8–8, Berkeley, CA, USA, 2005. USENIX Association.
18. J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene. Formal concept analysis in knowledge processing: A survey on applications. *Expert Syst. Appl.*, 40(16):6538–6560, 2013.
19. S. Barnum. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX). Technical report, The MITRE Corporation, 2013.
20. B. Sertkaya. A survey on how description logic ontologies benefit from formal concept analysis. *CoRR*, abs/1107.2822, 2011.
21. S. Staab and R. Studer. *Handbook on Ontologies*. Springer Publishing Company, Incorporated, 2nd edition, 2009.
22. R. Stanley, H. Astudillo, V. Codocedo, and A. Napoli. A conceptual-kdd approach and its application to cultural heritage. In *Concept Lattices and their Applications*, pages 163–174, 2013.
23. G. Stumme. Formal concept analysis. In *Handbook on Ontologies*, pages 177–199. 2009.
24. B. E. Ulicny, J. J. Moskal, M. M. Kokar, K. Abe, and J. K. Smith. Inference and Ontologies. In *Cyber Defense and Situational Awareness*, Advances in Information Security. 2014.