A Semantic Approach to Reachability Matrix Computation

Nicole Dalia Cilia
Dept. of Philosophy,
Sapienza University of Rome,
Via Carlo Fea 3, Rome, Italy,
nicole.cilia@uniroma1.it

Noemi Scarpato
University Telematica San Raffaele
RomaVia di Val Cannuta,247 Rome,
Italy,
noemi.scarpato@unisanraffaele.gov.it

Marco Romano
Epistematica S.r.l.
Via Ostiglia 10, 20133, Milano, Italy,
m.romano@epistematica.com

Abstract— **The Cyber Security is a crucial aspect of networks management. The Reachability Matrix computation is one of the main challenge in this field. This paper presents an intelligent solution in order to address the Reachability Matrix computational problem.**

Keywords*— *CyberSecurity;Ontologies; Reasoning.

I. Introduction

In this paper we describe our contribute in the PANOPTESEC[1] project. PANOPTESEC aims to deliver beyond-state-of-the-art prototype of a cyber defence decision support system, demonstrating the benefit of a risk based approach to automated cyber defence. PANOPTESEC takes into account of the dynamic nature of information and communications technologies (ICT) and of the constantly evolving capabilities of cyber attackers in order to propose a solution based on knowledge representation and reasoning.

Recently, various studies provided progress in the Cyber Defence domain with data models and methods to focus the security problem in large networks. Morin et al. [1] have provided "a data model for security systems to query and assert knowledge about security incidents and the context in which they occur. This model constitutes a consistent and formal that an organization implements with an ICT system".

In order to better assess the effect of countermeasures to cyber-attacks and better rank countermeasures, PANOPTESEC provides a list of requirements for a system for mission impact assessment. Information about ICT assets and their vulnerabilities is used in order to compute known ways to attack a system (so-called attack graphs). Reachability Matrix is the input for the Attack Graph Generator. The PANOPTESEC approach to cyber-security maintenance support is based on a model of relations between business services and the supporting ICT assets. Business services represent the mission assessment. Information about ICT assets and their vulnerabilities is used in order to compute known ways to attack a system (so-called attack graphs). Reachability Matrix is the input for the Attack Graph Generator.

The scope of this paper encompasses data collection and correlation for the ACEA use case, but also provides a generalized approach for cyber security domain. We propose a semantic approach that applying the formalism of Description Logics [1] to the Cyber Security domain.

Following we describe the Reachability Matrix Correlator (RMC) component, the Reachability Matrix Ontology (RMO) and the reasoning task. RMC provides algorithms for computing reachability information. RMO describes the Cyber Security domain. Reasoning task uses the RMO ontology in order to compute the Reachability Matrix. Our approach foresees that RMC populates the RMO with input data provided by PANOPTESEC Data Collection and Correlation system (see section II and III for more details) and applies a set of SWRL rules and SPARQL queries to compute the Reachability Matrix (see section IV for more details). Reachability Matrix is employed in PANOPTESEC to determine if a node can reach another node (via ISO/OSI layer protocols), this information is crucial to risk management.

II. Reachability Matrix Correlator

As shown in **Fig. 1** the PANOPTESEC architecture includes: Visualization System, data Collection and Correlation System, Dynamic Risk Management System, Integration Framework and Monitored System. The Data Collection and Correlation System (DCC) has the goal of providing suitable data to all other components required for building a cyber-security protection system. The Reachability Matrix Correlator component is part of the Data Collection and Correlation System.

The role of DCC in the PANOPTESEC project is to develop a data collection and a correlation engine for building an advanced cyber-security maintenance system.

The Data Collection and Correlation System, which also contains a model for describing the impact of cyber-attacks as well as corresponding countermeasures (based on a mission model), will provide the necessary input for other components of PANOPTESEC. DCC is composed by five main components: Data Collection Interface, Data Collection Collector, Low-Level Correlator, Reachability Matrix Correlator and Mission Impact Module (see **Fig. 1**). The DCC module needs to avoid duplicate data handling and complex synchronization principles. RMC provides the reachability matrix, useful for the Attack Graph Generation component of

[1] http://www.panoptesec.eu/

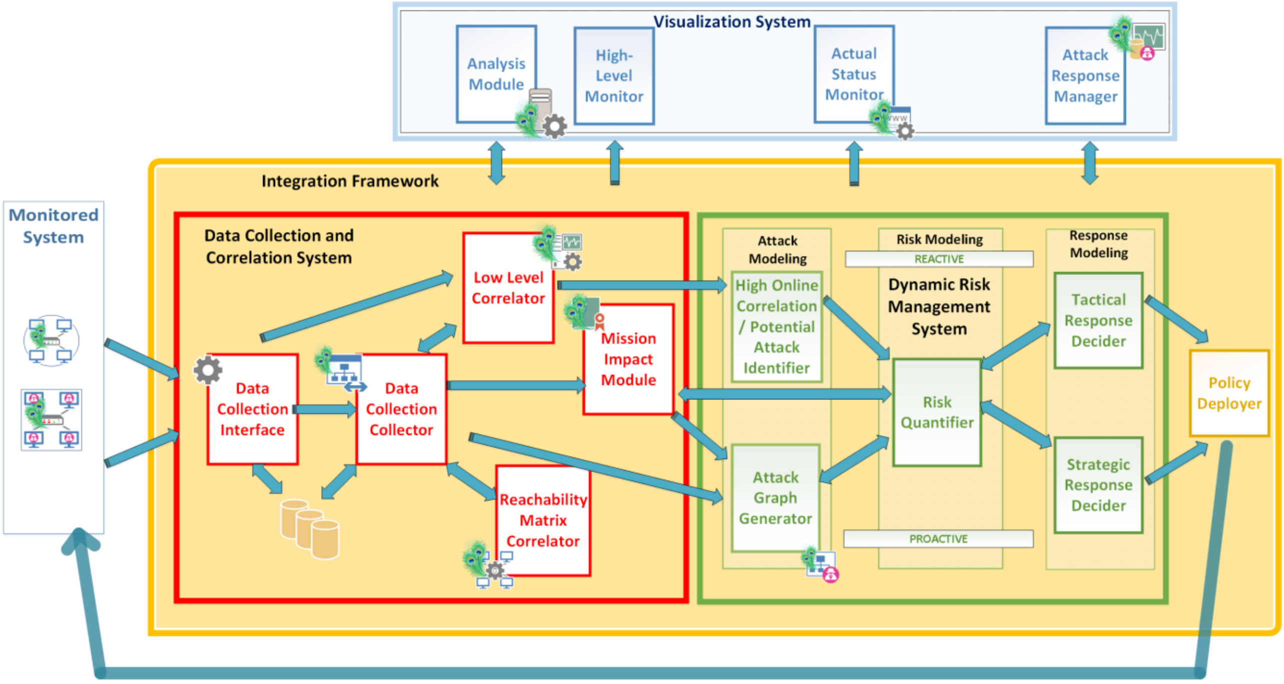

Fig. 1: PANOPTESEC global architecture and logical data flows.

the Dynamic Risk Management Response System. The reachability information are used to determine if a node can reach another node (via ISO/OSI layer protocols).

RMC performs the reachability computation across the monitored ICT network to deduct if two nodes are reachable from each other in the network, for all pairs of nodes representing ICT devices. To achieve the RMC goal we need to produce an abstract machine-readable representation of the knowledge, the RMO (see next section for details). As shown in **Fig. 2** RMO is imported from an external file and stored in the Graph Database Sesame [2] by the T-Box Loader, this operation happens once at the initialization of the RMC and successively only if RMO is changed. Then A-Box Loader populates the RMO with the information regarding the Network Inventory, Deploy Access Control Policy and Mitigation Action. Finally the reachability correlation engine computes the Reachability Matrix using information stored into Knowledge Base.

The RMC must:

- determine if a node is reachable from another node on a logical level. To provide at a logical level if a node can be reached from another node If in a network a node is reachable from another node, there is a possibility that an adversary might be able to infiltrate a network further. Such information is gatherable from, e.g., Firewall Rules, Mapping Rules, Firewall Logs and/or Traffic Captures.
- determine reachability in terms of Source-Port, Target-Port, Protocol to obtain a detailed view of reachability in a network and provide the most available information if a node is reachable over a specific port and protocol, there might exist vulnerabilities in such a protocol. Further a node might be reachable, but this reachability does not allow an adversary to progress further in a network.
- identify physical entities responsible for a reachability. Identify hardware entities, e.g. Firewalls, Switches, Routers, that route a reachability on a physical level. A logically non-existing hardware, e.g. a switch, but itself be prone to vulnerabilities, which might allow an adversary to broaden a reachability.
- consider that a node might be known via multiple addresses To identify a reachability on a logical level between unique devices in a subnetwork, entities might be addressed (e.g. IP) in another way, than from outside the subnetwork.

The requirements presented are secondary to:

a) Identifying and defining an adequate representation of knowledge (ontology), the proper Knowledge Base and the appropriate Knowledge Repository [2] to store the Knowledge Base,

b) Defining a proper mode to populate the Knowledge Base.

Several tasks are needed to achieve the RMC goal:

a) To study and produce a correct representation of the problem, using the most suitable available methods.

b) To perform applied research to determine the optimum methods to solve automatically the problem.

c) To guarantee that the representation of the knowledge, the representation of the problem and the solving method are abstract enough to be independent of any specific commercial networking devices or applications.

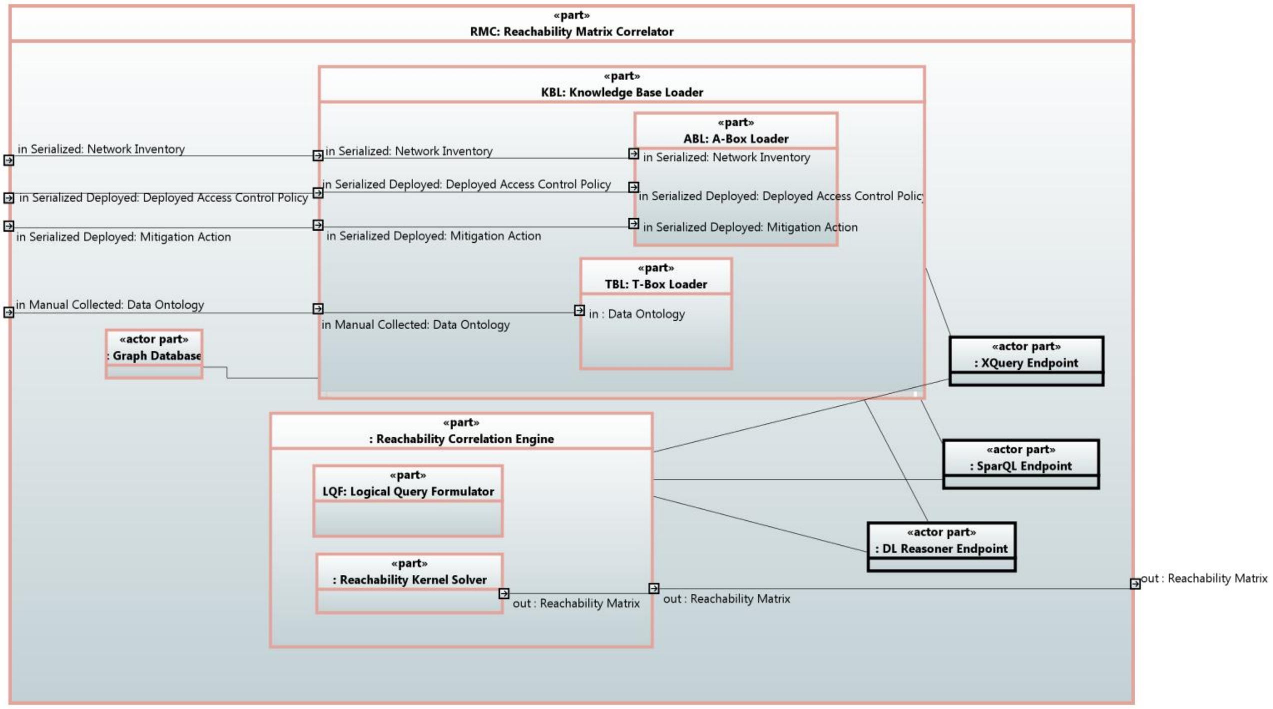
«part»
RMC: Reachability Matrix Correlator
«part»
KBL: Knowledge Base Loader
«part»
ABL: A-Box Loader
in Serialized: Network Inventory
in Serialized Deployed: Deployed Access Control Policy
in Serialized Deployed: Mitigation Action
in Serialized: Network Inventory
in Serialized Deployed: Deployed Access Control Policy
in Serialized Deployed: Mitigation Action
in Serialized: Network Inventory
in Serialized Deployed: Deployed Access Control Polic
in Serialized Deployed: Mitigation Action
«part»
TBL: T-Box Loader
in Manual Collected: Data Ontology
in Manual Collected: Data Ontology
in : Data Ontology
«actor part»
: Graph Database
«actor part»
: XQuery Endpoint
«part»
: Reachability Correlation Engine
«part»
LQF: Logical Query Formulator
«part»
: Reachability Kernel Solver
«actor part»
: SparQL Endpoint
«actor part»
: DL Reasoner Endpoint
out : Reachability Matrix
out : Reachability Matrix
out : Reachability Matrix

Fig. 2: Component View: Reachability Matrix Correlator (Internal Blocks Diagram)

d) To integrate the available tools and methods into a running prototype correlator that produces the correct Reachability Matrix.

e) To guarantee that the IT services provided by the Correlator are be aligned to the needs of the Attack Graph Generation component of the Dynamic Risk Management Response System .

f) To guarantee the compatibility of the technical solution with the overall PANOPTESEC system

g) To guarantee that the technical implementation of the solution within PANOPTESEC system is performing at an acceptable service level for a prototype within the project scope and quality requirements.

III. THE KNOWLEDGE BASE

As mentioned above RMO ontology [3] represents connectivity, Network Inventory, Access Control Policy and Mitigation Actions. RMO has the following characteristics:

- Capable of representing all kind of ICT devices, including terminal machines,
- Able to represent connectivity including sub-netting,
- Able to describe ICT devices grouping, according to filtering rules in machines,
- Capable of representing gateways and firewalling rules,
- Having the capacity to represent connectivity between every kind of nodes in the ICT network,
- Able to represent Deployed Access Control Policy Rules,
- Able to represent Deployed Mitigation Actions.

In order to design RMO used within the RMC component, we faced a thorough and deep study of IP networks in order to identify all objects that come into play in such a domain, with all their characteristics, the relationships between them, and the role each element plays in successful communications over IP networks. This study has gone in parallel with the analysis of the Data Model that we have been provided with, with special regards for the schemas concerning the Network Inventory and the Deployed Access Policies (which are basically the format of the data incoming into our component as its input), and the schema for the Reachability Matrix which determines the format for the data that exits our component.

The resulting knowledge representation, which is an OWL [4] ontology, provides a reconciled vision of these partial Data Models, in such a way that the ontology has “room” enough to receive all data from Network Inventory (and the other input files about routing tables, firewall rules and NAT rules) so as to compose a Knowledge Base (static T-box, plus the A-box reloaded over time), and to re-model the data so as to fit the output data format required by the Reachability Matrix data model. Of course, input and output formats described in the It Data Model do overlap, since are different views of the same matter. More precisely, the output that we produce contains a subset of all the information which is in the input that we receive, but enriched with some “new” information.

This is the information made explicit by automatic reasoning, thanks to the “logical embedding” of the additional knowledge about the functioning of IP networks (derived from our initial study) that is recorded in the ontology (in particular in the T-box). While designing the ontology, the classes described in the Data Model schemas, with all their attributes, needed to be re-modelled to fit the different representation paradigm of ontologies. The most typical cause of intervention is the need to distinguish objects - and their relationships with other objects (possibly with different types of objects) - from simple values that express attributes of the objects (sort of terminal, minimal points of information about which is not possible to say anything else). Briefly, at the present stage of development, the ontology has an expressivity well within OWL-DL [4] expressivity (allowing for good performance of reasoning). It counts with 37 named classes (i.e. concepts in the ontology T-box) that collect the objects accounted for in the Network Inventory and the other input files. There also 37 different relationships (object properties according to the OWL terminology) to represent the possible relationships among the objects of this classes, and other 55 (datatype) properties to account for all other characteristics of the objects.

The most important part for the function of our module (which at present is focused on reachability at the layer 3 of the OSI model) is the part that accounts for: nodes identification, network interfaces, network they belong to, and routing instructions to reach other networks, i.e. the routes and the complex information to describe them: the source (node&interface), the destination (network), and the gateway to pass through. Besides the classes that collect the objects of these various types, a set of 12 object properties allow to logically model the reachability between nodes. These (object) properties deal with:

- the network interfaces belonging to some node
- the network that each interface is connected to
- and, as a consequence, the networks that a node belongs to.

But also they deal with the other networks that can be reached by passing through one or more gateways, based on routing instructions. Finally, a set of 4 SWRL [5] (Semantic Web Rule Language) rules "force" the reasoner [6] to compute, for every interface of a node, every other node it can reach to (further details on this regard in the next section).

IV. About the Reasoning

The very first reasoning service used with regard to our ontology is the consistency check of the T-box, which is run at the design time of the ontology. Of course, the ontology passed this check. Subsequent check is the validation of the entire knowledged base. Once the A-box is loaded along with the T-box, and the whole KB is loaded into the framework of our component, this second service checks whether the A-box – produced based on the input data (Network Inventory and other files) – is consistent with respect to the T-box. Normally, a fail in this check would highlight an error in the way the input data is translated into the A-box, hence still an error in the ontology design.

The most interesting part is the reasoning triggered by the SWRL rules that rely on information stored in the Knowledge Base. These rules are typical logical rules of the form:

IF condition1 and … condition N THEN consequence.

The rules provided along with our ontology describe all possible scenarios to be investigated in order to detect all the nodes that are reachable from any given couple made of a node (with its specific routing instructions) and any of its network interfaces (connected each to one particular network).

The first rule covers the case of all reachable nodes within the same domain to which a given node belongs. The second rule covers the case of all reachable nodes within some known networks for which special routing instructions are given. The third rule covers the case of all other networks not known in advance, yet reachable through a series of "hops" to default gateways. Though absolutely necessary, the fourth rules does not cover any special case. It only enforces the reasoning in such a way that the transitivity of the relevant relationships (object properties in the ontology describing the functioning of "hopping" through gateways) is properly taken into account by the reasoner [6]. The execution of the reasoning based on all the information within the Knowledge Base and the four SWRL rules, allows to produce the set of all pairs made of a network interface and the nodes it can reach (discovered by looking at every network interface of a node, its direct connections and the routing instructions given to the node it belongs to). Here we have all information needed to produce the reachability matrix (as it is at present stage, at layer 3 of the OSI model).

Last step to produce our output – the Reachability Matrix – for use on the part of the other components is to explicitly point out, for each network interface of any given node, the set of all and only the other nodes that it can reach. However, this is not properly speaking reasoning, since it is just retrieval of triples (the form in which data are declared in OWL), and it is achieved by firing some SPARQL [7] queries (actually embedded in the APIs [8] of the persistence environment that we adopt). Other similar queries retrieve the rest of information that is available in the Knowledge Base and is expected in the Reachability Matrix according to the output format.

V. Acknowledgements

This paper has been supported by Epistematica (http://www.epistematica.com/) within the PANOPTESEC project.

References

[1] Morin, B, L Mé, H Debar, M Ducassé (2009). A logic-based model to support alert correlation in intrusion detection . *Information Fusion,* 10 (4), 285-299.

[2] Sesame (http://rdf4j.org/).

[3] Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing?. *International journal of human-computer studies*, *43*(5), 907-928.

[4] McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, *10*(10), 2004.

[5] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, *21*, 79.

[6] Haarslev, V., & Müller, R. (2001). RACER system description. In *Automated Reasoning* (pp. 701-705). Springer Berlin Heidelberg.

[7] Prud'Hommeaux, E., & Seaborne, A. (2008). SPARQL query language for RDF. *W3C recommendation*, *15*.

[8] Stellato A. OWLART API (http://art.uniroma2.it/owlart/).