Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Reachability Matrix Ontology: A Cybersecurity Ontology

Noemi Scarpato, Nicole Dalia Cilia & Marco Romano

To cite this article: Noemi Scarpato, Nicole Dalia Cilia & Marco Romano (2019): Reachability Matrix Ontology: A Cybersecurity Ontology, Applied Artificial Intelligence, DOI: 10.1080/08839514.2019.1592344

To link to this article: https://doi.org/10.1080/08839514.2019.1592344

APPLIED ARTIFICIAL INTELLIGENCE
https://doi.org/10.1080/08839514.2019.1592344

Taylor & Francis
Taylor & Francis Group

Reachability Matrix Ontology: A Cybersecurity Ontology

Noemi Scarpato [a], Nicole Dalia Cilia[b], and Marco Romano[c]

[a]Department of Human sciences and Promotion of the quality of life, San Raffaele Roma Open University, Rome, Italy; [b]Department of Electrical and Information Engineering, University of Cassino and Lazio meridionale; [c]Epistematica, Milano, Italy

ABSTRACT

In this paper, we describe the Reachability Matrix Ontology (RMO). RMO aims to describe the networks and the cybersecurity domain in order to compute the reachability information (reachability matrix). Reachability Matrix determines if a node can reach another node (via ISO/OSI layers protocol).

To achieve this objective RMO describes the network's elements, the network connectivity information, and the access control policies. RMO also provides some SWRL rules able to calculate the Reachability Matrix. Besides RMO and SWRL rules, there are also a set of SPARQL queries to refine the computation of the Reachability Matrix.

To the best of our knowledge, RMO represents an innovative approach to the computation of the reachability matrix. Following we will describe our approach based on a strategy that exploits a combination of OWL, description logic rules and SPARQL queries.

Introduction

In Risk Based Security (2016), the authors reported that there have been 2,991 breaches during the first nine months of 2016 exposing over 2.2 billion records.

Many of these cyberattacks aim to affect critical infrastructure, public web sites or most popular web sites, usually personal and sensitive data as financial or medical data (Guadagni et al. 2016) (Ferroni et al. 2016) and (Guadgni 2017) are the preferred goal of these kinds of attacks.

Prevent these situations and/or define the real-time defense strategies are the key tasks of cybersecurity systems.

Traditionally ontological approaches are successfully applied in various domains such as legal domain (Pazienza, Scarpato, and Stellato 2009), (Bianchi et al. 2009), (Boella et al. 2016), cultural Heritage domain (Accardi and Chiarenza 2016), (Pennacchiotti and Zanzotto 2008) and semantic web (Fallucchi et. al 2014).

CONTACT Noemi Scarpato noemi.scarpato@unisanraffaele.gov.it Department of Human sciences and Promotion of the quality of life, San Raffaele Roma Open University, Via Val Cannuta, 247, Rome, Italy

The Ontological approach to cybersecurity issues is an innovative trend in cybersecurity domain. Recently few attempts to develop a cybersecurity ontology are been performed (see section 2 for further details). To the best of our knowledge, existing approaches are related to the attack detection and to the definition of mitigation actions instead, in our approach, we carry out the computation of Reachability Matrix via a logical approach. We think that the identification of the correct Reachability Matrix is a key aspect in order to create a cybersecurity system because is the base for all other components.

In this paper, we wish to demonstrate that the computation of the reachability matrix can and should be performed through an ontological approach. In particular, we provide an adaptive method able to calculate reachability matrix in all networks configurations exploiting SWRL (Ian Horrocks et al. 2004) rules and SPARQL (Prud'hommeaux Eric and Seaborne Andy 2008) queries combined with RMO Ontology.

Following, we describe the state of art of cybersecurity ontological approaches and their application in real contexts. Further, we illustrate the latest version of RMO, this version is able to analyze networks features at layer four of ISO/OSI protocol, actually RMO was developed to represent all levels of ISO/OSI protocol through the definition of dedicated SWRL rules, but at the present stage of development, it describes the networks at level four of ISO/OSI protocol. Also, we describe the PANOPTESEC[1] system, a cybersecurity decision support system, and in particular, we illustrate the implementation of RMO ontology strategy into the Reachability Matrix Correlator component of PANOPTESEC. Finally, we provide our deduction about the ontological approaches in cybersecurity and we indicate future works in order to represent all levels of ISO/OSI protocol in our RMO ontology.

Cybersecurity Ontologies Background

Ontological approaches in cybersecurity are a relatively new task in the cybersecurity domain. Despite this, many attempts are been provided both to define single ontology and to define methodological approaches in the creation of cybersecurity ontologies.

An interesting approach is described in (Obrst, Chase, and Markeloff 2014), in this paper authors describe a method to develop a Cyber ontology and provide a prototype of this kind of ontology. To achieve this issue, authors investigate many aspects of the cybersecurity domain and indicate the possible ontologies and standards that could be used in the creation of a cybersecurity ontology.

In Oltramari et al. (2014), the authors describe an ontology framework to develop a set of ontologies able to describe cyber operations as complex entities. In this paper, the authors analyzed the complex cybersecurity domain

and provide a three-level ontology for the cybersecurity: CRATELO. Basic level of CRATELO is a domain ontology of cyber operations (OSCO), the middle-level ontology (SECCO) represents security-related concepts finally top level is DOLCE ontology in order to describe generic characteristics of world entities (shape, dimension, qualities, etc.).

In Balduccini, Kushner, and Speck (2015), the authors presents an architecture able extract semantically rich content from a human-readable file, this architecture exploits machine learning modules for automatic file format identification, tokenization, and entity identification. The process is driven by an ontology of cybersecurity domain-specific concepts.

With respect to creation of dedicated ontologies in cybersecurity domain, there are many examples in literature that aim to represent or a particular aspect of the considered domain ((Simmonds, Sandilands, and van Ekert 2004),(Oltramari et al. 2017)) or the domain in all of its peculiarities ((Iannacone et al. 2015), (Syed et al. 2017)).

At the best of our knowledge, no ontologies that are able to describe information needed to generate the Reachability Matrix has been provided yet. In this paper, we present our cybersecurity ontology (RMO), this ontology is able to represent all peculiarities of the Reachability Matrix, i.e the subspace of all the states reachable from a specific network, from the origin to a given time.

Reachability Matrix Ontology

Design of RMO

As mentioned above we design the RMO in order to allow the computation of reachability matrix via an ontological approach. To realize our ontology we adopt the so-called "middle-out" approach, a blended top-down and bottom-up approach. On the one hand, we study thoroughly and deeply the IP networks in order to identify all aspects of such domain, with all their characteristics, the relationships between them, and the role each element plays in successful communications over IP networks.

On the other, we perform a complete analysis of the data model provided by PANOPTESEC modules, with special regards for the schemas concerning the Network Inventory (NI) and the Deployed Access Control Policies (DACP), which are the expected input for RMC component, and the scheme of the Reachability Matrix which is the expected output of RMC.

The RMO represented via OWL (McGuinness and Harmelen 2004), provides a unified vision of these partial data models and of general domain information.

In such a way, the RMO is able to: represent all data provided by NI and DACP, create the Knowledge Base (static T-box, plus the A-box reloaded over time) and to calculate the Reachability Matrix. The Reachability Matrix

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 information has been made explicit by the automatic reasoning, performed exploiting the logical SWRL rules defined into RMO.

While designing the ontology, the classes described in the data model schemas, with all their attributes have been modeled to fit the objectives of PANOPTESEC project.

In particular, we solved the need to distinguish objects and their relationships with

other objects from simple values that express attributes of the objects (minimal points of information about which is not possible to say anything else).

At the present stage of the development, the ontology has an expressivity well within

OWL-DL expressivity (useful for good performance of the reasoning). RMO counts 25 named classes (i.e., concepts in the ontology) that collect the objects described in the Network Inventory and in the other input files. There are 16 different object properties able to represent the possible relationships among the objects of classes, and other 32 datatype properties to account for all other characteristics of the objects.

Quick Presentation of RMO Ontology

Key Elements

The ontology is all about a small set of elements (25 concepts), described by a number of datatype properties (32), which take plain literals as value, and object properties (16) which relate an element to another element.

Key elements are basically networks, nodes, interfaces (IP interfaces in particular), routes, ports (along with protocols) and DACP Rules (Deployed Access Control Policy Rules). Most of the data parsed from the input files and loaded into the Knowledge Base are arranged with respect to the ontology based on this subset of key elements.

Network

Networks are discovered by analyzing information about IP interfaces of the nodes and routing rules. Based on IP addresses of all the known nodes, indeed, and the data that fully describe every routing rule, all networks which are relevant to the Reachability Matrix are identified. One special instance of network is Internet, which does not belong to the monitored system, but nevertheless plays a role as far as external nodes in the Internet may be communicating with nodes within the monitored system.

Node

Nodes are the core objects of the RMC. A major part of the data provided in the NI input JSON

file are about them. Also, most of such data is represented by means of datatype properties directly attached to the node objects (e.g., its geographical location) or to 'parts' of a node (such as its Operating System for instance). Since ad hoc object properties relate a node to, e.g., its OS, it is straightforward to follow such relations and get all final data relevant to a given node. It is the case for instance of active users logged into a session of the OS on a given device. As a consequence, the template below lists only the properties which are relevant to the calculation of the Reachability Matrix.

A Node is completely described (as far as the focus is on reachability) by the following series of triples:

[URI of the node] nodeID [string datatype]
[URI] hostName [string datatype]
[URI] hasInterface [object] (points out all network interfaces of a node)
[URI] hasRoute [object] (links a node to any route it may use to send data over IP networks)
[URI] hopsTo [object] (points out another node(s) the node hops to in order to reach to other networks).
[URI] belongsTo [object] (spots the network the node belongs to).

A few subtypes of nodes are noteworthy: Internal nodes are all node described within the NI input file. They are assumed to belong to the set of networks that compose the monitored system.

External nodes are all other nodes. By definition, all nodes belonging to the Internet are external nodes.

Interface

Whereas nodes are the arriving points (terminal ends) of reachability, the IP interfaces are the starting points. Nodes and interfaces are linked in the ontology by the *hasInterface* object property, and every node may have multiple interfaces. An IP interface is completely described by the following series of triples:

[URI of the interface] interfaceID [string datatype]
[URI] connectedTo [network object] (spots out the network an interface is directly connected to)
[URI] startsRoute [route object] (links an interface to any route that can be followed from there)
[URI] hasPort [port object] (links an interface to any of its ports)
[URI] intReachesOut [IP interface object] (records reachability between an interface and the other IP interfaces it actually can reach to).

Note that values for the *intReachesOut* property are not filled in at parsing time of input JSON files (NI and DACP), rather they are filled in with the results of the reasoning.

Port

For every IPInterface known in the NI input file, a set of ports is represented. A Port is completely described by the following series of triples:

[URI of the port] ipPortID [string datatype]
[URI] portState [string (open|closed) datatype]
[URI] portNumber [integer (0–65535) – datatype]
[URI] portProtocol [string (TCP|IP) datatype] (links the port to the transport protocol it uses, either TCP or IP)
[URI] hasServiceProtocolDescription [string datatype] (spots the service protocol listening on that port)

Note that the pair of values for *portNumber* and *portProtocol* properties identifies the same real world Service Protocol as the *hasServiceProtocolDescription* datatype property alone. This double way to recover Service Protocols is demanded by the different use scenarios envisaged.

Route

A Route is completely described by the following series of triples:

[URI of the route] routingRuleID [datatype]
[URI of a node] hasRoute [object – this route] (links a routing rule with the node that may use it)
[URI of an IPInterface] startsRoute [object – this route] (links the routing rule to the IPInterface(s) that will actually send packets along that route)
[URI of the route] routeTo [object – network] (spots the network which is reachable via that route)
[URI] routeVia [object – node] (spots the node that acts as a gateway for that route).

DACPRule

DACPRules (Deployed Access Control Policy Rules) are divided into two subtypes. They are all generated, for every new input provision, based on the content of the DACP input document.

Most of the calculation of the Reachability Matrix at OSI layer 4 depends on information contained in the representation of the following elements:

FWRule

A FWRule (Firewall Rule) is completely described by the following series of triples:

[URI of the rule] DACPruleID [datatype]
[URI] registeredAt [node – object] (links a FWRule to the device that executes it)
[URI] allowsNode [node – object] (indicates the set of nodes matching the Destination value for this rule in the DACP input, which will be reachable if the rule is passed)
[URI] allowsIPProtocol [string representing the IPProtocol (TCP or UDP) allowed by this rule – datatype]
[URI] allowsPortNumber [integer (0–65535) representing the port numbers allowed by this rule – datatype]
[URI] appliesTo [IP interface – object] (indicates the set of IP interfaces matching the Source value for this rule in the DACP input, for which the rule shall be fired).

NAT Rule

A NATRule (Network Address Translation Rule) is completely described by the following series of triples:

[URI of the rule] DACPruleID [datatype]
[URI] registeredAt [node – object] (links a NAT Rule to the device that executes it)
[URI] allowsNode [node – object] (indicates the set of nodes matching the Destination value for this rule in the DACP input, which will be reachable if the rule is passed)
[URI] allowsIPProtocol [string representing the IPProtocol (TCP or UDP) allowed by this rule – datatype]
[URI] allowsPortNumber [integer (0–65535) representing the port numbers allowed by this rule – datatype]
[URI] appliesTo [IP interface – object] (indicates the set of IP interfaces matching the Source value for this rule in the DACP input, for which the rule shall be fired).

SWRL Rules in Reachability Matrix Ontology

As mentioned above RMO includes a set of SWRL rules in order to introduce reasoning in the ontology. These rules are able to compute the reachability matrix at level three of ISO/OSI protocol. In general, we have designed our ontology to represent all levels of ISO/OSI, but at present, RMO implements level three of ISO/OSI protocol by exploiting existing SWRL rules and level four by exploiting SPARQL queries (see the following section). To represent further levels only new SWRL rules or SPARQL queries are needed. Following we describe these rules in details.

Set_StartsRoute

This rule generates triples where each interface node gets paired with route node because each route needs to know what is the interface where it goes out to the network. Every route consists of hostname and interface name properties, based on those values Reasoner matches subject and object of this triple.

Route(?route) Node(?node) interfaceName(?route, ?ifc_name) ∧ interfaceName(?ifc, ?ifc_name) hostName(?route, ?host_name) hostName(?node, ?host_name) ∧ hasInterface(?node, ?ifc) → startsRoute(?ifc, ?route)

Set_RouteVia

This rule generates triples where each route node gets paired with a gateway node. Each route must have the gateway where the IP packet will be transferred.

Every route has the gateway value which is the IP address.

gateway(?route, ?ip) ∧ hasInterface(?gw, ?ifc) ∧ hasAddress(?ifc, ?addr) ∧ address(?addr, ?ip) → ∧ routeVia(?route, ?gw)

Set_hopsTo

This rule generates triples where each node gets paired with directly connected gateway node. Actually, this is the edge in the network graph. It uses two previous rules to generate a new triple.

hasInterface(?node, ?ifc) ∧ startsRoute(?ifc, ?route) ∧ routeVia(?route, ?gw) → ∧ hopsTo(?node, ?gw)

Trans_hopsTo

This rule is a transitive one. It creates an indirect connection between a node and a gateway.

hopsTo(?x, ?g1) ∧ hopsTo(?g1, ?g2) → hopsTo(?x, ?g2)

InSameNetwork-PerInterface

These rules generate the triples of interfaces which are reachable and are in the same network.

connectedTo(?int1, ?net) ∧ connectedTo(?int2, ?net) → intReachesOut(?int1, ?int2)
InOtherNetwork-PerInterface

This rule also generates the triples of interfaces which are reachable, but they are not in the same network. The new triple is generated if a gateway of one interface and gateway of another are directly or indirectly connected.

connectedTo(?int1, ?net1) ∧ connectedTo(?int2, ?net2)∧ startsRoute(?int1, ?r) ∧ routeVia(?r, ?g1) ∧ hopsTo(?g1, ?g2) ∧ belongsTo(?g2, ?net2) →intReachesOut(?int1, ?int2)

Sparql Queries in Reachability Matrix Ontology

To represent the level four of ISO/OSI protocol we defined a set of SPARQL queries able to compute the Reachability Matrix.

These queries are executed inside the Java code, and they were mostly used for easier parsing and preparing the data for the Reasoner.

Following we describe these queries in details.

Same Network

This query simply checks if two interfaces belong to the same network: *?net*. If such network does not exist, then the result set will be empty.

```
SELECT DISTINCT ?net WHERE {
ifc1 connectedTo ?net .
ifc2 connectedTo ?net . }
```

Node Adjacency

This query is used in finding Responsible paths. The result of this query is the list of (directly) adjacent nodes (gateways) of the node. These are the edges of the graph, while vertices are the network nodes. This is the subset of triples which have *hopsTo* property, i.e., before applying transitive *hopsTo* SWRL rule.

```
SELECT DISTINCT ?gw WHERE {
node hasInterface ?ifc .
ifc startsRoute ?route .
?route routeVia ?gw .}
```

Starts Route

This is a fix for SWRL rule *Set_StartsRoute*. Routing tables sometimes miss the information about the interface from that route starts from, so this SPARQL query resolves it. For each route in the system which does not have any value for a *interfaceName* property, we call this query. It finds all interfaces which belong to the gateway written in the *gateway* triple. After finding it, new triples with *startsRoute* can be added.

```
SELECT DISTINCT ?ifc WHERE {
?gw belongsTo ?net .
node hasInterface ?ifc .
?ifc connectedTo ?net .}
```

Firewall Ports and Protocols

This query returns all open ports for a pair of interfaces *ifc* and *ifc_reach* on a Firewall node. This query is called repetitively on the whole Responsible path.

```
SELECT DISTINCT ?ports ?protocols WHERE {
?fw appliesTo ifc .
```

```
?fw allowsNode node_reach .
?fw registeredAt fw_node_name .
?fw allowsPortNumber ?ports .
?fw allowsIPProtocol ?protocols .
?fw RDF.type FWRule .
?fw allowsIFC ifc_reach .}
NAT ports and protocols
```

This query returns all open ports for a pair of interfaces *ifc* and *ifc_reach* on a NAT node. This query is called repetitively on the whole Responsible path.

```
SELECT DISTINCT ?ports ?protocols WHERE {
?nat appliesTo ifc .
?nat allowsNode node_reach .
?nat registeredAt nat_node_name .
?nat allowsPortNumber ?ports .
?nat allowsIPProtocol ?protocols .
?nat RDF.type NATRule .
?nat allowsIFC ifc_reach .}
```

Panoptesec Project

"Panoptes" is an ancient Greek term meaning, "all eyes" or "all seeing". This term has incorporated into the project name to represent the PANOPTESEC consortium because the overall goal of the PANOPTESEC project is to deliver a continuous cybersecurity monitoring and response capability.

The PANOPTESEC project is a beyond-state-of-the-art prototype of a cybersecurity decision support system. PANOPTESEC realize a risk-based approach to automated cybersecurity that accounts for the dynamic nature of information and communications technologies (ICT) and the constantly evolving capabilities of cyber attackers.

As mentioned above the security issues related to networks and computer systems are a crucial aspect for organizations in order to protect their business operations and services. Unfortunately, the capabilities of the hackers to attack these systems are continuously increasing.

Nowadays the existing commercial solutions do not meet the demands of cybersecurity decision support systems able to: continuous monitoring the ICT systems, detect vulnerabilities and attacks and provide rapid incident responses.

The PANOPTESEC project aims to take on these challenges through the definition of the PANOPTESEC prototype. This prototype provides a continuous monitoring and response capability to prevent, detect, manage and react to cyberattacks in real-time. Also, it supports breach notifications and improves situation awareness while supporting the decision-making process in critical situations.

PANOPTESEC delivers this capability through a modular architecture developed via the integration of technologies that will collectively deliver the required capabilities.

The PANOPTESEC architecture is composed of:

- Visualization System,
- Data Collection and Correlation System,
- Dynamic Risk Management System,
- Integration Framework and Monitored System.

Following we describe the Reachability Matrix Correlator (RMC) component that is part of Data Collection and Correlation System. The role of Data Collection and Correlation System (DCC) in PANOPTESEC project is to develop a data collection and a correlation engine for building an advanced cybersecurity system. The DCC provides the necessary input for other components of PANOPTESEC.

DCC is made up of five main components:

- Data Collection Interface,
- Data Collection Collector,
- Low-Level Correlator,
- Reachability Matrix Correlator,
- Mission Impact Module.

Reachability Matrix Correlator (RMC) (Cilia, Scarpato, and Romano 2015) calculate the reachability matrix, that is one of the main inputs for the other modules of PANOPTESEC. Following we describe RMO ontology, it is the ontology exploited by RMC to compute the reachability matrix.

Conclusion and Future Works

In this paper, we analyzed the state of art of the cybersecurity ontologies creation and we have defined our approach to this issue. To aims this objective, we presented an innovative approach to the definition of the reachability matrix exploiting the power of semantic representation and reasoning. Our hypothesis is that a blended approach between RMO Ontology, SWRL rules, and SPARQL queries is the best solution in order to calculate the reachability matrix.

The reachability matrix provides basic information to design cybersecurity systems so it is a very key information for this kind of systems. To realize the reachability matrix computation we presented the RMO ontology and a set of SWRL rules and SPARQL queries able to calculate the reachability matrix compliant with level four of ISO/OSI protocol. Our ontology is designed to

represent all levels of this protocol via the definition of dedicated SWRL rules and SPARQL queries. We will define these rules and queries in the next version of RMO ontology.

Acknowledgments

This work has been partially supported by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 610416 (PANOPTESEC). The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

Funding

This work has been partially supported by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 610416 (PANOPTESEC).

Note

1. http://www.panoptesec.eu/dissemination/public_deliverables.shtml.

ORCID

Noemi Scarpato http://orcid.org/0000-0002-6573-8095

References

Accardi, A. R. D., and S. Chiarenza. 2016. Digital Museums of the Imagined Architecture : An Integrated Approach. *Disegnarecon* Vol 9, No 17 (2016).

Balduccini, M., S. Kushner, and J. Speck. 2015. Ontology-driven data semantics discovery for cyber-security. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9131:1–16. doi:10.1007/978-3-319-19686-2_1.

Bianchi, M., M. Draoli, G. Gambosi, M. T. Pazienza, N. Scarpato, and A. Stellato. 2009. ICT tools for the discovery of semantic relations in legal documents. CEUR Workshop Proceedings. Vol. 582. 2nd International Conference on ICT Solutions for Justice, Skopje, FYR Macedonia, September 24, 2009.

Boella, G., L. D. Caro, L. Humphreys, L. Robaldo, P. Rossi, and V. D. T. Leendert. 2016. Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. *Artificial Intelligence and Law* 24 (3):245–83. doi:10.1007/s10506-016-9184-3.

Cilia, N. D., N. Scarpato, and M. Romano. 2015. A semantic approach to reachability matrix computation. CEUR Workshop Proceedings. Vol. 1523. Tenth Conference on Semantic Technology for Intelligence, Defense, and Security, Fairfax VA, USA, November 18-20, 2015.

Fallucchi, F., E. Alfonsi, A. Ligi, and M. Tarquini. 2014. Ontology-Driven Public Administration Web Hosting Monitoring System. Lecture Notes in Computer Science

(Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Vol. 8842.

Ferroni, P., Fabio, M. Z, Noemi, S., Silvia, R., Umberto, N., Mario, R., and Fiorella, G. 2016. "Risk Assessment for Venous Thromboembolism in Chemotherapy-Treated Ambulatory Cancer Patients." Medical Decision Making 37 (2):234–42. https://doi.org/10.1177/0272989X16662654.

Guadagni, F., N. Scarpato, F. Patrizia, G. D'Ottavi, F. Boavida, M. Roselli, G. Garrisi, and A. Lisi. 2016. Personal and sensitive data in the E-health-IoT universe. Internet of Things. IoT Infrastructures: Second International Summit, IoT 360{\textdegree} 2015, Rome, Italy, October 27-29, 2015, Revised Selected Papers, Part II, edited by Benny Mandler, Johann Marquez-Barja, Miguel Elias Mitre Campista, Dagmar Cagávnová, Hakima Chaouchi, Sherali Zeadally, Mohamad Badra, et al., 504–14. Cham: Springer International Publishing. doi:10.1007/978-3-319-47075-7_54.

Guadagni, F., M. Z. Fabio, S. Noemi, R. Alessandro, R. Silvia, F. Patrizia, and R. Mario 2017. "RISK: A Random Optimization Interactive System Based on Kernel Learning for Predicting Breast Cancer Disease Progression." In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10208 LNCS:189–96. https://doi.org/10.1007/978-3-319-56148-6_16.

Horrocks, I., P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. 2004. SWRL: A semantic web rule language combining OWL and RuleML. https://www.w3.org/Submission/SWRL/#references.

Iannacone, M., S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, and J. Goodall. 2015. Developing an ontology for cyber security knowledge graphs. Proceedings of the 10th Annual Cyber and Information Security Research Conference on - CISR '15, ACM Press, New York, New York, USA, 1–4. doi:10.1145/2746266.2746278.

McGuinness, D. L., and F. Van Harmelen. 2004. OWL web ontology language overview. W3C Recommendation, 1–22. http://www.academia.edu/download/30759881/5.3-B1.pdf.

Obrst, L., P. Chase, and R. Markeloff. 2014. Developing an ontology of the cyber security domain. *CEUR Workshop Proceedings* 966:49–56. http://mx.franz.com/agraph/cresources/white_papers/STIDS2012_T06_ObrstEtAl_CyberOntology.pdf.

Oltramari, A., D. Henshel, M. Cains, and B. Hoffman. 2017. Towards a human factors ontology for cyber security. Accessed April 14. http://ceur-ws.org/Vol-1523/STIDS_2015_T04_Oltramari_etal.pdf.

Oltramari, A., L. F. Cranor, R. J. Walls, and M. Patrick. 2014. Building an ontology of cyber security. *CEUR Workshop Proceedings* 1304:54–61.

Pazienza, M. T., N. Scarpato, and A. Stellato. 2009. *STIA*: Experience of semantic annotation in jurisprudence domain. Frontiers in Artificial Intelligence and Applications* 205. doi:10.3233/978-1-60750-082-7-156.

Pennacchiotti, M., and F. M. Zanzotto. 2008. *Natural language processing across time: An empirical investigation on Italian*, 371–82. Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-85287-2_36.

Prud'hommeaux Eric, and Seaborne Andy. 2008. SPARQL query language for RDF. https://www.w3.org/TR/rdf-sparql-query/.

Risk Based Security. 2016. Data Breach QuickView Not Just Security, the Right, January, 1–15.

Simmonds, A., P. Sandilands, and V. E. Louis 2004. *An ontology for network security attacks*, 317–23. Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-30176-9_41.

Syed, Z., A. Padia, T. Finin, L. Mathews, and A. Joshi. 2017. UCO: A unified cybersecurity ontology.