Dissertation

Master´s Degree in Computer Engineering - Mobile Computing

A Decision Support System for Corporations Cyber Security Risk Management

Gabriela del Rocío Roldán Molina

Leiria, September 2017

Dissertation

Master´s Degree in Computer Engineering - Mobile Computing

A Decision Support System for Corporations Cyber Security Risk Management

Gabriela del Rocío Roldán Molina

MSc Thesis supervised by Professor Dr. Carlos Manuel da Silva Rabadão, Professor at School of Technology and Management from Polytechnic Institute of Leiria, MSc. Mario Giovanny Almache Cueva, Professor at Department of Computer Science from University of the Armed Force ESPE (Ecuador) and Dr. Vitor Manuel Basto Fernandes, Assistant Professor at Department of Information Science and Technology (ISTA) from Lisbon University Institute.

Leiria, September 2017

“If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.”

— Sun Tzu, The Art of War

This page was intentionally left blank

Dedication

This work is dedicated to all my family who has always been present, my parents Milton and Elsa whom I thank infinitely for giving me their constant support and love, to be an example of overcoming and for having formed me throughout my life to be the person I am today.

To my sisters Lili, Janeth and Andrea for motivating me always and be supportive at any time, thanks for those messages of motivation and concern throughout my training and completion of this work.

To God for giving me the strength to continue in any circumstance, to take care of me always and who has allowed each of the achievements achieved so far.

This page was intentionally left blank

Acknowledgements

First of all, I want to thank my supervisors Mario Almache, Carlos Rabadão for the support they have always shown and without any doubt to Professor Vitor Fernandes for his great availability at all times in the event of any doubt, for his patience and motivation for the culmination of this work in addition to Coordinator of the master's degree during the elaboration of the project Professor Carlos Grilo.

To the University of the Armed Forces and to the Department of Computer Science for allowing me to carry out the necessary tests for the completion of the work.

To the Polytechnic Institute of Leiria and Senescyt institutions that gave me the opportunity to continue to train academically and increase my skills with the acquisition of new knowledge.

On a more personal note, I can never leave unmentioned for their emotional help to my boyfriend Jorge Marcelino, family and friends.

This page was intentionally left blank

Previous Note

As the result of the work done in this thesis, the following articles were produced:

- Gabriela Roldán, Mario Almache, Carlos Rabadão, Iryna Yevseyeva and Vitor Fernandes, "A Decision Support System for Corporations Cybersecurity Management," in *12th Iberian Conference on Information Systems and Technologies*, Lisboa, 2017 (published).
- Gabriela Roldán, Mario Almache, Carlos Rabadão, Iryna Yevseyeva and Vitor Fernandes, " A Comparison of Cybersecurity Risk Analysis Tools," in *CENTERIS - Conference on ENTERprise Information Systems*, Barcelona, 2017 (to be published).

This page was intentionally left blank

Abstract

This thesis presents a decision aiding system named C3-SEC (Contex-aware Corporative Cyber Security), developed in the context of a master program at Polytechnic Institute of Leiria, Portugal. The research dimension and the corresponding software development process that followed are presented and validated with an application scenario and case study performed at Universidad de las Fuerzas Armadas ESPE – Ecuador.

C3-SEC is a decision aiding software intended to support cyber risks and cyber threats analysis of a corporative information and communications technological infrastructure. The resulting software product will help corporations Chief Information Security Officers (CISO) on cyber security risk analysis, decision-making and prevention measures for the infrastructure and information assets protection.

The work is initially focused on the evaluation of the most popular and relevant tools available for risk assessment and decision making in the cyber security domain. Their properties, metrics and strategies are studied and their support for cyber security risk analysis, decision-making and prevention is assessed for the protection of organization's information assets.

A contribution for cyber security experts decision support is then proposed by the means of reuse and integration of existing tools and C3-SEC software. C3-SEC extends existing tools features from the data collection and data analysis (perception) level to a full context-ware reference model.

The software developed makes use of semantic level, ontology-based knowledge representation and inference supported by widely adopted standards, as well as cyber security standards (CVE, CPE, CVSS, etc.) and cyber security information data sources made available by international authorities, to share and exchange information in this domain. C3-SEC development follows a context-aware systems reference model addressing the perception, comprehension, projection and decision/action layers to create corporative scale cyber security situation awareness.

Keywords: Decision making; cybersecurity; risk analysis.

This page was intentionally left blank

Figures index

Tables index

This page was intentionally left blank

Acronyms

Acronym	Meaning
AMI	Amazon Machine Images
API	Application Programming Interface
CCE	Common Configuration Enumeration
CISO	Chief Information Security Officer
CPE	Common Platform Enumeration
CSV	Comma-separated values
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
FDCC	Federal Desktop Core Configuration
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTML	HyperText Markup Language
IT	Information Technology
JDBC	Java Database Connectivity
JDK	Java Development Kit
JSF	Java Server Faces
MITRE	The MITRE Corporation
MVC	Model View Controller
NIST	National Institute of Standards and Technology
NVD	National Vulnerability Database
NVTs	Network Vulnerability Tests
OVAL	Open Vulnerability and Assessment Language
OWL	Ontology Web Language
RDF	Resource Description Framework
SCAP	Security Content Automation Protocol
SSH	Secure SHell

SWRL	Semantic Web Rule Language
URI	Uniform Resource Identifier
UWE	UML-based Web Engineering
XCCDF	eXtensible Configuration Checklist Description Format
XML	eXtensible Markup Language
XSG	Extensible Scene Graph
XSOM	XML Schema Object Model

Index

Introduction

Surely one of the greatest risks to an organization's information security is not often the weakness in the technology control environment. Rather it is the action or non-action by all the people that are using the technology. Recent reports have revealed the emergence of millions of computer security incidents per year and each year new records are reached. They refer that in 2014, 65% of companies, victim of intrusion and information theft, were notified after a late detection process that lasts 13 months on average [1].

1.1 Motivation

Current cyber security reports like the one mentioned in the previous section, motivates the development of new technologies that can augment human understanding and decision-making abilities to create situation awareness in cyber environments. Situation awareness in cyber environments is made possible by the process of deriving context knowledge (awareness) from a multitude of information sources. Generally, it comprises three main levels, perception, comprehension and projection, which feeds the decision and action cycle. Perception, involves sensory of significant information about the system itself and the environment it is operating in. This information can be obtained with the help of data collection tools related to the technological infrastructure of an organization (hardware, services, databases). Comprehension, encompasses more than simply sensing/perceiving data, it relates the meaning of the information with the system goal/purpose. It can be represented through an ontology for context knowledge representation. Projection, consists of predicting how system current state will evolve (in time) and how it will affect the future states of the operating environment.

Currently there are tools that comprise the different levels of situation awareness to help detect, prevent and recover from cyber incidents that could threaten the security of an organization. Nevertheless, many existing security tools and approaches focus on system and application levels. For this reason, security analysts need more up to date systematic methods to quantitatively evaluate network vulnerabilities, predict attack risk and potential

impacts, assess proper actions to minimize business damages, and ensure mission success in a hostile environment. As a natural descendant of this requirement, security metrics are of major importance for context security awareness, coordinated network defense, and mission assurance analysis. They can provide a better understanding of the adequacy of security controls, and help security analysts to effectively identify which critical assets to focus their limited resources on to ensure mission success [2].

1.2 Objectives

This work proposes a context-aware systems approach to identify, define, develop and apply a simple comprehensive security and business continuity assurance analysis. The research addresses existing security tools and metrics for the cyber security domain for systems and network operations analysis, along the context-aware system approach layers, perception, comprehension, projection and decision/action cycle.

The (software) decision support system resulting from this study is named C3-SEC (Contex-aware Corporative Cyber Security) and intends to provide cyber security decision makers the ability to make informed decisions, selecting the best course of action to mitigate identified vulnerabilities/threats and ensure business continuation in the actual hostile cyber environment.

1.3 Organization

The rest of the document is organized as follows, chapter 2 presents and compares the most relevant information and communication infrastructure data collection tools to support the perception level of the approach proposed in this thesis. This chapter also introduces the cyber security standards adopted by these tools. Chapter 3 elaborates on knowledge representation technologies and standards, such as ontology design and engineering using the Ontology Web Language (OWL) standard. Additionally, a software tool developed in this thesis to support the comprehension layer of the followed approach is presented. In chapter 4, the most relevant risk strategies and techniques adopted by cyber security risk analysis tools are studied and compared. Chapter 5 proposes an innovative customizable cyber security risk strategy to make the best use of corporations/business knowledge and expertise on information assets/value. A case study carried out at the Universidad de las

Fuerzas Armadas ESPE – Ecuador is presented in chapter 6, showing the support provided by the approach and software tools developed in the context of this thesis along the full perception-action cycle. Finally, chapter 7 summarizes the thesis contributions and points future research directions.

A literature revision is presented along the chapters of the thesis, according to the research topics addressed in each chapter.

This page was intentionally left blank

Perception Level

Currently there are several tools that provide data collection features about an Information and Communication Technologies (ICT) infrastructure. Some of them include vulnerability scanning to analyze the technological infrastructure of an organization, based on metrics and standards already established by international cyber security entities (MITRE, NIST). These tools can generate reports of threats found in the infrastructure and help security managers to identify risks that may affect business continuity. The output of these tools that operate at the perception layer, will be used as input for the comprehension level, more specifically to instantiate the semantic model designed at the ontology level.
These tools are identified, described and compared in the following sections, after the most relevant cyber security standards adopted by them are introduced in section 2.1.

2.1 Cyber security Standards and Metrics

This section introduces the most relevant standards and metrics proposed by international standardizing organizations and adopted by cyber security tools, which are of utmost importance for cyber security information sharing and exchange. Cyber security information sharing and exchange (vulnerabilities identification, vulnerabilities severity classification, exploits, etc.) is seen in this thesis and by all international public/private authorities in the cyber security domain of crucial importance to fight cyber-attacks and protect legitimate public and private information systems and information assets.

2.1.1 Common Configuration Enumeration (CCE)

CCE [3] defines a list that provides unique identifiers to security-related system configuration to facilitate fast and accurate correlation of configuration statements presents in disparate domains. In addition, CCE is also one of six existing open standards used by the National Institute for Standards and Technology (NIST) [4] in its Security Content Automation Protocol (SCAP) [5] program, which combines “a suite of tools to help automate vulnerability management and evaluate compliance with United States of America federal

information technology security requirements". Numerous products have been validated by NIST as conforming to the CCE component of SCAP.

2.1.2 Common Platform Enumeration (CPE)

CPE [6] is a standardized method of describing and identifying classes of applications, operating systems, and hardware devices present among an enterprise's computing assets. The National Vulnerability Database (NVD) [7] defines a dictionary with every single CPE existent, and since they are listed in order to make the technological world more standardized, many IT systems use CPE to improve correlation of test results, and ease gathering of metrics. A CPE usually consists on the prefix "cpe:" and then other three parts that begin with a slash. The three parts identify hardware, OS and Application.

NMAP [8] is one application that uses the CPE standard which allows to cross information with the NVD CVE file. The current version of CPE is 2.3 which is defined through a set of specifications in a stack-based model, where capabilities are based on simpler, more narrowly defined elements that are specified in lower levels of the stack. This design opens opportunities for innovation, as novel capabilities can be defined by combining only the needed elements, and the impacts of changes can be better compartmentalized and managed.

2.1.3 Common Vulnerabilities and Exposures (CVE)

CVE [9] is the industry standard for sharing/publishing vulnerabilities and exposure names. It was created in 1999, when almost every security tool used their specific database and their specific names. This was a problem since it was not possible to determine when different databases were referring the same product, the same vulnerabilities or even if a value for the vulnerability severity would mean the same in another database. This could result in security gaps coverage and in an ineffective integration of all the databases. CVE appeared to solve this problem, proposing standardized identifiers and now every vulnerability is described with the same attributes and metrics. A CVE possess a CVE-ID, this identifier is built based on a syntax that is CVE + YEAR + ARBITRARY DIGITS, this way the CVE database can be listed based on years, and by order of appearance.

2.1.4 Common Vulnerability Scoring System (CVSS)

CVSS [10] [11] is responsible to categorize (in a numerical score) the risk (severity) that a vulnerability imposes to a specific product. The numerical score can even be translated into a qualitative representation from low to critical to help organizations cyber security risk assessment, prioritization and planning. Figure 1 shows the three groups of metrics defined by CVSS the base group, the temporal group and the environmental group:

- Base metrics represent the intrinsic vulnerability characteristics that are constant over time and in the user's environment.
- Temporal metrics represent the characteristics of a vulnerability that are most likely to change over time but not in different user environments.
- The Environmental metrics are the ones that reflect the characteristics of a vulnerability concerning a particular environment.

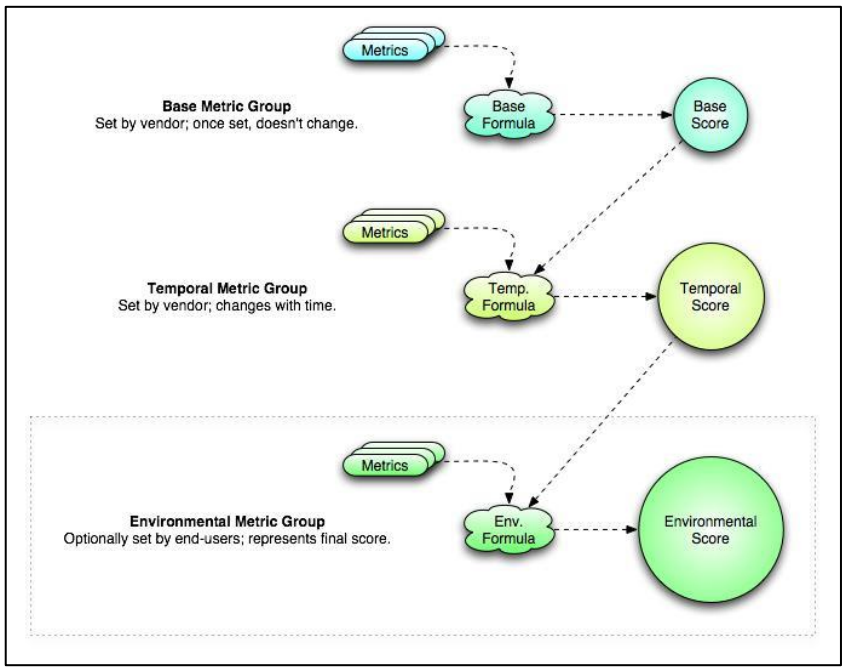

Figure 1 CVSS (Scoring View) [12]

CVSS scores can be used to rate security vulnerabilities (to get an indication of their relative severity) affecting a very wide range of software products: operating systems, web and legacy applications, security products (firewalls, antivirus software), databases, etc. [13]. Every application or service that uses the Common Vulnerability Scoring System (CVSS) should provide not only the CVSS score, but also a vector describing the components from

which the score was calculated. This allows decision makers to validate the score while providing a common set of vulnerability attributes to be disclosed [11].

CVSS vectors containing only base metrics take the following form (Table 1 presents the acronyms used in CVSS base metrics):

(AV: [L, A, N] /AC: [H, M, L] /Au: [M, S, N] /C: [N, P, C] / I: [N, P, C] /A: [N, P, C])

The following an example of vector definitions base:

(AV: L/AC: H /Au: N /C: N /I: P/A: C)

Metric	Description	Possible Values
AV	AccessVector (Related exploit range)	L= Local access, A = Adjacent network, N = Network
AC	AccessComplexity (Required attack complexity)	H= High, M = Medium, L = Low
AU	Authentication (Level of authentication needed to exploit)	M= Requires multiple instances, S= Requires single instance, N = None required
C	ConfImpact (Confidentiality impact)	N = None, P = Partial, C = Complete
I	IntegImpact (Integrity impact)	N = None, P = Partial, C = Complete
A	AvailImpact (Availability impact)	Possible Values: N = None, P = Partial, C = Complete

Table 1 Vector Definitions Base

2.1.5 Open Vulnerability and Assessment Language (OVAL)

OVAL [14] is an international and community effort to promote open and free cyber security content, free to the public. It includes a language to encode system details, and an assortment of content repositories held throughout the community. Software tools and services use OVAL for the three steps of system assessment: representing system information, expressing specific machine states, and reporting the results of an assessment. Use of OVAL [15] also provides for reliable and reproducible information assurance metrics and enables interoperability and automation among security tools and services. Through interoperability use of OVAL provides for automation, one example of which is the U.S. National Institute

of Standards and Technology's (NIST) Security Content Automation Protocol (SCAP) effort. OVAL is one of six existing standards SCAP uses to enable automated vulnerability management, measurement, and policy compliance evaluation. Besides, the OVAL Language and OVAL content are used in numerous information security products and services from around the world.

2.2 IT Infrastructure Data Collection Tools

This section provides a summary of relevant information about available tools for the collection of data related to the technological infrastructure of an organization, including vendor contact information. In addition, at the end of the section conclusions about the most appropriate tool for the development of the decision support system proposed in this thesis are presented.

A comparative study identifying the main features and properties of the most relevant tools for collecting information on the IT infrastructure is presented next, and the most suitable and convenient for the development of a decision support system in the cyber security domain is chosen. Tools properties are presented in tabular form in Table 3, properties definitions and tools assessment must be understood as follows. If a field in the table for a particular tool contains no information, that means either that the field was not relevant for that tool, or that the information could not be found on the supplier's Web site on one of the NIST or MITRE Web sites mentioned above.

Tool identifies the name of tool. *Purchase Type* identifies the way in which the tool can be purchased: Appliance or Software. If the tool is distributed on an appliance, unless explicitly noted, the appliance is presumed to include an operating system and hardware, so those fields in the tool's table will be left blank. *Free Version* identifies if the tool has a free version, the corresponding field of the table will be filled with the word "Yes" otherwise "No". *License* identifies the type of license under which the tool is distributed: Commercial, Shareware, Open Source, or Freeware. *CPE*, identifies if the tool supports this standard. The table will be filled with the word "Yes" or "No". *CCE*, identifies if the tool implements this type of specification. The table will be filled with the word "Yes" or "No". *Standards*, identifies relevant standards to which the tool is compliant with. This includes only standards directly relevant to vulnerability analysis, i.e., SCAP, OVAL, CVE, CWE, and CVSS. Standards for

configuration checking (e.g., XCCDF, FDCC) and other types of analyses are not included. For tools, not compliant with any such standards, this field is left blank. Entries in this field are based on supplier claims of standards compliance (some supplier claims are in the process of being validated by the responsible standards bodies and the tools do not yet appear on validated products lists). *OS Support*, identifies the operating system(s) (OS) on which a software tool runs. This field will also identify any other software that is required for the product to run (e.g., database, .NET framework, browser). *Supplier*, identifies the full name of the organization or individual that developed and distributes the tool. For suppliers that are non-U.S.-based, the country in which their headquarter are (or, for individuals, in which they reside) is noted in parentheses. *Decision Support*, identifies if the tool supports features of vulnerability detection and identification of remediation measures on a scale of prioritization. The table will be filled with the word "Yes" or "No". *Export Results*, identifies the format to which the results can be exported (information about the ICT infrastructure assets), for example XML, CSV, etc. *Information*, identifies the URL to the supplier's information about the tool. The set of tools studied are enumerated and described next:

2.2.1 Nessus Home

Nessus is one of the most popular and capable vulnerability scanners, particularly for UNIX systems. It was initially free and open source, but source code was closed in 2005 and removed the free "Registered Feed" version in 2008. A free "Nessus Home" version is also available, though it is limited and only licensed for home network use [16]. Nessus® Home allows to scan a personal home network (up to 16 IP addresses per scanner) with the same high-speed, in-depth assessments and agentless scanning convenience that Nessus subscribers enjoy [17].

Nessus is the most trusted vulnerability scanning platform for auditors and security analysts. Users can schedule scans across multiple scanners, use wizards to easily and quickly create policies, schedule scans and send results via email [18]. Reports generated by Nessus use standards as CPE, CVE and CVSS which can be exported in different formats as CVS, HTML, PDF, Nessus and NessusBD.

Nessus features color-coded indicators along with corresponding values, that allow to quickly assess scan's data, to help understand an organization's vulnerabilities. Each scan shows a vulnerabilities list, sorted by severity. It also includes compliance checks, this list

displays counts and details sorted by vulnerability severity. In addition, the scan's results include remediation information, this list displays all remediation details, sorted by the number of vulnerabilities.

2.2.2 Saint

SAINT (originally Security Administrator's Integrated Network Tool) [18] is a suite of integrated products that perform vulnerability scanning, assessment, and validation on network devices, operating systems, databases, desktop applications, Web applications, and other targets. The tool suite includes SAINTscanner, an agentless vulnerability assessment tool that can perform both authenticated and unauthenticated vulnerability scans that uncover areas of weakness on the target, and recommend remediation. SAINTscanner not only detects weaknesses but also identifies remediation that can be applied to them before those weaknesses can be exploited by intruders. It provides information on how to implement those remediation, including pinpointing the most exploitable vulnerabilities for which remediation should be applied first. SAINTscanner's database of vulnerability checks and exploits is automatically updated each day with new checks/exploits, enabling it to anticipate many common system vulnerabilities. It reports the presence of exploits, the detected vulnerabilities' CVSS score, the identification of the vendor whose product is found to harbor the vulnerability, and other useful information.

2.2.3 Nmap (ZenMap)

Nmap known as Network Mapper is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping) [8].

Through the execution of commands in Nmap, you can obtain an XML data file showing the main information of the scan such as host name, address, open ports, services and CPE (Common Platform Enumeration). This information is compared to CVE (Common Vulnerabilities and Exposures) allowing to check system vulnerabilities. However, Nmap by itself doesn't tell the existence of vulnerabilities on a system. Based on the scanning results and on user knowledge of computer networking and of the network baseline, the Nmap user may be able to figure out what vulnerabilities exist and address them to improve the overall security posture. Therefore, when compared to other tools, Nmap doesn't support corrective measures for the vulnerabilities found or prioritization of remediation measures.

2.2.4 eEye Retina

Retina Security Scanner [19] supports features to efficiently identify IT exposures and prioritize remediation measures in an enterprise-wide scale. The main features of this tool are:

- Continually monitor and improve enterprise security posture.
- Identify IT assets and sensitive data across disparate environments.
- Find security exposures in network, web, database and virtual assets.
- Prioritize remediation based on real risk to critical assets.
- Easily deploy and scale from small to large environments.
- Realize optimal performance via non-intrusive scanning.
- Get fast, frequent updates from the BeyondTrust Research Team.

Retina Network vulnerability [18] scanning is also offered in a free SaaS package, Retina Community, that allows free vulnerability assessments and SCAP [5] configuration compliance scans across the operating systems, applications, devices, and virtual environments at up to 32 target IP addresses, with reports generated in eXtensible Markup Language (XML) [20], comma-separated values (CSV) [21], and Portable Document Format (PDF). In addition, Retina contains in its reports suggestions for remediating the security weaknesses. Scan results can be sorted by machine (host), by vulnerability, or by CVE/IAV findings. Vulnerabilities can be sorted by name, risk, or severity code. It is also possible to specify the level of detail and display options such as page breaks and optional job metrics or detailed audit status [22].

2.2.5 GFI LANguard

GFI LANguard [18] is a network security scanner and patch management solution that assists in patch management, vulnerability management, network and software auditing, asset inventorying, change management, risk and compliance analysis. GFI LanGuard [23] supports machines across Microsoft®, MAC OS X® and Linux® operating systems as well as many third-party applications. It includes its own vulnerability assessment database that checks for 2,000+ CVEs and SANS Top 20 vulnerabilities. The database is regularly updated with information from Bugtraq, SANS, CVE, Microsoft security updates, and GFI Software's and other community-based information repositories.

Scan results can be exported in XML format. GFI also offers a freeware version, intended for personal use, and capable of scanning up to five IP addresses. The freeware version of GFI LANguard provides all functions found in the commercial version with the exception of patch management for non-Microsoft applications [18]. GFI LanGuard presents a functionality for Vulnerability Management through a graphic threat level indicator that provides a weighted assessment of the vulnerability status of a scanned computer or group of computers, and whenever possible, a Web link for more information on a particular security issue. Any detected vulnerabilities can be managed by selecting to remediate, ignore, acknowledge or re-categorize as appropriate.

2.2.6 nCircle IP360

nCircle IP360 [18] is a component of nCircle's security risk and compliance management suite. Using agentless technology, IP360 profiles all networked devices and tests for the presence of more than 40,000 conditions (OSs, applications, vulnerabilities, configurations). IP360 includes integrated Web application scanning to identify security risk in Web applications. IP360 provides, as an option, the nCircle Perimeter Profiler (a cloud-based virtualized appliance) to scan Internet facing assets for network, operating system, and Web application vulnerabilities, in the same way it scans assets on the internal network.

IP360 uses advanced analytics and a unique quantitative scoring algorithm based on several factors—including the vulnerability score and business-relevant asset value—to prioritize the vulnerabilities for remediation. The result is actionable data that enables IT security teams to focus on the tasks that will quickly and effectively reduce overall network risk with

the fewest possible resources [24]. Furthermore, IP360 has support for the following standards: SCAP, OVAL, CVE, CVSS.

2.2.7 Security System Analyzer 2.0 Beta

SSA (Security System Analyzer) [25] [18] is free non-intrusive OVAL, FDCC, XCCDF and SCAP scanner. It provides security testers and auditors with an advanced overview of the security policy level applied. It can identify vulnerabilities and security discrepancies through its OVAL interpreter and large database of OVAL vulnerability definitions. Findings can be output in CSV. The main features of this tool are:

- Fully support of open security standards and initiatives (CVE, OVAL, CCE, CPE, CWE, SCAP, CVSS).
- Perform Compliance and Security Checks using the XCCDF - The eXtensible Configuration Checklist Description Format.
- Qualifying the vulnerabilities using CVSS v2.0 scoring.

2.2.8 OpenVas

The Open Vulnerability Assessment System (OpenVAS) [26] is a framework of several services and tools. The core of this SSL-secured service-oriented architecture is the OpenVAS Scanner. The scanner very efficiently executes the actual Network Vulnerability Tests (NVTs) which are served via the OpenVAS NVT Feed or via a commercial feed service. All clients run on Windows, Linux, and other OSs. The third-party tools integrated into the OpenVAS framework are Nikto, Nmap, ike-scan, snmpwalk, amap, ldapsearch, Security Local Auditing Daemon, Ovaldi OVAL interpreter, pnscan, portbunny, strobe, and w3af [18].

As for support for making decisions OpenVAS allows assessment of vulnerabilities, access control and intrusion, and assessment risk using the CVSS scoring system. It allows us to analyze a PC or a local / remote server and perform various types of reports on detected vulnerabilities. In addition, adds a correlation engine to interlace everything that has been identified / detected and propose associated solutions. The standard adopted for OpenVas is OVAL.

2.2.9 Nexpose

Rapid7 Nexpose [27] is a vulnerability scanner that enables to focus on risk that matters while greatly reducing the time required to run a successful vulnerability management program. NeXpose is offered in four versions [18]:

1. NeXpose Enterprise®, intended for organizations with large, complex networks of more than 1,024 IP addresses; NeXpose Enterprise is intended to be installed on dedicated servers that host no other security software (e.g., no IPS, IDS, virus scanner, etc.).
2. NeXpose Consultant is intended for use by independent security consultants and auditors, and designed to run on a laptop; it also provides configuration features that tune the tool for one-time integrated scans/tests.
3. NeXpose Express is intended for small-to-medium sized businesses (Class C networks with 256 IP addresses or fewer), and also intended to be deployed on a laptop.
4. NeXpose Community is a free, single-user edition intended for single user or home business use on networks of 32 or fewer IP addresses; the Community version lacks custom scan and report configuration, email alert, Web application scanning, compliance/configuration scanning, and provides only limited reporting (XML format only).

Table 2 shows in more detail the functionalities presented by the Enterprise Nexpose version, which is the most complete versus the Community version.

Functionalities	Enterprise	Community
General		
Max Number of IPS	Unlimited	Up to 30
Number of users	Unlimited	One
Number of scan engines included	Unlimited	One
Licensing model	Perpetual	Free
Collect		
Run one scan for multiple compliance reports	Yes	Yes
Automatic vulnerability updates and Microsoft Patch Tuesday vulnerability updates	Yes	Yes
Scan scheduling and alerting	Yes	Yes
Web application scanning	Yes	
PCI compliance	Yes	
Advanced report and scan customization	Yes	
Open API™ and third-party Integrations	Yes	
Policy manager	Yes	
Virtual scanning (Vmware NSX)	Yes	
Dynamic discovery scanning (Vmware, Mobile)	Yes	
Distributed scanning	Yes	
Adaptive Security with automated actions	Yes	
Dynamic, live dashboards with 50+ cards	Yes	
Scan IP addresses belonging to third parties		
Prioritize		
Exception management	Yes	Yes
Interactive charting	Yes	Yes
Dynamic Asset Groups and Tagging	Yes	Yes
Custom Tags and System Criticality Tags	Yes	
Report Templates and Uploading	Yes	
Integrated vulnerability validation with Metasploit	Yes	

continue →

Customizable threat models	Yes	
Remediate		
Executive and remediation reporting	Yes	Yes
User Role Customization	Yes	
Remediation Workflow	Yes	
Deployment Options		
Software Installation	Yes	Yes
Virtual Appliance	Yes	Yes
Private Cloud	Yes	
Physical Appliance	Yes	
Managed Service	Yes	
Support		
Online Support	Yes	Community
Assigned Account Manager	Yes	
Phone Support	Yes	
2-hour response for severity 1 issues	Yes	

Table 2 Functionalities Enterprise and Community Nexpose Version [28]

The tool also provides detailed remediation guidance that includes time estimates, exploit risk score, and asset criticality. Nexpose prioritizes mitigation tasks to reduce overall risk as quickly as possible. For example, within Nexpose you can use the Prioritized Remediation report to determine which patches have the highest impact in reducing risk to your environment. Nexpose categorizes vulnerabilities with a CVSS score. The standards adopted for Nexpose are CPE, CCE, SCAP, CVE and CVSS. Nexpose also has an option of Reporting Data Model which is a dimensional model that allows customized reporting. The implementation of the Reporting Data Model is accomplished using the PostgreSQL relational database management system, version 9.0.13. As a result, the syntax, functions, and other features of PostgreSQL can be utilized when designing reports against the Reporting Data Model. The Reporting Data Model is available as an embedded relational schema that can be queried against using a custom report template. With Nexpose it is possible to apply tags to indicate the locations of the assets. It is possible then to create reports based in these tags and assess the risk of the assets by location. This option is known

as Applying Real Context with tags that allows the tracking of assets in an organization, to identify, group, and report on them according to how they impact the business [29].

Figure 2 User Added Tags Nexpose [29]

As we can see in Figure 2, a Nexpose user can easily gain context into a specific asset. He knows that this asset falls under PCI Compliance, lives in the DMZ somewhere, and this asset is really critical to his business. In addition, the asset is owned by "John Smith" and is located somewhere in Austin. This allows to gain real insight into how to tackle risks that are found on this asset now, and in the future. This also helps simplify the overall workflow. If a new risk is discovered on this asset in the future, it is known how to tackle the problem [29].

2.2.10 QualysGuard

The Qualys Cloud Platform [30], also known as QualysGuard, consists of an integrated suite of solutions to help organizations simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications. QualysGuard includes Vulnerability Management (VM), a cloud service that gives you immediate, global visibility into where your IT systems might be vulnerable to the latest Internet threats and how to protect them. It helps you to continuously secure your IT infrastructure and comply with internal policies and external regulations [31]. Furthermore, Qualys separates reporting from scanning, enabling to use a wide range of filters to explore the vulnerability findings. It is possible to look for specific types of vulnerabilities and use criteria from Qualys's KnowledgeBase such as severity, business risk, CVSS scores, existence of exploits or malware, and whether patches are available [32].

Tool	Purchase Type	Free Version	License	CPE	CCE	Standards	SO Support	Supplier	Decision Support	Export Results	Information
Nessus Home	Software	Yes	Freeware/ Commercial	Yes		CVE, CVSS, SCAP only for paid version	Windows, Mac OS X, Free BSD, Linux	Tenable Network Security®	Yes	HTML CSV Nessus DB (.db)	https://www.tenable.com/products/nessus-home
Saint8	Software, Appliance, or SaaS	No	Commercial	Yes	Yes	CVE, CVSS, OVAL, SCAP	Linux, Mac OS X	SAINT Corp.	Yes	CSV json-formatted	https://www.saintcorporation.com/products/SAINT8.html
Nmap (ZenMap)	Software	Yes	Open Source	Yes			Windows, Linux, Mac OS X	Nmap	No	XML	https://nmap.org/
eEye Retina	Software	Yes (Trial)	Commercial	Yes		CVE, CVSS, OVAL, SCAP	Windows	eEye Digital Security®	Yes	XML CSV	https://www.beyondtrust.com/products/retina-network-security-scanner/
GFI LANguard	Software	Yes (Trial)	Commercial			CVE, OVAL	Windows, Linux, Mac OS X	GFI Software	Yes	XML	http://www.gfi.com/lannetscan
nCircle® IP360	Appliance	No	Commercial	Yes		CVE, CVSS, OVAL, SCAP		nCircle Network Security, Inc.	Yes	CSV XML	https://www.tripwire.com/it-security-software/enterprise-vulnerability-management/tripwire-ip360/
Security System Analyzer 2.0 Beta	Software	Yes	Open Source	Yes		CVE, CVSS, XCCDF, OVAL, SCAP	Windows	NETpeas, Societe Anonyme (SA) (Morocco)	Yes	CSV	https://code.google.com/archive/p/ssa/

continue

OpenVas	Software, Appliance, or SaaS	Yes	OpenSource			OVAL	Windows, Linux	Atomic Corporation's OpenVAS Project (Germany)	Yes	XML, HTML, PDF	http://www.openvas.org/software.html
Nexpose	Software	Yes	Commercial	Yes	Yes	CVE, CVSS, SCAP	Windows, Linux, VMWare Virtual Appliance	Rapid7	Yes	XML, HTML, PDF.	https://www.rapid7.com/es/products/nexpose
QualysGuard	SaaS	Yes (Trial)	Commercial			CVE, CVSS, SCAP		Qualys, Inc	Yes	HTML, MHT, PDF, CSV, and XML	https://www.qualys.com/suite/vulnerability-management/features/

Table 3 IT Infrastructure Data Collection Tools.

2.3 Tools Comparison

This section presents a comparative study identifying the main features of the most relevant tools for collecting information on the ICT infrastructure, in order to choose the most suitable and convenient for the development of a decision support system in the cyber security domain. The criteria established for the characterization and evaluation of the tools are: If the tool has a free version (Free Version); Type of license under which the tool is distributed (License), Commercial, Shareware, Open Source, or Freeware; If the Common Platform Enumeration standard is supported (CPE); If the Common Configuration Enumeration standard is supported (CCE); Relevant standards to which the tool is compliant with (Standards), which includes only standards directly relevant to vulnerability analysis, i.e., SCAP, OVAL, CVE, CWE, and CVSS; The operating system(s) (OS) on which a software tool runs (OS Support); If the tool supports functionalities for vulnerability detection, identification and prioritization of remediation measures (Decision Support); Format to which the results can be exported (Export Results), e.g. information about the assets, exported to XML, CSV, etc.

For the selection of the most suitable tool to be adopted in the following stages of this thesis, each criteria was assigned a weight, according to its relevance in the context of the thesis. The criteria and their weights are presented next. Free Version, the value of “3” is assigned to the tool that has a free version available without a time limit, “2” for one that has a free version but has a limit number of days (usually 30 days) and “1” for one that does not have a free version. License, this metric is defined according to the type of license, in the case of being open source the assigned value is “3”, if it is freeware “2” and in case of being commercial the assigned value is “1”. CPE, CCE, Decision Support, in the case of the CCE, CPE and decision support criteria, value “2” or value “1” is assigned to indicate the corresponding tool compliance or not compliance, respectively. Standards, OS Support and Export Results, these criteria are quantified in Table 4, according to the number of standards used by the tool, the operating systems it supports or the number of formats available to export the results. Three is the highest value, for example in case a tool uses more than 3 security standards.

Value	Weighting
One	1
More than 2	2
More than 3	3

Table 4 Weighing Standards, SO Support, Export Results

Table 5 shows the overall results according to the established metrics.

Tool	Free Version	License	CPE	CCE	Standards	SO Support	Decision Support	Export Results	Total
Nexpose	3	1	2	2	3	3	2	3	19
Nessus Home	3	2	2	1	2	3	2	3	18
Security System Analyzer 2.0 Beta	3	3	2	1	3	1	2	1	16
OpenVas	3	3	1	1	1	2	2	3	16
Saint8	1	1	2	2	3	2	2	2	15
Nmap (ZenMap)	3	3	2	1	1	3	1	1	15
eEye Retina	2	1	2	1	3	1	2	2	14
QualysGuard	2	1	1	1	3	1	2	3	14
GFI LANguard	2	1	1	1	2	3	2	1	13
nCircle® IP360	1	1	2	1	3	1	2	2	13

Table 5 Results Comparison Tools

As described previously, Nexpose is ranked first with a total of 19 points, followed by Nessus Home with 18 points. In this way, it can be concluded that Nexpose is the most promising tool in the context of the study carried on in this thesis, because it fulfills most criteria in comparison with the other tools. Among several properties, we can emphasize that this tool supports operating systems such as Windows and Linux. Furthermore, the representation of the results (vulnerability reports) is based on standards such as CPE, CVE and CVSS. This information can be exported in various formats such as XML and HTML, allowing developers to obtain these data for manipulation and integration with other applications. Another important reason for choosing this tool is that it has several features for decision support, one of which is to get a full picture of risk across ICT assets, encompassing vulnerabilities and configuration issues, presented in easy-to-use customizable reports. This enables better decision-making and increases the credibility of the security team across the organization.

Comprehension Level

To implement the comprehension layer of the context aware system proposed in this thesis, an (OWL [33]) ontology based knowledge representation was adopted. The role of the ontology here is not only to represent/incorporate/integrate the data captured about the ICT infrastructure at the perception layer by the tool described in the previous section, but also to allow domain/corporations specific knowledge to be added by cyber security experts (e.g. Chief Information Security Officers - CISO). Experts are allowed to introduce new specific knowledge into the ontology using Protége [34] ontology editor. Assets characterization such as asset value and importance of each security dimension associated to that asset (privacy, integrity, availability) must be provided by experts and added to the ontology. This knowledge is essential to support corporation specific cyber risk analysis and management to be performed by the decision aiding software to be developed in our study.

The following sections in this chapter introduce the concepts of ontology, the Web Ontology Language (OWL) and related works about ontologies in the cyber security domain. One of the most used tools for ontology design and edition (Protégé [34]) is presented and the proposed ontology design to be used in the following stages of the thesis is described and explained. In order to integrate the ICT infrastructure data characterization generated by Nexpose (XML reports), with corporations cyber security experts specific knowledge (represented at the ontology level in OWL), existing XML to OWL conversion tools were studied and a software component for XML to OWL generation was developed and presented in this chapter.

3.1 Ontology Concepts

According to one of the most widely accepted definitions of ontology in computer science "An ontology is a formal explicit specification of a shared conceptualization for a domain of interest." [35]. It is formal and logic-based, which makes reasoning possible; it has explicit specification, which makes it easy for new learners of this domain; it is a shared conceptualization, which defines a common vocabulary for researchers who need to share

information in this domain. Web Ontology Language (OWL) [33] was approved by World Wide Web Consortium (W3C) to be one of the key Semantic Web technologies in 2004 [36].

An ontology is not a database, is not a program, and more than a conceptualization it represents a view of a knowledge domain. Ontologies allow to cover various targets to enable the exchange of data between programs. In addition, ontologies simplify the translation of different representations. An ontology is a method applied to a selected domain to formally represent the concepts and relationships in it. To develop an ontology is necessary to define classes, to establish the hierarchy of classes in taxonomies (subclass, superclass), to set relations (properties) and describing values and objects that are related.

One general proposal to the process of building ontologies is given by Noy [37] in the comment]: 1) Determine the scope and domain of the ontology; 2) Consider reusing existing ontologies; 3) Enumerate important terms in the domain; 4) Define the class hierarchy; 5) Define object properties; 6) Define data properties; 7) Create individuals; 8) Publish.
Noy process of building ontologies was generally followed to build the ontology used in this thesis to represent cyber security specific knowledge domain.

3.2 Web Ontology Language

One of the languages with great expressive power that has become standard for annotating Web ontologies is the Web Ontology Language (OWL). The Web Ontology Language (OWL) is an international standard for encoding and exchanging ontologies and is designed to support the Semantic Web [33]. In other words, OWL is a standard for the Semantic Web that lets to manage, integrate, share and reuse data on the Web. OWL is grounded on the Resource Description Framework (RDF) standard.

RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a “triple”). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications [38].

OWL provides more vocabulary for describing properties and classes than RDF: disjoint classes, relationships between classes, cardinality, equality, properties, characteristics of properties, and enumerated classes. OWL has 3 sub-languages, OWL Lite, OWL Description Logic (DL) and OWL Full. For the sake of simplicity, it can be said that OWL Lite is an extremely simple least expressive OWL dialect, OWL DL provides a grateful expressiveness and allows fairly advanced description logics reasoning capabilities, and OWL Full which is the most expressive OWL dialect, it allows free syntactical as RDF, but is highly unlikely that any reasoning software is able to support complete reasoning for every feature of OWL Full. In this thesis, like in most semantic web based applications, OWL DL was adopted to design and build the ontology used for the semantic level knowledge representation. The ontology is presented and explained in the following sections of the thesis.

3.3 Cyber Security Ontology Related Studies

Due to the advantages of representing knowledge in the form of ontologies, several studies in the cyber security domain made use of (OWL) ontologies. A literature revision was performed on cyber security ontologies in the context of the current thesis. The knowledge made available by the means of cyber security ontologies scientific publications, was taken into consideration in the process of building the ontology proposed in this thesis. This allowed to reuse cyber security domain specific vocabulary, concepts, relations and (description logics) rules published in this scientific area. The scientific publications that most influenced the design of the ontology proposed in this thesis are briefly presented and explained next.

“Ontologies for Modeling Enterprise Level Security Metrics” [39], the main goal of this paper is the development an ontology that has knowledge about which threats endanger which assets and which counter measures can reduce the probability of a damage. This ontology can enable a quantitative risk analysis so that the manager of an enterprise can choose the appropriate safeguard mechanism to reduce the threats to their enterprise. This work presents a model for Enterprise Level Security, discusses application of the ontology for collecting and querying data on security metrics.

“Ontology-Based Evaluation of ISO 27001” [40], in this paper a metamodel of the ISO 27001 security standard explaining its core concepts is presented. A comparison is also made

about the constructed metamodel with various information security ontologies. The paper discusses their application and present the basic ideas of applying qualitative data analysis (QDA), after a brief overview of related works.

"Formalizing Information Security Knowledge" [41], this paper describes a security ontology which provides an ontological structure for information security domain knowledge. Besides existing best-practice guidelines such as the "German IT Grundschutz Manual". An evaluation conducted by an information security expert team has shown that this knowledge model can be used to support a broad range of information security risk management approaches.

"An Ontology Based Approach to Information Security" [42], this paper presents a conceptual implementation model of an ontology defined in the security domain. The model presented contains the semantic concepts based on the information security standard ISO/IEC_JTC1, and their relationships to other concepts, defined in a subset of the information security domain.

"A Security Ontology for Security Requirements Elicitation" [43], this paper presents a core and generic security ontology for security requirements engineering. Its core and generic status is attained thanks to its coverage of wide and high-level security concepts and relationships. This work implemented the ontology and developed an interactive environment to facilitate the use of the ontology during the security requirements engineering process. The proposed security ontology was evaluated by checking its validity and completeness compared to other ontologies.

"Ontologies for Security Requirements: A Literature Survey and Classification" [44], this paper is a survey, it proposes an analysis and a typology of existing security ontologies and their use for requirements definition. This work is part of a larger project aiming to improve security requirement definition using ontologies. The main objective in this paper was to review, analyze, select and classify security ontologies, as a scope study but with a particular interest in the field of security requirements engineering.

"Towards a new generation of security requirements definition methodology using ontologies" [45]**,** this research proposes to include ontologies into the requirements engineering process. The main goal of this work was to take advantage of the existing security and domain ontologies, and propose mechanisms and techniques to use them in an approach that guides the definition and analysis of security requirements for a particular domain of activity.

As suggested by Noy [37] in one of the steps leading the process of building ontologies ("Consider reusing existing ontologies"), the ontology proposed in this thesis adopted as much as possible the cyber security knowledge represented in the above-mentioned ontologies, with adaptations specifically designed to serve the purpose of the current thesis.

3.4 Ontology Editor: Protégé

Protégé [34] is a free, open-source ontology editor and framework for building intelligent systems. It is one of the most widely used ontology editor, is supported by a strong community of academic, government, and corporate users, who use Protégé to build knowledge-based solutions in areas as diverse as biomedicine, e-commerce, and organizational modeling. Moreover, Protégé's plug-in architecture can be adapted to build both simple and complex ontology-based applications. Developers can integrate the output of Protégé with rule systems or other problem solvers to construct a wide range of intelligent systems. Protégé fully supports the latest OWL 2 Web Ontology Language and RDF specifications from the World Wide Web Consortium. Protégé was adopted in this thesis for ontology design and edition.

3.5 Study of Tools for Generation of OWL ontology from XML Data Source

Although the data generated by Nexpose is represented using the eXtensible Markup Language (XML) which can be used as data exchange format in different domains, XML covers only the syntactic level and lacks support for semantic representation and reasoning. Ontologies can provide a semantic representation of domain knowledge which supports efficient reasoning and expressive power. One of the most popular ontology languages is the Web Ontology Language (OWL). It can represent domain knowledge using classes, properties, axioms and instances for the use in a distributed environment such as the World

Wide Web [46]. There are different methods and tools that enable the generation of an ontology from an XML resource. The main tools found in the literature for this purpose are described next:

- **JXML2OWL**

JXML2OWL [47] is a framework divided in two sub projects: JXML2OWL API and JXML2OWL Mapper. The API is a generic and reusable open source library for mapping XML schemas to OWL ontologies for the Java platform while the Mapper is an application with a graphical user interface (GUI) developed in Java Swing that uses the API and eases the mapping process. JXML2OWL supports manual mappings from XML, XSD or DTD documents to an OWL ontology, thus supporting all the kinds of mappings such as many-to-many. Currently, conditional mappings through XPath predicates are not implemented within the framework. According to the mapping performed, JXML2OWL generates mapping rules wrapped in an XSL document that allows the automatic transformation of any XML data, that is, any XML document validating against the mapped schema, into instances of the mapped ontology. Figure 3 represents such process.

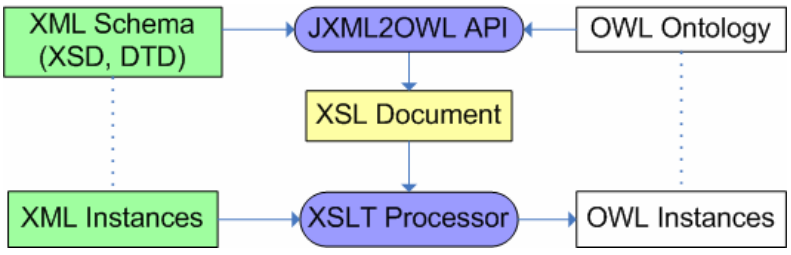

Figure 3 JXML2OWL Supports Mappings and Instances Transformation [47]

With JXML2OWL, the mapping process requires several steps. The first step consists in creating a new mapping project and loading both the XML Schema related file (XSD or DTD) and the OWL ontology. If an XML schema in not available, it is possible to load an XML document. In this case, JXML2OWL extracts a possible schema. In the second step, the user creates class mapping between elements of the loaded XML schema and classes of the ontology. Once these mappings are created, it is possible to relate them to each other to create object property mappings, or to relate them with elements of the XML schema to create datatype property mappings. Finally, in the last step, it is possible to export the transformation rules, generated according to the mapping performed, as an XSL document. With this XSL document, it is possible to transform any XML document which validates

against the mapped XML schema into individuals of the mapped OWL ontology. Obviously, both the API and the Mapper support all these steps.

- **X2OWL**

X2OWL [48] [49] is a tool implemented within OWSCIS framework to handle the wrapping of single XML data sources to local ontologies. This tool is deployed inside a data provider to tackle two tasks: 1) Create a local ontology from a single XML data source, and 2) Translate SPARQL queries over the local ontology into XQuery queries over the local XML data source. This method is based on XML schema to automatically generate the ontology structure, as well as, a set of mapping bridges. The method also includes a refinement step that allows to clean the mapping bridges and possibly to restructure the generated ontology.

This process is based on some mapping rules that indicate how to convert each component of the XML schema to a semantically corresponding ontology component. During ontology generation process, X2OWL also generates a mapping document that describes the correspondences between the XML data source and the generated local ontology. The mapping document is expressed using the proposed mapping specifications: XOML.

The created ontology is described in OWL-DL language. It plays the role of the local ontology within the data provider. The generated ontology only describes the concepts and properties but not the instances. Data instances are retrieved and translated as needed in response to user queries. During ontology generation process (see Figure 3), X2OWL also generates a mapping document that describes the correspondences between the components of the XML data source and those of the generated local ontology.

- **Automatic Generation of OWL Ontology from XML Data Source** [46]

This method uses the same notations used in [48] with some modifications to apply on multiple XML data sources. The approach is based on XML schema to build the ontology. If the schema does not exist, it can be automatically generated from the source XML document, this method copes with all possible complex cases arising from different XML schema design styles. The generation of OWL ontology from XML data sources could be described in 4 steps (see Figure 4):

1. The XML document is transformed to XML-Schema using the Trang API for java. The Trang takes as input a schema written in XML syntax and produces as output a schema written in XML-Schema.
2. The XML-Schema is analyzed using XML-Schema Object Model (XSOM). XSOM is a Java library that allows applications to easily parse XML Schema documents and inspect information in them. It is expected to be useful for applications that take XML Schema as an input.
3. The output of XSOM is used as input for the Java Universal Network/Graph framework (JUNG) [16]. The JUNG is used for graph-based manipulations. It generates XML- Schema Graph (XSG) that describes the schema in the same way whatever its design style is. An XSG is composed of a vertex set, and an edge set. The vertex set contains all elements, attributes, nonprimitive types, element groups and attribute groups. The edge set contains the edges established:
    - From each element to its type (if not primitive).
    - From each type, element group or attribute group to their contained elements and/or attributes.
4. The Jena API [50] uses XSG as input to generate OWL entities. Basically, OWL Classes emerge from complex types, element group declarations, and attribute-group declarations according to the mapping rules. Object properties emerge from element-sub element relationships. Datatype properties emerge from attributes and from simple types.

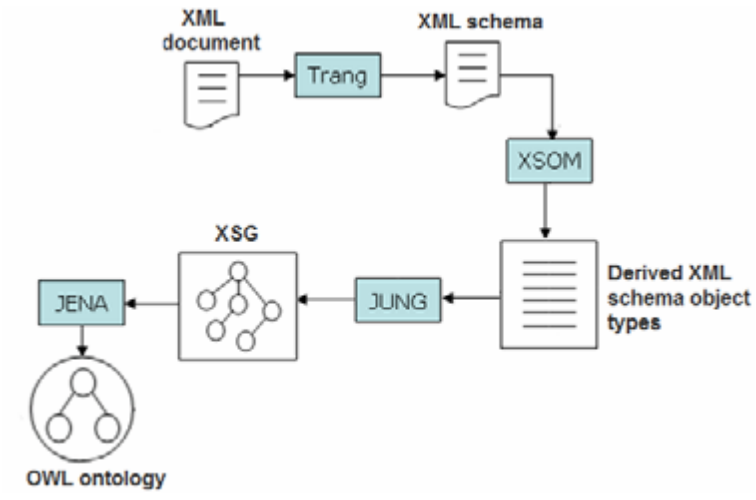

Figure 4 The Generation Process of OWL Ontology from each XML Data Source [46]

The method is based on three types of mappings:

1. OWL classes: two kinds of complex types are distinguished: 1) global, named complex types, and 2) local anonymous complex types.
2. Object properties: Elements (global or local) are not mapped directly to the ontology, but the element-sub element relationship in the schema is translated as the proposed object property in the ontology.
3. Datatype properties: Elements of simple types are mapped to the proposed datatype properties. When a complex type (global or local) contains an element of a simple type (primitive or defined) having as domain the class corresponding to the complex type. If the simple type is a primitive XML Schema Definition (XSD) datatype (xsd:string, xsd:integer, ….) then the range of the proposed datatype property is this datatype.

- **Topbraid composer**

TopBraid Composer is a visual modeling environment from industry experts for creating and managing domain models and ontologies in the Semantic Web standards RDF, RDFS and OWL [51]. TopBraid Composer is based on the Eclipse platform and the Jena API. Composer seamlessly integrates logical and rule-based reasoning engines. It offers a convenient drag-and-drop, form-based user interface with the ability to view and edit ontologies in a variety of serialization formats. Testing, consistency checking and debugging is supported by built-in OWL Inference engine, SPARQL query engine and Rules engine. TopBraid Composer makes it easier for an enterprise to move to Semantic Web standards by importing legacy models including XML Schemas, UML, RDB Schemas and spreadsheets. Open APIs are available and it can run with the dase back-end for improve scalability.

This tool can automatically generate an OWL/RDF ontology from any XML file. Each distinct XML element name is mapped into a class, and the elements themselves become instances of those classes. A datatype property is generated for each attribute. The nesting of the XML elements is stored by means of the composite:child property described in a recent blog entry. TopBraid can be used to import arbitrary XML documents into OWL so that they can be queried and processed with semantic web tools. The mapping is bi-directional and lossless so that files can be loaded, manipulated and saved without losing

structural information. The conversion occurs automatically, users do not have to worry about writing any rules for commonly needed mappings. However, those users that need to make further transformations can use SPARQL Rules and SPARQLMotion to customize their generated OWL ontology or further transform RDF triples representing the XML data [52].

- **XML to OWL Tools Comparison Summary**

As described above some of the tools require more steps than others for generating an ontology from a XML resource. With JXML2OWL the mapping process has a number of steps, including loading some files as XML Schema related file (XSD or DTD), XML instances and the OWL ontology. This way it is possible to relate each of the elements in the XML file with the ontology, but this work can be a little confusing and tedious. X2OWL works from an XML Schema for the generation of the ontology, being necessary to convert the XML file to an XSD format. In addition, the generated ontology only describes the concepts and properties but not the instances. Data instances are retrieved and translated as needed in response to user queries.

TopBraid however is a tool that allows the automatic generation from the XML file for ontology, its graphical interface allows easy handling of each of the generated elements such as classes, objects and datatype properties. This tool also has other features such as logical and rule-based reasoning engines, and it offers a convenient drag-and-drop, form-based user interface with the ability to view and edit ontologies in a variety of serialization formats. One of the disadvantages of TopBraid is that it is a commercial tool, for this reason a method was developed specifically for this thesis, for the transformation of the XML ouptut provided by the Nexpose into an OWL ontology. The method proposed in this thesis is presented in the next section.

3.6 Building Semantic Level Cyber Security Context Awareness

A method to build/instantiate the initial (OWL) ontology automatically is proposed in this section. The generation method is based on the XML-Schema of Nexpose [53] for the construction of the (OWL) ontology. As shown in Figure 5, the generation of OWL ontology from XML standards and data sources could be described in 3 steps:

1. From the XML-Schema the design of the base ontology (Appendix 1) is performed, using Protégé, a tool that provides a graphical interface for the construction of ontologies in OWL language.
2. The Nexpose results XML file is analyzed using the Document Object Model (DOM), an application programming interface for Java. In this way Nexpose dynamically generated data is obtained and added into the ontology.
3. Finally, each of the individuals obtained from parsing the XML file is added in the base ontology, with the help of the OWL-API for java.

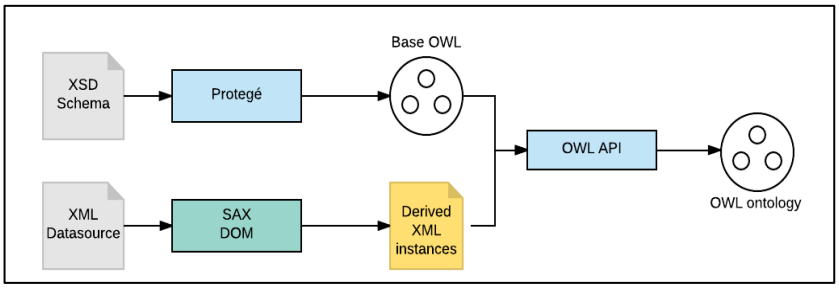

Figure 5 Generation Process of OWL Ontology

- **XML to OWL Mapping**

This section defines a notation to specify mappings between elements of Nexpose XML Schema and resources of an OWL ontology, which is mainly defined by classes, datatype and object properties [47] [54]. Three types of mappings are presented as follows:

1. Class mapping: Maps an XML node to an OWL concept.
2. Datatype property mapping: Maps an XML node to an OWL datatype property.
3. Object property mapping: Relates two class mappings to an OWL object property.

In Table 6 it is possible to observe the notation of the mapping of the vulnerability node in relation to the data of the XML schema.

Mappings	**Schema Node XML**
Class	Vulnerability
Datatype property	id, title, severity, pciSeverity, cvssScore,cvssVector,published,added,modified,riskScore.
Object property	hasVulnerability (between Device and Vulnerability class)

Table 6 Vulnerability Mapping

The generated OWL ontology is shown in Figure 6. In this ontology, there are twelve locals complex types defined within the Location, Device, Software, OperatingSystem, Vulnerability, SecurityPillars, Risk, Exploit, Tag, Reference, Description, Solution and Malware.

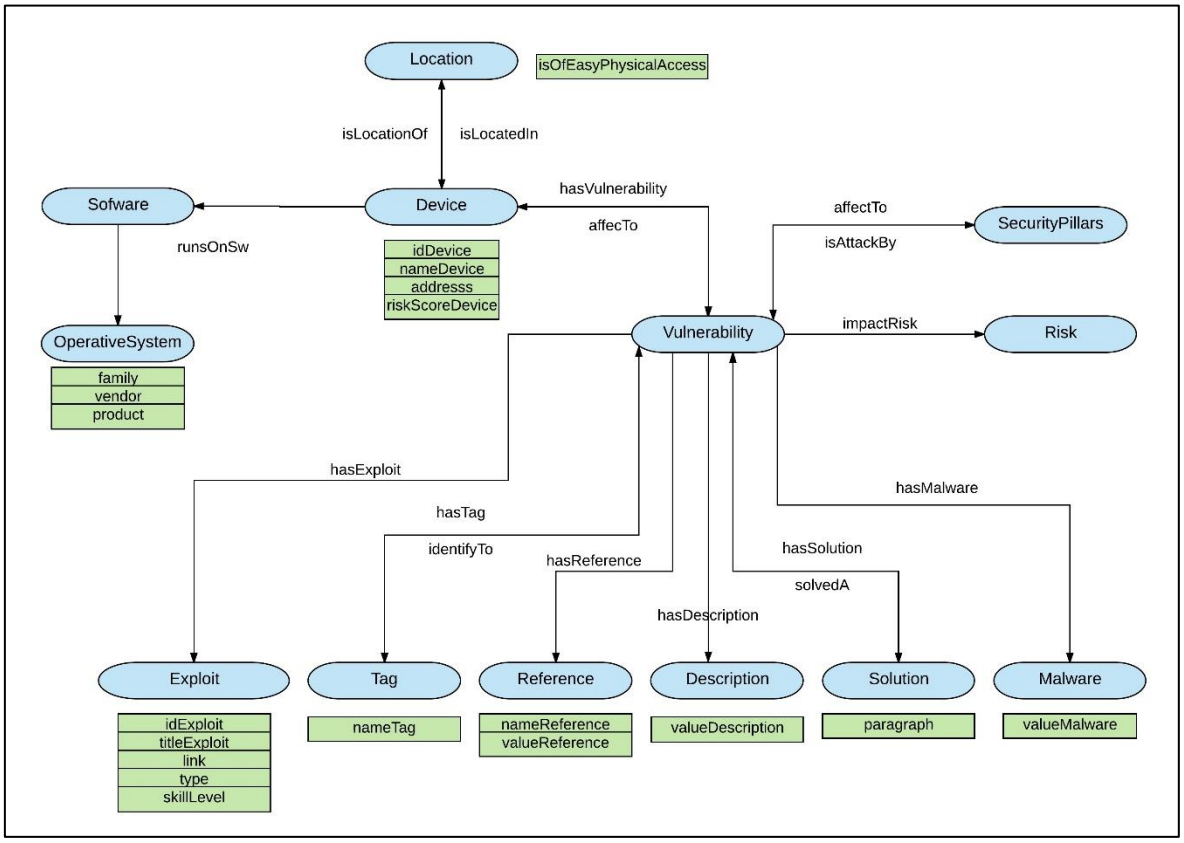

Figure 6 OWL Ontology Structure

Tables 7 and 8 show the object properties and datatype properties of the generated ontology, respectively.

Datatype Property	**Domain**	**Range**
id	Vulnerability	string
title	Vulnerability	string
cvssScore	Vulnerability	float
cvssVector	Vulnerability	string
pciSeverity	Vulnerability	integer
severity	Vulnerability	integer
riskScore	Vulnerability	float

continue

published	Vulnerability	string
added	Vulnerability	string
modified	Vulnerability	string
idExploit	Exploit	string
title	Exploit	string
type	Exploit	string
link	Exploit	string
skillLevel	Exploit	string
nameReference	Reference	string
valueReference	Reference	string
nameTag	Tag	string
valueTag	Tag	string
valueDescription	Description	string
paragraph	Solution	string
valueMalware	Malware	string
isOfEasyPhysicalAccess	Location	string
idDevice	Device	string
riskScoreDevice	Device	string
address	Device	string
nameDevice	Device	string

Table 7 The Datatype Properties

Object Property	**Domain**	**Range**
hasDescription	Vulnerability	Description
hasExploit	Vulnerability	Exploit
hasReference	Vulnerability	Reference
hasSolution	Vulnerability	Solution
hasTag	Vulnerability	Tag
hasMalware	Vulnerability	Malware
affecTo	Vulnerability	SecurityPillars
impactRisk	Vulnerability	Risk
hasVulnerability	Device	VulnerabilityDefinitions
isLocatedIn	Device	Location
runsOnSw	Device	Software
solvedA	Solution	Vulnerability
identifyTo	Tag	Vulnerability
isAttactBy	SecurityPillars	Vulnerability
isLocationOf	Location	Device

Table 8 The Object Properties

This page was intentionally left blank

Projection Level - Risk Analysis

This chapter analyzes different tools for ICT infrastructure data collection, vulnerability scanning and the support they can provide for cyber security risk assessment and decision making in organizations. The criteria used to evaluate, compare and select the most suitable tools for this study include cyber security metrics, standards and risk strategies. In addition, they are classified and contextualized with respect to the situation awareness layer they belong to (perception, comprehension, projection and decision/action). The following sections in this chapter introduce a detailed literature review about the tools and a comparative analysis of these tools with respect to risk assessment.

4.1 ICT Infrastructure and Cyber Security Data Collection Tools

Following a detailed literature review on most relevant ICT infrastructure and cyber security data collection tools, and having proceeded with an initial shortlisting process, it was concluded that a set of nine tools of interest are worth to be addressed in this thesis: Nessus, Saint8, Retina Security Scanner, GFI LANGuard, nCircle® IP360, Security System Analyzer 2.0, OpenVas, QualysGuard, Nexpose. These tools were analyzed according to the following criteria, which are assumed as the most relevant for the tools comparison: cyber security metrics (confidentiality, integrity impact, etc.), standards (CVE, CVSS, etc.) and risk strategies supported (real, temporal, weighted). These tools and the corresponding analysis are presented in detail next.

Nessus [55] supports the Common Vulnerability Scoring System (CVSS) standard [10], including metrics from versions v2 and v3 simultaneously. If both CVSS2 and CVSS3 attributes are present, both scores are computed. However, when computing risk factor, the CVSS2 score takes precedence. Besides, Nessus includes a risk factor based on CVSS which filters results based on the vulnerabilities detected in the ICT infrastructure (e.g., Low, Medium, High, Critical). The severity ratings are derived from the associated CVSS score,

where 0 is Info, less than 4 is Low, less than 7 is Medium, less than 10 is High, and a CVSS score of 10 will be flagged as Critical [56].

Saint8 [57] deals with assets, such as data, personnel, devices, systems and facilities that enable the organization to achieve business goals. Stakeholders are involved in risk identification and in providing data for computing both technical and business-related cyber security metrics, such as business unit, function, criticality and business cost impact. In addition, Saint uses CVSS score to create a risk profile to classify (prioritize) vulnerabilities. CVSS scores are grouped by severity levels: less than 4 corresponds to Potential risk factor, 4-7 scores map to Concern risk factor and 7-10 score to Critical.

Retina Security Scanner [19] assess risk and prioritizes remediation based on Real Risk strategy [58] in business context considering assets criticality and vulnerability exploitability (evaluated with the help of Core Impact®, Metasploit® and Exploit-db tools), CVSS, and other factors [59]. It is available as a standalone application or as part of Retina CS Enterprise Vulnerability Management. Retina CS version 5.7 [60] introduces new asset risk analysis, allowing the decision maker to "weight" the asset score based on either threat risks (i.e. vulnerabilities and attacks) or exposure risks (i.e. ports, shares, services, accounts). To normalize the risk according to a company's priorities a scale between 0 and 10 is introduced, with lowest score (0) corresponding to asset with lowest risk and with highest score (10) corresponding to asset with highest priority.

GFI LANGuard [61] [62] [63] scans the ICT infrastructure (hardware, network, operating systems, services, and applications), performs vulnerability analysis, risk assessment, and identifies and prioritizes remediation actions using databases such as Open Vulnerability and Assessment Language (OVAL) [14] and SANS Top 20 [64]. The tool also provides executive and technical reports for business and technical decision support.

nCircle IP360 and Tripware IP360 [65] [66] perform hosts data collection, vulnerability scoring and prioritization. Moreover, it also suggests remediation measures and prioritizes them. These tools make use of exploitability and vulnerability data from Tripwire's Vulnerability and Exposure Research Team (VERT). Business context is taken into account within risk assessment.

Security System Analyzer 2.0 (SSA) [67] [68] defines a patch management deployment strategy using CVSS scores to qualify the vulnerabilities. Also, SSA identifies vulnerabilities and discrepancies using the OVAL interpreter and performs compliance and security checks using the XCCDF - The eXtensible Configuration Checklist Description Format [69].

OpenVas [70] [26] scanner shows the results of the vulnerabilities prioritized according to the impact on the systems (high, medium or low) and indicates the number of vulnerabilities found for each impact category. Besides OpenVAS is an official OVAL Adopter and OpenVAS-5 is registered as 'Systems Characteristics Producer'.

QualysGuard [71] manages cyber security vulnerability risks taking into account severity, business risk, CVSS scores, existence of exploits, malware and available patches. It provides easy and flexible ways for ICT infrastructure scanning and cyber risk reporting.

Nexpose [72] associates CVSS metrics to calculate the risk of a vulnerability on an asset. It has different risk strategies which are based on the formula in which factors such as likelihood of compromise, impact of commitment, and asset importance are calculated. Each formula produces a different range of numeric values. Many of the available risk strategies use the same factors in assessing risk, each strategy evaluating and aggregating the relevant factors in different ways. The common risk factors are grouped into three categories: vulnerability impact, initial exploit difficulty, and threat exposure. The factors that comprise vulnerability impact and initial exploit difficulty are the six-base metrics employed in the Common Vulnerability Scoring System (CVSS). Threat exposure data come from three variables: Vulnerability age which is a measure of how long the security community has known about the vulnerability, Exploit exposure which is the rank of the highest-ranked exploit for a vulnerability that measures how easily and consistently a known exploit can compromise a vulnerable asset, and Malware exposure which is a measure of the prevalence of any malware kits, also known as exploit kits, associated with a vulnerability. The risk assessment strategies are: real risk, temporal plus risk, temporal risk, weighted risk and PCI ASV risk [73] [74].

- Real Risk, Equation 1 shows the formula used to calculate the Real Risk scoring model [58]:

$$\text{Risk} = \frac{\text{CVSS Impact Metrics}}{\text{CVSS Likelihood Metrics}} \times \text{Exposure}\left(\frac{\text{Malware Kits}}{\text{Exploit Rank}}, \text{time}\right) \quad (1)$$

- Temporal Plus, Equation 2 shows the formula used to calculate the Temporal Plus scoring model [75]:

$$\text{Risk} = \sqrt{t} \times \frac{(1+AV+C+I+A)}{(AC+Au)^2} \quad (2)$$

Where (t) is the time-based likelihood and represents the number of days since the vulnerability was publicly disclosed. The overall score increases with the number of days. The "CVSS" values refer to the various base component vectors of the CVSS version 2 which is broken down into 6 metrics, including: Access Vector (AV); Access Complexity (AC); Authentication Required (Au); Confidentiality Impact (C); Integrity Impact (I) and Availability Impact (A) [74].

- Temporal, Equation 3 shows the formula used to calculate the Temporal scoring model [75]:

$$\text{Risk} = \sqrt{t} \times \frac{(AV+C+I+A)!}{(AC+Au)^2} \quad (3)$$

- Weighted [72] [73], the Weighted risk model is based primarily on asset data and vulnerability types, and it emphasizes the following factors: 1) Vulnerability severity, ranging from 1 to 10; 2) Number of vulnerability instances; 3) Type of asset, such as a computer, router, or wireless access point (WAP); 4) Number and types of services on the asset; 5) The level of importance, or weight, that is assigned to a site when you configure it (e.g. low, high). Equation 4 shows the formula defined in the Nexpose configuration files for the Weighted scoring mode, this file can be found as "vulnsev-scvtype-devclass.xml" [76].

$$\text{Risk} = \text{vulnSeverity} \times 0{,}02 \quad (4)$$

- PCI ASV 2.0 [75] [77], this strategy applies a score based on the Payment Card Industry Data Security Standard (PCI DSS) Version 2.0 to every discovered vulnerability. PCI DSS specifies twelve requirements for compliance, among the requirements for risk assessment is defined "Vulnerability Categorization" to assist in prioritizing the solution or mitigating identified issues. Approved Scanning Vendors (ASVs) must assign a severity level to each identified vulnerability (1 = lowest severity, 5 = highest severity) and must use two tools to categorize and rank vulnerabilities, and determine scan compliance: 1. The Common Vulnerability

Scoring System (CVSS) version 2.0 and 2. The National Vulnerability Database (NVD). Any vulnerability with a CVSS base score of 4.0 or higher will result in a non-compliant scan.

4.2 Risk Assessment Tools Comparison

As described in the previous section each tool uses various techniques or strategies for risk-based prioritization. Most of these tools use CVSS score metrics to assess the risk that a vulnerability may pose to the business, either in the tool's own strategies or by adding new metrics that allow the user a better understanding of what is happening in the environment. In addition, to have more complete data for risk management, many of the tools have integration mechanisms with other commercial technology partners to further enhance the management of vulnerabilities that can affect an organization. Table 9 shows the tools comparison in terms of metrics, proposed strategies and if they support integration mechanisms with technology partners.

Tool	**Metric**	**Strategy**	**Integration mechanisms with**
Nessus Home	CVSS2, CVSS3	Results based on the risk factor of the vulnerability (e.g., Low, Medium, High, Critical)	Kenna, ThreatConnect, Cisco ISE, ForeScout
Saint8	Business unit, Criticality, Business cost, CVSS	Prioritization and the application of resources to assets based on metrics of importance to the organization.	Cisco FireSIGHT Management Center
EyeRetina	Business impact, Core Impact, Metasploit, Exploit-db, CVSS	Real risk to critical assets and exploitability	Kenna, IBM QRadar SIEM, LogRhythm
GFILanguard	OVAL, CVE	Security issues are rated by their severity level and each computer is given a risk and vulnerability rating.	Core Security Technologies
nCircle® IP360	CVE, CVSS OVAL,SCAP	Prioritizes vulnerabilities, manages risk and improves security efficacy by combining	Kenna, IBM QRadar, Bringa, LockPath, Trusted Integration

continue

		business context with vulnerability intelligence.	
Security System Analyzer	CVE, CVSS, OVAL, SCAP	-	-
OpenVAs	OVAL	The results of the vulnerabilities prioritized according to the impact on the systems.	Kenna, Greenbone, SecPod
QualysGuard	CVSS, CVE, SCAP, Severity	Risk-based approach to prioritizing the remediation efforts and fixing those vulnerabilities that would impact the business.	Bringa, Modulo, Kenna, ForeScout LogRhythm
Nexpose	CVE, CVSS, SCAP	Real Risk, Temporal Plus, Temporal Weighted, PCI ASV 2.0	Kenna, ForeScout,LogRhythm Bringa,LockPath, Modulo,RSA Security Analytics, Risk I/O, TraceSecurity, Agiliance, R.sam

Table 9 Comparison of Cyber Security Risk Management Tools.

Although most of the tools use the CVSS metrics for prioritization and risk management, some of them incorporate other metrics considered important to an organization. For example, Saint8 incorporates “Business unit”, “Criticality” and “Business cost” to know the impact that a vulnerability may have on the business. Eye Retina uses “Business impact”, “Core Impact Metasploit and Exploitdb” as other metrics to assess risk, and QualysGuard uses severity levels based on the CVSS score. It is possible to emphasize that some of the tools pose their own risk assessment strategy to support decision making. Among them are nCircle® IP360 that combines business context with vulnerability intelligence, Saint8 that associates not only the base metrics but also the environment metrics to measure the real risk impact on the organization, and Nexpose that incorporates different risk strategies adapted to the needs of the business. Another feature to note is the support for integration with other technology partners that different tools have. The technology partners provide a specialized service for risk assessment and decision support that also incorporates the results of the vulnerability scanning tool in a format compatible like XML - eXtensible Markup Language, making it more powerful for security and business value analyses. Most of the solutions provided by these technology partners are commercial or have a limited trial time, which represents a strong constraint for many companies.

C3-SEC Requirements, Architecture, Integration and Implementation

This chapter describes the software development dimension of the thesis, i.e., C3-SEC architecture, design, implementation and integration with Nexpose. First, a high abstraction level of the architecture is presented, followed by the technologies used to develop the C3-SEC decision support system and the corresponding implementation decisions made along the software development process.

5.1 C3-SEC Requirements and Development Methodology

UWE, UML-based Web Engineering, is applied as a web application oriented methodology in the present work. UWE is a methodology for the development of web applications focused on the systematic design, customization and semi-automatic generation of scenarios that guide the development process. Among the modeling activities of the methodology, the following activities belonging to the requirements analysis stage were adapted in the context of the current work: functional and non-functional requirements, where the functionalities of the system are described in detail and the realization of activity diagrams in which the responsibilities and actions of the actors involved are delimited [78].

5.1.1 Functional Requirements

The functional requirements of the application are described in Tables 10,11,12,13 and 14.

Id. Requirement	FR01
Name	User Login
Description	Enter username and password to access the application
Inputs	Username, password
Outputs	Admission to the application after verifying that it is a valid user.

continue →

Process	Authenticating the input data in the database application.
Preconditions	Be a valid user.
Postconditions	Access to the application is granted or not.
Collateral effect	If the username and password are incorrect, warning messages will be displayed and will not allow access
Priority	High
Role executes	Ciber security decision maker

Table 10 Functional Requirement 1

Id. Requirement	**FR02**
Name	Upload XML file
Description	The user must upload the XML file generated in Nexpose.
Inputs	XML file
Outputs	Message with notification of the status of the load.
Process	Transformation of XML file information to OWL ontology.
Preconditions	First login to the application with username and password.
Postconditions	Ontology created based on the information provided by the XML file to make the corresponding reports on the vulnerabilities that affect the company.
Collateral effect	In case of a problem in the loading process, the user will be notified by a warning message.
Priority	High
Role executes	Ciber security decision maker

Table 11 Functional Requirement 2

Id. Requirement	FR03
Name	Company Information Report
Description	The user can view the most relevant information on the state of the company's assets as well as graphical reports that indicate the total vulnerabilities by category and number of vulnerabilities that affect the security pillars.
Inputs	Ontology created by C3-SEC.
Outputs	Report about company information.
Process	Query for the ontology to generate the report.
Preconditions	FR02
Postconditions	The results generated by the inference of the ontology are presented in the report.
Collateral effect	In case of a problem in the reporting process, the user will be notified by a warning message.
Priority	Medium
Role executes	Ciber security decision maker

Table 12 Functional Requirement 3

Id. Requirement	FR04
Name	Vulnerabilities by Impact Risk Report
Description	User can view vulnerabilities categorized by risk impact (very low, low, medium, high, very high).
Inputs	Ontology created by C3-SEC.
Outputs	Vulnerabilities found by impact risk.
Process	Query for the ontology to generate the report.

continue →

Preconditions	FR02
Postconditions	The results generated by the inference of the ontology are presented in the report.
Collateral effect	In case of a problem in the reporting process, the user will be notified by a warning message.
Priority	Medium
Role executes	User

Table 13 Functional Requirement 4

Id. Requirement	**FR05**
Name	Vulnerability Description
Description	The user can see a more detailed description of the vulnerability as well as remediation measures and which security pillars it affects.
Inputs	Selected vulnerability of report “Vulnerabilitiess by rsisk impact”
Outputs	Description about selected vulnerability.
Process	Query for the ontology to generate the report.
Preconditions	Select vulnerability of the report "Vulnerabilities by risk impact"
Postconditions	The results generated by the inference of the ontology are presented in the report.
Collateral effect	In case of a problem in the reporting process, the user will be notified by a warning message.
Priority	Medium
Role executes	User

Table 14 Functional Requirement 5

5.1.2 Non-Functional Requirements

Non-functional requirements of this work are as follows:

a) Requirements Interface
    1. The web application language interface will be in English.
    2. The web application lets the user to visualize all ontology required information.
    3. The web application should minimize the ontology topology complexity.
b) Requirements Navigation
    1. The web application will use consistent and coherent navigation mechanisms, improving its the usability of the web application. Allows users to easily identify navigation patterns and possible disorientation navigation is prevented.
    2. The web application will have standard navigation buttons (home, back, etc.).
    3. The web application will prevent the opening of pop-ups (pop-ups, because these can become cause disorientation in the time window is changed and can cause unpredictable results user interface behavior in devices that do not support multiple opening more than one windows interfaces).
c) Usability requirements
    1. The web application will have an attractive and user-friendly interface.
    2. The web application display error messages according to its the activities.
    3. The web application does not allow users to run unfinished operations.
d) Scalability requirements
    1. The web application will be able to allow maintenance changes and, with new features upgrades.
e) Operational requirements
    1. The web application will have mandatory fields.
    2. The web application restricts invalid data entry for all existing fields.
    3. The web application will validate passwords for user access.
f) Safety requirements
    1. The web application will handle information with integrity.
g) Hardware requirements
    1. To implement the Web application there is no restriction in terms of hardware, as it is not required to install the application on specific purpose devices or computers.
    2. The web application allows proper display in all screen resolutions, however the resolution of 1024x768 and higher is recommended to view the entire scene on the screen.

5.2 C3-SEC System Architecture

Figure 7 shows a high-level view of the decision support system architecture and all the components that take part of it. This architecture is based on the layers of the context-aware systems stack (Perception, Comprehension, Projection and Decision/Action). The system is composed of five main components that are described next:

- Component one corresponds to the perception layer, in which the data about the technological infrastructure of the organization and its vulnerabilities are obtained through the selected tool (Nexpose). The information provided by Nexpose is based and compliant with international security standards and metrics (CVSS, CVE, CPE) proposed by entities such as NIST and MITRE.
- Component two and three correspond to the comprehension layer that is in charge of the transformation of the information provided by Nexpose in XML format to an OWL ontology format. At this level, cyber security experts of an organization are allowed to introduce new specific knowledge into the ontology using the Protégé ontology editor. Assets characterization such as asset value and importance of each security dimension associated to that asset (privacy, integrity, availability) must be provided by the organization experts and added to the ontology. This knowledge is essential to support corporation specific cyber risk analysis and management. In addition, the ontology is extended with new business-related metrics such as cost, weight, impact and security pillars (confidentiality, integrity, availability), benefiting from the formal logics inference and reasoning and decision aiding features made possible by semantic technologies.
- Finally, part four and five corresponds to the projection and decision / action layer, all layers being supported by a web application that provides decision aiding for chief information security officers to take appropriate decisions and actions in maintaining the security of the organization.

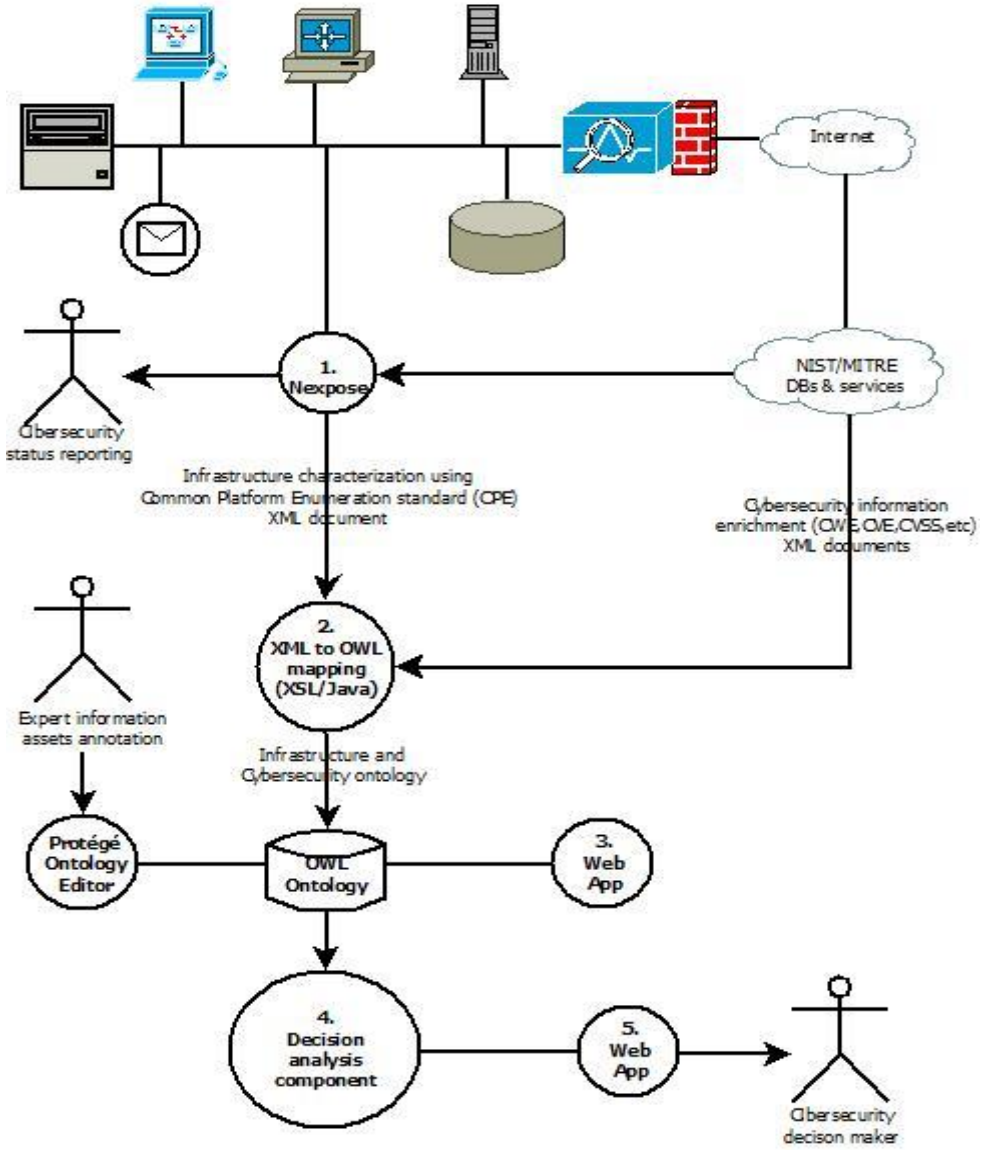

Figure 7 C3-SEC System Architecture

5.3 C3-SEC Integration with Nexpose

Among the possible approaches for software applications data integration (file transfer, shared database, remote procedure invocation and messaging), a XML file transfer/sharing approach was adopted and implemented for C3-SEC integration with Nexpose. Additionally, a presentation layer integration framework needs to be used for *single sign-on* and transparent, unified graphical user interface, use of Nexpose and C3-SEC. The integrated workflow of C3-SEC and Nexpose is currently based on a sequence of steps to produce results on threats that can affect the enterprise environment and help at security expert to make informed decisions about cybersecurity actions to take. The UML activity diagram of

Figure 8 shows the activities that are carried out in the C3-SEC and Nexpose integration workflow.

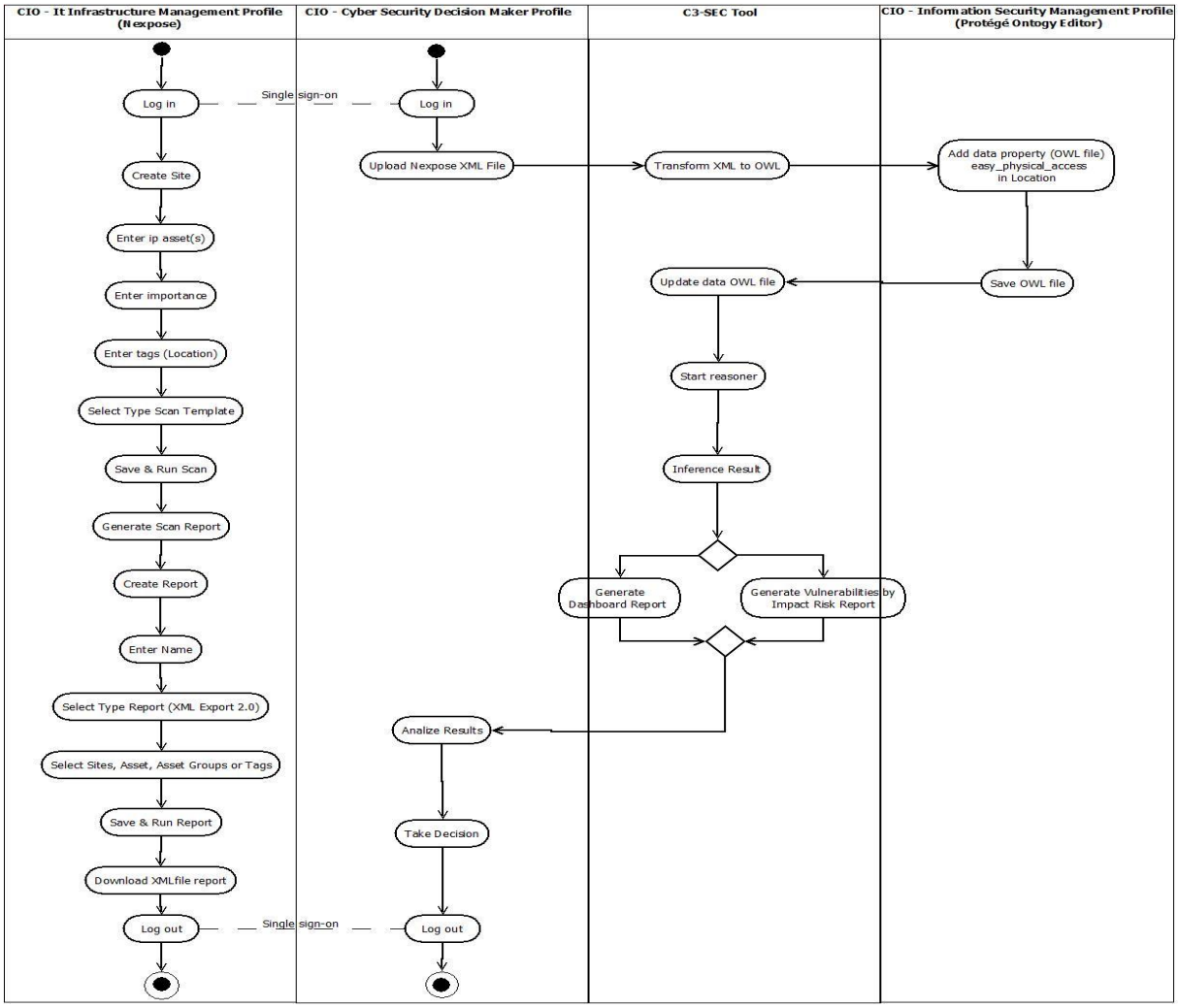

Figure 8 Activities Diagram Activities (C3-SEC Integration with Nexpose)

The activities represented in the diagram of Figure 8 belonging to Nexpose and C3-SEC are described in detail in the following sections.

5.4 Nexpose Features

Therefore, to obtain the report in XML format containing the information about the technological infrastructure of an organization and its vulnerabilities, we must create a site with Nexpose (Figure 9). The assets of the organization to be scanned are specified, i.e., named, a corresponding IP address is assigned (Figure 10) and eventually extra tags are added that help to identify the importance of the asset for the organization.

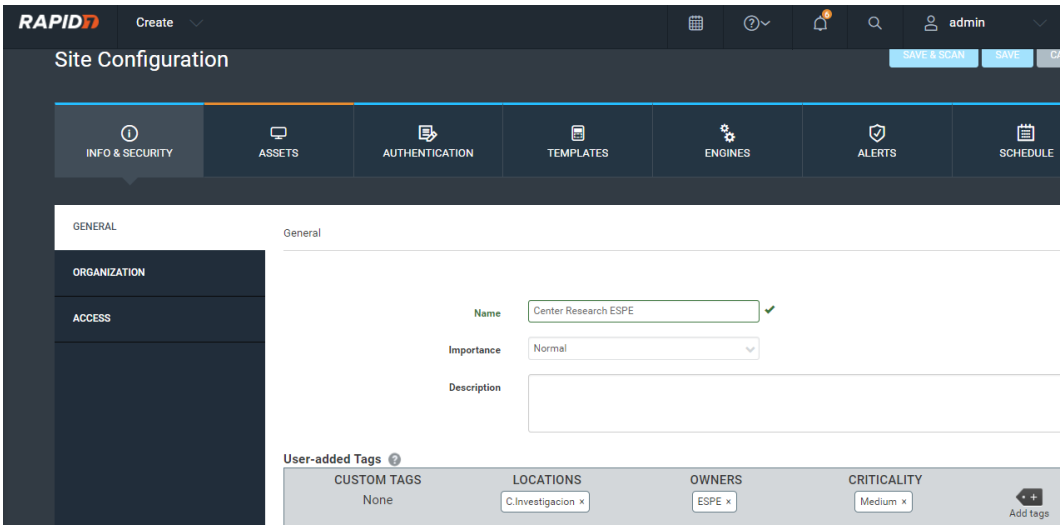

Figure 9 New Site Configuration

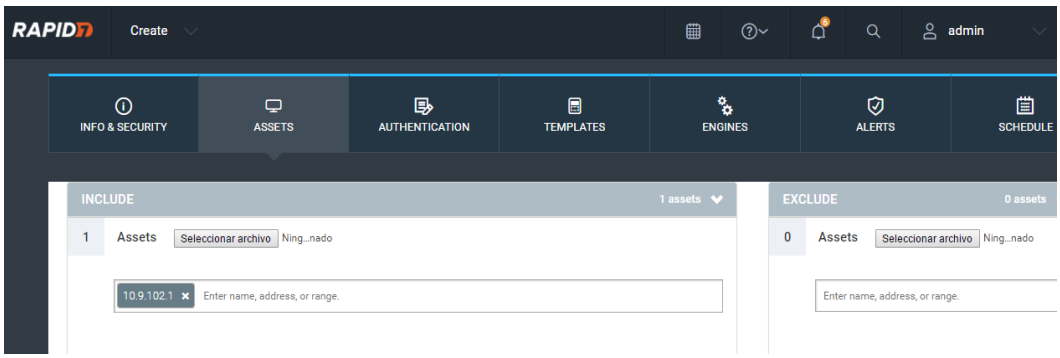

Figure 10 Adding IP Address

Figure 11 shows the last step of the site configuration which is to save and run the scanning process.

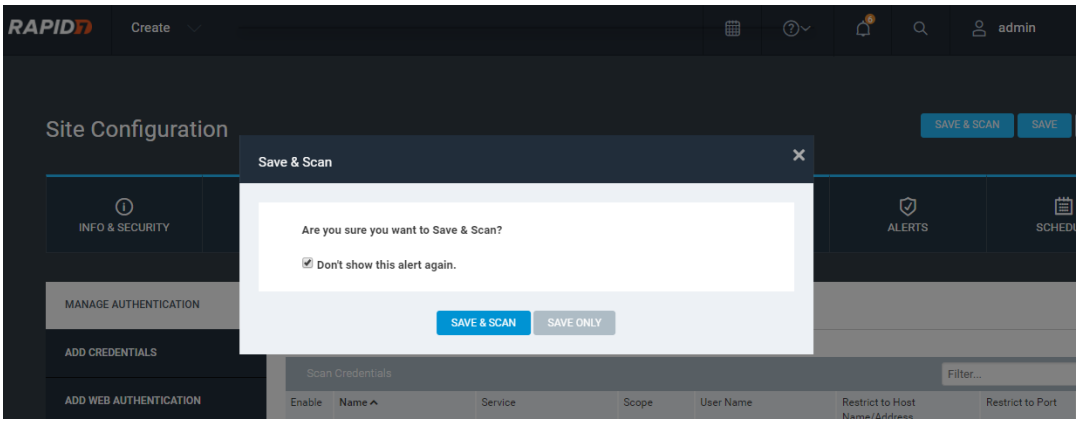

Figure 11 Save and Scan

Once the scan is executed, Nexpose displays a graphical report of the asset status as is shown in Figure 12.

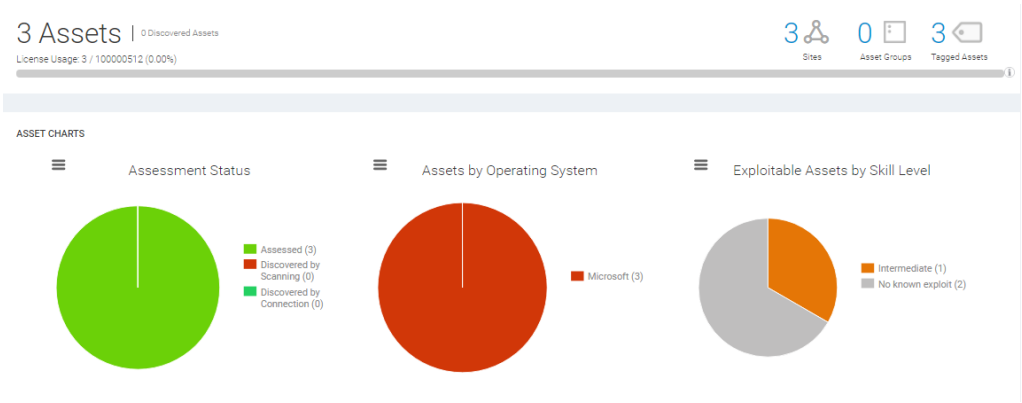

Figure 12 Scan View of the Computer Science Research Infrastructure Center at ESPE-Ecuador

To generate the report of the scan performed we must usego to the module “Reports” in the “Create a report” option and the “Export” tab. Nexpose offers different formats to export the results, among them are ARF (Asset Reporting Format), XML format and, Database Export. In our study this case we select “XML Export 2.0” must be selected, which contains all data available in XML, as well as additional risk fields, associated vulnerabilities and malware kits, PCI compliance, site information and scan information. A name mustmay be assigned to identify the report (Figure 13).

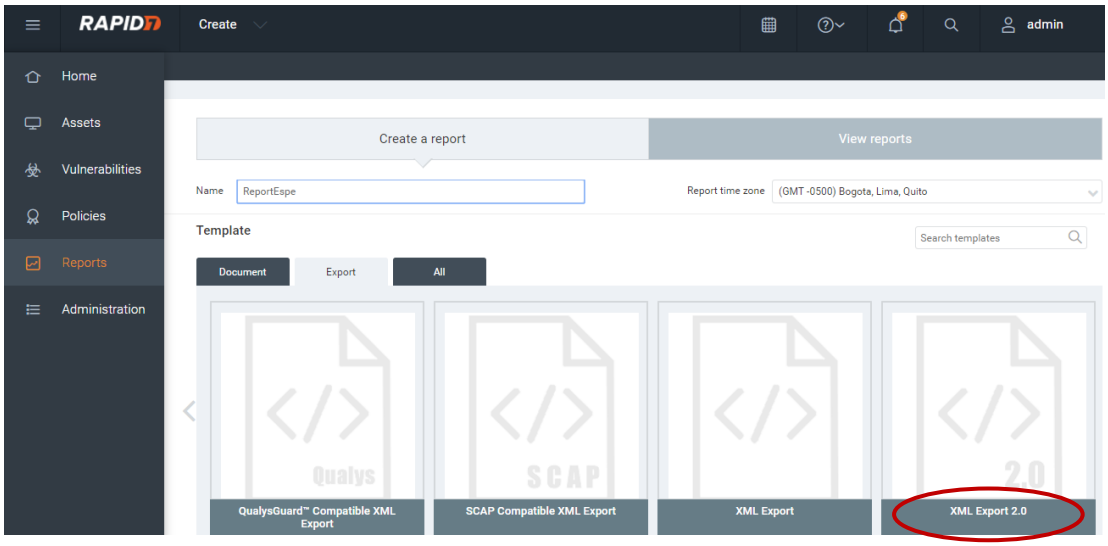

Figure 13 Scan Report XML

The next step is to select the scan to be used for the report generatation (“Select Scan” in Figure 14).

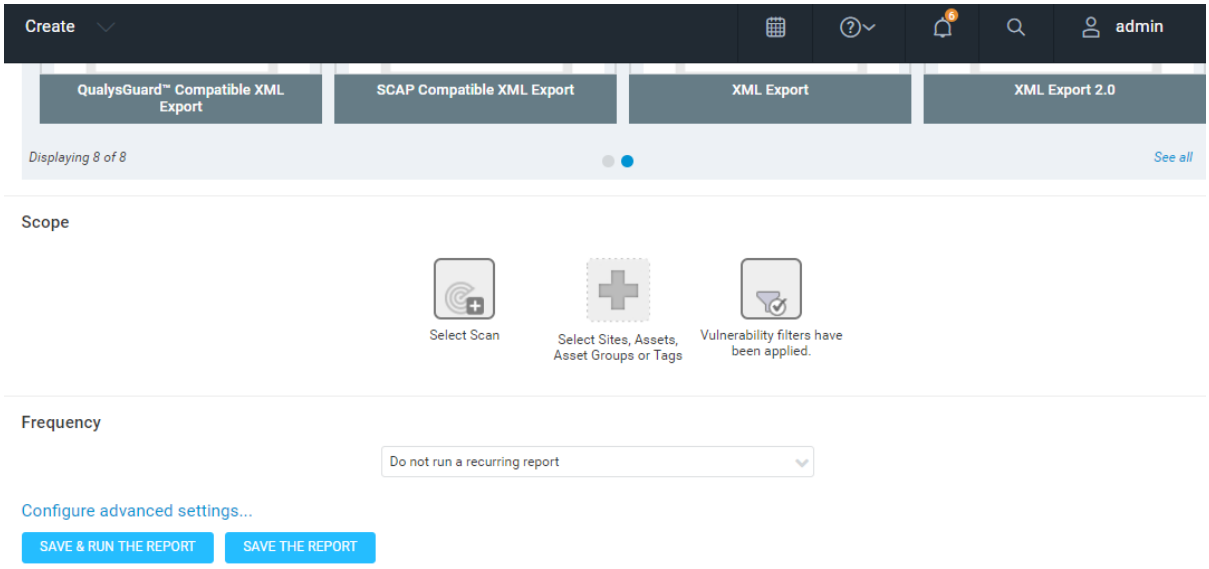

Figure 14 Select Scan Option

As shown in Figure 15, the window indicates that must be selected the site in which the scan was run must also be selected. Once selected the site click on the “Select Scan” button.

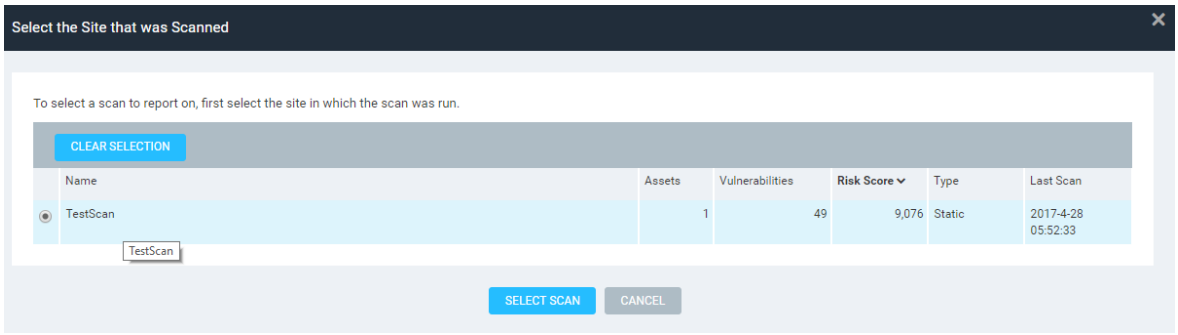

Figure 15 Selection Site that was Scanned

Next, in the following window it must select the specific scan instance that serves as input for the report has to be selectedthat it wants to report on (Once selected the scan click on the “OK” button as shown in Figure 16).

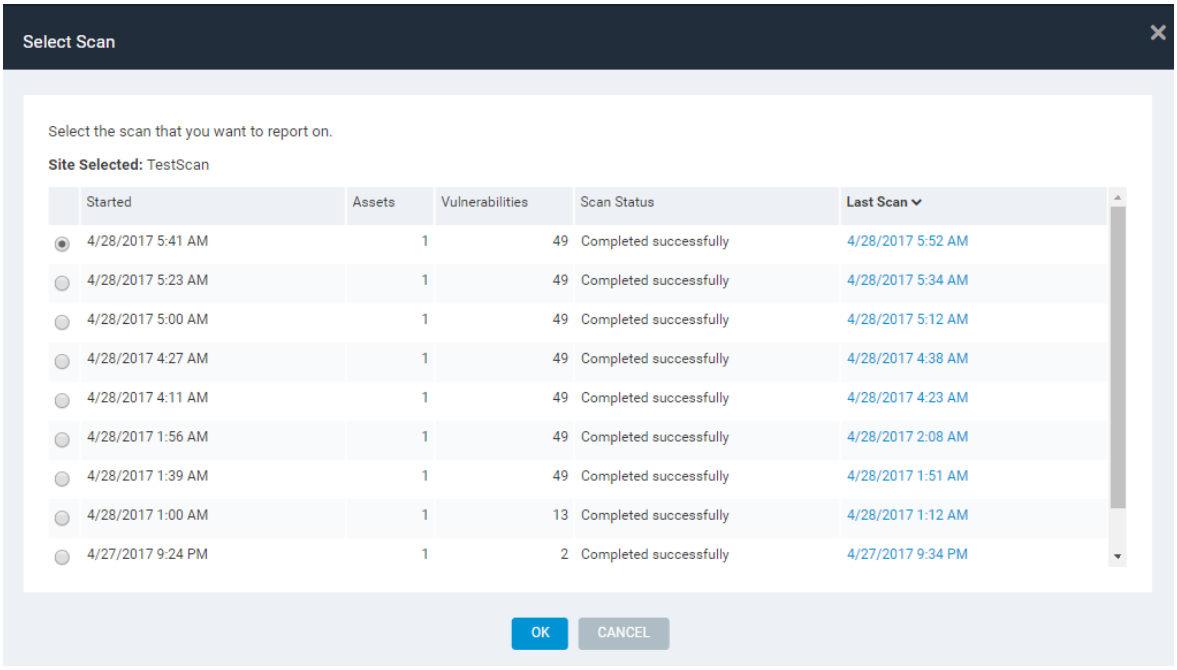

Figure 16 Selection Scan Window

Figure 17 shows that the scan selection process is correct and that it is possible tocan proceed to save and generate click on the “Save & Run Report” button to generate the report.

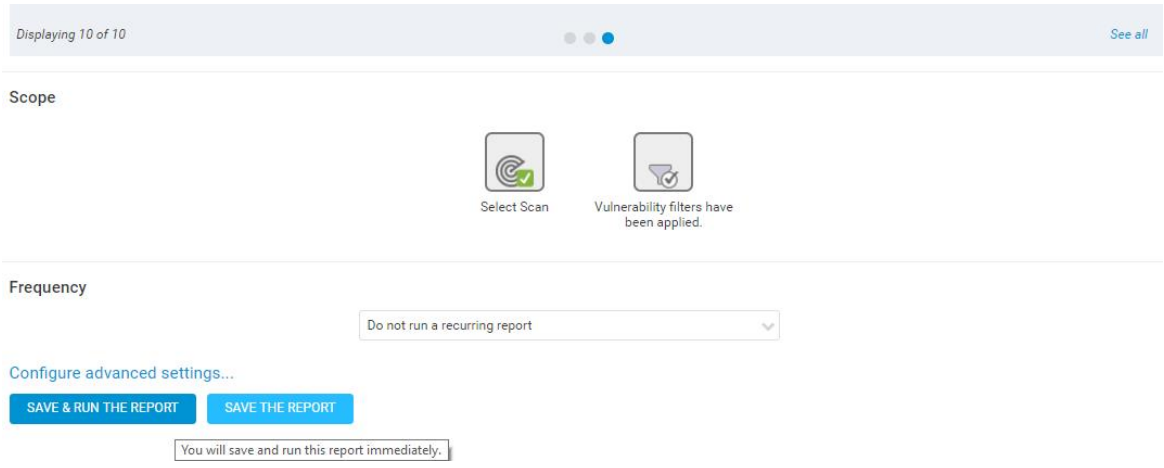

Figure 17 Save and Run Report Option

Once the execution is finished, it can see a link that directsto the report generated in XML format is shown. Figure 18 shows a list of previously generated reports as example.

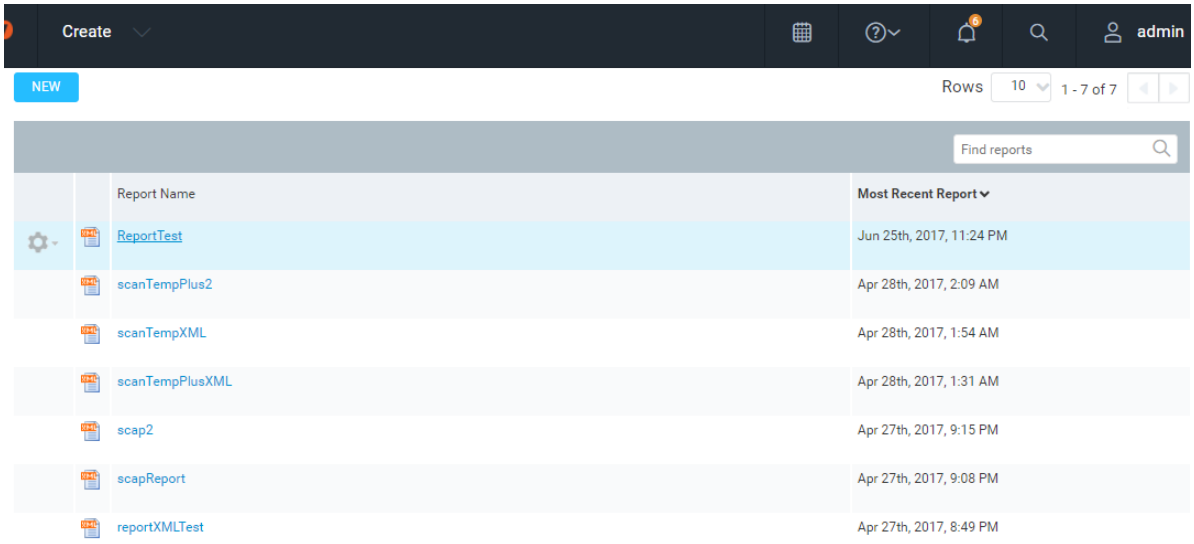

Figure 18 List Scan Reports

The following section explains each of the modules of the application, in addition the integration of the results file in XML format generated by Nexpose.

5.5 C3-SEC Features System Modules

This section presents C3-SEC features, including the integration process of XML Nexpose reports with C3-SEC.

5.5.1 User Login

Figure 19 displays the login page where the user must type his user name and password to access the application.

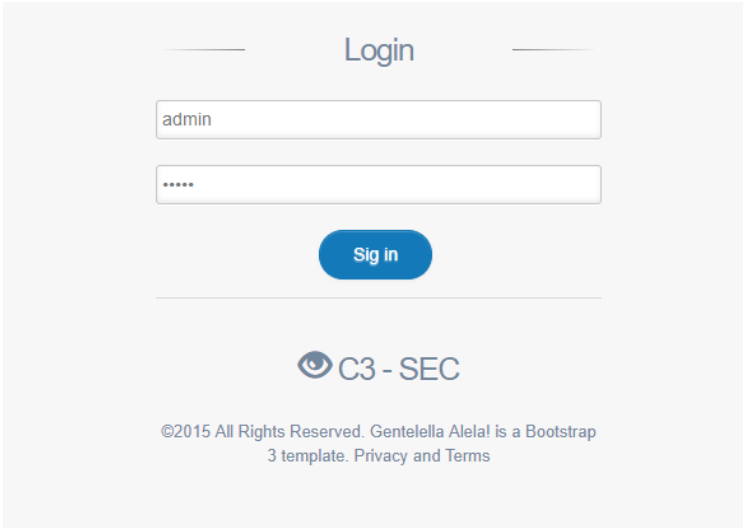

Figure 19 User Login

5.5.2 Upload Module

When the system is accessed for the first time, the application automatically forwards the user to the "Upload" module. This module allows loading the file generated by Nexpose with the information about the technological infrastructure and vulnerabilities found in the organization, which will be automatically transformed by the application and incorporated into the designed ontology. The C3-SEC gives the possibility to create a new ontology based on the loaded file or to add the information of the file to the ontology previously created. Figure 20 shows the interface to load the Nexpose XML file into C3-SEC. If the load is correct, a message will be displayed notifying the user that the load has been successful.

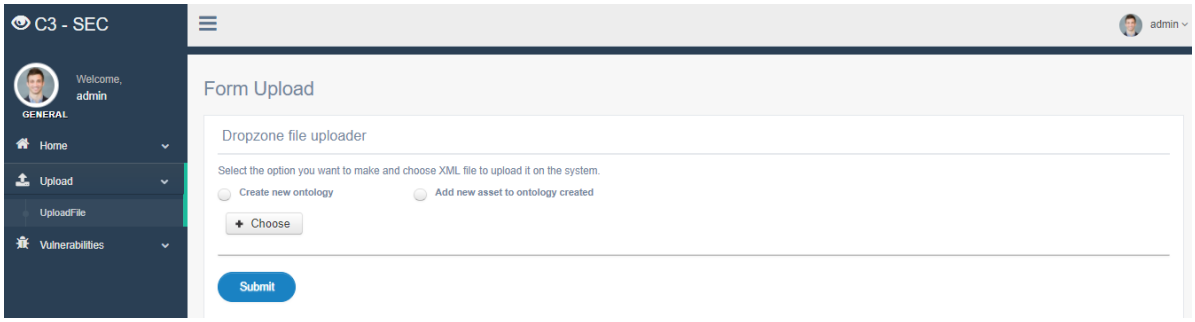

Figure 20 Upload Module

5.5.3 C3-SEC Dashboard Module

The C3-SEC dashboard module shown in Figure 21 presents shows a general corporative cyber security situation awareness overview view in terms of security to know the state of the company. Figure 21 shows the main asset(s) information such as alias, ip address, total vulnerabilities and total risk score, and two reports about the total number of vulnerabilities by category and the number of vulnerabilities by security pillar.

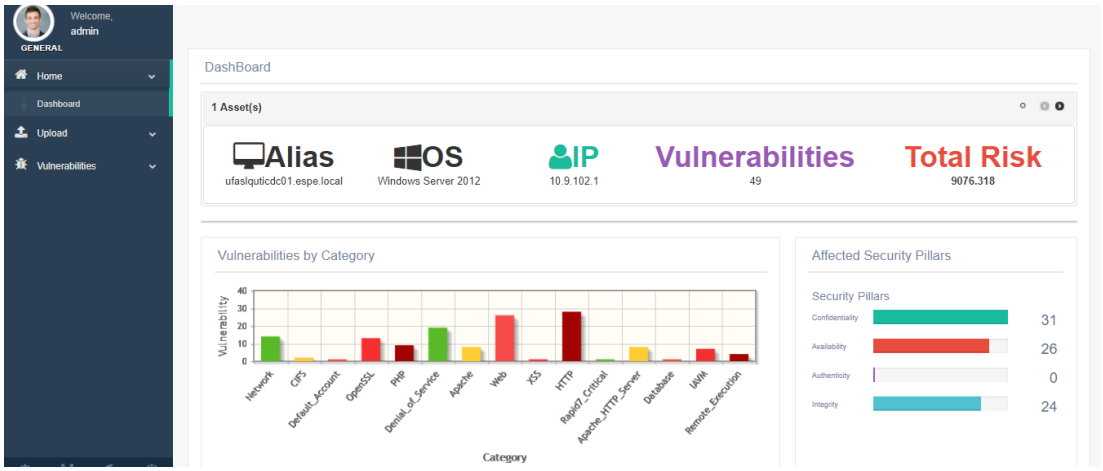

Figure 21 Dashboard Module

5.5.4 Impact Risk Module

The C3-SEC Impact Risk Module This module provides insights of shows a view of the corporations vulnerabilities according to the risk impact they represent for in their businesses company (very low, low, medium, high, very high). Each of the vulnerabilities is shown in a table categorized by its risk impact (see Figure 22 and Figure 23).

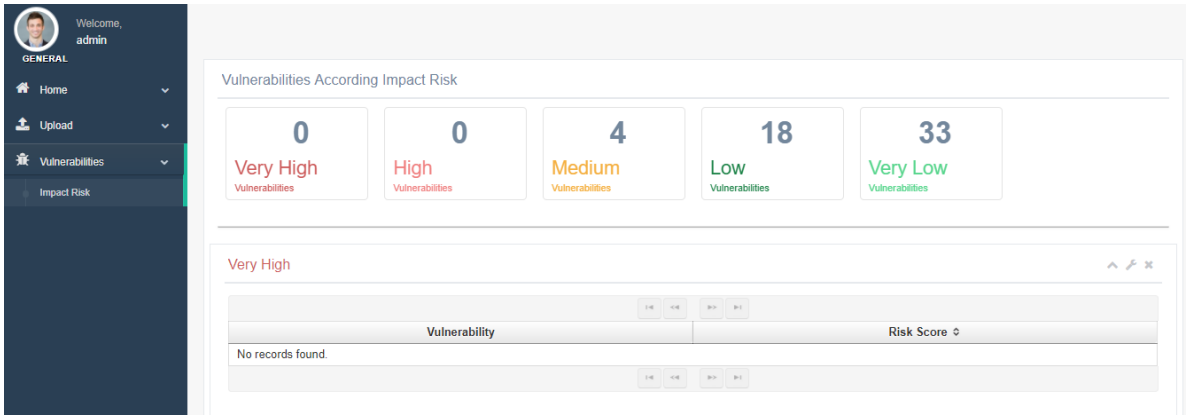

Figure 22 Impact Risk Module

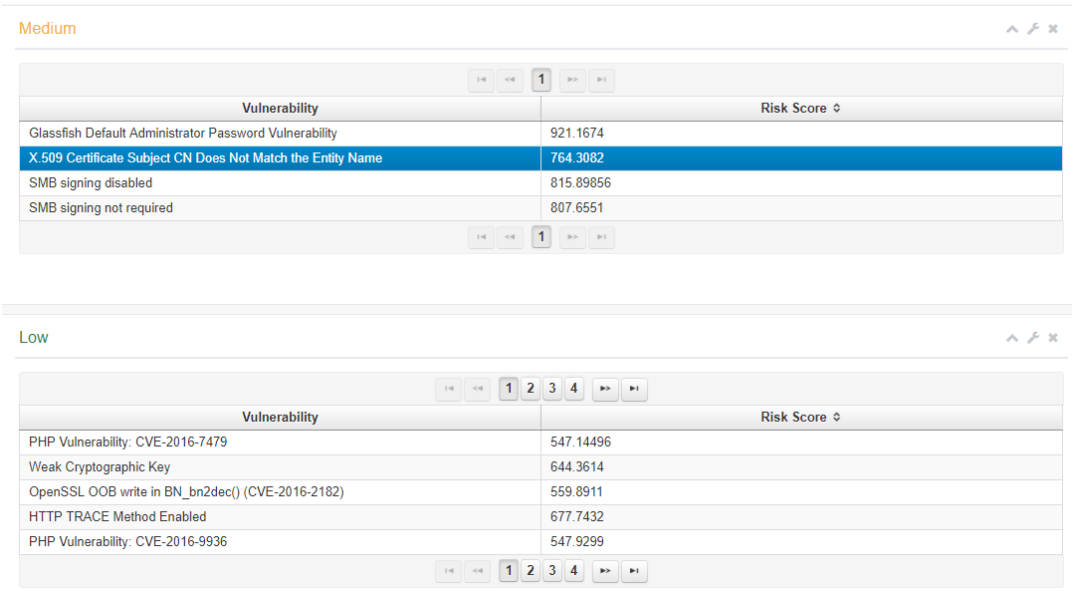

Figure 23 Vulnerabilities by Impact Risk

In addition, when selecting one of the shown vulnerabilities that are in the tables, C3-SEC displays a window with the vulnerability information such as CVSS score, Risk score, Description, Security Pillars that it affects and corresponding its remediation actions (, i.e., Figure 24).

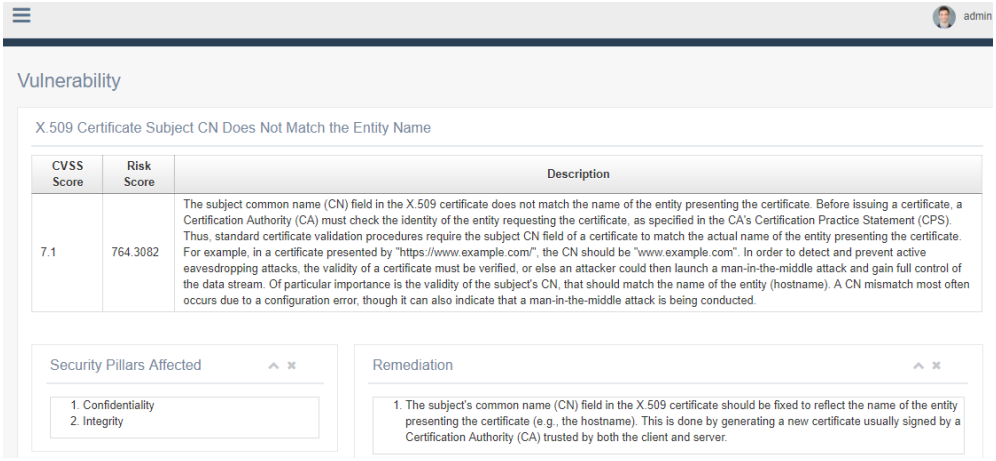

Figure 24 Vulnerability Information Window

5.6 Implementation

A web application was developed to make the decision support system features available to the cyber security professionals (users/decision makers) via a web browser. For the development of the web application Java EE [79] development and execution technologies were used. Figure 25 shows the core Java EE components adopted in the software developed for this thesis.

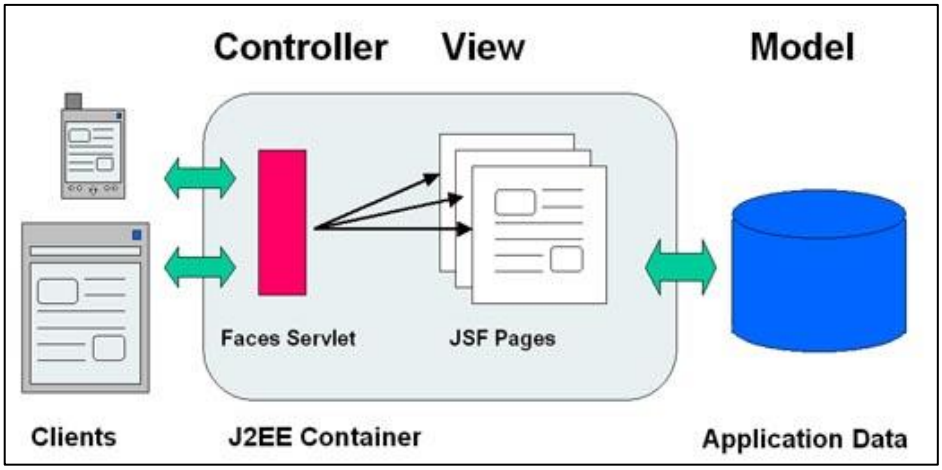

Figure 25 Java EE Technologies for Web Application Development

Among the technologies offered by Java EE framework, the following were especially useful and used in this thesis:

- **JSF (Java Server Faces)** [80]**:** The web application was made with JSF 2.2, a framework of user interfaces based on Architecture Model View separating its components to provide greater control over every part of the application, facilitating their development and maintenance.

- **PrimeFaces** [81]**:** PrimeFaces framework latest version (5.3) was used, which contains open source visual components for the whole Java Server Faces 2.2, for the creation and design of the web application.
- **Template Bootstrap** [82]**:** For the visual interface of the application Gentelella Bootstrap Admin Template was used, which is available for free on its official website.

Model-View-Controller (MVC) design pattern was adopted for the software development, which presents well known robust properties and software quality attributes. MVC separates business logic with respect to the data (model) and user interface (view / GUI). It allows independent changes in each of the parts without affecting the other. In other words, changes in the user interface (GUI) do not affect data handling, and data can be reorganized without changing the user interface. The description of the MVC components is presented next.

- **Model**

The development of this layer implied the definition/implementation of classes showing the model of the entities that interact with the application. This allows to access the attributes or fields of the ontology and to work with data as objects. The code fragment shown next is a representative code fragment of a class model.

```
public class OperativeSystem {
    private String vendor;
    private String family;
    private String product;

    public OperativeSystem(String vendor, String family,
String product) {
        this.vendor = vendor;
        this.family = family;
        this.product = product;
    }
    public String getVendor() {
        return vendor;
    }
    public void setVendor(String vendor) {
        this.vendor = vendor;
    }

   public String getFamily() {
        return family;
    }
```

```
    public void setFamily(String family) {
        this.family = family;
    }
    public String getProduct() {
        return product;
    }
    public void setProduct(String product) {
        this.product = product;
    }
}
```

- **Controller**

The controller receives user requests and in response returns the corresponding view. Among the relevant classes of this layer is "ManagedBean", which contains the get and set methods, business logic or even unbacking bean methods. For the management of the ontology, the following Maven Project dependencies were set:

1. owlapi-distribution-5.0.4.jar.
2. owlapi-api-5.0.4.jar.
3. jfact-5.0.1.jar.
4. openllet-owlapi-2.5.1.jar
5. openllet-core-2.5.1.jar

In each of the "Bean" it is necessary to declare the following parameters that allow to upload and manage the ontology, as shown in the following code fragment.

```
private OWLOntologyManager manager;
private IRI documentIRI;
private OWLOntology ontology;
private OWLReasonerFactory factory = null;
private OWLReasoner reasoner;
private OWLDataFactory dataFactory;
```

The class constructor should initialize each of the declared objects as shown in the next code fragment.

```
public ImpactRiskBean() {
    try {
        // Load an ontology from local
        PathOntology path = new PathOntology();
```

```
            File file = new File(path.getPath());
            manager = OWLManager.createOWLOntologyManager();
            ontology =
manager.loadOntologyFromOntologyDocument(file);
            documentIRI =
manager.getOntologyDocumentIRI(ontology);
            factory = new JFactFactory();
            OWLReasonerConfiguration config = new
SimpleConfiguration(500);
            // Create a reasoner that will reason over our
ontology and its imports
            // closure. Pass in the configuration.
            reasoner = this.factory.createReasoner(ontology);
            // Ask the reasoner to classify the ontology
reasoner.precomputeInferences(InferenceType.CLASS_HIERARCHY);
            dataFactory = manager.getOWLDataFactory();

        } catch (OWLOntologyCreationException ex) {
Logger.getLogger(ImpactRiskBean.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
```

For data visualization and ontology reasoning in the web application, different methods were developed, which allow for reading of individuals, classes, subclasses and properties of the ontology. Among the several available reasoners, FaCT ++ a reasoner covering OWL and OWL 2 DL-based ontology languages was selected for this project because it is the reasoner that best fits the version of the OWL API adopted in the thesis. Pellet (Openllet) reasoner was also used, specifically for the Semantic Web Rule Language (SWRL) support. A fragment of the method/code that allows to get “individuals” with their respective properties of a particular class is given next.

```
    public List<String> printIndByClass(OWLOntology ont, String
clase, OWLDataFactory dataFactory, OWLReasoner reasoner) {
        List<String> listIndByClass = new ArrayList<>();
        String base = "http://www.co-
ode.org/ontologies/testont.owl";
        OWLClass claseInd = dataFactory.getOWLClass(IRI
                .create(base + "#" + clase));
        NodeSet<OWLNamedIndividual> individualsNodeSet =
reasoner.getInstances(
                claseInd, true);
        Set<OWLNamedIndividual> individuals =
individualsNodeSet.getFlattened();
        String individualClass = "";
        for (OWLNamedIndividual ind : individuals) {
```

```
            individualClass = pm.getShortForm(ind).replaceAll(":",
"");
            individualClass = individualClass.replaceAll("_", "
");
            listIndByClass.add(individualClass);
        }
        return listIndByClass;
    }
```

- **View**

The view is basically responsible for the user interface and interactions, accepting her/his requests and displaying the answers to those requests. Next a code fragment of the view component is shown.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:h="http://java.sun.com/jsf/html"
                xmlns:f="http://java.sun.com/jsf/core"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:p="http://primefaces.org/ui"
                template="template.xhtml">
    <ui:define name="titlePanel">
        Form Upload
    </ui:define>
    <ui:define name="title2">
        Dropzone file uploader
    </ui:define>
    <ui:define name="contenido">
        <h:form   enctype="multipart/form-data">
            <p>Choose XML file and click "Submit" button to upload
file.</p>
            <p:growl id="messages" showDetail="true" />
            <p:fileUpload value="#{fileUploadView.file}"
mode="simple" skinSimple="true" />
            <p:separator />
            <p:commandButton class="btn btn-round btn-primary"
value="Submit" ajax="false"
actionListener="#{fileUploadView.upload}"  />
        </h:form>
    </ui:define>
</ui:composition>
```

5.6.1 Implementation in the Cloud

For the implementation and deployment of the web application in the cloud, Amazon Web Services EC2 was selected because it provides all the necessary support for the correct configuration of the required infrastructure and is one of the services with highest robustness, scalability and storage capacity in comparison to other providers of this service [83]. The tools, resources and steps described next were used/followed in the C3-SEC software project, to make use of AWS, Amazon Web Services, specifically EC2, Amazon Elastic Compute Cloud.

- Java Development Kit (JDK) [84], provides the necessary tools for the development and coding of programs in Java (e.g. Java applications and applets).
- Amazon Machine Image (AMI) [85], is a template that contains the software configuration (operating system, application server and applications) that are required to launch an instance of the virtual machine. It can select an AMI provided by AWS, the user community, or the AWS market, or make use of customized AMI.
- PUTTY [86], it's a free Telnet and SSH implementation for Windows and Linux platforms, along with an xterm terminal emulator.
- Secure SHell (SSH) [87], is a protocol that facilitates secure communications between two systems using a client / server architecture and allows users to remotely connect to a host. Unlike other remote communication protocols such as FTP or Telnet, SSH encrypts the connection session, making it difficult for anyone to obtain unencrypted passwords.
- Amazon Web Services (AWS) provides a scalable, high-reliability, low-cost cloud infrastructure that drives hundreds of thousands of businesses in 190 countries around the world. Thanks to data centers located in the US, Europe, Brazil, Singapore, Japan and Australia, customers from all economic sectors can benefit [88].
- Glassfish Server: It is an open source application server that offers advanced features such as application version control, application scope resources, and great support for NetBeans 7.0 development tools, and higher versions such as Eclipse and Other popular IDEs [89].
- JDBC: Java Database Connectivity, sends SQL commands to a relational database engine, which can be Oracle, Infomix, SyBase, etc. JDBC is a low-level API for high-level APIs, also providing an integration of SQL into Java, i.e. SQL statements

mixed with Java (e.g. a Java variable can be used in an SQL statement to receive or return results) [90].

- EC2: Amazon Elastic Compute Cloud, are the virtual servers in the cloud provided by Amazon [91].

The steps described next are required to run the application in the cloud:

1. In *http://aws.amazon.com* web page (see Figure 26) a "Register" operation must be performed with an email account, and a password.

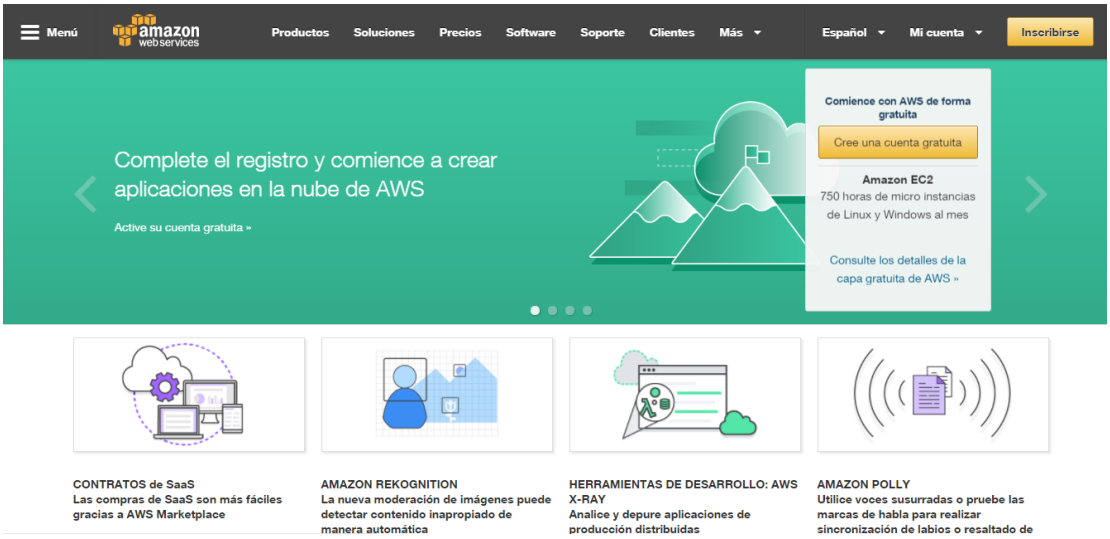

Figure 26 Home Amazon Web Services

2. Once the subscription and "Log in" has been done the data about all the Web Services Amazon provides is presented (see Figure 27). For the software project of this thesis EC2 was selected (Amazon Elastic Compute Cloud).

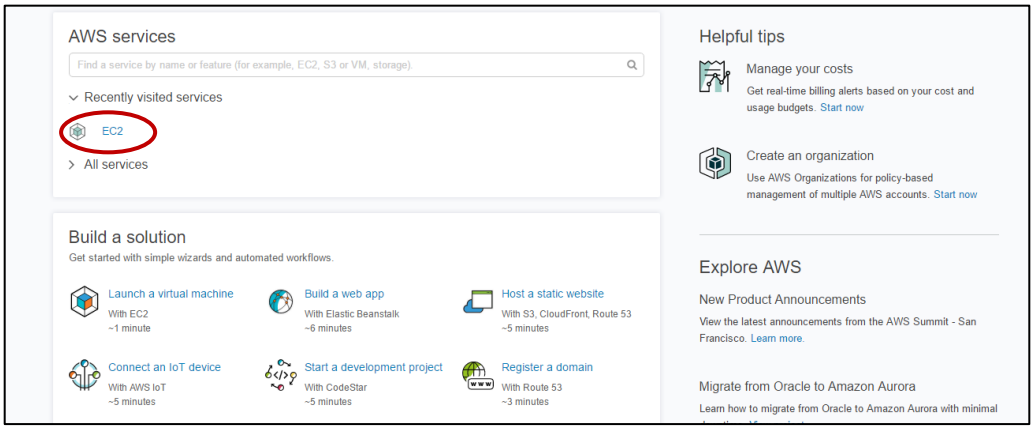

Figure 27 Amazon Web Services

3. Selecting the “EC2 Dashboard” option allows for the creation of the required instance and “Launch Instance” button (Figure 28) for running it.

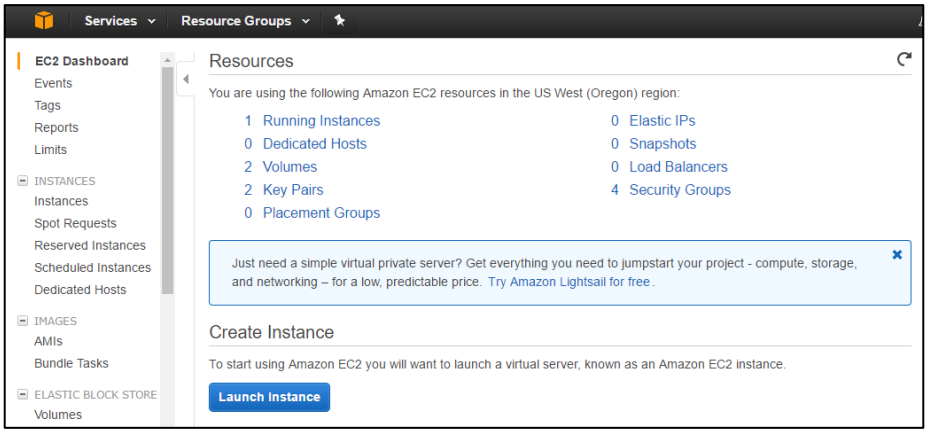

Figure 28 EC2 Dashboard

4. To create the new instance, it is necessary to set the parameters that comply with the tools needed to upload the application to the cloud (Mysql, Glassfish server). In the present project Ubuntu Server 16.04 free version was selected (see Figure 29).

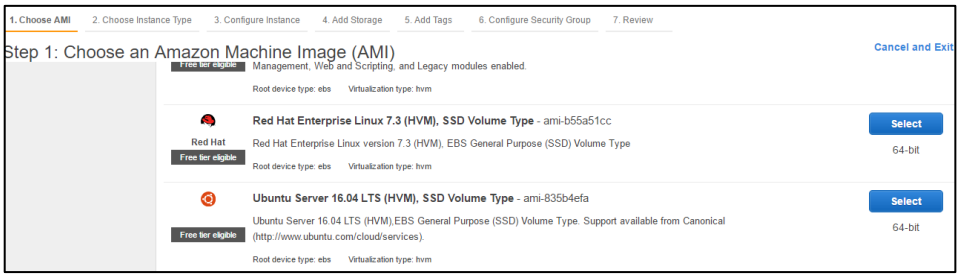

Figure 29 AMI in AWS

The AMI has the following characteristics:

- Operating System: Ubuntu Server 16.04
- Instance Type: t2.micro
- Memory: 1GiB = 1.07 GB
- Processor: Intel Xeon High Frequency, Turbo up to 3.3 GHz
- Layer: Free

5. In this instance, it is necessary to add in the “Configure Security Group”, the rule “Custom TCP Rule” (TCP protocol is necessary to guarantee ordered delivery of data packets) and reserve the 4848 port for this service. Being a resource that is widely

required in the project “Source” configuration must be set to “Anywhere”. Once the instance is configured “Review and Launch” can be performed (see Figure 30).

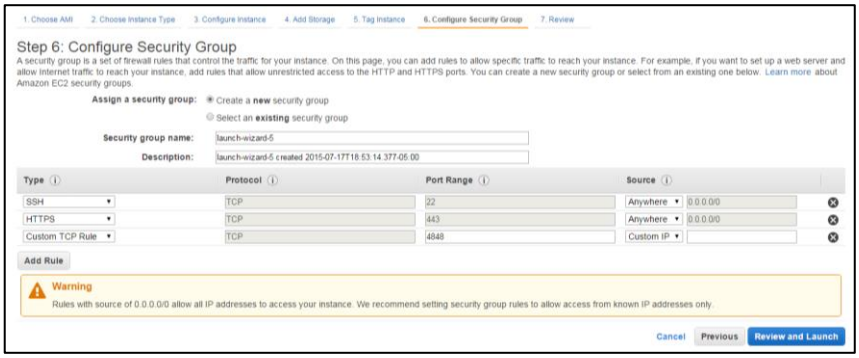

Figure 30 Configure Security Group

6. For security purposes a public key cryptography “Key Pair” must be generated and a corresponding label/description assigned for key management tasks support, (“Download Key Pair” button as shown in Figure 31).

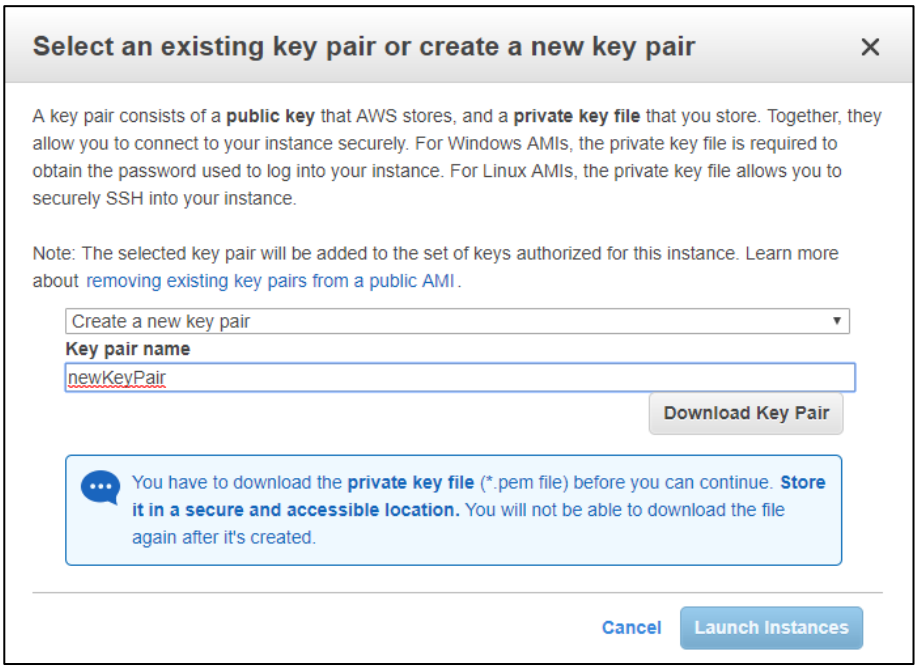

Figure 31 Creation New Key Pair

7. Once the Key Pair is generated, the instance is launched and an automatic notification with a message with the state of the instance is generated (as shown in Figure 32).

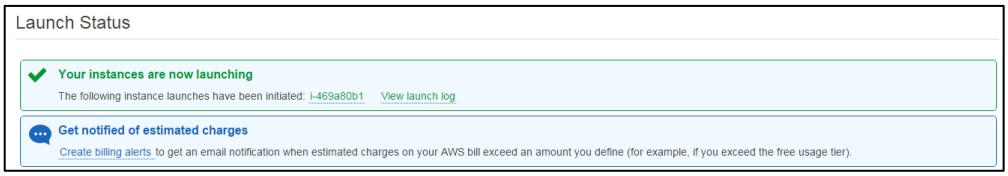

Figure 32 Instance Launch Status

The next step is to connect to the instance with the Java SSH client provided by Amazon or with the Putty tools [92].

8. As shown in Figure 33 Putty allows the generation of the private key ".ppk" by reading the previously downloaded KeyPair .pem.

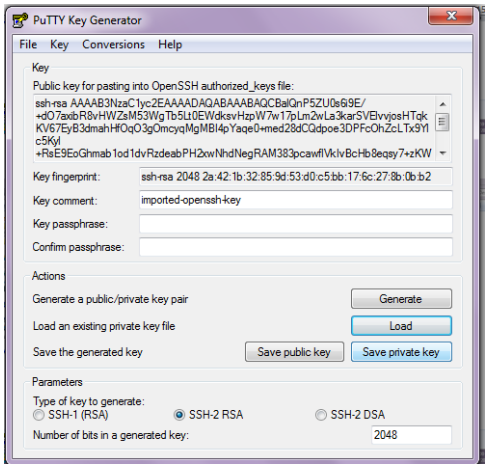

Figure 33 PuttyGen Tool

9. Using Putty tool with the Public IP address generated for the Amazon instance and the corresponding private key (.ppk) is possible to access and manage the cloud service via a SSH (secure) connection (Figure 34) .

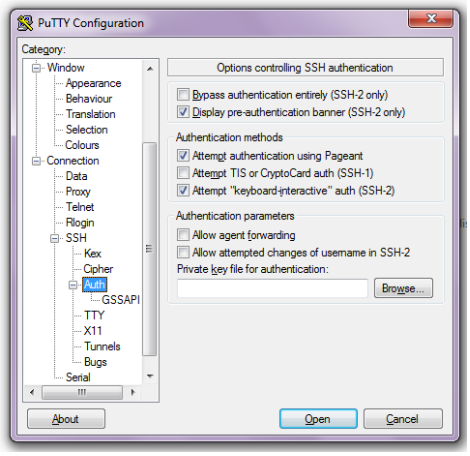

Figure 34 Putty Panel

10. After the SSH connection to the virtual machine is made with root user access (sudo -i), various required updates can be done using the following sequence of commands:
    - *apt-get update* (to update the virtual machine).
    - *apt-get install openjdk-7-jdk* (for JDK Version 7 Installation).
    - *wget download.java.net/glassfish/4.0/release/glassfish-4.0.zip* (for Glassfish Installation).
    - *apt-get install unzip* (for Unzip Installation).
    - *cp glassfish-4.0.zip /opt* (to copy Zip Glassfish to the machine applications folder).
    - *unzip glassfish-4.0.zip* (to Unzip Glassfish).
    - *glassfish4/glassfish/bin/asadmin start* (to start domain Glassfish).
    - *glassfish4/glassfish/bin/asadmin change-admin-password* (to set Glassfish password).
    - *glassfish4/glassfish/bin/asadmin enable-secure-admin* (to enable Glassfish Web Administrator Login).
    - *glassfish4/glassfish/bin/asadmin restart-domain* (to restart domain Glassfish).
    - *apt-get install MySql-server* (for MySql Server installation).
    - *MySql –h localhost –u root –proot* (to connect to MySql Server).
    - *create database BaseName* (to create the database).
    - *use BaseName* (to mount the database).
    - exit (copy databasesScript and exit MySql).
    - *wgethttp://cdn.MySql.com/Downloads/Connector-J/MySql-connector-java-5.1.36.zip* (download MySql Connector – Java).
    - *glassfish4/glassfish/bin/asadmin start* (restart Glassfish).
    - *unzip MySql-connector-java-5.1.36.zip* (Unzip Conector MySql – Java).
    - *glassfish4/glassfish/bin/asadmin asadmin restart-domain* (restart domain Glassfish).

Once the machine has been configured correctly, access *http://34.209.91.214:4848/* for managing Glassfish via a web administrator interface. As a final step, we enter the "Applications" module in the "deploy" option and load the application's .war file so that it is deployed to the glassfish server. In Figure 35 we can see that the application has been

deployed correctly and we can access it using the following link: *http://34.209.91.214:8080/DecisionMakerTool-1.0.0-SNAPSHOT.*

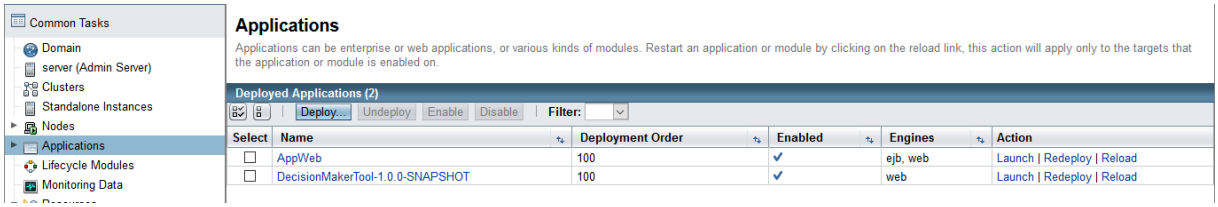

Figure 35 Glassfish Console

5.7 Editing Ontology in Protégé

The ontology created by C3-SEC (described in previous sections) is a key component of the decision aiding software proposed in this thesis, which can be edited and extended with specialized, corporations custom knowledge provided by the cyber security expert using the Protégé ontology editor. The ontology is initially created by C3-SEC and placed in the application directory in the resources folder as "OntologyNexpose.owl". This file can be edited and updated by the CISO with the Protégé editor, for instance to add new attributes or environment variables to allow the calculation of the risk of the organization assets. Protégé editor version 5.00 was used in this thesis to edit the OWL ontology (Figure 36). In chapter 6 (Case Study) an example of the advantages of (ontology level) knowledge provided by the cyber security expert using Protégé is presented.

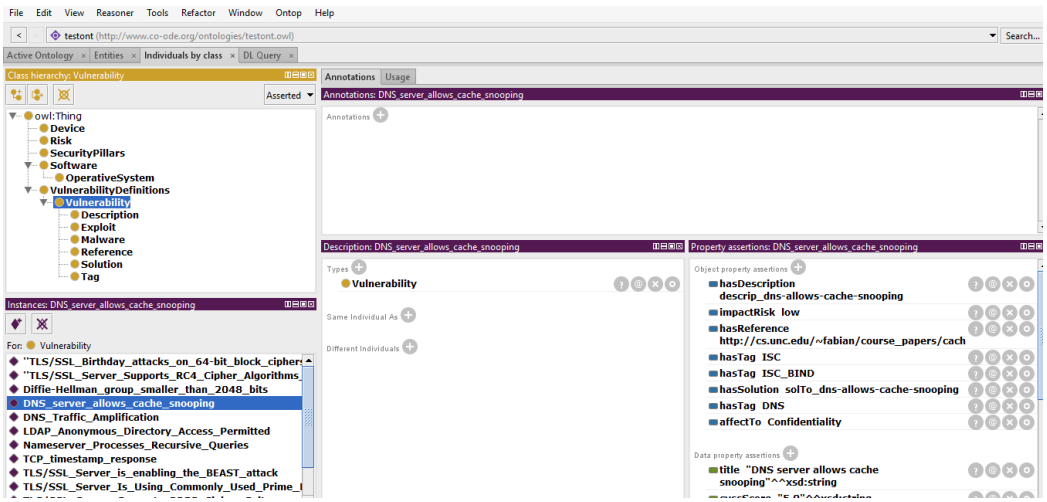

Figure 36 View of the Ontology in Protégé

Figures 37 to 41 show the user-friendly interface of Protégé to add new individuals, data and object properties, and axioms/rules, respectively.

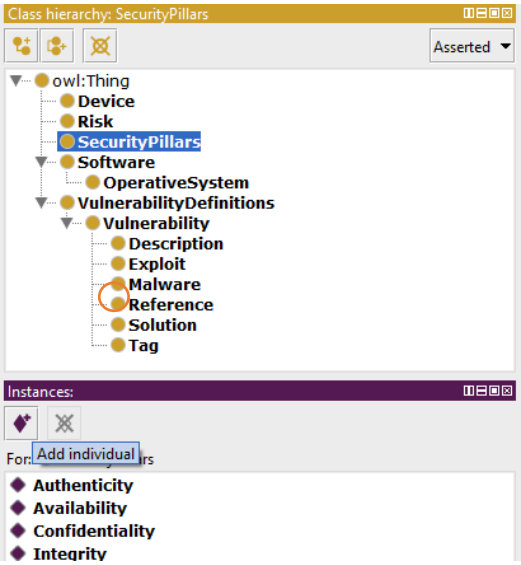

Figure 37 Adding individuals in Protégé

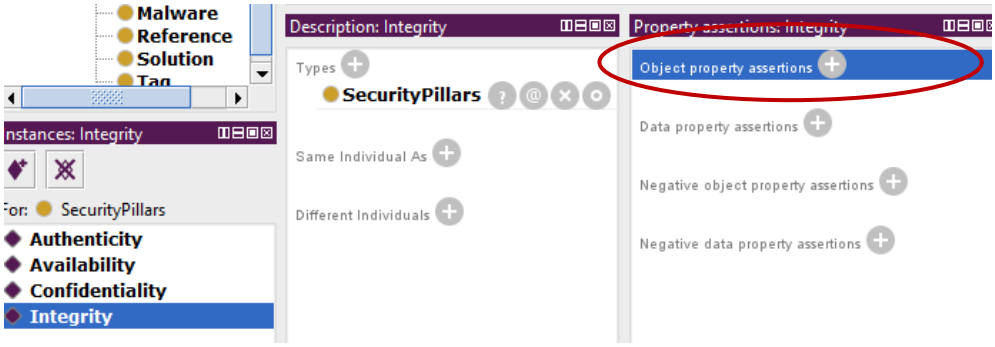

Figure 38 Adding Object Properties in Protégé

Figure 39 Adding Data Properties in Protégé

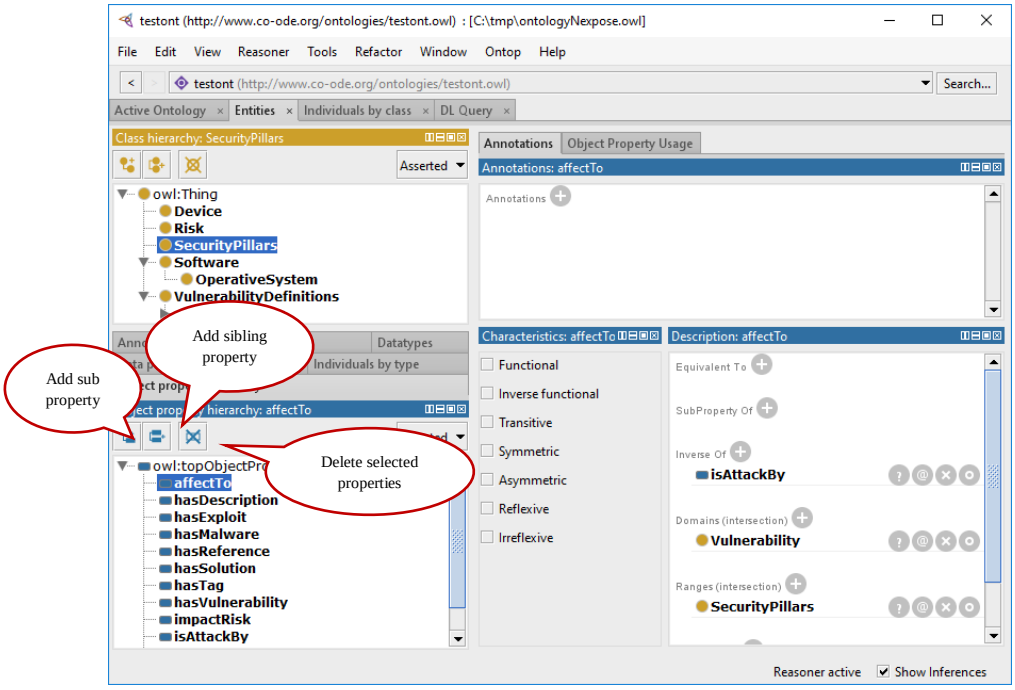

Figure 40 Adding New Object Property in Protégé

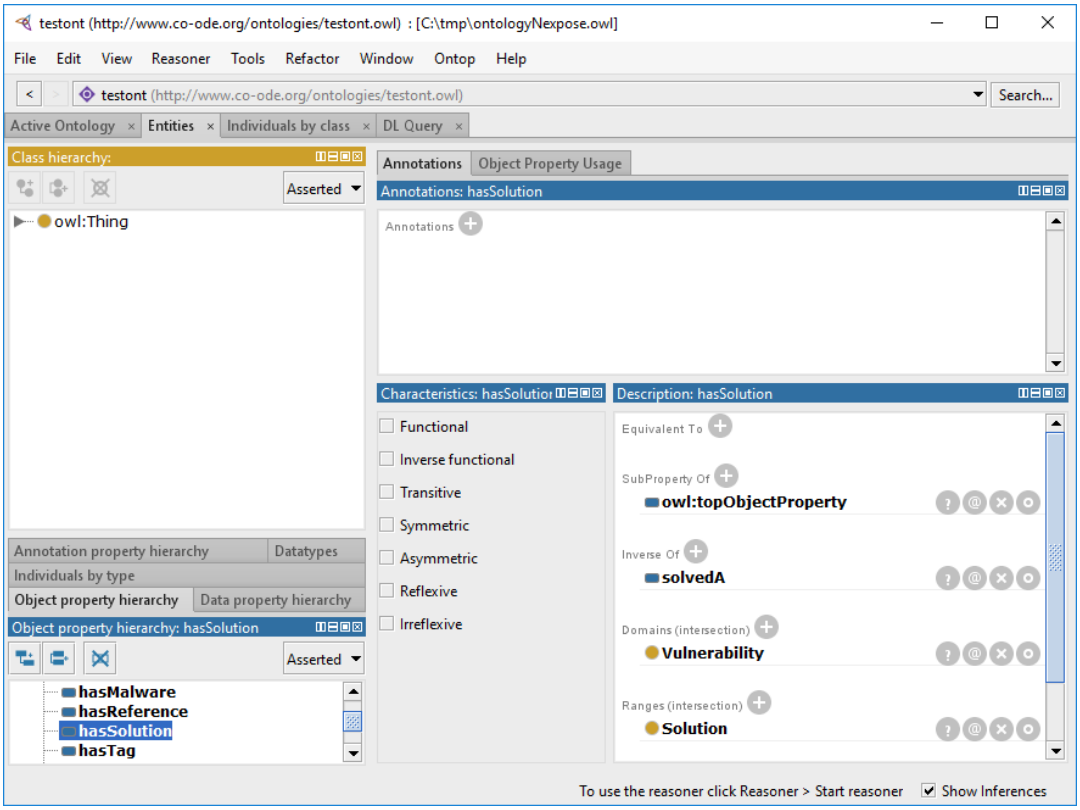

Figure 41 Characteristics Object Property in Protégé

Case Study

The case study presented in this thesis was carried out at the "Research Center of the Department of Computer Science of Universidad de las Fuerzas Armadas ESPE" in Ecuador. Nexpose was used to carry out a scan in the research center ICT infrastructure that allowed to analyze possible threats and to perform cyber security risk assessment, followed by a C3-SEC analysis.

6.1 Case Study Nexpose Cyber Security Risk Analysis

As described in previous sections, Nexpose offers the possibility to calculate risk using different strategies adjusted to the organization's environment, helping to prioritize the vulnerabilities that need to be addressed first. The study is focused on the comparison of different risk assessment strategies applied within the same case study. Table 15 shows the risk calculated by Nexpose, with a total of 49 vulnerabilities found, not considering the criticality factor (CVSS environmental metrics).

Strategy	Risk Score Original
RealRisk	17,920
TemporalPlus	48,048
Temporal	43,227
Weighted	10.0
PCI ASV 2.0 Risk	5.0

Table 15 Nexpose Risk Scores

The criticality factor shows the importance of an asset or its impact on business. In Nexpose this is identified by the "Criticality Tag". Each criticality tag has an associated risk score modifier. The listed risk modifiers will be included in asset risk score calculations when "Risk Score Adjustment" is enabled. These values can be adjusted according to the specific needs of the business. Figure 42 shows Nexpose default values form adopted for the case study.

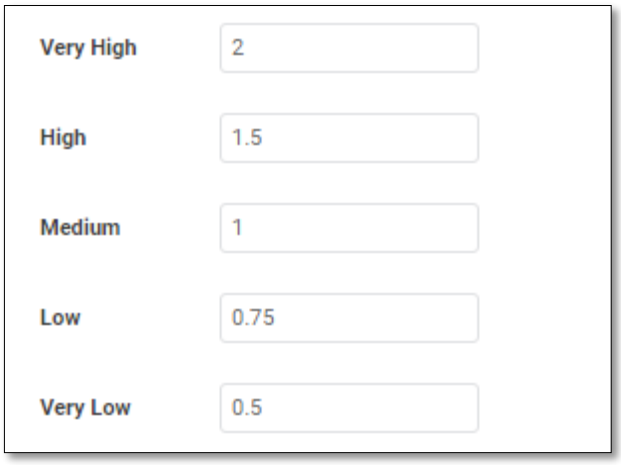

Figure 42 Risk Score Adjustment

In Nexpose, the risk score is applied to a site (asset or collection of assets that are targeted for a scan) or asset group. The calculation used to determine the risk for the entire site or group depends on the risk strategy. In addition, the criticality gets applied to each asset and the total risk score for the group is calculated based upon the individual asset risk scores. "To calculate the risk score for an individual asset, Nexpose uses the algorithm corresponding to the selected risk strategy. If 'Risk Score Adjustment' is set and the asset has a criticality tag applied, the application then multiplies the risk score determined by the risk strategy by the modifier specified for that criticality tag" [93]. The values presented in Table 16 were applied to a site with an asset (server), in each column can be observed the difference between the risk scores with respect to the applied criticality tag (see Figure 42) and the selected strategy. In case of having more than one asset to be compared, the asset with the highest risk score will have higher priority.

	Criticality				
Strategy	Very High	High	Normal	Low	Very Low
RealRisk	35,841	26,881	17,920	13,440	8,960
TemporalPlus	96,096	72,072	48,048	36,036	24,024
Temporal	86,454	64,840	43,227	32,420	21,613
Weighted	20.1	15.0	10.0	7.5	5.0
PCI ASV 2.0 Risk	10.0	7.5	5.0	3.8	2.5

Table 16 Risk Score Comparison

Complementarily, Figure 43 shows the report generated by Nexpose about the vulnerabilities found. This report allows the identification of vulnerabilities that may affect the organization most critically based on some most relevant criteria such as CVSS score, according to risk strategy and severity.

EXCLUDE RECALL RESUBMIT Total Vulnerabilities Selected: 0 of 49

Title	CVSS	Risk	Published On	Modified On	Severity	Instances	Exceptions	Solution
X.509 Certificate Subject CN Does Not Match the Entity Name	7.1	5.0	Fri Aug 03 2007	Wed Jan 28 2015	Severe	2	Exclude	
SMB signing disabled	7.3	5.0	Mon Nov 01 2004	Thu Jul 12 2012	Severe	2	Exclude	
PHP Vulnerability: CVE-2016-10160	7.5	5.0	Tue Jan 24 2017	Mon Feb 27 2017	Critical	2	Exclude	
PHP Vulnerability: CVE-2016-9936	7.5	5.0	Wed Jan 04 2017	Tue Jan 10 2017	Critical	2	Exclude	
PHP Vulnerability: CVE-2016-9935	7.5	5.0	Wed Jan 04 2017	Tue Jan 10 2017	Critical	2	Exclude	

Figure 43 Report Vulnerabilities Found

In addition, Nexpose offers different graphical reports to gain insights into what is happening in the organization environment as well as to understand how the vulnerabilities are affecting and jeopardizing the company's assets. One of the useful reports for an organization's cyber security team is the “Vulnerabilities by CVSS score”, which shows the amount of vulnerabilities group by CVSS score ranges. Figure 44 shows a Nexpose graphical report from the case study.

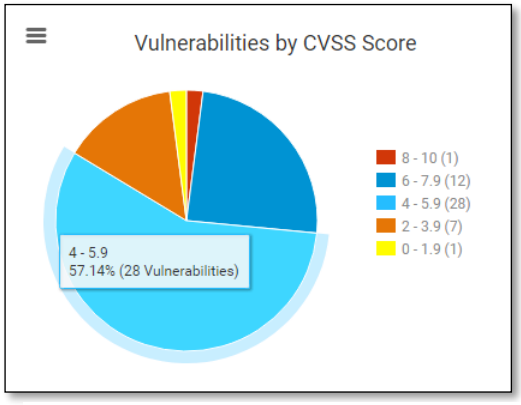

Figure 44 Vulnerabilities by CVSS Score

6.2 Case Study C3-SEC Cyber Security Risk Analysis

In this section, the C3-SEC features which extend Nexpose risk analysis possibilities are highlighted. For this case study, the scanning of two assets of the institution were taken so that two XML files of the Nexpose tool were generated and load into C3-SEC through the integration mechanisms developed for C3-SEC (presented in previous sections of the thesis).

The first file was loaded with the option “Create new ontology” as shown in Figure 45 because this was the first time the information about the IT infrastructure of the institution was loaded into C3-SEC.

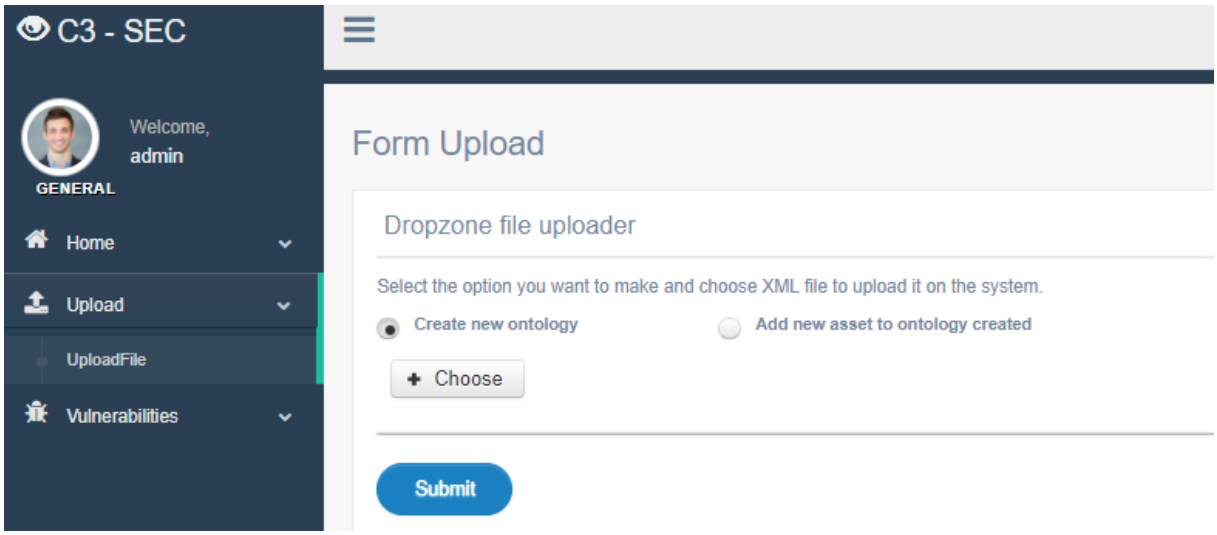

Figure 45 Form Upload C3-SEC

In the “Dashboard” module shown in Figure 46, we can see the cyber security risk analysis generated by C3-SEC. The graphics that can be observed in Figure 46 show clearly the vulnerabilities by category as well as the number of vulnerabilities that affect each pillar, being these reports a specific characteristic (cyber security risk analysis extension) of C3-SEC.

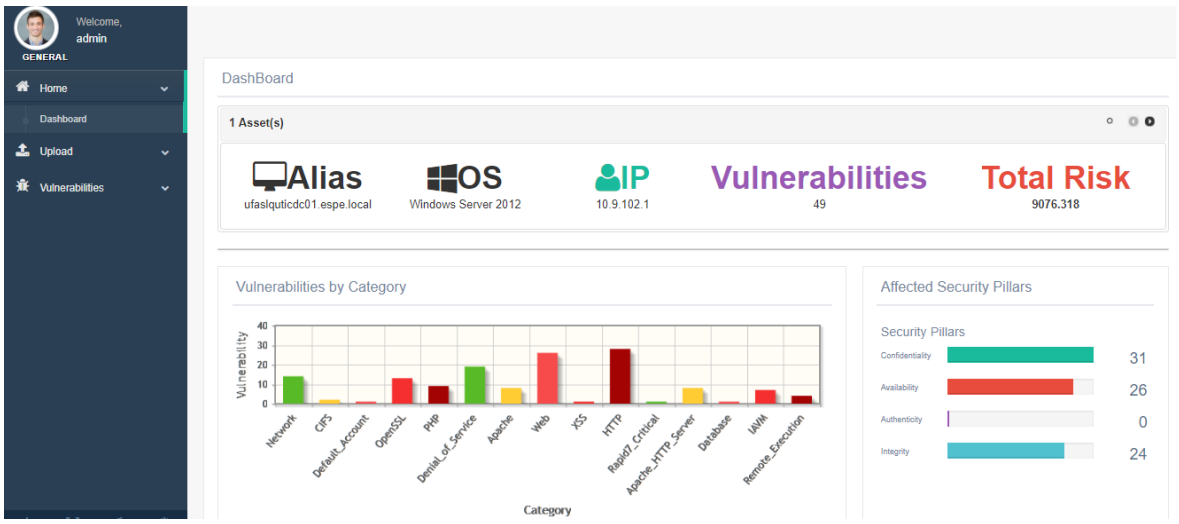

Figure 46 C3-SEC Report (One asset)

To load the second file, the option “Add new asset to ontology created” of the Upload Module was chosen. This allows adding a new institution asset to the ontology already created. The results presented by C3-SEC relate the information of each of the assets through a query made to the ontology to generate the report that shows the vulnerabilities that affect the company in the different security pillars. Figure 47 shows a view of the case study.

Figure 47 C3-SEC Result Report

For cyber security risk analysis with C3-SEC, the institution security expert added ontology level knowledge related to the location of the assets using the Protégé ontology editor. C3-SEC is able to use the knowledge (assertions and rules) provided by the expert to reason (by the means of description logics inference) about assets, cyber security properties and corresponding cyber security risk computation. In this case study, the expert added the following knowledge to the ontology:

- Creation of a location individual under the ontology class “Location”, which represents the location in which the asset is located.
- Insertion of the object property “isLocationOf” that allows to relate the location to the asset.
- Insertion of the data property “isOfEasyPhysicalAccess” which establishes the facility with which an asset can be accessed in a certain location.
- Once the knowledge has been added to the ontology by the expert, the RDF / XML file must be saved and named “OntologyNexpose.owl” so that it can be read by C3-SEC.

The knowledge inserted by the expert in the ontology in this example allows C3-SEC reasoning in situations such as: an asset located in a location that was assigned easy physical access is also of easy physical access and therefore is subject of security risks, namely with reflexes on the availability pillar. In the current case study, the location of the Research Center was added with the name “C.Investigacion”, being the place where the server of the case study is located. To relate the asset to the location, the object property “isLocationOf” was added and linked to the individual of the “Device” class representing the asset. Protégé autocompletion features creates an user friendly user interface which automatically loads the names of individuals from the ontology, then it is only necessary for the user to enter the first few letters of the asset name or search from the list of individuals.

Finally, to specify the ease of access of the location the user must enter the value “yes” in case the location is easily accessible or “no”, otherwise, in the data property “isOfEasyPhysicalAccess”. In the case study, “yes” was selected stating that this location is of easy physical access, having security reflexes on all assets located at “C.Investigacion” and in the subsequent C3-SEC cyber security risk analysis. Figure 48 shows the (SWRL) rule stating that (individuals) ICT equipment located in a (physically) easily accessible location are also of easy physical access and have therefore its security properties (e.g. availability pillar) affected. This rule is an example of the reasoning support the OWL ontology provides to C3-SEC. Figure 49 shows the knowledge added to the ontology, involving the practical application of this rule in the case study adopted in the thesis.

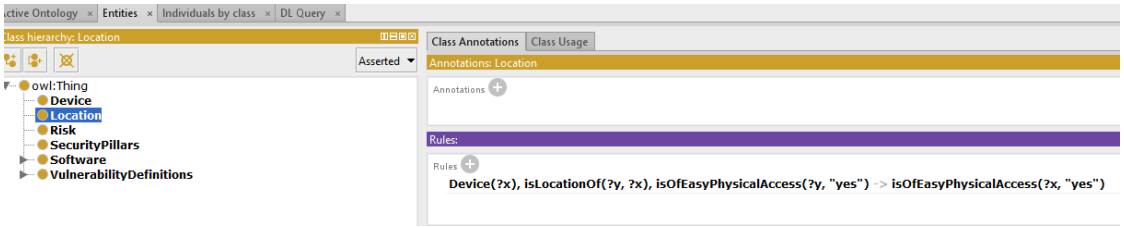

Figure 48 Rule of Location-Related to Ontology Properties Characteristics

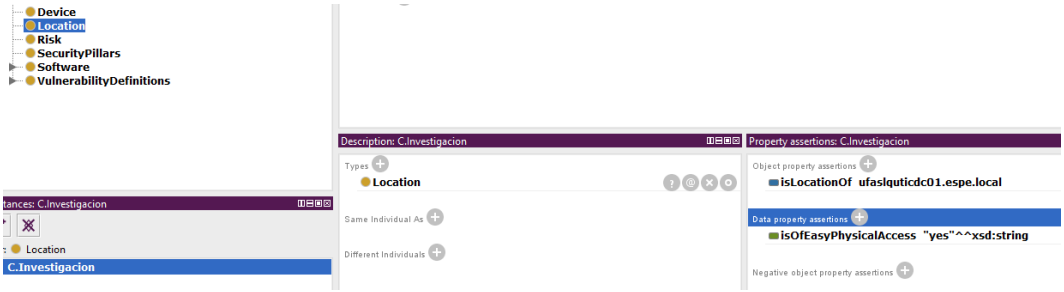

Figure 49 Adding Properties about Location in Protégé

When the dashboard module is updated, as shown in the Figure 50, we can see that the number of vulnerabilities affected has changed when compared to the previous result (see Figure 47). This is due to the inference made by the Pellet reasoner in the ontology, followed by a query in C3-SEC to obtain the vulnerabilities that affect the institution. The transitive characteristic of the "isOfEasyAccess" property turns all assets located in an easy access location of easy access, affecting its security properties (e.g. availability) and corresponding C3-SEC cyber security risk analysis reports. The query to obtain the new "Availability" security pillar indicator, used to generate the results shown in Figure 50, is as follows:

```
Vulnerability that affectTo some (isOfEasyPhysicalAccess value
"yes") or affectTo value Availability.
```

This query allows to find all the vulnerabilities that affect an asset in a certain location and that are also easily accessible, which directly affects the Availability. The rules added to the ontology allow to relate each individual of the classes "Device", "Vulnerability" and "Location" through the inverse object properties to each other such as "affectTo" and "hasVulnerability" for the classes "Device" and "Vulnerability" and the "isOfEasyPhysicalAccess" datatype property for the Device and Location classes. The result will be that all the vulnerabilities complying with this condition are added to the previous calculation (Nexpose calculation without inference support), taking into account eventual redundant vulnerabilities found with respect to the security pillar "Availability".

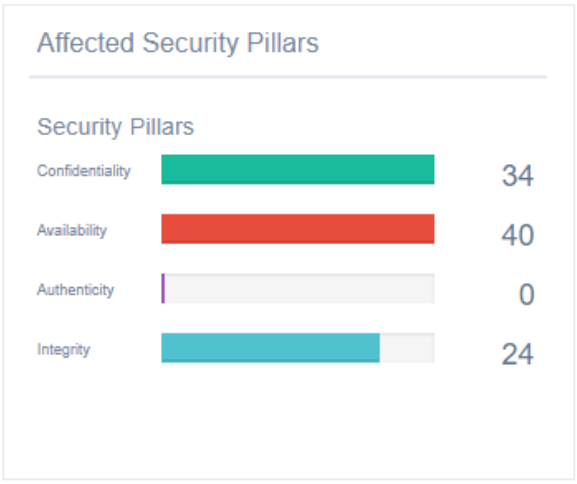

Figure 50 Report Affected Security Pillars

This page was intentionally left blank

Conclusions

This thesis was motivated by the fast growth of cyber security threats and incidents widely documented by scientific, technical, business and governmental entities and authorities.
The thesis contribution is targeted to corporation level cyber security risk management and follows a context-aware systems approach to provide corporations cyber security situation awareness. C3-SEC addresses a cyber security scope and features that are missing in state of the art existing tools for this domain. It is based on a contextual knowledge representation approach that allows to identify, define, develop and apply a simple, comprehensive security analysis and assurance of business continuity.

The context-aware systems reference model (perception, comprehension, projection and decision/action layers) lead the analysis, design, development and implementation of C3-SEC project. For the first level (perception), a comparative analysis was carried out between the main cyber security tools for scanning a company's technological infrastructure assets and vulnerabilities: Nessus Home, Saint8, Nmap (ZenMap), eEye Retina, GFI LANguard, nCircle® IP360, Security System, Analyzer 2.0 Beta, OpenVas, Nexpose, QualysGuard. A comparison framework and metrics (operating system, support for cyber security data exchange standards, etc.) was defined an applied, resulting in the selection of Nexpose as the best option to be integrated as a component of C3-SEC context-aware systems model.
At the comprehension level, an OWL ontology was designed taking into account the data and semantic models of Nexpose. Complementary knowledge (e.g. description logics rules) was added to the ontology for C3-SEC decision making support (e.g. RDF/OWL queries). None of the state-of-the-art tools studied addressed the comprehension layer by the means of (OWL) semantic knowledge representation and knowledge management. C3-SEC is proposed in the current thesis to fill this gap and to take advantage of all the benefits made available by semantic web standards and technologies.

For the projection level, a comparative analysis was performed between the scanning tools in relation to the risk analysis features. Based on this comparison, missing features were identified and the lack of assets physical location consideration for cyber security risk

analysis purposes was selected as the feature to introduce and highlight in C3-SEC. The influence of assets physical location in the security pillar “Availability” was illustrated in the case study adopted in the thesis to validate C3-SEC approach and software.
As future work, it is intended to incorporate new risk management strategies in C3-SEC and simplify (make more transparent) Nexpose and C3-SEC integration, adding presentation layer integration mechanisms, single sign on features, unified configuration files and graphical interfaces, etc.

C3-SEC revealed in the thesis case study to provide valuable help for cyber security decision-makers to make informed decisions, by combining international authorities cyber security technical data about vulnerabilities and corporations experts specific knowledge, and suggesting the best course of action to mitigate vulnerabilities and ensure business continuity in today's hostile cyber environment.

References

[1] A. Oltramari, L. Faith, C. R. J. Walls and P. McDaniel, "Building an Ontology of Cyber Security," November 2014. [Online]. Available: http://ceur-ws.org/vol-1304/stids2014_t08_oltramarietal.pdf.

[2] Y. Cheng, J. Deng, J. Li, S. DeLoach, A. Singhal and X. Ou, "Metrics of Security," 2014. [Online]. Available: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=917850.

[3] NIST, "About CCE - Archive," 22 March 2013. [Online]. Available: https://cce.mitre.org/about/.

[4] NIST, "About NIST," 25 August 2016. [Online]. Available: https://www.nist.gov/about-nist.

[5] Nist, "The Security Content Automation Protocol (SCAP)," 16 December 2016. [Online]. Available: https://scap.nist.gov/.

[6] The MITRE Corporation, "CPE Common Platform Enumeration," 28 November 2014. [Online]. Available: https://cpe.mitre.org/.

[7] NIST, "Computer Security Resource Center," 20 March 2017. [Online]. Available: https://nvd.nist.gov/.

[8] G. Lyon, Nmap Network Scanning, 2009.

[9] The MITRE Corporation, "CVE Common Vulnerabilities and Exposures," 23 February 2017. [Online]. Available: https://cve.mitre.org/.

[10] FIRST, "Common Vulnerability Scoring System v3.0: Specification Document," 2017. [Online]. Available: https://www.first.org/cvss/specification-document.

[11] FIRST.Org, Inc, "The Common Vulnerability Scoring System (CVSS)v2," 2007. [Online]. Available: https://www.first.org/cvss/cvss_basic-2.0.pdf.

[12] FIRST, "CVSS Frequently Asked Questions," 2017. [Online]. Available: https://www.first.org/cvss/v2/faq.

[13] E. P. Maurice-Oracle, "Understanding the Common Vulnerability Scoring System (CVSS)," 5 April 2011. [Online]. Available: https://blogs.oracle.com/security/entry/understanding_the_common_vulne_2.

[14] The Mitre Corporation, "OVAL Open Vulnerability and Assessment Language," 9 February 2016. [Online]. Available: https://oval.mitre.org/.

[15] Mitre, "About OVAL," 13 May 2014. [Online]. Available: https://oval.mitre.org/about/.

[16] G. Lyon, "SecTools.Org: Top 125 Network Security Tools," 2011. [Online]. Available: http://sectools.org/.

[17] tenable, "Nessus Home," 2017. [Online]. Available: https://www.tenable.com/products/nessus-home.

[18] IATAC, "Vulnerability Assessment," 2 May 2011. [Online]. Available: https://www.csiac.org/wp-content/uploads/2016/02/vulnerability_assessment.pdf.

[19] BeyondTrust, "Retina Network Vulnerability Scanner," 2017. [Online]. Available: https://www.beyondtrust.com/products/retina-network-security-scanner/.

[20] W3C, "Extensible Markup Language (XML)," 11 October 2016. [Online]. Available: https://www.w3.org/XML/.

[21] Edoceo, "Comma Separated Values (CSV) Standard File Format," 2016. [Online]. Available: http://edoceo.com/utilitas/csv-file-format.

[22] D. Vitale, "Doug Vitale Tech Blog," 13 February 2012. [Online]. Available: https://dougvitale.wordpress.com/2012/02/13/retina-network-security-scanner.

[23] GFI Software, "Patch management for operating systems," 2017. [Online]. Available: http://www.gfi.com/products-and-solutions/network-security-solutions/gfi-languard/specifications/patch-management-for-operating-systems.

[24] nCircle, "nCircle IP360TM," 2010. [Online]. Available: http://www.base-camp.cc/wp-content/download/ncircle/nCircle-DS-IP360-1004-05.pdf.

[25] SSA, «SSA - Security System Analyzer 2.0,» 2010. [En línea]. Available: https://code.google.com/archive/p/ssa.

[26] OpenVAS , "About OpenVAS Software," [Online]. Available: http://www.openvas.org/software.html. [Accessed 2017].

[27] Rapid7, "The Evolution of Nexpose: Get a Free InsightVM Trial," 2017. [Online]. Available: https://www.rapid7.com/products/nexpose/download.

[28] Rapid7, "Rapid7 Vulnerability Scanner Tools," 2017. [Online]. Available: https://www.rapid7.com/products/nexpose/download/editions/.

[29] Rapid7Community, "Driving Risk Reduction through RealContext™ in Nexpose 5.9," 26 March 2014. [Online]. Available: https://community.rapid7.com/community/nexpose/blog/2014/03/26/driving-risk-prioritization-through-realcontext-in-nexpose-59.

[30] Qualys, Inc., "QualysGuard® is the Qualys Cloud Platform," 2017. [Online]. Available: https://www.qualys.com/qualysguard.

[31] Qualys, Inc., "The Market Leader," 2017. [Online]. Available: https://www.qualys.com/suite/vulnerability-management.

[32] Qualys, Inc., "Features," 2017. [Online]. Available: https://www.qualys.com/suite/vulnerability-management/features.

[33] W3C Semantic Web, "Web Ontology Language (OWL)," 11 December 2013. [Online]. Available: https://www.w3.org/OWL/.

[34] Stanford Center for Biomedical Informatics Research, "Protégé," 2016. [Online]. Available: http://protege.stanford.edu/.

[35] D. Man, "Ontologies in Computer Science," June 2013. [Online]. Available: http://www.math.ubbcluj.ro/~didactica/pdfs/2013/didmath2013-06.pdf.

[36] Y. I. B.-F. V. T. H. J. N. E. M. Li L., "Building and Using an Ontology of Preference-Based Multiobjective Evolutionary Algorithms. In: Trautmann H. et al. (eds) Evolutionary Multi-Criterion Optimization," in *EMO 2017 9th International Conference on Evolutionary Multi-Criterion Optimization, vol 10173*, Münster, Germany, 2017.

[37] N. F. Noy and D. L, "Ontology Development 101: A Guide to Creating Your First Ontology," 2001. [Online]. Available: http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

[38] W3C, "Resource Description Framework (RDF)," 25 February 2014. [Online]. Available: https://www.w3.org/RDF/.

[39] A. Singhal and D. Wijesekera, "Ontologies for Modeling Enterprise Level Security Metrics," in *CSIIRW '10 Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, Tennessee, USA, 2010.

[40] D. Milicevic and M. Goeken, "Ontology-based Evaluation of ISO 27001," 11 August 2017.

[41] S. Fenz and A. Ekelhart, "Formalizing Information Security Knowledge," in *ASIACCS'09*, Sydney, Australia, 2009.

[42] T. Pereira and H. Santos, "An Ontology Based Approach to Information Security," in *Metadata and Semantic Research*, Springer, 2009, pp. 183-192.

[43] A. Souag, C. Salinesi, R. Mazo and I. Comyn-Wattiau, "A Security Ontology for Security Requirements Elicitation," 15 April 2016.

[44] "Ontologies for Security Requirements: A Literature Survey and Classi," 19 Junho 2012.

[45] A. Souag, "Towards a new generation of security requirements definition methodology using ontologies," 2012. [Online]. Available: http://ceur-ws.org/Vol-863/paper3.pdf.

[46] N. Yahia, S. A. Mokhtar and A. Ahmed, "Automatic Generation of OWL Ontology from XML Data Source," *International Journal of Computer Science Issues,* vol. 9, no. 2, 2012.

[47] T. Rodrigues, P. Rosa and J. Cardoso, "Mapping XML to existing OWL ontologies," *ResearchGate,* July 2008.

[48] R. Ghawi, "Ontology-based cooperation of information systems :contributions to database-to-ontology mapping and XML-to-ontology mapping," 24 January 2011.

[49] R. Ghawi and N. Cullot, "Building Ontologies from XML Data Sources," in *Database and Expert Systems Application, 2009. DEXA '09. 20th International Workshop on*, Linz, Austria, 2009.

[50] The Apache Software Foundation,, "Apache Jena," 2017. [Online]. Available: https://jena.apache.org/tutorials/index.html.

[51] W3C, "TopBraid," 9 January 2011. [Online]. Available: https://www.w3.org/2001/sw/wiki/TopBraid.

[52] TopQuadrant, Inc., "Living in The XML and OWL World – Comprehensive Transformations of XML Schemas and XML Data to RDF/OWL," 29 September 2011. [Online]. Available: http://www.topquadrant.com/2011/09/29/living-in-the-xml-and-owl-world-comprehensive-transformations-of-xml-schemas-and-xml-data-to-rdfowl/.

[53] Rapid7Community, "Report_XML_Export_Schema_2.0.zip," 3 July 2013. [Online]. Available: https://community.rapid7.com/docs/DOC-2148. [Accessed 2017].

[54] N. Yahia, S. A. Mokhtar and A. Ahmed, "Automatic Generation of OWL Ontology from XML Data Source," *IJCSI International Journal of Computer Science Issues,* vol. 9, no. 2, pp. 77-83, 2012.

[55] Tenable, "Nessus 6.9 User Guide," 8 May 2017. [Online]. Available: https://docs.tenable.com/nessus/6_9/Content/Resources/PDF/Nessus_6_9.pdf.

[56] Tenable Community, "Risk Factor," 2010 July 2010. [Online]. Available: https://community.tenable.com/thread/2567.

[57] SAINT Corporation, "Asset Management," 2017. [Online]. Available: http://www.saintcorporation.com/products/asset-management.

[58] Rapid7, "Leveraging Security Risk Intelligence," July 2011. [Online]. Available: https://information.rapid7.com/rs/495-KNT-277/images/rapid7-whitepaper-leveraging-security-risk-intelligence.pdf.

[59] BeyondTrust, "Retina CS Enterprise Vulnerability Management," February 2017. [Online]. Available: https://www.beyondtrust.com/wp-content/uploads/ds-retina-cs.pdf?1486141809.

[60] BeyondTrust, "Retina Enterprise Vulnerability Management Solutions," 2017. [Online]. Available: https://www.beyondtrust.com/wp-content/uploads/new-features-retina-cs-5-7-rnss-5-23.pdf?1448922701.

[61] GFISoftware, "GFI LanGuard," 2017. [Online]. Available: http://www.gfi.com/products-and-solutions/network-security-solutions/gfi-languard.

[62] Bitwork Technologies, "Network Security Scanner and Patch Management GFI LANguard," 2012. [Online]. Available: http://www.bitworktech.com/our-solutions/information-security/patch--vulnerability-management.

[63] Insight Technology Solutions APS, "GFI LANguard," 2017. [Online]. Available: http://dk.insight.com/shop/gfi/languard.

[64] SANS™ Institute, "SANS Continuous Monitoring Poster," 2016 . [Online]. Available: https://www.sans.org/media/critical-security-controls/SANS_CSC_Poster.pdf.

[65] Tripwire, "Tripwire IP360 8.0 Datasheet," 2017. [Online]. Available: https://www.tripwire.com/products/tripwire-ip360/tripwire-ip360-datasheet-register/.

[66] Tripwire, "Prioritize Vulnerabilities, Manage Your Risk," 2017. [Online]. Available: https://www.tripwire.com/solutions/vulnerability-and-risk-management/.

[67] BrotherSoft, "Security System Analyzer 2.0 Beta002," 2014. [Online]. Available: Security System Analyzer 2.0 Beta002.

[68] Google Code, "SSA - Security System Analyzer 2.0," 2010. [Online]. Available: https://code.google.com/archive/p/ssa.

[69] NIST, "XCCDF - The Extensible Configuration Checklist Description Format," 2017. [Online]. Available: https://scap.nist.gov/specifications/xccdf/.

[70] M. Á. Mendoza, "Cómo utilizar OpenVAS para la evaluación de vulnerabilidades," ESET, 18 November 2014. [Online]. Available: https://www.welivesecurity.com/la-es/2014/11/18/como-utilizar-openvas-evaluacion-vulnerabilidades/.

[71] Qualys, Inc., "Vulnerability Management," 2017. [Online]. Available: https://www.qualys.com/suite/vulnerability-management/features/#prioritize..

[72] Rapid7Community , "Nexpose User's Guide (English)," 22 March 2017. [Online]. Available: https://community.rapid7.com/docs/DOC-1387?cs=web.

[73] Rapid7Community, "Working with risk strategies to analyze threats," 14 December 2016. [Online]. Available: https://help.rapid7.com/nexpose/en-us/Files/Working_with_risk_strategies_to_analyze_threats.html.

[74] G. Roldán, M. Almache, C. Rabadao, I. Yevseyeva and V. Fernandes, "A Decision Support System for Corporations Cybersecurity Management," in *12th Iberian Conference on Information Systems and Technologies*, Lisboa, 2017.

[75] Rapid7Community, "PCI, CVSS, & risk scoring frequently asked questions," 14 Decembber 2016. [Online]. Available: https://help.rapid7.com/nexpose/en-us/Files/Risk_scoring_FAQ.html.

[76] Rapid7Community, "Creating custom NeXpose risk scoring strategies," 10 April 2011. [Online]. Available: https://community.rapid7.com/docs/DOC-1136.

[77] PCI Security Standards Council LLC, "Payment Card Industry (PCI) Approved Scanning Vendors," May 2013. [Online]. Available: https://www.pcisecuritystandards.org/documents/ASV_Program_Guide_v2.pdf.

[78] D. Minguez Sanz and E. J. García Morales, "Metodologías para el Desarrollo de Aplicaciones Web: UWE," 2011. [Online]. Available: https://jorgeportella.files.wordpress.com/2011/11/analisis-diseo-y-desarrollodeaplicacionesweb.pdf.

[79] Oracle, "Java EE at a Glance," 2017. [Online]. Available: http://www.oracle.com/technetwork/java/javaee/overview/index.html.

[80] O. C. Chris Schalk, "Introduction to Javaserver Faces - What is JSF?," April 2005. [Online]. Available: http://www.oracle.com/technetwork/topics/index-090910.html.

[81] "Primefaces," 2017. [Online]. Available: https://www.primefaces.org/.

[82] GrayGrids Inc. , "Gentelella – Free Bootstrap Admin Template," 2017. [Online]. Available: https://graygrids.com/item/gentelella-free-bootstrap-admin-template/.

[83] G. Roldan and L. A. M. Chandi, "Implementación de un Aplicactivo Web como Servicio SaaS, bajo un infraestructura en la nube IaaS, para la Cooperativa San Vicente del Sur - Matriz," 2015. [Online]. Available: http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/10779/T-ESPE-049264.pdf?sequence=1&isAllowed=y.

[84] Oracle, "Oracle," julio 2015. [Online]. Available: https://www.java.com/es/download/faq/develop.xml .

[85] Amazon, "Amazon," Julio 2015. [Online]. Available: http://aws.amazon.com.

[86] T. Simon, "PuTTY: A Free Telnet/SSH Client," 21 Junio 2015. [Online]. Available: http://www.chiark.greenend.org.uk/~sgtatham/putty/.

[87] R. H. Enterprise, "Red Hat Enterprise Linux 4: Manual de referencia," 2015. [Online]. Available: http://www.gb.nrao.edu/pubcomputing/redhatELWS4/RH-DOCS/rhel-rg-es-4/ch-ssh.html.

[88] Amazon, "Acerca de AWS," julio 2015. [Online]. Available: http://aws.amazon.com/es/about-aws/.

[89] GlassFish, "GlassFish Server Open Source Edition - 3.1.2," 2015. [Online]. Available: https://glassfish.java.net/downloads/3.1.2-final.html#v3_licensing.

[90] F. Fernández and Y. Muñoz, "JDBC," 2015. [Online]. Available: http://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/jfernand/.

[91] Amazon, "Amazon Web Services," 2015. [Online]. Available: http://aws.amazon.com/es/.

[92] PuTTY, "Download PuTTY," 2017. [Online]. Available: http://www.putty.org.

[93] Rapid7Community, "Adjusting risk with criticality," 9 August 2017. [Online]. Available: https://help.rapid7.com/insightvm/en-us/index.html#Files/Adjusting_risk_with_criticality.html#criticality_strategy_interaction.

This page was intentionally left blank

Appendix 1 – Base Ontology

Following the OWL file made as the base ontology for the C3-SEC application.

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.co-ode.org/ontologies/testont.owl#"
     xml:base="http://www.co-ode.org/ontologies/testont.owl"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <owl:Ontology rdf:about="http://www.co-
ode.org/ontologies/testont.owl"/>

    <!--
////////////////////////////////////////////////////////////////////////
/////////////
    //
    // Object Properties
    //
////////////////////////////////////////////////////////////////////////
/////////////
     -->

    <!-- http://www.co-ode.org/ontologies/testont.owl#affectTo -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#affectTo">
        <owl:inverseOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#hasVulnerability"/>
        <owl:inverseOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#isAttackBy"/>
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#SecurityPillars"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#hasDescription -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#hasDescription">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Description"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#hasExploit -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#hasExploit">
```

```
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Exploit"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#hasMalware -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#hasMalware">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Malware"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#hasReference -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#hasReference">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Reference"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#hasSolution -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#hasSolution">
        <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
        <owl:inverseOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#solvedA"/>
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Solution"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#hasTag -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#hasTag">
        <owl:inverseOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#identifyTo"/>
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Tag"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#hasVulnerability --
>
    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#hasVulnerability">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Device"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#VulnerabilityDefinitions"/>
```

```
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#identifyTo -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#identifyTo"/>

    <!-- http://www.co-ode.org/ontologies/testont.owl#isAttackBy -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#isAttackBy"/>

    <!-- http://www.co-ode.org/ontologies/testont.owl#isLocatedIn -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#isLocatedIn">
        <owl:inverseOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#isLocationOf"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#isLocationOf -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#isLocationOf">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Location"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Device"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#runsOnSw -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#runsOnSw">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Device"/>
        <rdfs:range rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Software"/>
    </owl:ObjectProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#solvedA -->

    <owl:ObjectProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#solvedA"/>

    <!--
///////////////////////////////////////////////////////////////////////
//////////////
    //
    // Data properties
    //
///////////////////////////////////////////////////////////////////////
//////////////
     -->

    <!-- http://www.co-ode.org/ontologies/testont.owl#added -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#added">
```

```
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#address -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#address">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Device"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#cvssScore -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#cvssScore">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#cvssVector -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#cvssVector">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#family -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#family">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#OperativeSystem"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#id -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#id">
        <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2002/07/owl#topDataProperty"/>
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#idDevice -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#idDevice">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Device"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#idExploit -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#idExploit">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Exploit"/>
```

```
    </owl:DatatypeProperty>

    <!-- http://www.co-
ode.org/ontologies/testont.owl#isOfEasyPhysicalAccess -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#isOfEasyPhysicalAccess">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Location"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#link -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#link">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Exploit"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#modified -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#modified">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#nameDevice -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#nameDevice">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Device"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#nameReference -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#nameReference">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Reference"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#nameTag -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#nameTag">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Tag"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#paragraph -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#paragraph">
        <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2002/07/owl#topDataProperty"/>
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Solution"/>
    </owl:DatatypeProperty>
```

```
    <!-- http://www.co-ode.org/ontologies/testont.owl#pciSeverety -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#pciSeverety">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range>
            <rdfs:Datatype>
                <owl:onDatatype
rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
                <owl:withRestrictions rdf:parseType="Collection">
                    <rdf:Description>
                        <xsd:minInclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</xsd:minInclusi
ve>
                    </rdf:Description>
                    <rdf:Description>
                        <xsd:maxInclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">5</xsd:maxInclusi
ve>
                    </rdf:Description>
                </owl:withRestrictions>
            </rdfs:Datatype>
        </rdfs:range>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#product -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#product">
        <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2002/07/owl#topDataProperty"/>
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#OperativeSystem"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#published -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#published">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#riskScore -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#riskScore">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#riskScoreDevice -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#riskScoreDevice">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Device"/>
```

```
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#severity -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#severity">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
        <rdfs:range>
            <rdfs:Datatype>
                <owl:onDatatype
rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
                <owl:withRestrictions rdf:parseType="Collection">
                    <rdf:Description>
                        <xsd:minInclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</xsd:minInclusi
ve>
                    </rdf:Description>
                    <rdf:Description>
                        <xsd:maxInclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">10</xsd:maxInclus
ive>
                    </rdf:Description>
                </owl:withRestrictions>
            </rdfs:Datatype>
        </rdfs:range>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#skillLevel -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#skillLevel">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Exploit"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#title -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#title">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#titleExploit -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#titleExploit">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Exploit"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#type -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#type">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Exploit"/>
    </owl:DatatypeProperty>
```

```
    <!-- http://www.co-ode.org/ontologies/testont.owl#valueDescription --
>
    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#valueDescription">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Description"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#valueMalware -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#valueMalware">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Malware"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#valueReference -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#valueReference">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Reference"/>
    </owl:DatatypeProperty>

    <!-- http://www.co-ode.org/ontologies/testont.owl#vendor -->

    <owl:DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/testont.owl#vendor">
        <rdfs:domain rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#OperativeSystem"/>
    </owl:DatatypeProperty>

    <!--
///////////////////////////////////////////////////////////////////////
//////////////
    //
    // Classes
    //
///////////////////////////////////////////////////////////////////////
//////////////
     -->

    <!-- http://www.co-ode.org/ontologies/testont.owl#Description -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Description">
        <rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:Class>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Device -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Device"/>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Exploit -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Exploit">
```

```
        <rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:Class>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Location -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Location"/>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Malware -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Malware">
        <rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:Class>

    <!-- http://www.co-ode.org/ontologies/testont.owl#OperativeSystem -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#OperativeSystem">
        <rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Software"/>
    </owl:Class>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Reference -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Reference">
        <rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:Class>

    <!-- http://www.co-ode.org/ontologies/testont.owl#SecurityPillars -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#SecurityPillars"/>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Software -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Software">
        <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    </owl:Class>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Solution -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Solution">
        <rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
    </owl:Class>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Tag -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Tag">
        <rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability"/>
```

```
    </owl:Class>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Vulnerability -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Vulnerability">
        <rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#VulnerabilityDefinitions"/>
    </owl:Class>

    <!-- http://www.co-
ode.org/ontologies/testont.owl#VulnerabilityDefinitions -->

    <owl:Class rdf:about="http://www.co-
ode.org/ontologies/testont.owl#VulnerabilityDefinitions">
        <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    </owl:Class>

    <!--
///////////////////////////////////////////////////////////////////////
//////////////
    //
    // Individuals
    //
///////////////////////////////////////////////////////////////////////
//////////////
     -->

    <!-- http://www.co-ode.org/ontologies/testont.owl#Authenticity -->

    <owl:NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Authenticity">
        <rdf:type rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#SecurityPillars"/>
    </owl:NamedIndividual>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Availability -->

    <owl:NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Availability">
        <rdf:type rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#SecurityPillars"/>
    </owl:NamedIndividual>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Confidentiality -->

    <owl:NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Confidentiality">
        <rdf:type rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#SecurityPillars"/>
    </owl:NamedIndividual>

    <!-- http://www.co-ode.org/ontologies/testont.owl#Integrity -->

    <owl:NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/testont.owl#Integrity">
        <rdf:type rdf:resource="http://www.co-
ode.org/ontologies/testont.owl#SecurityPillars"/>
    </owl:NamedIndividual>
```

```
</rdf:RDF>

<!-- Generated by the OWL API (version 4.2.1.20160306-0033)
https://github.com/owlcs/owlapi -->
```