Call: FP7-SEC-2013-1
Activity: SEC-2013.2.5-4: Protection systems for utility networks – Capability Project
Project Number: 608090

HyRiM

Hybrid Risk Management for Utility Networks

Collaborative Project

D1.3
Report on categorisation to support standardisation efforts of utilities according to Hybrid Risk Metrics

Due date of deliverable: March 2016
Actual submission date: May 2016

Start date of project: April 1, 2014 Duration: 36 months

Organisation name of lead contractor for this deliverable
ETRA Investigation y Desarrollo S.A

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

HISTORY

Version	Date	Reason	Reviewed by
vv.0.0	30-10-2015	Definition of Structure	Ana María Arias - ETRA
vv.0.1	27-11-2015	First contributions from partners	Ana María Arias - ETRA
vv.0.2	15-02-2016	First version of the deliverable	Ana María Arias - ETRA
vv.0.3	06-04-2016	First revision of the deliverable	Antonios Gouglidis – ULANC Stefan Schauer - AIT
vv.0.4	26-04-2016	Integration of changes and further contributions	Ana María Arias - ETRA
vv.0.5	23-05-2016	Revision of the deliverable	Antonios Gouglidis – ULANC
vv.0.6	27-05-2016	Final version	Ana María Arias - ETRA

AUTHORS LIST

Organization	Name
ETRA	Ana María Arias (amarías.etraid@grupoetra.com; +34963134082)
ULANC	Antonios Gouglidis (a.gouglidis@lancaster.ac.uk ; +44 1524 510380)
UPASSAU	Xiaobing He (xiaobing.he@uni-passau.d ; phone number: +49 851509-3056)
AKHLELA	Paolo Giacalone (paolo.giacalone@akhela.com ; phone number: +39 070 2466 1415)

Table of Contents

Index of Figures

Index of Tables

EXECUTIVE SUMMARY

Vulnerability risk assessment is a crucial process in security management. In this deliverable, we examine different existing standards, methodologies and categorization frameworks in order to get an insight of how vulnerabilities are assessed and classified according to different methods and criteria. Based on a comparative analysis of the examined approaches, we suggest a suitable and intuitive scoring system to be used for the categorization of vulnerabilities within HyRiM. We also outline the reasons of this selection and point out the main benefits of adopting a standard and universal categorization framework.

In detail, we select and examine nine approaches for the categorization of vulnerabilities in the context of cyber security risks. For each approach, firstly we provide a general overview and then go deeper in terms of particular specifications, classification method and used metrics (if applicable), and finally their suitability for HyRiM purposes. Although not all the examined approaches include metrics for categorizing vulnerabilities, most of them provide useful resources to facilitate utility providers to speak the same language and therefore to specify the vulnerabilities and threats by using standardized identifiers and languages. In fact, many are complementary to each other. It has also been found that some categorization approaches use non-common metrics to characterize vulnerabilities and are based on its own severity rates applicable for their own products or specific users. Hence, these are considered out of scope for the HyRiM categorization approach.

After analyzing each approach separately, we proceed to compare them with the objective of identifying which one offers the most suitable vulnerability measurement characteristics and prioritization techniques to be used within HyRiM. It is important to highlight that the aim is not to propose a new categorization system but instead to promote a common understanding of vulnerabilities and their impact through the selection of a universal standard method for the assessment of its criticality. In this context, the main parameters taken into account for the selection are the openness, universality and flexibility of the vulnerability scoring systems. At the end of the document, we provide a rationale of the chosen standard framework, giving further details about the metrics applied, the method to calculate impact scores, the main strengths and benefits and a list of useful resources and references to be further used.

1 ABBREVIATIONS

Term	Meaning
CB	Circumstances of Breach
CCE	Common Configuration Enumeration
CNA	CVE Numbering Authority
CPE	Common Platform Enumeration
CVE	Common Vulnerabilities and Exposures
CVRF	Common Vulnerability Reporting Framework
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
CWSS	Common Weakness Scoring System
DoD	Department of Defense (U.S)
DPA	Data Protection Authorities
DPC	Data Processing Context
EI	Ease of Identification
ENISA	European Union Agency for Network and Information Security
EU	European Union
FIPS	Federal Information Processing Standard Publication
ICASI	Industry Consortium for Advancement of Security on the Internet
NCP	National Checklist Program (U.S)
NIST	National Institute of Standards and Technology (U.S)
NVD	National Vulnerability Database (U.S)
OS	Operating System
OVAL	Open Vulnerability and Assessment Language
PPE	Personal Protective Equipment
RAC	Risk Assessment Code
S&P 500	Stock market index, maintained by S&P Dow Jones Indices
SCAP	Security Content Automation Protocol
SE	Systems Engineering
SE	Severity
XCCDF	eXtensible Configuration Checklist Description Format
XML	Extensible Markup Language

2 INTRODUCTION

Security vulnerability represents an important issue for network security. Usually, attackers take advantage of existing vulnerabilities and by entering in the network, affect the system in terms of availability, confidentiality and integrity. For this reason, as soon as a vulnerability is identified, it is very important to patch it so the system remains protected from any potential attack [1]. In order to protect the security of the systems, IT administrators should continuously work on the identification and assessment of security vulnerabilities usually coming from software or system implementation blemishes. These system flaws are very attractive for attackers who are able to exploit the vulnerabilities and therefore cause damage in the system data destroying its confidentiality, integrity or availability. Nowadays, IT managers use different platforms and tools to identify and assess the vulnerabilities. Prioritization of vulnerabilities is required in order to act on those representing the major risk but when the amount of vulnerabilities to take care of is very high and each one is assessed using different score scales [2] [3] [4], the question is how IT administrators can translate this raft of vulnerability data into tractable information?

How to prioritize vulnerabilities is a topic that has been widely debated in the existing literature. What is clear is that it is of critical importance for organizations to have adequate techniques for assessing and prioritizing its system vulnerabilities. In practice, organizations use diverse forms to measure the vulnerabilities due to the differences in context and the potential impact [5]. During the last years, different techniques and tools for scoring and ranking system vulnerabilities have been developed and implemented by both commercial and noncommercial entities. Although these systems and tools currently available for use have diverse advantages, the main disadvantage is that the assessed things and the measure units are often different, e.g. the approach of some is solely to measure the impact degree under the hypothesis that it is uniform for all organizations. The lack of standardization and interoperability between the different existing systems and the limited scope of what they cover represent the main shortcoming of the current situation. In effect, one of the most relevant flaws is the fact that most of the existing systems are internet-centric taking care only about vulnerabilities related to computers connected to the internet network. In this context, the main objective of the present work is to identify and propose the use of an open and universal approach to address the above mentioned shortcomings and therefore promote communicability and comprehensibility in the categorization of vulnerabilities and their criticality. Along the document, different existing standards, methodologies and categorization frameworks have been analyzed to finally conclude with the selection of the most suitable for HyRiM purposes. The result is the definition of the Common Vulnerability Scoring System (CVSSv3) as the universal standard severity ratings of software vulnerabilities to be used within HyRiM. CVSSv3 (greatly improved compared to CVSSv2) is an open vulnerability measurement tool currently used by many organizations from a variety of industries to assess security risks. Being a global framework designed to help IT managers to understand the criticality of system vulnerabilities and assess the priority given to security patching, it provides a solid base with clear definitions and guidance.

In practice, the scores provided by the CVSS are used to give a rating of the security vulnerabilities and obtain an estimation of their severity. CVSS scores are applicable to a wide range of systems and security products including software, firewalls, antivirus, databases, webs, legacy applications, etc. In the specific case of HyRiM, CVSS scores are expected to be used by the utility providers, helping them to classify their systems vulnerabilities in terms of criticality by using a common vocabulary. The distinguishing characteristic of the CVSS compared to other existing scoring systems is the provision of an open technique, which can be used by different organizations as a model to rank vulnerabilities in a standard way and enabling at the same time to customize the assessment according to context and each user specifications. The CVSS makes it possible through the scoring of three different metric groups *(Base, Temporal, and Environmental)*, which combined provides users with a representation of a vulnerability and its associated risk allowing therefore IT managers to make informed decisions for its mitigation.

As previously pointed out, the present document provides an overview of the existing standards, methodologies and categorization frameworks, which are enabled by different means to assess the criticality of vulnerabilities and in some cases to prioritize them in terms of actions required to be carried out. The document is built around two big building blocks: on one hand, Section 3 includes a review of seven approaches for the categorization of vulnerabilities. Each approach is divided in three subsections including a general overview of it; a detailed description of the categorization strategy including specification of the used metrics (if applicable), and its suitability for HyRiM purposes. On the other hand, Section 4 is focused on identifying the most suitable categorization approaches to be used within HyRIM. For this purpose, a comparative analysis of the previously reviewed categorization strategies is included. A set of concluding remarks are provided in the last section.

3 EXISTING STANDARDS, METHODOLOGIES AND CATEGORIZATION FRAMEWORKS

3.1 Common Vulnerability Scoring System (CVSS)

3.1.1 General overview

The Common Vulnerability Scoring System v3.0 (CVSS) [6] is considered to be a robust and useful scoring system mainly for IT vulnerabilities. Specifically, it provides an open framework that is used for communicating the characteristics and the severity of software vulnerabilities as well as for the prioritization of vulnerability remediation activities [7]. Due to its openness, CVSS is considered to be well suited for application in various environments, including the industry, governmental organizations, etc. In its current version, i.e., v3.0, CVSS is claimed to provide more clear and consistent approaches towards a better understanding, description and comparison of IT vulnerabilities. This is mostly the result of a few amendments in the scoring system, i.e., the provision of a consistent scoring system, the replacement of "Scoring Tips", and, the capability of being able to consider the system itself. In the following, we provide brief information with regard to the main specifications of CVSS v3.0 [8] and examine the suitability of it in the context of HyRiM.

3.1.2 CVSS Specification

CVSS in version 3.0 includes three main groups of metrics, viz. Base, Temporal and Environmental. Specifically, intrinsic qualities of a vulnerability are represented by the Base group, and time depended changes of vulnerabilities are reflected by the temporal group. Moreover, the Environmental group is concerned with vulnerabilities' characteristics that are unique to the environment of a user. In the following, we provide a high-level description of the metrics that reside in the aforementioned groups.

The Base group is composed of the Exploitability and Impact metrics. The former is capable of reflecting information regarding the ease and technical means by which a vulnerability can be exploited, while the latter reflects the direct consequence of a successful exploit. With regard to the Temporal group, the set of its metrics reflect how the various characteristics of vulnerabilities might change over time as well as provide awareness regarding the existence of automated methods that may result in increasing the scoring in CVSS, e.g., exploit kits. Lastly, the Environmental group include metrics that allow the incorporation of security controls (part of user's environment), which eventually may mitigate any consequences. The above-mentioned groups of metrics are assigned with a value by analysts in order to compute the overall vulnerability score. This is done using specific formula, which results in a score ranging from 0.0 to 10.0. The overall score might include only metrics provided by the Base group, which is considered to be mandatory, while the Temporal and Environmental groups are not. Nevertheless, the provision of a score for the latter two metric groups can refine the score of vulnerabilities. In general, security product vendors, etc. provide the Base and Temporal metric groups, while Environmental metrics are provided by organizations. CVSS

provides, apart from a score for vulnerabilities, a string vector with textual information. This information refers to the value of each metric that was used for the computation of the overall CVSS score.

For reasons of completeness, we further list in Table 1 the various metrics that exist under each of the examined groups.

Base metrics	Temporal Metrics	Environmental Metrics

• Exploitability metrics
  ○ Attack Vector (AV)
  ▪ Network (N)
  ▪ Adjacent (A)
  ▪ Local (L)
  ▪ Physical (P)
  ○ Attack Complexity (AC)
  ▪ Low (L)
  ▪ High (H)
  ○ Privileges Required (PR)
  ▪ None (N)
  ▪ Low (L)
  ▪ High (H)
  ○ User Interaction (UI)
  ▪ None (N)
  ▪ Required (R)
  ○ Scope (S)
  ▪ Unchanged (U)
  ▪ Changed (C)
• Impact Metrics
  ○ Confidentiality Impact (C)
  ▪ High (H)
  ▪ Low (L)
  ▪ None (N)
  ○ Integrity Impact (I)
  ▪ High (H)
  ▪ Low (L)
  ▪ None (N)
  ○ Availability Impact (A)
  ▪ High (H)
  ▪ Low (L)
  ▪ None (N)
	• Exploit Code Maturity (E)
	  ○ Not Defined (X)
	  ○ High (H)
	  ○ Functional (F)
	  ○ Proof-of-Concept (P)
	  ○ Unproven (U)
	• Remediation Level (RL)
	  ○ Not Defined (X)
	  ○ Unavailable (U)
	  ○ Workaround (W)
	  ○ Temporary Fix (T)
	  ○ Official Fix (O)
	• Report Confidence (RC)
	  ○ Not Defined (X)
	  ○ Confirmed (C)
	  ○ Reasonable (R)
		• Security Requirements (CR, IR, AR)
		  ○ Not Defined (X)
		  ○ High (H)
		  ○ Medium (M)
		  ○ Low (L)
		• Modified Base Metrics[1]
		  ○ Modified Attack Vector (MAV)
		  ○ Modified Attack Complexity (MAC)
		  ○ Modified Privileges Required (MPR)
		  ○ Modified User Interaction (MUI)
		  ○ Modified Scope (MS)
		  ○ Modified Confidentiality (MC)
		  ○ Modified Integrity (MI)
		  ○ Modified Availability (MA)

Table 1 - List of CVSS v3.0 metrics

For more information about the individual metrics, we recommend the reader to refer to the CVSS specification document [8], where information is also provided about the equations used for the computation of the CVSS score. Moreover, an online CVSS scores calculator is available at [9].

3.1.3 Suitability for HyRiM purposes

In this section, we provide information regarding the suitability of CVSS v3.0 as a scoring system suitable for application in the HyRiM project. The provided information is the result of examining the main functionalities of CVSS in its latest version. As already discussed, CVSS is considered to be applicable in various environments, including industrial ones. Having a look at the main differences between the current and previous version of the scoring system [10], it is apparent that CVSS in version 3.0 is capable of being aware of situations in which a vulnerability in one application may have an impact to other applications of the

[1] Same values for all metrics, i.e., Not Defined (default); High; Medium; and, Low.

systems. This provides us with indications regarding the potential to examine cascading vulnerabilities in utility networks. This new functionality is introduced in CVSS through the Scope metric. Specifically, scope is defined in [10] as the *"collection of privileges defined and managed by an authorization authority when granting access to computing resources"*. In addition to the former functionality, the potential to score multiple vulnerabilities with Vulnerability Chaining appears also to be appropriate for the HyRiM project, too. Vulnerability chaining is related with the successful sequential exploit of multiple vulnerabilities, which will eventually lead to a successful attack on an IT system. CVSS v3.0 provides the potential to score the individual vulnerability chains. Such functionality appears to be effective in complex systems, as utility networks, where a lot of dependencies exist amongst them.

3.2 National Vulnerability Database (NVD)

3.2.1 Database description

National Vulnerability Database (NVD) [11] is the U.S. government repository of standards-based vulnerability management data. NVD is managed by the Department of Homeland Security (DHS), the National Institute of Standards and Technology (NIST), and the MITRE Corporation. Vulnerability standards included in the NVD are presented in Figure 1.

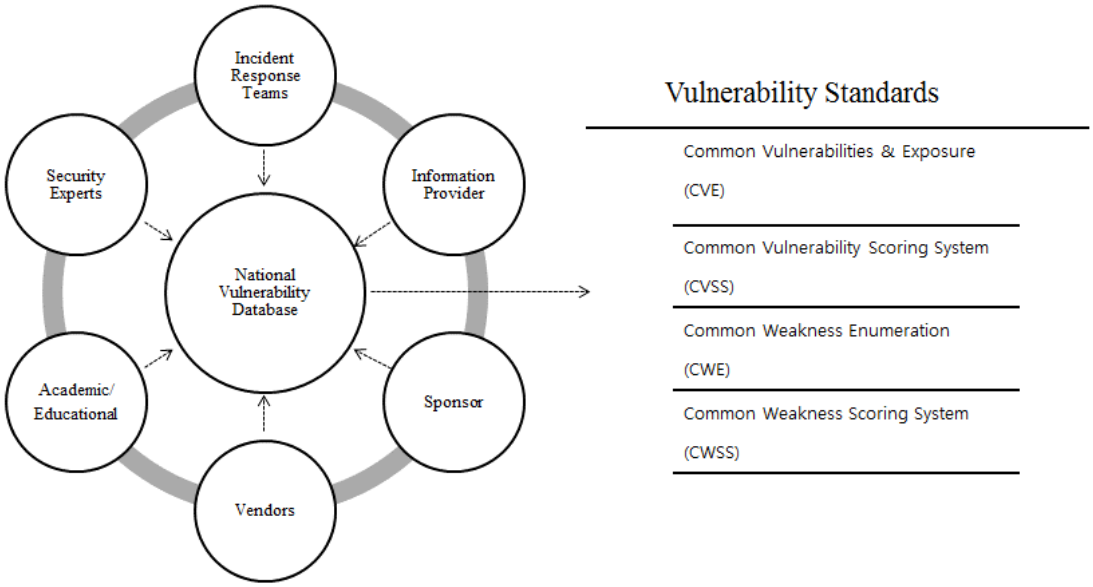

Figure 1. Vulnerabilities collecting system of NVD [12]

NVD is used as the repository for security-related contents for NIST's security content automation protocol (SCAP). The data in NVD can enable vulnerability management automation, security measurement and compliance. NVD collects security checklists, security related software flaws, misconfigurations, product names and impact metrics, databases of vulnerabilities, etc. Some of those databases are listed in the following:

- **Security checklists**
  The national checklist program (NCP) is the U.S. government repository of publicly available security checklists (or benchmarks). NCP provides low guidance on setting the security configuration of operating systems and applications, in the eXtensible Configuration Checklist Description Format (XCCDF) and Open Vulnerability and Assessment Language (OVAL).
- **Vulnerability search engine**
  If a CVE standard vulnerability name or OVAL query was tried, vulnerabilities that match ALL keywords would be presented. Vulnerabilities description associated with software flaws (CVE)

and common configuration enumeration (CCE) misconfigurations will be returned by this search engine.

- **SCAP**
  SCAP is a set of programs and protocols that NVD supports. SCAP aims at organizing, expressing, and measuring a) security-related information in standardized ways, as well as b) related reference data (provided by NVD) for vulnerabilities [13]
- **Product dictionary**
  Common Platform Enumeration (CPE) is a structured naming scheme for information technology systems, software, and packages. CPE includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name. The dictionary
  a) is provided in XML format
  b) is available to the general public
  c) is updated nightly when modifications or new names are added.
- **Impact metrics (CVSS)**
  NVD receives feeds from CVE and analysts assign CVSS scores (Base only).The NVD provides CVSS scores for almost all known vulnerabilities and provides a CVSS score calculator to customize vulnerability impact scores based on FIPS 199 system ratings [11]

3.2.2 Common Vulnerabilities and Exposures (CVE)

CVE is a dictionary of security vulnerabilities. Established in 1999, its creation responds to the lack of standardization of names of vulnerabilities (different repositories referring to the same vulnerability using a different name). CVE [14] consist of standard identifiers for vulnerabilities, which help to find information about a vulnerability, including mechanisms and existing products for removing the vulnerability. It also helps to define if some specific tools are useful for detecting attacks that are based on specific vulnerabilities. Once a security vulnerability is identified, the CVE Numbering Authority (CNA) assigns it an identifier. Then the information is posted on the CVE list by the CVE Editor. MITRE Corporation is the main CNA and the unique CVE Editor. Other CNAs are software vendors such as Apple or Adobe, third-party coordinators such as CERT/CC, or researchers such as Core Security Technologies.

3.2.3 Common Weakness Enumeration (CWE)

CWE [15] includes a collection of software weakness types described and stored as .xml, .xsd and .pdf documents. The main four (4) types of CWE IDs are:

1) Category ID - aggregates types of weaknesses
2) Compound Element ID - aggregates a group of several events that together can result in a successful attack
3) View ID - is assigned to predefined perspectives with which one might look at the weaknesses in CWE
4) Weakness ID

Each individual CWE can have many CWE children associated with it in a hierarchical structure. The NVD provides a cross section of the global CWE structure integrating therefore the CWE into the scoring of CVE vulnerabilities. Currently, CWE is used in NVD as a mechanism that classify CVEs according to the type of vulnerability they represent. Around CWE, different relevant body of knowledge such as Common Weakness Scoring System (CWSS), Common Attack Pattern Enumeration and Classification (CAPEC) and Common Vulnerabilities and Exposures (CVE) are active. They are used by different organizations, including DoD, to identify and alleviate the most critical types of vulnerabilities in the software [16]

3.2.4 Common Weakness Scoring System (CWSS)

Ranking software weaknesses using numerical scores is important due to the need for such an operation from both software developers and consumers, which eventually will lead to a prioritization of the actions that are required to either avoid or eliminate the potential weaknesses. The CWSS [17] provides a mechanism for prioritizing software weaknesses in a flexible, consistent and open manner. Therefore, a number of targeted, generalized, context-adjusted, and aggregated methods make such a ranking possible. CWSS 0.8 is based on the Targeted scoring method. This method is applicable to a particular package. The CWSS 0.8 scoring formula includes 18 factors divided into 3 groups: The Base Finding Group, the Attack Surface Group, and the Environmental Group.
The CWSS severity score is similar to the CVSS in the sense that both scores are obtained by a successive calculation of three metrics. The difference is that the final score of the CWSS is a value between 0 and 100, while the CVSS score is a value between 0 and 10. CWSS and CVSS are not competitors but on the contrary can be leveraged together.

3.2.5 Suitability for HyRiM purposes

The suitability of NVD for HyRiM can be twofold. Vulnerability search engine could return CVE identifier, summary and CVSS severity of a vulnerability. The CVSS score calculator allows utility operators to calculate scores (base scores, temporal scores or even environmental scores) to reflect the impact of the vulnerability on their organization. More details about how CVSS could fit to HyRiM can be found in Section 3.1.3

3.3 Red Hat Product Security rates.

3.3.1 General overview

Red Hat is an American multinational company providing open-source software solutions to the enterprise community. Known as the world's open source leader and member of the S&P 500 index, Red Hat provides high-performing, reliable and secure technological solutions commonly used in mission-critical systems of different sectors (e.g. financial, telecommunications, transport, defense, etc.) and companies all around the world [18]. In terms of security, Red Hat´s mission is to assist its customers on the protection from security issues and to identify, track and solve any security problems that may affect users when using red hat products and services. Red Hat provides the guidance, stability and security required to confidently deploy corporative solutions [19] .

In this context, the Red Hat Security Response Team assigns an impact rating to the different security issues identified in Red Hat solutions according to the severity of the problem. Specifically, a four-point scale including Low, Moderate, Important, and Critical levels is used for this purpose. Additionally, the Common Vulnerability Scoring System (CVSS) base scores are applied for rating the identified security issues. Combined, these scores are useful to determine the impact of security issues and therefore help users to properly plan and prioritize protection measures such as upgrade strategies of their systems. [20]. It is worth mentioning that the obtained scores indicate the potential risk of a specific vulnerability based on the analysis of the bug but not on the threat level, which means that the impact rate is not altered if an attack is released for a particular flaw [21]

3.3.2 Issue Severity Classification

As previously indicated, Red Hat rates the impact of security issues found in its products by using a four-point scale (i.e., “Low impact”, “Moderate impact”, “Important impact” and “Critical impact”). The four-point scale reflects severity of an issue and the level of its consequences. The obtained score helps customers to judge the severity of the problem and to prioritize the updates to be carried out in the system. The rating, based on a technical analysis of the specific flaw and its type, reflects the potential risk of the security issue. However, since the current threat level is not taken into account the rate do not change if an exploit or worm

is later released for a flaw, or if one is available before the release of a fix [22]. Table 2 presents the description of the four-point scale used to rate issues having a security impact.

LEVEL	DESCRIPTION
Critical impact	This rating is given to flaws that **could be easily exploited by a remote unauthenticated attacker and lead to system compromise** (arbitrary code execution) without requiring user interaction. These are the types of vulnerabilities that can be exploited by worms. Flaws that require an authenticated remote user, a local user, or an unlikely configuration are not classed as Critical impact.
Important impact	This rating is given to flaws that **can easily compromise the confidentiality, integrity, or availability of resources.** These are the types of vulnerabilities that allow local users to gain privileges, allow unauthenticated remote users to view resources that should otherwise be protected by authentication, allow authenticated remote users to execute arbitrary code, or allow remote users to cause a denial of service.
Moderate impact	This rating is given to flaws that **may be more difficult to exploit but could still lead to some compromise of the confidentiality, integrity, or availability of resources, under certain circumstances.** These are the types of vulnerabilities that could have had a Critical impact or Important impact but are less easily exploited based on a technical evaluation of the flaw, or affect unlikely configurations.
Low impact	This rating is given to **all other issues that have a security impact.** These are the types of vulnerabilities that are believed to require unlikely circumstances to be able to be exploited, or where a successful exploit would give minimal consequences.

Table 2. Red Hat Issue Severity Ratings [22]

The use of this four-point scale along with the Common Vulnerability Scoring System (CVSS) base scores allows to provide a prioritized risk assessment helping customers to understand and schedule upgrades to their systems and thus enabling informed decisions on the risk each issue places on a unique environment.
It is worth to remark that Red Hat does not use the CVSS to determine the priority with which flaws are fixed. Instead, CVSS is used as a guideline to identify key metrics of a flaw and the priority to fix flaws is defined by overall impact of the flaw calculated using the described four-point scale.

A Red Hat security advisory may contain fixes for more than one vulnerability and for packages including more than one product (e.g. Red Hat Enterprise Linux 5 and 6). Each issue in a security advisory has an impact rate for each product. The global severity of an advisory is the highest severity out of all the individual issues, across the different products targeted by the advisory. In order to make things simpler, advisories solely show the global severity (except kernel advisories, which list the severity of each issue). The advisories include links to the key entries in Red Hat's bug-tracking system, where individual impacts and additional commentary are available. The severity level is adjusted when a technology – usually enabled and used by default – completely blocks the exploitation of a particular vulnerability across all architectures. Moreover, when a technology reduces the risk of a vulnerability, the severity level is also adjusted and an explanation of the decision in the bug-tracking entry is provided [22].

Base Score Variations across products
Depending on the product, version, and architecture, a CVE-named vulnerability can have different CVSS metrics. Some examples are provided below:

- A vulnerability affecting only a small number of architectures.
- A vulnerability mitigated by source code protection mechanisms on some platforms.
- A vulnerability affecting more than one application

When CVSS base scores are significantly different across products, they are assigned to each product separately, wherever possible. If the score is not split, the worst-case outcome is taken (the metric giving the highest CVSS base score)

3.3.3 Suitability for HyRiM purposes

Red Hat provide security ratings based on a four-point scale indicating how critical a security issue is and helps users to judge its severity guiding them to prioritize the required systems updates. These rates only apply for security issues that may affect users of Red Hat products and services. Basically, as other IT vendors (such as Microsoft, Cisco, Oracle, etc.) and security organizations (such as Secunia, Symantec, etc.), Red Hat has created its own ratings to assess the severity of vulnerabilities of their products but without following a unified approach taking into account other existing vulnerability assessments or data bases. Within this approach, each vendor only records their own products so users cannot make a comparison of severity among different vendors. In this context, Red Hat Security rates do not represent an objective categorization and therefore are not suitable for HyRiM purposes of promoting the standardization of vulnerabilities assessment.

3.4 Mishap Severity Categories (MIL–STD–882E)

3.4.1 General overview

The MIL-STD-882E is a military standard drafted by the U.S. Department of Defense's (DoD) in which internal common practices for conducting system safety and means of evaluating risks are described. The standard outlines the DoD Systems Engineering (SE) approach to eliminate hazards and minimize risks (in case it is not possible to eliminate hazards). The MIL-STD-882E covers all types of hazards applying to systems, equipment, products, infrastructure (including hardware and software) throughout design, development, test, production, use, and disposal of defense systems. The standard includes the system safety requirements throughout the life-cycle of any system, which when properly applied enable the identification and management of hazards (and its respective risks) during system development and the related engineering activities [23].

The system safety process consists of eight elements. Figure 2 depicts the typical logic sequence of the process. However, iteration between steps may be required.

1. **Document the system safety approach for managing hazards as an integral part of the SE process.**
2. **Identify and document hazards** through a systematic analysis process that includes system hardware and software, system interfaces (to include human interfaces), and the intended use or application and operational environment.
3. **Assess and document risk -**The severity category and probability level of the potential mishap(s) for each hazard across all system modes are assessed using specific definitions further explained in section 3.4.2
4. **Identify and document risk mitigation measures** - when a hazard cannot be eliminated, the associated risk should be reduced to the lowest acceptable level within the constraints of cost, schedule, and performance by applying the system safety design order of precedence.
5. **Reduce risk** considering and evaluating the cost, feasibility, and effectiveness of candidate mitigation methods.
6. **Verify, validate, and document risk reduction** through appropriate analysis, testing, demonstration, or inspection.
7. **Accept risk and document** - before exposing people, equipment, or the environment to known system-related hazards, the risks shall be accepted by the appropriate authority.
8. **Manage life-cycle risk** considering any changes to include, but not limited to, the interfaces, users, hardware and software, mishap data, mission(s) or profile(s), and system health data.

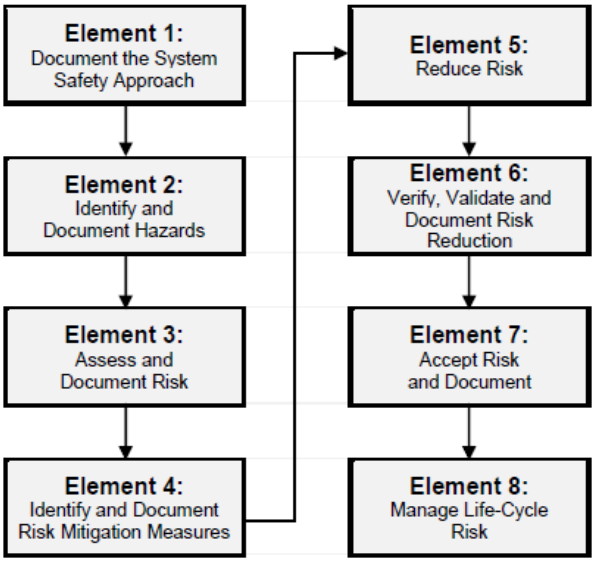

Figure 2. Elements of the system safety process according to MIL–STD–882E [23].

3.4.2 Risk assessment metrics

The severity and the level of probability of a potential mishap for each hazard across all system modes are assessed using the criteria defined in Table 3 and Table 4. Specifically, metrics of Table 3 are used to define the severity category of a hazard at a given point in time (i.e., "Catastrophic", "Critical", "Marginal" and "Negligible"), determining the potential for death or injury and the environmental impact and the economic impact. A hazard can have the potential to affect one or all of these mentioned areas.

SEVERITY CATEGORIES		
Description	**Severity Category**	**Mishap Result Criteria**
Catastrophic	1	Could result in one or more of the following: death, permanent total disability, irreversible significant environmental impact, or monetary loss equal to or exceeding $10M.
Critical	2	Could result in one or more of the following: permanent partial disability,injuries or occupational illness that may result in hospitalization of at least three personnel, reversible significant environmental impact, or monetary loss equal to or exceeding $1M but less than $10M.
Marginal	3	Could result in one or more of the following: injury or occupational illness resulting in one or more lost work day(s), reversible moderate environmental impact, or monetary loss equal to or exceeding $100K but less than $1M.
Negligible	4	Could result in one or more of the following: injury or occupational illness not resulting in a lost work day, minimal environmental impact, or monetary loss less than $100K.

Table 3. Severity categories according to MIL–STD–882E [23].

Metrics of Table 4 include six levels (A to F) used to determine the probability for a given hazard at a given point in time and therefore assess the likelihood of occurrence of a mishap. Level F is used to relate cases in which the hazard is no longer present. No amount of warning, caution, training or Personal Protective Equipment (PPE) can move a mishap probability to level F.

PROBABILITY LEVELS			
Description	Level	Specific Individual Item	Fleet or Inventory
Frequent	A	Likely to occur often in the life of an item.	Continuously experienced.
Probable	B	Will occur several times in the life of an item.	Will occur frequently.
Occasional	C	Likely to occur sometime in the life of an item.	Will occur several times.
Remote	D	Unlikely, but possible to occur in the life of an item.	Unlikely, but can reasonably be expected to occur.
Improbable	E	So unlikely, it can be assumed occurrence may not be experienced in the life of an item.	Unlikely to occur, but possible.
Eliminated	F	Incapable of occurence. This level is used when potential hazards are identified and later eliminated.	Incapable of occurence. This level is used when potential hazards are identified and later eliminated.

Table 4. Probability levels according to MIL–STD–882E [23]

For quantitative analysis, it is preferable to use quantitative data indicating the rate of occurrence or frequency for the hazard. The provability level of not happening is usually less than one in a million. As regards the definition, the frequency (numerator) is the expected or real number of mishap during a specified exposure (denominator). The exposure is based on things such as number of hours of flights, number of missile firings, number of miles driven, number of years of service, etc. In those cases where quantitative data is not available then it is required to rely on the qualitative descriptions provided above in Table 4.

A Risk Assessment Code (RAC) is used to express the assessed risks. The RAC consist of one severity category and one probability level (e.g. a RAC of 2B is the combination of Critical severity category and a probable probability level). The risk assessment matrix presented in Table 5 assigns each RAC a specific risk level (High, Serious, Medium, or Low)

RISK ASSESSMENT MATRIX				
SEVERITY / PROBABILITY	Catastrophic (1)	Critical (2)	Marginal (3)	Negligible (4)
Frequent (A)	High	High	Serious	Medium
Probable (B)	High	High	Serious	Medium
Occasional (C)	High	Serious	Medium	Low
Remote (D)	Serious	Medium	Medium	Low
Improbable (E)	Medium	Medium	Medium	Low
Eliminated (F)	Eliminated			

Table 5. Risk assessment matrix according to MIL–STD–882E [23]

When systems hazards have associated risks related to software causes and controls, it may acceptable based on evidence that the causes and mitigations have been identified, implemented, and verified according to the DoD customer requirements. The evidence supports the idea that hazard controls offer the required level of mitigation and therefore the associated risks can be accepted by the appropriate risk acceptance authority. Following this approach, there is not difference between software and hardware and operators. If the

software design does not comply with the safety requirements, then it contributes to the associated risks with inadequacy verified software hazards causes and controls. Generally, risk assessment takes into account both qualitative and quantitative judgment and evidence. An illustration of how these principles can be applied to provide an assessment of risk associated with software causal factors is provided in Table 6.

Risk Levels	Description of Risk Criteria
	A software implementation or software design defect that upon occurring during normal or credible off-nominal operations or tests:
High	• Can lead directly to a catastrophic or critical mishap, or • Places the system in a condition where no independent functioning interlocks preclude the potential occurrence of a catastrophic or critical mishap.
Serious	• Can lead directly to a marginal or negligible mishap, or • Places the system in a condition where only one independent functioning interlock or human action remains to preclude the potential occurrence of a catastrophic or critical hazard.
Medium	• Influences a marginal or negligible mishap, reducing the system to a single point of failure, or • Places the system in a condition where two independent functioning interlocks or human actions remain to preclude the potential occurrence of a catastrophic or critical hazard.
Low	• Influences a catastrophic or critical mishap, but where three independent functioning interlocks or human actions remain, or • Would be a causal factor for a marginal or negligible mishap, but two independent functioning interlocks or human actions remain. • A software degradation of a safety critical function that is not categorized as high, serious, or medium safety risk. • A requirement that, if implemented, would negatively impact safety; however code is implemented safely.

Table 6. Software hazard causal factor risk assessment criteria [23]

3.4.3 Suitability for HyRiM purposes

The MIL-STD-882 is the safety standard from the Department of Defense of U.S (DoD). This risk based safety standard covers hazards applicable to systems, products, equipment and/or infrastructure (including software and hardware) throughout the entire life cycle (including design, development, test, production, use, and disposal); and defines general safety requirements to enable the identification and understanding of known hazards and their associated risks. The MIL-STD-882 standard is focused on safety failures and is not always adequate to handle malicious threads. However, in spite of this limitation and the fact that this standard is designed to be used and applied by DoD Acquisition Programs, it offers a good risk ranking system, which overall structure and metrics could eventually be extrapolated and adapted to fit with a secure development life within HyRiM purposes.

3.5 Methodology of the assessment of severity of personal data breaches

3.5.1 Overall methodology

The European Union Agency for Network and Information Security (ENISA) provides in [24] recommendations for a methodology that is capable of assessing the severity of personal data breaches. The latest version of the methodology consists of an update of the initial approach proposed in the context of a study on the technical implementation of the Article 4 of the ePrivacy Directive. The main objectives of this methodology include the provision of a quantitative tool in order to assess the severity of data breaches by data controllers and national competent authorities; to provide support for analyses and statistics with regard to the reported

personal data breaches; and to contribute towards a common severity assessment methodology in the European Union (EU).

In that methodology, ENISA defines three core elements to be taken into consideration when assessing the severity of data breaches, viz. the context of data processing, the individual's ease of identification, and the circumstances of the breach. Specifically, the methodology is able to guide the data controller and make the overall assessment via the use of the aforementioned three quantitative criteria. For the calculation of the severity score, the following formula is used: *SE = DPC x EI + CB*, where SE: Severity, DPC: Data Processing Context, EI: Ease of Identification, and CB: Circumstances of Breach [24]. Specifically, DPC is used to evaluate a given's data criticality under a specific context; EI operates as a correction factor of DPC, and CB is used to quantify specific circumstances of the breach – a more detailed information with regard to the scoring of these criteria is provided in [24]. When the process of evaluating each of the three criteria is finished, the severity score is calculated to provide a number that shows the severity level of the breach. The methodology provides four levels of severity: low, medium, high and very high – detailed information with regard to the description of the severity levels is provided in [24]. After the definition of the severity level, the result can be flagged in order to indicate certain elements of the breach that are important for the final assessment. In [24], two flags are defined, viz. number of individuals breached exceeds 100, and data unintelligible. The former indicates that an individual's data can be disclosed more easily in the context of a bigger incident, and that the high number of affected individuals influences the overall scale of the breach. Moreover, data unintelligence refers to the decrement of the impact to individuals due to the decreased possibility of unauthorized parties to access data [24].

3.5.2 Suitability for HyRiM purposes

The data breach severity methodology appears to be used mostly by Data Protection Authorities (DPA) and data controllers. Therefore, the suitability of this methodology in the context of the HyRiM purposes is not completely clear. Nevertheless, assuming that data controllers may operate in utility networks for protecting consumers' data will make this applicable to HyRiM. In this case, this methodology would provide important tools for the protection of consumers' data in several directions. Specifically, it would be in position to protect consumers' simple data (e.g., names, postal addresses, etc.), behavioral data (e.g., used plans), financial data (e.g., credit card information), sensitive data (e.g., creditworthiness), and credentials (e.g., credentials used for on-line services) [24].

3.6 Open Vulnerability and Assessment Language (OVAL)

3.6.1 General overview

OVAL [25] is sponsored by the office of cybersecurity and communications at the U.S. department of homeland security. The content of OVAL is a result of the collaborative effort of MITRE Corporation, OVAL board (representatives from numerous organizations such as operating system, security tool vendors, academic institutions and government) and the information security community. The OVAL repository uses the publicly known vulnerabilities identified in CVE list as the basis for most of the OVAL definitions, which are collaboratively developed by computer security researchers, software vendors and system administrators. OVAL allows for sharing technical details regarding how to identify the presence or absence of vulnerabilities on a computer system. OVAL allows personally reviewing individual OVAL definition to see exactly how the vulnerability determination was made. OVAL definitions help users determine the presence of vulnerabilities or configuration issues on systems before they can be exploited. Each OVAL definition includes metadata, a high-level summary, and the detailed test. In the following subsections, OVAL definitions and OVAL adoption program will be presented.

3.6.2 OVAL definitions

OVAL definitions, written in Extensible Mark-up Language (XML), detect the presence of software vulnerabilities, configuration issues, programs, and patches in terms of system characteristics and configuration information, without requiring software exploit code [26]. The previously mentioned system characteristics include a device's operating system and its settings, the installed software applications and their respective settings. Additionally, there are configuration attributes of software, encompass registry settings, file system attributes, as well as configuration files. OVAL definitions are standardized and machine-readable tests that check the presence of software vulnerabilities, configuration issues, programs, and patches in computer systems. Table 7 shows the information that is included in an OVAL definition. The OVAL define the vulnerabilities present in certain systems and whether the configuration of a system conforms to the security policies in place. Additionally, a check if patches, which may need to be applied, are appropriate for a system, is performed.

OVAL definitions encode the details of a specific machine state (when is a system vulnerable, in compliance, etc.) enabling testing of a system to be automated.

There are four main classes of OVAL definitions [26]:

- OVAL vulnerability definitions: tests that determine the presence of vulnerabilities on systems;
- OVAL compliance definitions: tests that determine whether the configuration settings of a system meet a security policy;
- OVAL inventory definitions: tests that whether a specific piece of software is installed on the system;
- OVAL patch definitions: tests that determine whether a particular patch is appropriate for a system.

NAME	DESCRIPTION
Metadata	OVAL-id, status of the definition, versions of the OVAL definition schema that the definition works with, a brief description of the security covered, the main author, a list of significant contributors
High-level summary	The specific OS, the name of the file with the vulnerability in it, application version, patch status
Detailed test	The logic for checking for the system characteristics and configuration attributes

Table 7. Information included in OVAL definitions, taken from [26]

3.6.3 OVAL adoption program

On one hand, using products and services that have adopted OVAL can not only provide users with a standard against how to measure tools when making purchasing decisions, but also provide vendors with something that distinguishes them from competition, thus encouraging adoption through the industry. On the other hand, integrating OVAL into vendor's products and services enables interoperability among their products and services and gain a competitive edge over companies that are not participating. The OVAL adoption program is based on five different capabilities:

- **Authoring tool:** aids in the process of creating new OVAL files.
- **Definition evaluator:** uses an OVAL definition to guide evaluation and produces OVAL results.
- **Definition repository:** makes OVAL definitions available to the community.
- **Results consumer:** accepts OVAL results as input (either displays those results to the user, or uses the results to perform some actions).
- **System characteristics producer:** generates a valid OVAL system characteristics document based on the details of an endpoint using one or more of the OVAL-supported assessment methods.

The OVAL adoption program is aiming at:

- educating vendors on best practices regarding the use and implementation OVAL and
- providing vendors with an opportunity to make formal self-assertions about how their products utilize OVAL and
- helping MITRE gain deeper insights into how OVAL is or could be utilized.

The four phases of the adoption program are listed in Table 8.

PHASES	PROCEDURES
1. Declaration to adopt OVAL	1.1. review the "OVAL technical use cases guide" 1.2. email MITRE to request "OVAL adoption declaration form" 1.3. emails from review authority 1.4. complete the declaration form and return it to MITRE 1.5. form is reviewed by the review authority and the product or service is added to the list of OVAL adoption program participants
2. Implementation	2.1. ensure proper integration of OVAL into the product or service 2.2. complete the integration of OVAL into the product or service 2.3. provide feedback on your experience to the OVAL moderator and OVAL community
3. Questionnaire	3.1. Email MITRE to request "OVAL adoption questionnaire form" 3.2. Emails from review authority 3.3. Complete the adoption questionnaire and email it to MITRE 3.4. Completed adoption questionnaire reviewed by the review authority and posted on OVAL website
4. Recognition	4.1. MITRE contacts the organization and the product or service is listed as an "official OVAL adopter"

Table 8. Phases of OVAL adoption program, taken from [27]

3.6.4 Suitability for HyRiM purposes

OVAL is a preventative measure, enabling only to define if there are vulnerabilities or configuration issues present on a computer system. One possible way to utilize OVAL to analyze cascading failures in critical infrastructure (the context of HyRiM) is to take advantages of its CVE identifier in OVAL definitions. Once a CVE identifier is found, CVSS can be used to score the severity of existing vulnerabilities (the suitability of CVSS for HyRiM is discussed in section 3.1.3. OVAL is restricted to publicly known configuration issues and vulnerabilities, which means that zero-day vulnerabilities could not be identified in OVAL. When applying OVAL for HyRiM, the risk of zero-day vulnerability could not be assessed. However, different from CVSS, the OVAL results schema allows applications to consume this data, interpret it, and take the necessary actions to mitigate the vulnerabilities and configuration conflicts.

3.7 Common Vulnerability Reporting Framework (CVRF).

3.7.1 General overview

Common Vulnerability Reporting Framework (CVRF) [26] is an XML-based language that allows different persons in different organizations to share important security-related information in a single format, making information exchange quicker. CVRF represents a very useful framework to exchange vulnerability information as well as all other types if security documentation. The present version is CVRF 1.1. CVRF was originally conceived to fill an important gap concerning vulnerability standardization, represented by the lack of a standard framework for creating vulnerability report documentation. As a matter of fact, it was very clear that a standard was necessary in this area and this was particularly true in all vulnerability reports, best practice documents and security bulletin released by vendors and coordinators. CVRF represents an important improvement since it allows to replace all non-standard reporting reports used in the past, making it possible to speed up information processing and exchange. CVRF makes available an XML format which

could be used by any vendor to publish important information concerning vulnerabilities, also including useful information such as CVE# to identify vulnerability, CVSS score to rate the relative severity of a vulnerability, mitigation instructions as well as affected products and versions. CVRF has two key features that make it very useful. Firstly, it provides a consistent way to depict security information thus simplifying the interpretation of the advisories. Secondly, it provides a machine-readable format for interpreting security advisories, making automation easier.

3.7.2 CVRF 1.1 characteristics and mind map

CVRF 1.1 offers a complete and flexible format and reduces duplication and the possibility of errors. Its main features are [27]:

- The Product Tree, a new method for specifying products in a hierarchical manner, has been created. It has been separated from the vulnerability section, reducing XML duplication. The Product Tree supports the construction of logical groups to further cut down on redundant XML by allowing logical groups to be referenced using a single identifier
- When possible, a consistent type/value construct has been implemented, enabling future updates or modifications without too many changes to CVRF parsers. Using a more generic construct also reduces the overall number of elements, by combining existing (similar) containers into one
- when possible, all elements that were similar and existed in several areas of the document have been aligned
- more constraints make it more difficult to build invalid documents
- thanks to a better use of optional elements, CVRF is more flexible for different document producers
- In order to guarantee a coherent look and feel, many of the optional meta-containers that use a plural word as the top-level identifier are now followed by a container that makes use of the singular version of the same identifier.

The ICASI CVRF Working Group keeps CVRF an updated framework that will be enhanced and revised as necessary. The current mind map is presented in Figure 3.

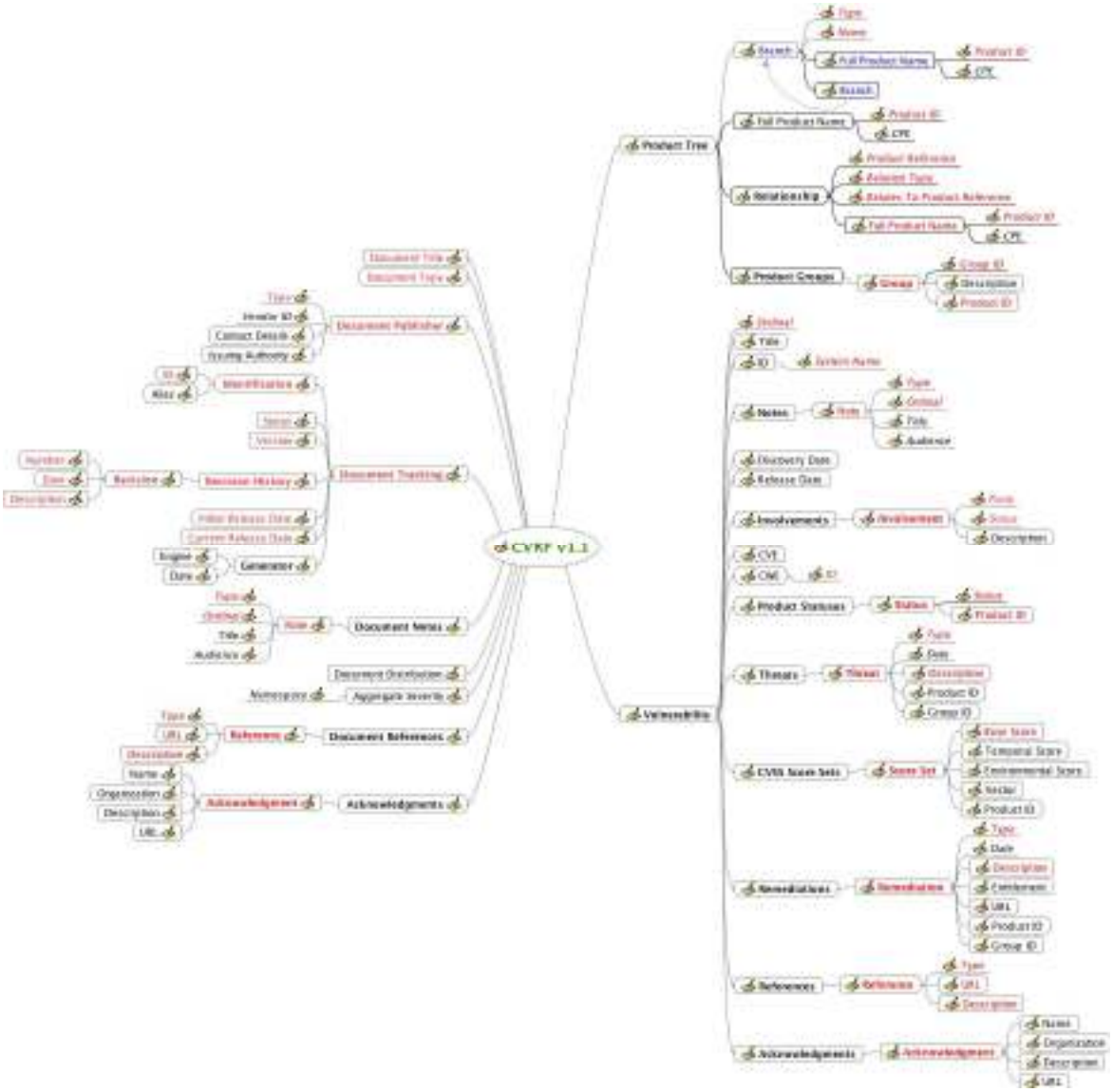

Figure 3. CVRF 1.1 mindmap [28]

3.7.3 Suitability for HyRiM purposes

We believe that CVRF can be useful for the HyRiM purposes since it offers not only vulnerability information, but any security-related documentation. What makes CVRF a very useful tool, is that it makes available vulnerability information in a single, standardized format, which speeds up information exchange and digestion, while also enabling automation. In a certain way, this is also the aim of the HyRiM project and this is why we believe it can be suitable for the project.

4 SUITABILITY ANALYSIS OF CATEGORIZATION APPROACHES

In this section, we provide a comparative analysis of the existing standards, methodologies and categorization frameworks revised in the previous sections of the document. Although for many of them the suitability for HyRiM purposes has been identified, the aim of this section is to pave the way towards the identification of a unique categorization system to be used within HyRiM. It is important to highlight that the aim is not to propose a new categorization of vulnerabilities but instead to support standardization efforts by the selection of a standard method enabling the classification of service parameters of utility network in terms of criticality for the infrastructures. It will contribute to solve the problem of multiple, incompatible

scoring systems in use today. The final goal is to establish a common framework to be used within HyRiM and through which utility providers can measure and compare vulnerabilities and risks, so as to assure communicability and comprehensibility of results.

The first studied approach is the **Common Vulnerability Scoring System (CVSS)** which is a free and open framework for assessing the severity of computer system security vulnerabilities. CVSS allows to assign severity scores to vulnerabilities, enabling responders to prioritize responses and resources according to threat. Scores are calculated based on a formula that depends on several metrics and range from 0 to 10, with 10 being the most severe. The CVSS base score has been adopted as primary method for quantifying the severity of vulnerabilities by a wide range of organizations and companies, including the **National Vulnerability Database (NVD)** [28], the Open Source Vulnerability Database (OSVDB) [29], CERT Coordination Center [30], Qualys [31] and Cisco [32]. As regards to the NVD (also revised above), it is not a vulnerability categorization system on its own but instead is a repository of standards-based vulnerability management data. NVD provides severity rankings of "Low", "Medium," and "High" but these qualitative rankings are simply mapped from the numeric CVSS scores. NVD provides CVSS scores for almost all known vulnerabilities. The NVD feeds itself from data coming from the CVE website and in response performs analysis to determine impact metrics (CVSS), vulnerability types (CWE), and applicability statements (CPE), along with other related metadata. The **Common Vulnerabilities and Exposures (CVE)** neither is a categorization system but instead it is simply a dictionary of publicly known information security vulnerabilities. The CVE catalog's main purpose is to standardize the way each known vulnerability or exposure is identified. This is important because standard IDs allow security administrators to quickly access technical information about a specific threat across multiple CVE-compatible information sources. The widespread use of CVE includes a vulnerability scoring based also on CVSS. While the CVE is a standard way to identify a vulnerability with standard naming convention, the CVSS is a standard way to measure vulnerability severity rating. The **Common Weakness Scoring System (CWSS)** is also a categorization system for prioritizing software weaknesses in a consistent, flexible and open manner. Although CVSS and CWSS are very similar in concept, there are some limitations with CWSS. In CWSS, the formula would need to be refined in order to guarantee that the range of potential scores is more evenly distributed. There are probably unexpected interactions between factors that must be identified and resolved. CVSS scoring contains built-in adjustments that prevent many factors from affecting the score excessively, while also giving some priority to impact over exploitability; similar built-in adjustments may need to be included in CWSS. Moreover, CWSS does not include any strategy to give higher priority to vulnerabilities related to flaws in design or architecture versus those related to implementation. Scores resulting from CWSS and CVSS scores are not always comparable. Even if CWSS scores (with a maximum of 100) are normalized by dividing by 10 to a CVSS range (resulting in CVSS-equivalent scores within the range of 0 to 10), this does not necessarily mean that a CVSS score of 7 is equivalent to a CWSS 70. Although some users might desire this feature to be applicable, equivalence in scores might not be feasible because CWSS is often measuring completely different characteristics than CVSS does.

As regards to categorization systems aimed at specific end-users (own rating systems), **Red Hat Product Security rates** the impact of security issues found in its products by using its own four-point scale (Low, Moderate, Important, and Critical), and complementarily using the CVSS base scores. Red Hat does not use CVSS to define the priority with which flaws are fixed but Instead it is used as a guideline to help identifying key metrics of a flaw while the priority for which flaws are fixed is defined by the overall impact of the flaw using the own Red Hat four-point scale. Although Red Hat provides a categorization of impact severities, these rates only apply to its products/services, which is not suitable for HyRiM purposes of promoting the standardization of vulnerabilities assessment. Another categorization approach reviewed is the military standard **MIL-STD-882E** which includes "**Mishap Severity Categories".** Although it provides categories and definitions to assess the severity category and probability level of potential mishap(s) for hazards across all system modes, they are aimed to be used by Military Departments and Defense Agencies within the DoD. Again, it does not fit with HyRiM purposes of supporting standardization efforts.

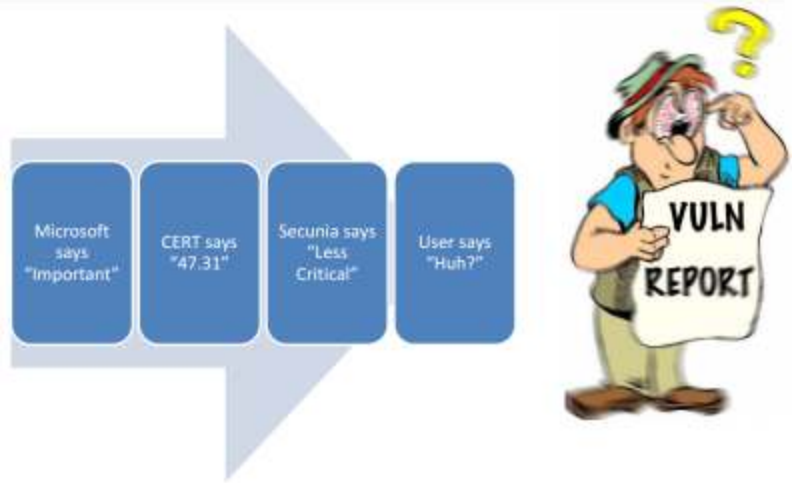

Figure 4. Scoring Discrepancy of vulnerabilities [34]

Further, the **Methodology of the assessment of severity of personal data breaches** developed by ENISA is another approach providing its own scoring and severity levels. The aim of this methodology is to provide data managers with a quantitative tool to assess the severity of personal data breaches and according to it notify the competent authorities. The tool is also useful as a means for data managers to quickly determine the required mitigation measures. Even though the methodology have four data categories for ranking, the categorization itself is not a general ranking of the types of data at hand. Furthermore, the methodology does not always cover all the possible casuistic, including impacts on specific groups of people or special cases for which a general methodology is not appropriate.

Finally, other approaches that despite not directly including categories or scoring systems are useful for the application of standard language and data in the assessment of vulnerabilities have been studied. In this context, we have revised, on the one hand, the **Open Vulnerability and Assessment Language (OVAL).** This is a XML-based language used to encode system details and standardize the three main steps of the assessment process: testing, analyzing and reporting. OVAL is not a vulnerability scanner as such but instead an open language to express checks for defining if software vulnerabilities and configuration issues, programs, and patches exist on a system. OVAL enables the sharing of technical details concerning how to determine the presence or absence of vulnerabilities. On the other hand, the **Common Vulnerability Reporting Framework (CVRF)** provides a common XML framework for reporting and sharing vulnerability information among multiple organizations. With CVRF, different parties including vendors, users, and coordinators of security response efforts around the world, are able to share critical vulnerability-related data in a standard, non-vendor specific format, thereby facilitating information dissemination, exchange, and incident resolution in a fast and secure way.

A comparison of the nine approaches analyzed above is presented in Table 9. As we can see, not all of them include metrics for categorizing vulnerabilities. However, many of them represent useful resources (e.g. CVE, NVD, OVAL, CVRF, etc.) that facilitate utility providers to speak the same language and therefore to specify the vulnerabilities and threats by using standardized identifiers and languages. Although only one scoring system is selected for use within HyRiM, all complementary and suitable standards will be also used. On the contrary, approaches covering specific products, services or organizations will be not taken into account for HyRiM purposes.

Nº	Assessment / Categorization Approach	Managed by	Type	Main characteristics	Metrics used	Link
1	Common Vulnerability Scoring System (CVSS)	FIRST.Org (U.S)	Scoring system	Open and standard framework. Consists of three metric groups to produce a score ranging from 0 to 10. Allows to prioritize responses and resources according to threat. Widespread use across organizations.	Own (CVSS)	https://www.first.org/cvss
2	Common Vulnerabilities and Exposures (CVE)	US CERT	Standard	Dictionary of publicly known information security vulnerabilities and exposures. Identifiers enable data exchange between security products and provide a baseline index point for evaluating coverage of tools and services.	CVSS	https://cve.mitre.org/
3	National Vulnerability Database (NVD)	NIST (U.S)	Repository of standards	Includes databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics (CVSS). The NVD is the CVE dictionary augmented with additional analysis, a database, and a fine-grained search engine.	CVSS	https://nvd.nist.gov/
4	Common Weakness Scoring System (CWSS)	MITRE (U.S)	Scoring system	Mechanism for prioritizing software weaknesses. CWSS is organized into three metric groups, which contains multiple metrics - also known as factors - that are used to compute a CWSS score for a weakness.	Own (CWSS)	https://cwe.mitre.org/cwss/cwss_v1.0.1.html
5	Red Hat Product Security Rates	Red Hat	Severity rating	Rates the impact of security issues found in Red Hat products using a four-point scale (Low, Moderate, Important, and Critical), as well CVSS base scores. These provide a prioritized risk assessment to help red hat customers understand and schedule upgrades to their systems.	Own + CVSS	https://access.redhat.com/security/updates/classification
6	Mishap Severity Categories (MIL–STD–882E)	Department of Defense - DoD (U.S)	Safety standard practice	Provides categories and definitions to assess the severity category and probability level of potential mishap(s) for hazards across all system modes. The standard is for use by all Military Departments and Defense Agencies within the DoD.	Own	http://www.system-safety.org/Documents/MIL-STD-882E.pdf
7	Methodology of the assessment of severity of personal data breaches	ENISA (EU)	Assessment methodology	Includes the provision of a quantitative tool to assess the severity of data breaches by data controllers and national competent authorities. Defines three quantitative criteria and four levels of severity for the assessment of the severity of data breaches.	Own	https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/dbn-severity

Nº	Assessment / Categorization Approach	Managed by	Type	Main characteristics	Metrics used	Link
8	Open Vulnerability and Assessment Language (OVAL)	MITRE (U.S)	Open Standard language (XML)	Open language to express checks for determining whether software vulnerabilities and configuration issues, programs, and patches exist on a system. Includes a language used to encode system details, and an assortment of content repositories. The language standardizes the three main steps of the assessment process.	n/a	https://oval.mitre.org/
9	Common Vulnerability Reporting Framework (CVRF)	ICASI	Standard language (XML)	XML-based language that enables different stakeholders across different organizations to share critical security-related information in a single format, speeding up information exchange and digestion.	n/a	http://www.icasi.org/cvrf/

Table 9. Comparison of existing categorization approaches

4.1 CVSS in HyRiM

For the purpose of HyRiM categorization of vulnerabilities, the **Common Vulnerability Scoring System (CVSS)** has been chosen as the most suitable scoring system due to its universal open and standardized method for rating IT vulnerabilities and determining the urgency of response. Being an industry open standard and vendor agnostic system, CVSS offers a universal language, usable and understandable by anyone, to assess vulnerability severity and determine priority of response solving therefore the problem of multiple and incompatible scoring systems. Within HyRiM, CVSS will be used as a standard scoring system enabling to compare and determine the relative importance of identified vulnerabilities in different systems.

The selection answers the need of avoiding the use of multiple scoring systems allowing decision makers/ security managers to use the same metrics to compare vulnerabilities and therefore make informed and timely decisions on the relative impact to their environments. In this context, the CVSS quantitative model allows repeatable accurate measurement while enabling users to identify the underlying vulnerability characteristics used to generate the scores. This standard measurement system is perfectly suitable for organizations, critical infrastructure managers and governments needing accurate and consistent vulnerability impact scores; and following this approach, also the most appropriate system for HyRiM.

CVSS provides a standard and easy way to capture the principal characteristics of a vulnerability, and produce a numerical score reflecting its severity, along with a textual representation of that score. It is possible to translate the numerical score into qualitative representations (low, medium, high, and critical) to help entities properly evaluate and prioritize their vulnerabilities and management processes. One of CVSS' strengths lies in its simplicity. The overall score in CVSS is divided into 15 separate characteristics within three metric groups: Base, Temporal, and Environmental. Each characteristic is divided into two or more distinct values. For example, the Access Vector indicates the location from which an attacker must exploit a vulnerability, with probable values of Local, Remote or Network Adjacent. “Local” means that are on the same physical or logical network, “remote” that are authenticated to the local system and “network” that are on the same physical or logical network. Usually, on top of the CVSS score, a vector is provided that identifies the selected values for each characteristic.

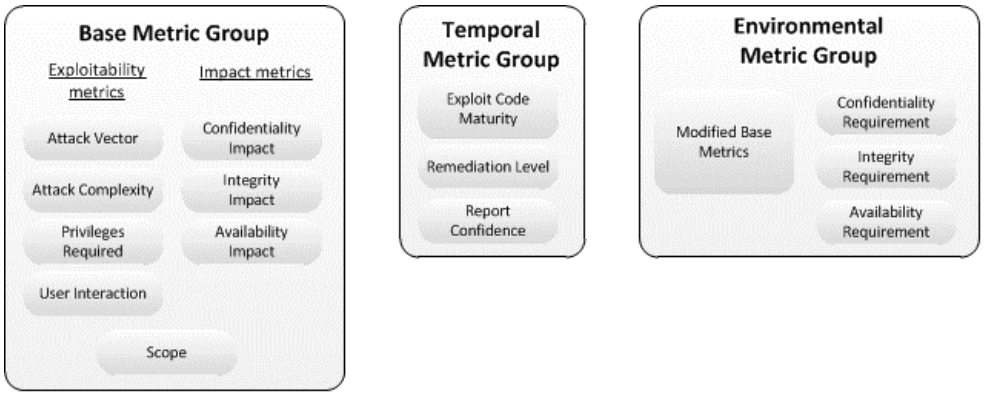

Figure 5. CVSS v3.0 Metric Groups [8]

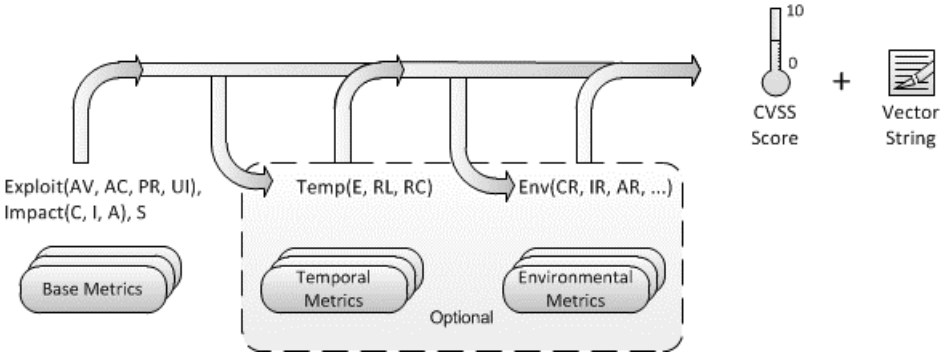

Figure 6. CVSS Metrics and Equations [8]

When properly applied and using the related documentation, CVSS scores are repeatable, i.e., different users will typically generate the same scoring for a specific vulnerability. However, different scores can be reached when information is incomplete, and significant variations are possible if an analyst does not carefully follow documentation. In comparison with the Confidentiality, Integrity, Availability model (CIA) which does not provide the depth and flexibility required by security experts, CVSS does provide the consistency that is useful for non-expert system and network administrators for prioritizing vulnerabilities. CVSS is widely adopted, particularly the base scores from the Base metric group.

Other reasons of this choice include that the system is flexible enough to manage both the recent challenges in vulnerability scoring, as well as those that we will see in the years to come, representing therefore a robust and useful scoring system that is fit for the future. Furthermore, the impact rates will be the same even when the vulnerabilities are discovered by multiple security tools used in different entities. By watching the CVSS scores of discovered vulnerabilities over time, organizations can more easily identify vulnerability trends. Then with an effective security program implemented, organizations will see improvements in their vulnerability metrics over time [33].

In short, **CVSS offers three main benefits** that justify its selection [34].

1) CVSS provides **standardized vulnerability scores**. When an organization uses a common algorithm for scoring vulnerabilities across all IT platforms, it can leverage a single vulnerability management policy defining the maximum allowable time to validate and remediate a given vulnerability.
2) CVSS provides an **open framework**. Users may be confused when a vulnerability is assigned an arbitrary score by a third party. With CVSS, the individual characteristics used to derive a score are transparent.
3) CVSS **enables prioritized risk.** When the environmental score is computed, the vulnerability becomes contextual to each organization, and helps provide a better understanding of the risk posed by this vulnerability to the organization.

After this final selection it is worth to remark that although we recognize that many other metrics could have been considered in the CVSS, it is also true that it is impossible to perfectly fit everyone's needs. However, we consider that CVSS metrics are the best compromise between completeness, accuracy and ease-of-use. Also important to highlight is the fact that as CVSS progress, the current metrics may expand or adjust, making the scoring more accurate, flexible and representative of future vulnerabilities and their risks.

4.2 CVSS resources and links

Below are useful references to additional CVSS v3.0 documents.

- Specification Document

Includes metric descriptions, formulas, and vector string. Available at:
https://www.first.org/cvss/specification-document

- User guide

Includes further discussion of CVSS v3.0, a scoring rubric, and a glossary. Available at:
https://www.first.org/cvss/user-guide

- Example document

Includes examples of CVSS v3.0 scoring in. practice. Available at:
https://www.first.org/cvss/examples

- CVSS v3.0 Calculator Use & Design

This guide covers the following aspects of the CVSS Calculator: Calculator Use, Changelog, Technical Design and XML Schema Definition. Available at:
http://www.first.org/cvss/use-design

- CVSS v3.0 calculator

Reference implementation of the CVSS v3.0 equations, available at:
http://www.first.org/cvss/calculator/3.0

- XML schema

Schema definition available at
https://www.first.org/cvss/cvss-v3.0.xsd

CONCLUSIONS

Security vulnerability is critical to network security. When exploited by attackers, vulnerabilities can result in important disruptions of the confidentiality, integrity and availability of the system. With the increasing number of vulnerabilities, vulnerability severity assessment becomes more and more important. Previously, many IT vendors assessed the severity of vulnerabilities of their products (e.g. Oracle, Microsoft, Red Hat, etc.) without unifying the used assessment method. At the same time, some security organizations (e.g. Symantec, Secunia, OSVDB, etc.) also developed their own Vulnerability Databases and assessment systems, which in many cases are mutually contradicted and therefore cannot be shared. This report proposes the use of a unified Vulnerability Assessment Standard, specifically the Common Vulnerability Scoring System (CVSS) has been selected as the most suitable for covering the needs within the HyRiM framework related to the categorization and prioritization of response when vulnerabilities are detected.

CVSS promotes the standardization of the vulnerability assessment and solves the problem of multiple incompatible scoring systems. It is a vendor agnostic, and an industry open standard designed to reflect vulnerability severity and determine urgency and priority of response in a way that can be usable and understandable by anyone. In fact, CVSS is currently approved and widely adopted by IT managers, security organizations, IT vendors and researchers across the world. In particular, CVSS provides a standard method for utility providers to rate the severity of vulnerabilities within their systems resulting in easier and more fluid remediation processes. Within HyRIM, CVSS complemented with the use of other international standards of classification (e.g., CVE and CWE) represents the common language to be used for vulnerabilities assessment and responses prioritization.

Using CVSS, which is based on context metrics (base, temporal an environmental) will help utilities to improve overall vulnerability management, assign resources and thus save costs. Reviewing CVSS scores of previous discovered vulnerabilities, can also help utilities to identify vulnerability trends and ideally, complemented with a good security program, improve their vulnerability metrics over time. Moreover, the CVSS in version 3.0 (through the scope metric) is capable of being aware of situations in which a vulnerability in one application may have an impact to other applications of the systems which will allow to examine cascading vulnerabilities in utility networks; which is of particular relevance for HyRiM project.

The results of this deliverable and the proposed standard will be further used in other tasks of the project. Specifically, in WP3, an analytical framework and associated metrics will be developed in order to understand how socio-technical systems can change over time and how temporal processes create vulnerabilities. The metrics and their underpinning assumptions and framework will be compared with those of CVSS in order to characterize the specific contributions we will make in that task. Moreover, the results will optimally support the standardization efforts required in WP5.

REFERENCES

[1] [Online]. Available: http://www.jocm.us/uploadfile/2015/0525/20150525041419219.pdf.

[2] [Online]. Available: http://www.microsoft.com/technet/security/bulletin/rating.mspx..

[3] [Online]. Available: http://www.kb.cert.org/vuls/html/fieldhelp. .

[4] [En línea]. Available: http://www.sans.org/newsletters/cva/.

[5] [Online]. Available: http://mintu-t.blogspot.com.es/2011_06_01_archive.html.

[6] [Online]. Available: https://www.first.org/cvss.

[7] [Online]. Available: https://nvd.nist.gov/cvss.cfm.

[8] «FIRST, "Common Vulnerability Scoring System v3.0: Specification Document," 2015.».

[9] [Online]. Available: https://nvd.nist.gov/cvss.cfm?calculator&adv&version=2#score.

[10] «FIRST, "Common Vulnerability Scoring System v3.0: User Guide," 2015.».

[11] [Online]. Available: https://nvd.nist.gov/cvss.cfm.

[12] [Online]. Available: http://www.naun.org/main/NAUN/communications/2014/a102019-105.pdf .

[13] [Online]. Available: http://csrc.nist.gov/publications/drafts/800-117-R1/Draft-SP800-117-r1.pdf.

[14] [Online]. Available: https://cve.mitre.org/.

[15] MITRE, “CWE Common Weakness Enumeration,” [Online]. Available: http://cwe.mitre.org.

[16] «Software Assurance Countermeasures in Program Protection Planning.,» [En línea]. Available: www.acq.osd.mil/se/docs/SwA-CM-in-PPP.pdf.

[17] [Online]. Available: https://cwe.mitre.org/cwss/cwss_v1.0.1.html.

[18] «http://www.ntia.doc.gov/files/ntia/redhat_27may2015.pdf».

[19] https://access.redhat.com/security/overview.

[20] [Online]. Available: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_the_Red_Hat_Customer_Portal.html .

[21] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Keeping_Your_System_Up-to-Date-Additional_Resources.html.

[22] «https://access.redhat.com/security/updates/classification».

[23] [Online]. Available: http://www.system-safety.org/Documents/MIL-STD-882E.pdf.

[24] «ENISA, "Recommendations for a methodology of the assessment of severity of personal data breaches," 2013.».

[25] [Online]. Available: http://oval.mitre.org.

[26] T. M. Corporation, «OVAL Repository Overview,» The MITRE Corporation, [En línea]. Available: https://oval.mitre.org/repository/about/overview.html. [Último acceso: 29 April 2016].

[27] T. M. Corporation, «OVAL Adoption Program Process,» The MITRE Corporation, [En línea]. Available: https://oval.mitre.org/adoption/process.html. [Último acceso: 29 April 2016].

[28] [Online]. Available: http://www.icasi.org/cvrf/.

[29] [Online]. Available: http://www.icasi.org/the-common-vulnerability-reporting-framework-cvrf-v1-1/.

[30] [Online]. Available: http://www.icasi.org/cvrf-1-1-mindmap/.

[31] [Online]. Available: https://nvd.nist.gov/home.cfm.

[32] [Online]. Available: http://www.osvdb.org/.

[33] [Online]. Available: https://insights.sei.cmu.edu/cert/2012/04/-vulnerability-severity-using-cvss.html.

[34] [Online]. Available: https://qualysguard.qualys.com/qwebhelp/fo_help/setup/cvss_scoring.htm.

[35] [Online]. Available: http://www.cisco.com/web/about/security/intelligence/Cisco_CVSS.html.

[36] [Online]. Available: https://scap.nist.gov/events/2010/itsac/presentations/day1/SCAP_101-CVE_and_CVSS.pdf.

[37] [Online]. Available: http://csrc.nist.gov/publications/nistbul/Oct-2007.pdf.

[38] [Online]. Available: https://www.first.org/cvss.

[39] [Online]. Available: http://www.system-safety.org/Documents/MIL-STD-882E.pdf.

[40] [Online]. Available: http://packetfactory.openwall.net/papers/CVSS/guide/index.html.

[41] [Online]. Available: https://www.first.org/cvss/specification-document.