Comparing and Evaluating CVSS Base Metrics and Microsoft Rating System

[1]Awad A. Younis and [1]Yashwant K. Malaiya
[1]Computer Science Department, Colorado State University, Fort Collins, CO 80523, USA
{younis,malaiya}@cs.colostate.edu

Abstract— **Evaluating the accuracy of vulnerability security risk metrics is important because incorrectly assessing a vulnerability to be more critical could lead to a waste of limited resources available and ignoring a vulnerability incorrectly assessed as not critical could lead to a breach with a high impact. In this paper, we compare and evaluate the performance of the CVSS Base metrics and Microsoft Rating system. The CVSS Base metrics are the de facto standard that is currently used to measure the severity of individual vulnerabilities. The Microsoft Rating system developed by Microsoft has been used for some of the most widely used systems. Microsoft software vulnerabilities have been assessed by both the Microsoft metrics and the CVSS Base metrics which makes their comparison feasible. The two approaches, the technical analysis approach (Microsoft) and the expert opinions approach (CVSS) differ significantly. To conduct this study, we examine 813 vulnerabilities of Internet Explorer and Windows 7. The two software systems have been selected because they have a rich history of publicly available vulnerabilities, and they differ significantly in functionality and size. The presence of actual exploits is used for evaluating them. The results show that exploitability metrics in either system do not correlate strongly with the existence of exploits, and have a high false positive rate.**

Keywords—Severity; Risk assessment; Software Vulnerability; Exploits; CVSS Base Metrics; Microsoft Rating System; Microsoft Exploitability Index; Empirical Software Engineering.

1. INTRODUCTION

Evaluating the risk associated with software vulnerabilities is crucial. In spite of the recent advances in vulnerability avoidance, vulnerability identification and removal, and intrusion prevention, vulnerabilities continue to be a critical threat [1]. Risk evaluation involves assessment of appropriate metrics. A security metric is a quantifiable measurement that indicates the level of security for an attribute of a system [2]. In addition to FIRST, an international confederation, some of the major software developers have developed rating systems for assessing the risk of software vulnerabilities. The rating systems include: CVSS [3] by FIRST, Microsoft [4], IBM ISS X-Force [5], Symantec [6], etc. Each one of them rates vulnerabilities' risk (severity) based on varieties of metrics and obtain a single overall score by assessing these metrics.

The accuracy of such rating systems is very important as they are intended to help decision makers in resource allocation, patch prioritization, program planning, risk assessment, and product and service selection. According to Verendel [7], the lack of validation and comparisons among such metrics makes their usability risky. To that end, in this study, we compare and evaluate the performance of the CVSS Base metrics and Microsoft Rating systems. The system using the CVSS Base metrics has been selected because it is the de facto standard that is currently widely used to measure the severity of individual vulnerabilities. On the other hand, the latter has been chosen because Microsoft software vulnerabilities have been evaluated using both Microsoft's own metrics and CVSS Base metrics and that makes their comparison feasible. Besides, it performs in-depth technical analysis of the vulnerabilities and this helps in comparing the effectiveness of the technical analysis approach (Microsoft) with the expert opinions approach (CVSS). The presence of actual exploits is used for their comparison.

In this study, we have examined 813 vulnerabilities of Internet Explorer browser and Windows 7 operating system. The two software systems have been selected because their vulnerabilities severity has been measured using the two selected rating systems, their rich history of publicly documented vulnerabilities, and their diversity in size and functionality.

This paper is organized as follows. Section 2 presents the related work. In Section 3, the background of the vulnerabilities, vulnerability databases, and the exploit database are discussed. In the following section, the selected vulnerabilities rating systems are discussed. In sections 5, the selected datasets are presented. In section 6, the applicability of CVSS and MS-Exploitability metrics is examined. Section 7 presents the discussion. Finally, concluding comments are given and the issues that need further research are identified.

2. RELATED WORK

A few researchers have started to examine CVSS critically. Bozorgi et al. [8] have studied the exploitability metrics in CVSS Base Score metric. They have argued that the exploitability measures in CVSS Base Score metric cannot tell much about the vulnerability severity. They attributed that to the fact that CVSS metrics rely on expert knowledge and static formulas. They have proposed a machine learning and data mining technique that attempts to predict the possibility of vulnerability exploitation. Bozorgi et al. have used the distributions resulting from the two approaches for evaluation. In contrast, in this paper, we evaluate the performance of CVSS exploitability metrics using well defined performance measures and using the presence of actual exploits to compare it with the performance of MS-Exploitability metric. In addition, we take into consideration the type of software when comparing the performance of the two metrics, as that might reveal some significant insight.

Allodi and Massacci in [9] and [10] have proposed the black market as an index of risk of vulnerability exploitation. Their approach assesses the risk of vulnerability exploitation based on the volumes of the attacks due to the vulnerability exploits sold in the black market. In their study, they conducted a thorough analysis of the CVSS Base metrics: exploitability and impact metrics. They compared CVSS metrics' performance against the existence of exploits in the EDB, Symantec's Attack Signature (wild), and OSVDB by using sensitivity and specificity measures. In contrast, in this paper, we compare CVSS exploitability metrics performance with MS-Exploitability metric.

Eiram in [11] has reviewed the value of the rating systems of the Microsoft Exploitability Index, Adobe Priority Rating system, and CVSS exploitability Temporal metric. The aim was determining if these rating systems are meeting the goal of easing prioritization of applying security updates. To evaluate the rating systems, Eiram counted the number of vulnerabilities that have a high severity value (1) and used it as a method of evaluation. He has suggested that this number should not be high. He only used vulnerabilities that were reported in 2012. In this paper, however, we choose to evaluate CVSS exploitability metrics instead of CVSS exploitability temporal metric because the former is the one

978-1-4673-7989-2/15 $31.00 © 2015 IEEE
DOI 10.1109/QRS.2015.44

that is always reported in well-known vulnerability databases such as NVD. Moreover, we do not only use a larger dataset, but also use well defined performance measures. Moreover, we are using the existence of the exploit as a method of evaluation instead of counting the vulnerabilities that have a high severity value.

3. BACKGROUND

3.1 Vulnerability

A vulnerability is defined as weakness in an asset that might be exploited by an adversary causing loss or harm [12]. More specifically, a software vulnerability is defined as "an instance of [a mistake] in the specification, development, or configuration of software such that the execution can violate the [explicit or implicit] security policy" [13]. Software vulnerability may result from input validation, authentication, authorization, configuration management, exception management, parameter manipulation, or cryptography. A vulnerability is located in an *asset* and if it is not controlled by a *countermeasure*, it could be exploited by a *threat* and hence it leads to a *risk*.

- A **risk** is defined as the potential for *loss* or *harm* as a result of the *likelihood* of unwanted event "threat" and its adverse *consequences* "impact" [14].
- A **threat** is defined as the probability of *an adversary* (threat agent) attempting *an attack* (threat event) and the probability of this attack being successful in exploiting a vulnerability. An *adversary* is "any person or a thing that acts (or has the power to act) to cause, carry, transmit, or support a threat" [15]. *An attack* is an attempt to break into an asset and take harmful actions and it may or may not be successful [12].
- A **countermeasure** is defined as the action, process, tool, device, procedure, technique or a measure that puts in place to reduce a risk of loss [16]. It is used to minimize or eliminate the probability of a threat exploiting a vulnerability in an asset.
- An **asset** is defined as anything that is of a value and importance to the owner, which includes information, programs, data, network, and communication infrastructures [17].

3.2 Vulnerability Databases

The databases for the vulnerabilities are maintained by several organizations such as National Vulnerability Database (NVD) [18], Open Source Vulnerability Database (OSVDB) [19], BugTraq [20], etc., as well as the vendors of the software. Vulnerabilities are assigned a unique identifier using MITRE Common Vulnerability and Exposure (CVE) service. CVE provides a reference-method for publicly known vulnerabilities.

NVD is the U.S. national vulnerability database which supports CVE standards. It is a public data source that maintains standardized information about reported software vulnerabilities. NVD provides information such as vulnerability' CVE number, publish date, location, type, CVSS measures (impact and exploitability), software affected, etc. On the other hand, Microsoft Security Bulletins [21] address security vulnerabilities in Microsoft software. Every bulletin contains a General Information that provides an executive summary and the affected and non-affected software. Besides, the bulletin has the Vulnerability Information section in which the vulnerability CVE number can be found along with a severity rating. From this section, a link to the vulnerability Exploitability Index score is provided. By clicking on this link, information such as bulletin ID, CVE number, Exploitability Index value, and keynotes can be attained.

3.3 Exploit Database (EDB)

An *Exploit* is a method: a piece of software, a chunk of data, or a sequence of commands, that identifies and takes advantage of a vulnerability in an asset [22]. EDB is an exploit database that records exploits and corresponding vulnerable software [23]. It is used by penetration testers, vulnerability researchers, and security professionals. It reports vulnerabilities for which there is at least a proof-of-concept (an exploit). EDB is considered as a regulated market for the exploits. EDB contains around 24075 exploits as the time of writing this paper. Most of its data are derived from the Metasploit Framework (a tool for creating and executing exploit code against a target machine). It provides a search utility that uses a CVE number to find vulnerabilities that have an exploit.

4. VULNERABILITY RATING SYSTEMS

Recently, several rating systems for assessing the severity of computer system security vulnerabilities have been developed. The rating systems include: CVSS [3], Microsoft [4], IBM ISS X-Force [5], Symantec [6], etc. Each one of them rates vulnerabilities' severity based on varieties of metrics and assigns a single overall score by assessing these metrics. They are intended to help decision makers to patch prioritization and risk assessment. In this section, CVSS Base metrics and Microsoft Rating systems that are selected for this study are briefly introduced. The two approaches differ significantly. CVSS depends on the opinions of experts whereas Microsoft's approach depends on the technical factors it considers significant.

4.1 CVSS Base Metrics

CVSS Base Score measures severity based on exploitability (the ease of exploiting a vulnerability) and impact (the effect of exploitation) as shown by the following:

$$\textbf{\textit{Base score}} = \text{Round to 1 decimal } \{[(0.6\times\text{Impact}) + (0.4\times\text{Exploitability}) - 1.5] \times f\,(\text{Impact})\}$$

The formula for the base score, as well as for the exploitability and impact sub-scores are based on expert opinion and are not based on formal derivations. The constants are chosen to yield the maximum values of 10. The base score is rounded to one decimal place and it is set to zero if the impact is equal to zero regardless of the formula. The CVSS scores for known vulnerabilities are readily available in the majority of public vulnerability databases. The CVSS score is a number in the range [0.0, 10.0]. This score represents the intrinsic and fundamental characteristic of a vulnerability and thus the score does not change over time. The two CVSS sub-scores exploitability and impact also range between [0.0, 10.0]. CVSS score from 0.0 to 3.9 corresponds to Low severity, 4.0 to 6.9 to Medium severity and 7.0 to 10.0 to High severity.

The impact sub-score measures how a vulnerability will directly affect an IT asset as the degree of losses in Confidentiality (I_C), Integrity (I_I), and Availability (I_A) as is shown by the following:

$$\textbf{\textit{Impact}} = 10.41 * (1 - (1 - I_C) * (1 - I_I) * (1 - I_A))$$

The impact sub-scores, on the other hand, are all assessed in terms of None (N), Partial (P), or Complete (C) *by security experts* and assigned one of the mentioned qualitative letter grades. Exploitability, on the other hand, is assessed based on three metrics: Access Vector (AV), Authentication (AU), and Access Complexity (AC) as is shown by the following:

$$\textbf{\textit{Exploitability}} = 20 \times AV \times AU \times AC$$

The AV reflects how a vulnerability is exploited in terms of local (L), adjacent network (A), or network (N). The AC measures the complexity of an attack required to exploit the vulnerability (once an attacker has gained an access to a target system) in terms of High (H), Medium (M), or Low (L). The AU counts the number of times an attacker must authenticate to reach a target (in order to exploit a vulnerability) in terms of Multiple (M), Single (S), or None (N). The CVSS system uses lookup tables to provide the numerical values needed for the subscores.

4.2 Microsoft Rating System

Microsoft (MS) rating system measures vulnerabilities' risk based on two variables: *Impact* and *Probability* [4]. The latter means the potential effect of a vulnerability being exploited, and the former means the likelihood of that exploitation taking place. These two variables are assessed using severity rating and Exploitability Index.

The impact factor, on one hand, is captured using the *severity rating* that indicates the "worst case" scenario of an attack that exploits a vulnerability. The Impact factor can take one of the following values: *Critical* (A vulnerability whose exploitation could allow code execution without user interaction), *Important* (A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of user data, or of the integrity or availability of processing resources), *Moderate* (Impact of the vulnerability is mitigated to a significant degree by factors such as authentication requirements or applicability only to non-default configurations), and *Low* (Impact of the vulnerability is comprehensively mitigated by the characteristics of the affected component).

On the other hand, the probability factor is assessed using the *Exploitability Index* that measures the likelihood that a specific vulnerability would be exploited within the first 30 days after bulletin release. The Exploitability Index can be assigned a score 1, 2, and 3 for any vulnerability with a severity rate of *Important or Critical*. Here, 1 means Consistent exploit code is likely, 2 means Inconsistent exploit code is likely, and 3 means Function exploit code unlikely.

5. DATASETS

5.1 Data Collection

In this study, the data about vulnerabilities and exploits of Microsoft Windows 7 and Internet Explorer (IE) during the period January 2009 to October 2014 were collected. The reason why we considered collecting data starting from 2009 is because the Microsoft Exploitability Index is introduced in October 2008 and has been included in Microsoft Bulletin starting 2009. This data was collected as follows. First, the vulnerabilities and their metrics' values were collected from NVD [18] and Microsoft Security Bulletin [21]. Second, the exploits were collected from EDB [23]. Table 1 shows the number of the selected vulnerabilities and their exploits.

Table 1: Internet Explorer and Windows 7 Vulnerabilities

Software	Exploit Exist	No Exploit Exist	Total Each
IE	33	436	459
Windows 7	52	302	354
Total All	**85**	**738**	**813**

It should be noted that the total number of the IE vulnerabilities is 482. However, out of the 482, 23 vulnerabilities were not selected because we could not find information about their CVSS Base metrics values or MS Exploitability Index values.

- Three out of the 23 vulnerabilities (CVE-2014-4066, CVE-2014-4112, and CVE-2014-4145) could not be found in the NVD even though they have CVE number.
- For the remaining 20 vulnerabilities, we could not find their MS Exploitability Index values. These vulnerabilities are shown in Table 2.
- None of these unselected vulnerabilities has an exploit.

Table 2: Unselected Vulnerabilities in Microsoft Internet Explorer

CVE-2010-0808	CVE-2010-3327	CVE-2010-3342	CVE-2010-3348
CVE-2012-0010	CVE-2011-1244	CVE-2011-1246	CVE-2011-1258
CVE-2011-1962	CVE-2011-2383	CVE-2011-3404	CVE-2011-0038
CVE-2012-0168	CVE-2012-1872	CVE-2012-1882	CVE-2013-3126
CVE-2013-3186	CVE-2013-3192	CVE-2014-2817	CVE-2014-4123

It should also be noted that the total number of vulnerabilities of Windows 7 is 380. However, out of the 380 vulnerabilities, 26 vulnerabilities were unselected because we could not find their MS Exploitability Index values. Out of the 26 vulnerabilities, four vulnerabilities that have an exploit were removed (CVE-2010-1890, CVE-2010-1887, CVE-2010-2554, and CVE-2010-3227). Table 3 shows these 26 vulnerabilities.

Table 3: Unselected Vulnerabilities in Microsoft Windows 7

CVE-2010-0252	CVE-2010-0481	CVE-2010-0811	CVE-2010-1890
CVE-2010-1887	CVE-2010-2554	CVE-2010-3227	CVE-2010-0811
CVE-2011-1971	CVE-2011-1978	CVE-2011-2002	CVE-2011-2004
CVE-2011-3415	CVE-2012-0156	CVE-2012-0174	CVE-2012-1850
CVE-2012-1851	CVE-2012-2531	CVE-2013-0013	CVE-2013-1291
CVE-2013-1293	CVE-2013-1336	CVE-2013-1337	CVE-2013-3172
CVE-2013-2556	CVE-2014-0295		

The attributes of every selected vulnerability were collected using the following steps.

1. From Microsoft Security Bulletin, the vulnerabilities' CVE numbers for Windows 7 and IE were collected.
2. Next, for every existed CVE number in Microsoft Security Bulletin, we collected the vulnerability' severity rating and Exploitability Index values.
3. Then, we searched for the same vulnerabilities' CVE numbers found in Microsoft Security Bulletin in the NVD.
4. After that, for every vulnerability's CVE number found in the NVD, the CVSS' impact Subscore and exploitability Subscore values were collected.
5. Lastly, for every selected vulnerability we used the CVE number to verify whether it has an exploit reported in the EDB or not.

Table 4 shows a part of the selected vulnerabilities because showing the whole datasets is limited by the number of pages allowed. Unlike CVSS Base Score, Microsoft rating system does not provide the total value of a vulnerability risk, but rather it provides an individual value and let the person in charge to make the decision based on the provided values. It has been noticed that for some vulnerabilities the severity rating and the Exploitability Index are assigned a combination of values instead of one value, for example Critical/Moderate or Important / Low, or, 1/2 or 1/3. This is because some vulnerabilities exist in older versions of Microsoft products that are no longer supported by Microsoft security updates, and hence are assigned a higher severity rating or

Table 4: The obtained measures of Microsoft Rating System and CVSS Base Score Metrics

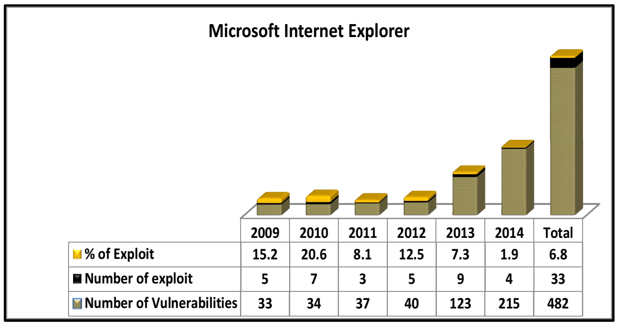
CVE	MS Security Bulletin	Microsoft		CVSS Base Score			Exploit Existence
		Severity Rating	*Exploitability Index*	*Impact Subscore*	*Exploitability Subscore*	*Total*	
CVE-2009-1547	MS09-054	Critical	2	10	8.6	9.3	EE
CVE-2010-0492	MS10-018	Critical/Moderate	1	10	8.6	9.3	NEE
CVE-2011-1992	MS11-099	Important /Low	3	2.9	8.6	4.3	NEE
CVE-2012-1858	MS12-037	Important /Low	3	2.9	8.6	4.3	EE
CVE-2013-1312	MS13-037	Critical/Moderate	2	10	8.6	9.3	NEE
CVE-2014-4141	MS14-056	Critical/Moderate	1	10	8.6	9.3	NEE

Exploitability Index values. On the other hand, the new releases of Microsoft products have a regular security updates and hence are assigned a lower severity rating or Exploitability Index values.

5.2 Analysis of Vulnerabilities and Exploits

5.2.1 Internet Explorer Dataset

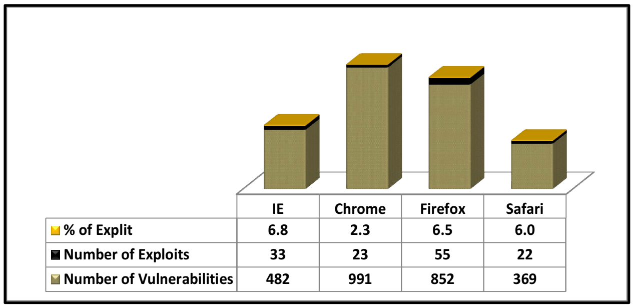
Fig.1 shows the distribution of the vulnerabilities and their exploits of IE from 2009 to 2014. As can be seen, the year 2010 has the highest percentage of exploit, which is a result of dividing the number of reported exploits per year by the total number of vulnerabilities reported in that year. Besides, a noticeable increase in the number of reported vulnerabilities has been noticed in 2013 and 2014. This could be attributed to *Microsoft Bounty Programs* that has started in 2013. It should be noted that there are more vulnerabilities than exploits. According to [24], Cybercriminals are more likely to exploit vulnerabilities that don't need any special conditions, or that offer particularly dangerous opportunity to execute malware on the compromised computer.



	2009	2010	2011	2012	2013	2014	Total
🟨 % of Exploit	15.2	20.6	8.1	12.5	7.3	1.9	6.8
■ Number of exploit	5	7	3	5	9	4	33
▨ Number of Vulnerabilities	33	34	37	40	123	215	482

Figure 1: Distribution of the number of reported vulnerabilities and exploits and percentage of exploits.

The total percentage of the exploits in relation to the number of reported vulnerabilities is 6.8. To evaluate this number based on IE' counterparts web browsers, we collected the vulnerabilities and exploits from 2009 to 2014 for Google Chrome, Firefox, and Apple Safari from NVD and EDB and the result is shown in Fig.2. As can be seen from Fig.2, the total number of reported vulnerabilities in Chrome and Firefox are the highest and this could be attributed to their vulnerability *Bounty Program.* Even though Chrome has the highest number of vulnerabilities, it has the lowest exploit percentage. This could be attributed to the quick patching, which in turn complicate the exploitation process. Although investigating this observation is really important, it is considered beyond the scope of this paper.

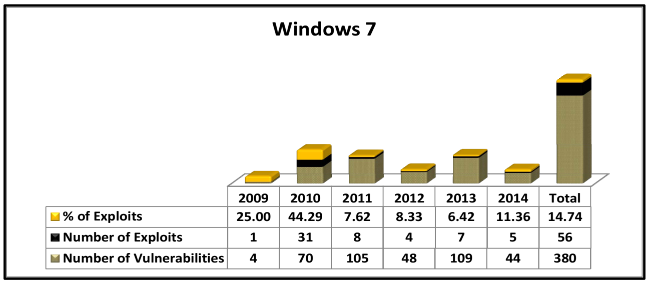


	IE	Chrome	Firefox	Safari
🟨 % of Explit	6.8	2.3	6.5	6.0
■ Number of Exploits	33	23	55	22
▨ Number of Vulnerabilities	482	991	852	369

Figure 2: Web Browsers distribution of the number of reported vulnerabilities and exploits and the percentage of exploits.

5.2.2 Windows 7 Dataset

The distribution of the vulnerabilities and exploits for Windows 7 is shown in Fig.3. As can be seen, 2010 has the most number of exploits, 31. We did some investigation and found out that some of the vulnerabilities that were discovered in Windows Vista are inherited by Windows 7 and this is because of the code reuse concept. Hence, Windows Vista vulnerabilities' exploit can also be used to exploit the inherited vulnerabilities in Windows 7. However, more vulnerability was reported in 2011 and 2013. According to [25], the reason behind the increase in 2011 is an effort of one researcher who looked at win32k.sys and discovered 20 vulnerabilities in 2010 and 59 in 2011. On the other hand, the increase in 2013 could be explained by Microsoft Bounty Programs.



	2009	2010	2011	2012	2013	2014	Total
🟨 % of Exploits	25.00	44.29	7.62	8.33	6.42	11.36	14.74
■ Number of Exploits	1	31	8	4	7	5	56
▨ Number of Vulnerabilities	4	70	105	48	109	44	380

Figure 3: Windows 7 distribution of the number of reported vulnerabilities and exploits and the percentage of exploits.

The total percentage of the exploits in Windows 7 is 14.74. To evaluate this number, we collected the reported vulnerabilities and exploits from 2009 to 2014 for the Mac OS X from NVD and EDB databases. The reason why we selected Mac OS X is because of its market share and popularity. Fig.4 shows that even though the number of the reported vulnerabilities in Mac OS X is more than

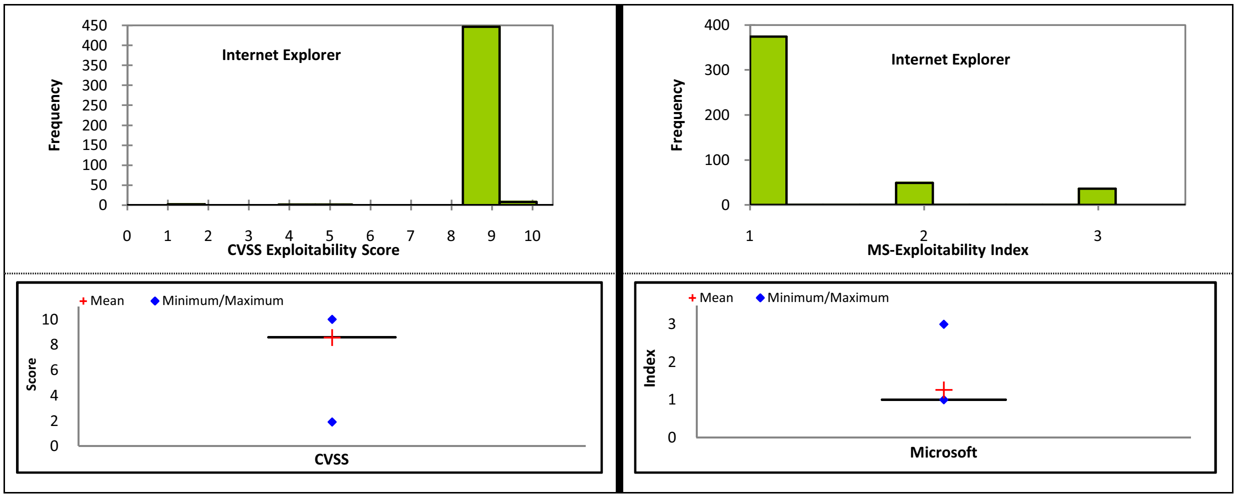
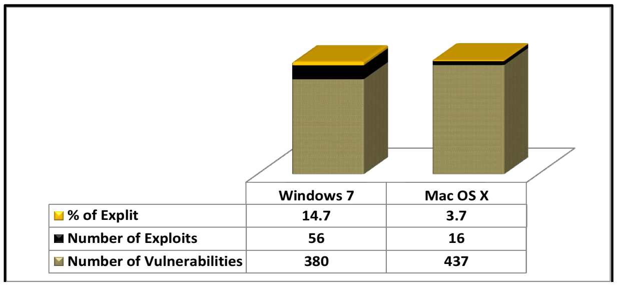


Figure 5: The histograms on the top represent the frequency distribution of the CVSS and Microsoft Exploitability values. The boxplots on the bottom describe the distribution of values around the median, which represented by a horizontal line.

Windows 7, the former has a higher number of exploit percentages. One possible explanation could be that Windows 7 has larger market share (40.81%) than Mac OS X (6.64%) in Desktop and Laptop computers [26], which could cause more impact and in turn attracts attackers as a target.



	Windows 7	Mac OS X
🟨 % of Explit	14.7	3.7
■ Number of Exploits	56	16
▨ Number of Vulnerabilities	380	437

Figure 4: Microsoft Windows 7 and Mac OS X number of reported vulnerabilities and exploits and the percentage of exploit.

5.3 *Analysis of Vulnerabilities' Rating Systems*

The selected rating systems, CVSS Base Score and Microsoft Rating system, measure vulnerability severity based on two factors the *impact of exploitation* and the *possibility of exploitation*. In this section we will first explore their exploitability values and then investigate their impact measures using the two selected datasets.

5.3.1 Exploitability Factor

Fig.5 shows the distribution of the exploitability values for CVSS and Microsoft for the IE dataset. For CVSS exploitability score, almost all vulnerabilities have a value between eight and nine (0.97, relative frequency), whereas for Microsoft Exploitability Index the values are almost one (0.82, relative frequency). Even though Microsoft Exploitability Index shows some variations, looking at the boxplot in Fig.5, it is clear that the distribution of the two metrics is indistinguishable for IE dataset. The median is 8.6 for CVSS and 1 for Microsoft. This shows that the exploitability factor for both metrics is almost a constant and not a variable. It should be noted that in the CVSS boxplot the minimum and maximum points are plotted as outliers by the software we used to create them (XLSTAT) whereas in Microsoft boxplot only the maximum point is plotted as an outlier.

Fig.6 illustrates the distribution of the exploitability values for CVSS and Microsoft for Windows 7 dataset. Unlike IE dataset, the distribution varies for the two metrics. For the CVSS Exploitability scores, almost half of vulnerabilities have a Low (0 to 3.9) exploitability values, whereas the other half has a High (7 to 10) exploitability values. Only a few vulnerabilities have a Medium

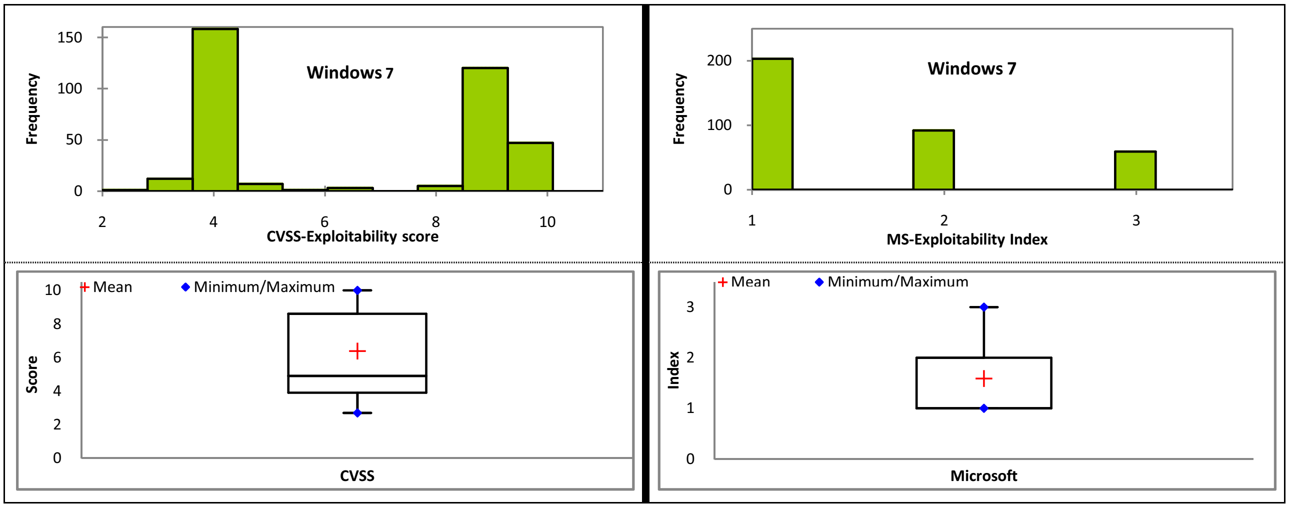


Figure 6: The histograms on the top represent the frequency distribution of the CVSS and Microsoft Exploitability values. The boxplots on the bottom describe the distribution of values around the median, which represented by a horizontal line.

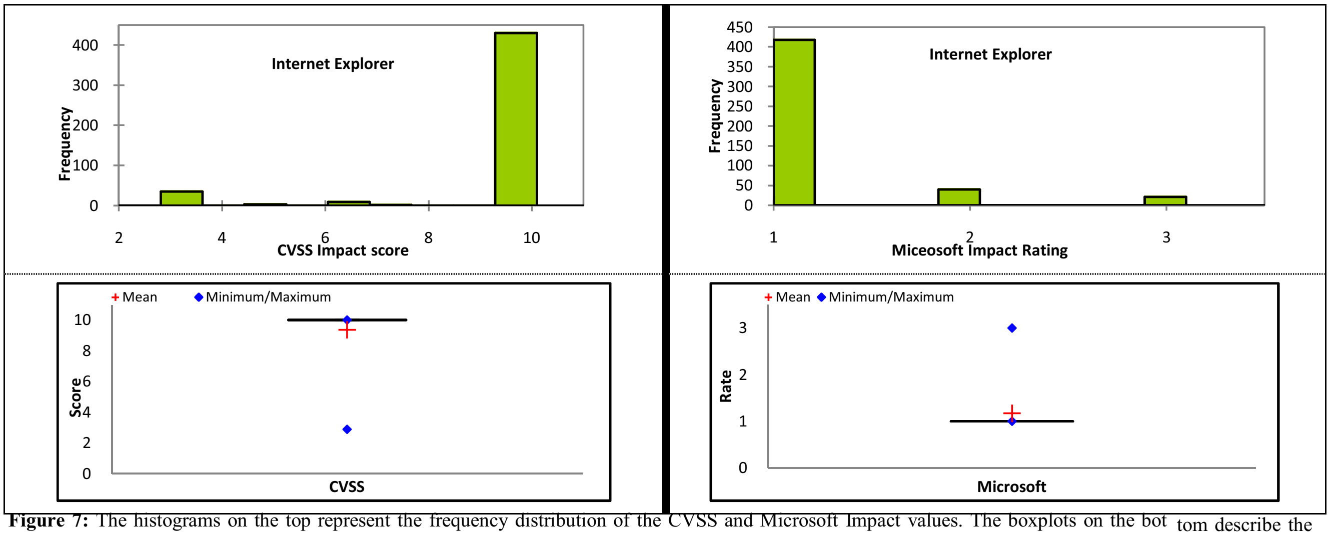


Figure 7: The histograms on the top represent the frequency distribution of the CVSS and Microsoft Impact values. The boxplots on the bottom describe the distribution of values around the median, which represented by a horizontal line.

(4 to 6.9) exploitability values. On the other hand, for the Microsoft Exploitability Index almost all vulnerabilities have an exploitability value between 1 and 2, which means exploit is likely. Only a few have an exploit value 3 (around 59), which means exploit is unlikely. The variability of the exploitability factor for both metrics is apparent in the reported boxplot.

To understand why the two metrics show completely different results when applied to different software type's datasets, we decompose the CVSS exploitability metrics. As can be seen in Table 5, we find that almost half of the vulnerabilities (172) in the Windows 7 dataset have a Local AV value and that has led to a Low CVSS exploitability subscore. On the other hand, there are only three vulnerabilities in the IE dataset that have a Local AV value. It has also been noted that the AV has a significant effect on the CVSS exploitability subscore, regardless of the values of the AU and AC. This can be clearly observed from IE dataset where almost all the vulnerabilities have a Network AV value and hence have been assigned High CVSS exploitability Subscore. It has also been observed that almost all the vulnerabilities in Windows 7 (94.19%) that have a Local AV value also have a Low AC value. This explains the increase in the number of vulnerabilities (60.45%) that have a Low AC.

Table 5: CVSS Exploitability Metrics Subscore for IE and Windows 7

Exploitability Metrics	Value	IE (%)	Windows 7 (%)
AV	Network	99.35	**51.41**
	Adjacent	0	0
	Local	**0.65**	**48.59**
AU	None	**100**	**95.76**
	Single	0	3.95
	Multiple	0	0
AC	High	1.31	1.98
	Medium	**98.04**	37.29
	Low	0.65	**60.45**

5.3.2 Impact Factor

Fig.7 shows the distribution of the Impact values for CVSS and Microsoft for the IE dataset. For CVSS Impact score, almost all vulnerabilities have a value between nine and ten (0.90, relative frequency). On the other hand, Microsoft Impact rating score values are one for almost all vulnerabilities (0.87, relative frequency). It should be noted that, for the sake of comparison, Microsoft Impact values have been mapped into numbers as follows: Critical=1, Important=2, Moderate=3 and Low=4. The medians as shown in boxplot are ten for the CVSS impact values and one for Microsoft Impact values. This shows that the impact values for both metrics for almost all vulnerabilities are high. This can be explained by the fact that 428 (93.24%) vulnerabilities are of the type Execute Code.

Table 6: CVSS Impact factor Metrics for IE and Windows 7

Software	Value	Complete (%)	Partial (%)	None (%)
IE	Confidentiality (C)	92.81	5.66	1.53
	Integrity (I)	92.59	3.05	4.36
	Availability (A)	92.81	1.96	5.23
Windows 7	Confidentiality (C)	87.29	3.67	8.19
	Integrity (I)	76.27	2.54	20.34
	Availability (A)	81.92	2.54	14.69

As it can be seen from Fig.8, more than half of the vulnerabilities have been assigned a high (10) CVSS impact values (0.72, relative frequency), whereas more than half of the vulnerabilities have been assigned an Important (2) Microsoft impact values (0.68, relative frequency). To understand the difference in the impact values between the two metrics, we decomposed the CVSS impact Metrics as shown in Table 6. The CIA values are dominated by the value *complete* for the majority of vulnerabilities especially for IE. We also find that around 270 vulnerabilities (76.27%) that have been assigned a value complete

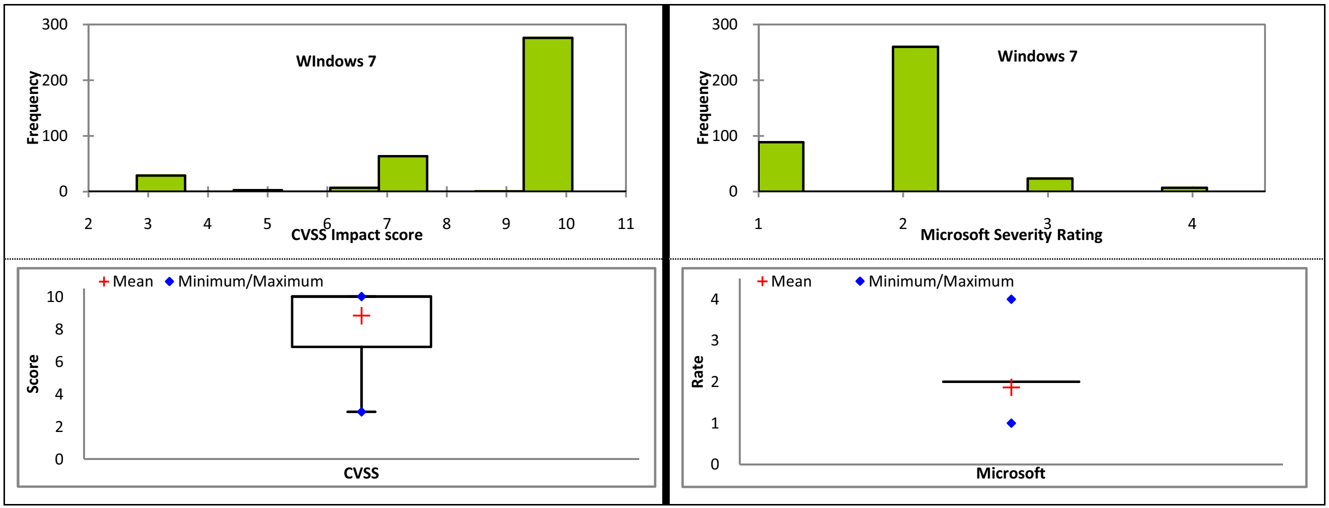


Figure 8: The histograms on the top represent the frequency distribution of the CVSS and Microsoft Impact values. The boxplots on the bottom describe the distribution of values around the median, which represented by a horizontal line.

for their C, I, and A metrics in Windows 7. We also find that there are 180 vulnerabilities of the type Gain Privilege and 169 of them have Microsoft impact value 2 (93.89%). The reason why Microsoft has relaxed the impact value of this type of vulnerabilities is possibly because this type of vulnerabilities depends on user configurations, which is most of the time have fewer user rights on the system.

6. VALIDATION OF CVSS AND MS-EXPLOITABILTY METRICS

Since data is available about the existence of exploits, we can only evaluate the Microsoft Exploitability Index metric with the CVSS exploitability metrics based on the availability of exploits. Fig.9 shows the CVSS and Microsoft Exploitability measures for IE and Windows 7 vulnerabilities against the Existence of Exploit (EE) and No-Exploit Existence (NEE). For the IE dataset, the CVSS exploitability median is 8.6 for both vulnerabilities with EE and for those with NEE. Besides, the median for the Microsoft Exploitability Index is 1 for both classes (EE, NEE). On the other hand, for the Windows 7 dataset, the CVSS exploitability median is 3.9 for vulnerabilities with NEE and 8.6 for vulnerabilities with EE. Moreover, the median for the Microsoft Exploitability Index is 1 for both classes (EE, NEE).

6.1 Methodology

To evaluate the performance of these two metrics, we used statistical measures termed sensitivity, precision, and F-measure. These measures are explained using a *confusion matrix* as shown in Table 7. The confusion matrix table shows the actual vs. the predicted results. It should be noted that a vulnerability is considered exploitable if it has a reported exploit, but that does not mean the otherwise. For the two class problem (a vulnerability is either exploitable or not exploitable), the following is defined based on Table 7.

Table 7: Confusion matrix

		Prediction	
		Exploitable	***Not Exploitable***
Actual	**Exploitable**	TP= True Positive	FN= False Negative
	Not Exploitable	FP= False Positive	TN= True Negative

- True Positive (TP): the number of the vulnerabilities predicted as exploitable, which do in fact have an exploit.

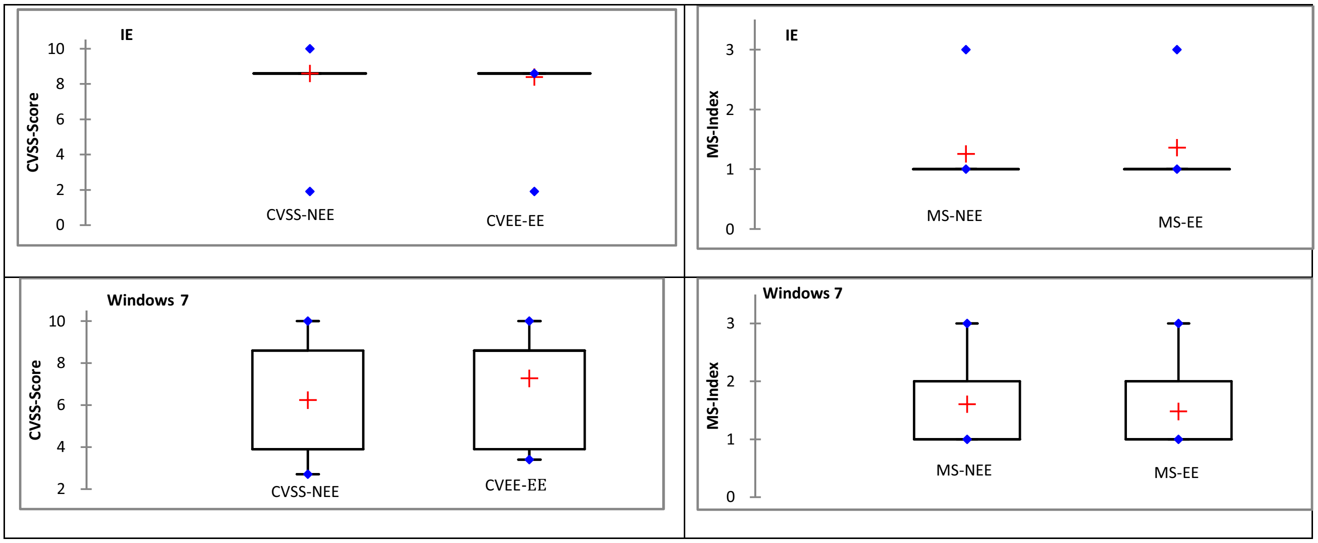


Figure 9: Vulnerability Exploitability measures using CVSS in the Left column and MS-Exploitability Index in the right Column compared to Exploit Exist (EE) and No-Exploit Exist (NEE) for IE and Windows 7 datasets.

- False Negative (FN): the number of vulnerabilities predicted as not exploitable, which turn out to have an exploit**.**
- False Positive (FP): the number of vulnerabilities predicted as exploitable when they have no exploit.
- True Negative (TN): the number of vulnerabilities predicted as not exploitable when there is no exploit.

The selected performance measures can be derived as follows.

6.1.1 Sensitivity (Recall)

Sensitivity, which also termed recall, is defined as the ratio of the number of vulnerabilities correctly predicted as exploitable to the number of vulnerabilities that are actually exploitable as shown by the following:

$$Sensitivity = \frac{TP}{TP + FN}$$

6.1.2 Precision

Precision, which is also known as the correctness, is defined as the ratio of the number of vulnerabilities correctly predicted as exploitable to the total number of vulnerabilities predicted as exploitable as shown by the following:

$$Precision = \frac{TP}{TP + FP}$$

For convenient interpretation, we express these two measures in terms of percentage, where a 100% is the best value and 0% is the worst value. Both precision and sensitivity should be as close to the value 100 as possible (no false positives and no false negatives). However, such ideal values are difficult to obtain because sensitivity and precision often change in opposite directions. Therefore, a measure that combines sensitivity and precision in a single measure is needed. Hence, we will introduce the F-measure in the following section. We believe that it is more important to identify exploitable vulnerabilities even at the expense of incorrectly predicting some not exploitable vulnerabilities as exploitable vulnerabilities. This is because a single exploitable vulnerability may lead to serious security failures. Having said that, we think more weight should be given to sensitivity than precision. Thus, we include F2-measure, which weights sensitivity twice as precision, to evaluate the two metrics.

6.1.1 F-measure

F-measure can be interpreted as the weighted average of sensitivity and precision. It measures the effectiveness of a prediction with respect to a user attached β times as much importance to sensitivity as precision. The general formula for the F-measure is shown by the following:

$$F_{\beta} - \text{Measure} = \frac{(1 + \beta^2) \times Precision \times Senetivity}{(\beta^2 \times Precision) + Senetivity}$$

β is a parameter that controls a balance between sensitivity and precision. When β = 1, F-measure becomes to be equivalent to the harmonic mean whereas when β < 1 it becomes more precision oriented. However, when β > 1, F-measure becomes more sensitivity oriented. In this paper β has been chosen to be 2.

6.2 Results

To calculate the above mentioned performance measures we need to obtain the confusion matrix for the two datasets. Using the data about the availability of exploits and the exploitability measures for CVSS exploitability metrics and Microsoft Exploitability Index, the confusion matrix was determined as shown in Table 8. Using the values in this matrix, the performance measures have been calculated as shown in Table 9. It should be noted that there is an imbalance in the two datasets. For instance, there are 33 vulnerabilities with an exploit compared to 436 vulnerabilities without an exploit in the IE dataset, whereas there are 52 vulnerabilities with an exploit and 302 vulnerabilities without an exploit in the Windows 7 dataset. We considered this imbalance in our performance analysis. For every dataset, we selected all vulnerabilities that have an exploit and at the same time we randomly (using random with replacement technique) selected the same number of vulnerabilities from those that have no exploit and calculated the performance measures and the results are shown in Table 9.

Table 8: Confusion matrix of CVSS exploitability metrics and Microsoft Exploitability Index

Internet Explorer			
CVSS		**Prediction**	
		Exploitable	***Not Exploitable***
Actual	**Exploitable**	TP= 32	**FN= 1**
	Not Exploitable	FP= 423	TN= 3
Microsoft Exploitability Index		**Prediction**	
		Exploitable	***Not Exploitable***
Actual	**Exploitable**	TP= 28	**FN= 5**
	Not Exploitable	FP= 395	TN= 31
Windows 7			
CVSS		**Prediction**	
		Exploitable	***Not Exploitable***
Actual	**Exploitable**	TP= 34	**FN= 18**
	Not Exploitable	FP= 141	TN= 161
Microsoft Exploitability Index		**Prediction**	
		Exploitable	***Not Exploitable***
Actual	**Exploitable**	TP= 43	**FN= 9**
	Not Exploitable	FP= 253	TN= 49

From the IE dataset and when the whole sample is considered, it is clear that the two metrics have a high sensitivity values and that comes at the cost of having a very low precision. It is also apparent that the false positive rate is very high, which is almost a 100% for CVSS exploitability metrics and around 93% for Microsoft Exploitability Index. This makes the two metrics behave like a random predictor. Looking at CVSS exploitability metrics values, the high positive rate can be explained by the fact that almost all the vulnerabilities in IE dataset have a Network AV value and hence have been assigned a high CVSS exploitability value and that has led to assessing those vulnerabilities as exploitable. On the other hand, when the imbalanced sample is considered, it can be seen that the precision of the two metrics has dramatically changed and that is in turn has changed the F1 and F2 measures too, which expected as both of them rely on the precision value. Besides, the false positive rate for Windows Exploitability Index has noticeably reduced.

Looking at the performance of the two metrics using the whole sample of Windows 7 dataset, it is clear that both metrics performed differently when compared to their performance using IE dataset. This difference is more apparent in CVSS exploitability

Table 9: Prediction Performance of CVSS Exploitability Metrics and Microsoft Exploitability Index

Software	Performance Measures	CVSS Exploitability Metrics		MS-Exploitability Index	
		Whole Sample (%)	*Balanced Sample (%)*	*Whole Sample (%)*	*Balanced Sample (%)*
IE	**Sensitivity**	97	97	85	85
	Precision	7	**50**	7	**57**
	F1-Measure	13	33	12	34
	F2-Measure	27	**82**	25	**77**
	False Positive Rate	99.30	97	92.70	**64**
Windows 7	**Sensitivity**	65.38	65	82.69	82
	Precision	19.43	**50**	14.53	**50**
	F1-Measure	29.96	29	24.71	31
	F2-Measure	44.39	**62**	42.66	**73**
	False Positive Rate	46.69	**62**	83.77	81

metrics. First, the false positive rate has significantly dropped to less than 50%. This could be explained by the fact that (unlike the IE dataset where almost all the vulnerabilities have a Network AV value and hence have been assigned a high exploitability value) in Windows 7 almost half of the vulnerabilities have been assigned Local AV values and hence have been assigned a low exploitability value. In other words, CVSS exploitability factor is highly influenced by the AV values. Second, the sensitivity has noticeably dropped and this is 6because there are 18 vulnerabilities that have been assessed as not exploitable and they turn out to have an exploit. However, when the imbalanced sample is taken into account, it can be seen that the precision of the two metrics has noticeably changed and that is in turn has also changed F2 measure. Besides, F1 measure slightly decreased and that is because of the drop in the sensitivity of the CVSS.

From the performance analysis, 6the following has been observed.

- The sensitivity measure of the Microsoft Exploitability Index has not been affected by the change of the type of software.
- The sensitivity measure of CVSS exploitability measure has been noticeably affected by the change of software type. This has led to a change in AV values and that in turn has made CVSS exploitability measure predicts 18 vulnerabilities as not exploitable where is in fact there are exploits for those vulnerabilities.
- Both metrics are very sensitive and that has led to a lower precision and a high false positive rate and this could lead to a waste of resources and effort.
- CVSS exploitability factor is highly influenced by the AV values regardless of the other two factors, AC and AU.
- Taking into consideration the imbalance in the datasets between the number of vulnerabilities with an exploit and those without an exploit has shown an improvement in the precision and F2 measures for the two metrics.
- It is unclear how the Microsoft Exploitability Index is assessed.

6.3 Threats to Valididity

In this paper, we have considered the datasets for only two products, Internet Explorer and Windows 7. However, the two selected software have a rich history of reported vulnerabilities and exploits (IE has 436 reported vulnerabilities, 33 with reported exploits, and Windows 7 has 354 reported vulnerabilities, 52 with reported exploits). One of our observations is that the false positive rate is high for both metrics. This could be a result of other factors that have not been considered in this study. Only publicly reported vulnerabilities and exploits have been considered here.

7. DISCUSSION

An important question that arises from the results of this study is *why both the technical approach (Microsoft) and the expert opinion approach (CVSS) did not perform well*? One possible reason could be that some of the chosen metrics do not correlate well with the factors that contribute to vulnerability exploitability in actual reality and hence new metrics are needed to be identified and added to the two rating systems. It will require extensive investigations to identify new metrics that well correlated. There may be some randomness in how potential exploit developers identify the vulnerabilities for which exploit development may be worthwhile. It is also possible that the software developers may work more aggressively towards developing patches for highly exploitable vulnerabilities making it less attractive for external exploit developers to develop exploits. However the results suggest that these may not provide the complete explanation for why the exploitability measures have not performed well. Another possible reason could be that the two metrics did not carefully consider the threat (the *external* factor) and mainly focus on the *internal* factors, which are alone not enough to capture the whole risk presented by a vulnerability. Looking at the formal theory of risk, we find that risk is defined as [27]:

$$\boldsymbol{Risk} = \text{Likelihood of an adverse event} \times \text{Impact of the adverse event}$$

$$\boldsymbol{Likelihood\ of\ an\ Adverse\ Event} = \text{Threat} \times \text{R\_Vulnerability}$$

$$\boldsymbol{Risk} = \text{Threat} \times \textbf{R}\text{\_Vulnerability} \times \text{Impact}$$

What the two rating systems have considered are the R_vulnerability (in the risk theory the term “vulnerability” is a number given by the probability of an attack’s success or an ease of exploitation) and the impact (losses that occur given a successful exploitation) and they assume the threat as either there or not adaptive, whereas a threat (an adversary), unlike accidents or acts of nature, is intelligent and may dynamically adapt to the used defensive measures. Therefore, vulnerabilities are dangerous if and only if someone (adversary) is interested in exploiting them (motive) and has the means (capability) to do it and that is shown by [28].

$$\boldsymbol{Threat}_{\boldsymbol{(Attacker)}} = \boldsymbol{Motive \times Capability}$$

A motive is a measure of how far an adversary is willing to go and what he is willing to risk to reach his objectives. A motive can be influenced by the sensitivity of data, desire for monetary gain, or the potential publicity effects of an attack. One way of measuring a motive in cyber security is based on the following factors [29]: a cost of attempting an attack, a payoff of a successful attack, a probability of successfully completing an attack, and a probability of detection. A capability, on the other hand, is the degree to which an adversary is able to execute an attack. To execute an attack, the skills, knowledge, tools, and techniques should be possessed by an adversary. According to [30], the following factors are a key in measuring a capability of a hacker or cracker: group size, history of relevant activity, technical expertise, and target selection. Some of the capability factors have been considered to a limited extent, especially by MS rating system. Quantifying and including the attackers' motive and some other capability factors as a part of the two studied rating systems may significantly advance the vulnerability risk assessment by increasing the accuracy and reducing the false positive rate.

8. CONCLUSION AND FUTURE WORK

This study compares and evaluates the performance of the CVSS Base metrics and Microsoft rating system using 813 vulnerabilities of IE and Windows7. The results show that the two measures have a very high false positive rate. It was observed that the sensitivity measure of CVSS exploitability metrics is noticeably affected by the software type. Besides, CVSS Exploitability factor is highly influenced by the AV values regardless of the other two factors (AC and AU). However, unlike the CVSS Base metrics where the metrics (factors) used for measuring vulnerabilities' risk are provided, Microsoft rating system does not provide such metrics but rather provides the values and their definition. Hence it was hard to conduct a thorough investigation trying to correlate the two sets of metrics.

Even though the two selected software have a rich history of reported vulnerabilities, considering other Microsoft products could increase the size of the dataset and that might reveal significant information. The study suggests that a simple measure of vulnerabilities exploitability using few metrics may not be sufficient. Hence, identifying new metrics that capture attributes that have not been yet considered and adding them to the two rating systems is needed. Younis et al. in [31] have proposed some distinctive metrics based on the software structure. In addition, identifying and including the external factors, such the attacker behavior, to the two selected rating system could improve their precision and reduce their false positive rate.

REFERENCES

[1] S. Farrell, "Why didn't we spot that? [Practical Security]," Internet Computing, IEEE, vol. 14, no. 1, pp. 84 –87, Feb. 2010.

[2] W. Jansen, "Directions in security metrics research," NIST, NISTIR 7564, 2009, pp.1-26.

[3] P. Mell, K. Scarfone, and S. Romanosky, "A complete guide to the common vulnerability scoring system version 2.0," Published by FIRST-Forum of Incident Response and Security Teams, 2007, pp.1–23.

[4] MSRC, "Microsoft security response center security bulletin severity rating system,"Available: https://technet.microsoft.com/en-us/security/ff943560.aspx. [Accessed: 24-March-2015].

[5] X-Force, "X-Force frequently asked questions," Available: http://www-935.ibm.com/services/us/iss/xforce/faqs.html. [Accessed: 24-March-2015].

[6] "Symantec Security Response, Threat severity assessment," Available: http://www.symantec.com/avcenter/threat.severity.html. [Accessed: 24-March-2015].

[7] V. Verendel, "Quantified security is a weak hypothesis: a critical survey of results and assumptions," In Proceedings of the 2009 workshop on new security paradigms workshop NSPW 09, 2009, pp. 37–50.

[8] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond heuristics: learning to classify vulnerabilities and predict exploits," in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010, pp. 105–114.

[9] L. Allodi and F. Massacci, "A preliminary analysis of vulnerability scores for attacks in wild," ACM Proc. of CCS BADGERS, 2012, pp.17-24.

[10] L. Allodi and F. Massacci, "My Software has a Vulnerability, should I worry?,", 2013 , arXiv preprint arXiv:1301.1275.

[11] C. Eiram, "Exploitability/Priority Index Rating Systems (Approaches, Value, and Limitations)," 2013. Available: https://www.riskbasedsecurity.com/reports/RBS-ExploitabilityRatings-2013.pdf. [Accessed: 24-March-2015].

[12] C. P. Pfleeger and S. L. Pfleeger, Security in computing. Prentice Hall PTR, 2006.

[13] A. Ozment, "Improving vulnerability discovery models," in Proceedings of the 2007 ACM workshop on Quality of protection, 2007, pp. 6–11.

[14] L. A. T. Cox Jr, "Some limitations of 'Risk = Threat x Vulnerability x Consequence' for risk analysis of terrorist attacks," Risk Anal., vol. 28, no. 6, pp. 1749–1761, Dec. 2008.

[15] L. Marinos and A. Sfakianakis, "ENISA surveys evolving threat landscape," Computer Fraud & Security, vol. 2013, no. 1, pp. 1–3, Jan. 2013.

[16] M. Sahinoglu, Trustworthy Computing: Analytical and Quantitative Engineering Evaluation. John Wiley & Sons, 2007.

[17] C. Onwubiko and A. P. Lenaghan, "Managing Security Threats and Vulnerabilities for Small to Medium Enterprises", In Intelligence and Security Informatics, 2007, pp. 244-249.

[18] "National Vulnerability Database Home," Available: http://nvd.nist.gov/. [Accessed: 24-March-2015].

[19] OSVDB: "Open Sourced Vulnerability Database," http://osvdb.org/. [Accessed: 24-March-2015].

[20] SecurityFocus. Available: http://www.securityfocus.com/archive/1. [Accessed: 24-March-2015].

[21] Microsoft Security Bulletin. Available: https://technet.microsoft.com/en-us/security/bulletin/dn602597.aspx. [Accessed: 24-March-2015].

[22] S. Frei, D. Schatzmann, B. Plattner, and B. Trammell, "Modeling the Security Ecosystem - The Dynamics of (In)Security," in Economics of Information Security and Privacy, 2010, pp. 79–106.

[23] EDB: "Exploits Database by Offensive Security," Available: http://www.exploit-db.com/. [Accessed: 24-March-2015].

[24] Kaspersky Lab, "Kaspersky Security Network Report: Windows usage & vulnerabilities 2014,". Available: https://securelist.com/files/2014/08/Kaspersky_Lab_KSN_report_windows_usage_eng.pdf. [Accessed: 24-March-2015].

[25] Secunia, "Secunia Vulnerability Review, Key figures and facts from a global IT-Security perspective, 2013,". Available: http://secunia.com/?action=fetch&filename=Secunia_Vulnerability_Review_2013.pdf. [Accessed: 24-March-2015].

[26] W3Counter. Available: http://www.w3counter.com/. [Accessed: 24-March-2015].

[27] G. Stoneburner, A. Goguen, and A. Feringa, "Risk management guide for information technology systems," Nist special publication, vol. 800, no. 30, pp. 800–30, Jul. 2002.

[28] L. A. Kuznar, A. Astorino-Courtois, and S. Canna, "From the Mind to the Feet: Assessing the Perception-to-Intent-to-Action Dynamic," AIR UNIV MAXWELL AFB AL, 2011.

[29] E. LeMay, M. D. Ford, K. Keefe, W. H. Sanders, and C. Muehrcke, "Model-based Security Metrics Using ADversary VIew Security Evaluation (ADVISE)," in 2011 Eighth International Conference on Quantitative Evaluation of Systems (QEST), 2011, pp. 191–200.

[30] S. Vidalis and A. Jones, "Analyzing Threat Agents and Their Attributes," In ECIW, 2005, pp. 369-380.

[31] A. Younis, Y.K. Malaiya and I. Ray, "Assessing Vulnerability Exploitability Risk Using Software Proprieties", Software Quality Journal: 1-44, Mar 2015.