My Software has a Vulnerability, should I Worry?

Luca Allodi and Fabio Massacci
DISI - University of Trento
Trento, Italy
Email: lastname@disi.unitn.it

***Abstract*—(U.S) Rule-based policies to mitigate software risk suggest to use the CVSS score to measure the individual vulnerability risk and act accordingly: an HIGH CVSS score according to the NVD (National (U.S.) Vulnerability Database) is therefore translated into a "Yes". A key issue is whether such rule is economically sensible, in particular if reported vulnerabilities have been *actually exploited in the wild*, and whether the risk score do actually match the risk of actual exploitation.**

We compare the NVD dataset with two additional datasets, the EDB for the white market of vulnerabilities (such as those present in Metasploit), and the EKITS for the exploits traded in the black market. We benchmark them against Symantec's threat explorer dataset (SYM) of actual exploit in the wild. We analyze the whole spectrum of CVSS submetrics and use these characteristics to perform a *case-controlled analysis* of CVSS scores (similar to those used to link lung cancer and smoking) to test its reliability as a risk factor for actual exploitation.

We conclude that (a) fixing just because a high CVSS score in NVD only yields negligible risk reduction, (b) the additional existence of proof of concepts exploits (e.g. in EDB) may yield some additional but not large risk reduction, (c) fixing in response to presence in black markets yields the equivalent risk reduction of wearing safety belt in cars (you might also die but still...). On the negative side, our study shows that as industry we miss a metric with high specificity (ruling out vulns for which we shouldn't worry).

I. Introduction

Software vulnerabilities assessments usually rely on the National (US) Vulnerability Database[1] (NVD for short). Each vulnerability is published with its "technical assessment" given by the Common Vulnerability Scoring System[2] (CVSS) which rate diverse aspects of the vulnerability [14].

Despite not being designed to be a metric for risk, the CVSS score is often used as such. For example, the US Federal government with QTA0-08-HC-B-0003 reference notice specified that IT products to manage and assess the security of IT configurations must use the NIST certified S-CAP protocol [20], which explicitly says: "*Organizations should use CVSS base scores to assist in prioritizing the remediation of known security-related software flaws based on the relative severity of the flaws.*" In other words, a rule-based policy is enforced: if the vulnerability is marked as "high risk" by the CVSS assessment, it must be fixed with high priority.

This interest from the industry is matched by many academic studies. On one side, Vulnerability Discovery Models [2], [13] try to predict the number of vulnerabilities that affect a software at a certain point in time, while empirical studies try to identify trends between open and closed source software [8], [24]. On the other, attack graphs [25] and attack surfaces [11] aim at assessing in which ways a system is "attackable" by an adversary and how easily he/she can succeed. Foundational to both approaches is calculating a) the number of vulnerabilities in the system and b) their individual "risk assessment".

Beside NVD, many datasets are used in vulnerability studies, but are they the right databases? For example, Bozorgi et al. [4] showed (as a side result) that the exploitability CVSS subscore distribution do not correlate well with existence of known exploit from the ExploitDB. There are two ways to interpret this result: the exploitability of CVSS is the wrong metric, or Bozorgi and his co-authors used the wrong DB. ExploitDB could just be used by security researchers to show off their skills (and obtain more contracts as penetration testers) but might not have a correlation with actual attacks by hackers. The same problem is faced in [24] where a large majority of "exploits" are reported as zero-days[3]. The "exploit" time in OVSDB only measures the time when a proof-of-concept exploit becomes known. Security researchers normally submit proof-of-concept exploits to vendors and vulnerability white markets in order to prove that the vulnerability is worth the bounty [15]. So it is not surprising that there are a lot of "zero-day" exploits; still, this does not mean that a bad hacker really exploited those vulnerabilities.

A. Our Contribution

In this work we address the following questions:

1) To what extent can public vulnerability datasets be used to measure software security?
2) Are the rule-based policies enforced, for example, by the US Government effective in decreasing risk of attacks?

In other words, when new vulnerabilities are found, are we measuring the rate at which security researchers try to extract bounties from vendors (and should not worry)? or there is a concrete risk that bad guys end up exploiting our systems (and should worry)? This is particularly interesting for the majority of *internet users at large* (individuals or corporations) who have not enough individual value to justify a targeted attack[4]. To this aim we analyzed three datasets:

- NVD, the benchmark universe of vulnerabilities;

[1] http://nvd.nist.gov

[2] http://www.first.org/cvss

[3] A zero-day exploit is present when the exploit is reported before or on the date that the vulnerability is disclosed.

[4] Obviously, for a nuclear power plan any proof-of-concept exploit is a problem as even a software crash may lead to a national emergency.

- EDB (Exploit-DB), which contains information on the existence of proof-of-concept exploits;
- EKITS, our database containing vulnerabilities used in exploit kits sold in the black market.

No previous study, to the best of our knowledge, extensively looked at CVSS *sub*scores throughout different datasets. We benchmark these DBs against the vulnerabilities exploited in the wild that we collected from Symantec's Threats and Attack Signatures databases (SYM). To make statistically sound conclusions, we perform a case-controlled randomized experiment in which we build random samples of the NVD, EDB and EKITS datasets according to the characteristics of exploits in SYM; our goal is to understand the conditional probability that a CVSS (sub)score would lead to an attack.

The conclusion of our analysis is the following: the NVD and EDB databases are not a reliable source of information for exploits in the wild, and the CVSS score doesn't help. The CVSS score only shows a significant sensitivity (i.e. prediction of attacks in the wild) for vulnerabilities bundled in exploit kits in the black market (EKITS). Unfortunately, it does not show a high specificity in any of our datasets, which means that it is not suitable, as a metric, to rule out "un-interesting" vulnerabilities (i.e. those not exploited).

The fact that EKITS vulnerabilities are actually exploited in the wild is interesting in its own sake. "Malware sales" are often scams for wanna-be scammers, such as credit-card numbers sold over IRC channels [10]. Surprisingly, while the final products (card numbers) sold on the black market are bad, the software tools to get them from the source look good.

In the rest of the paper we introduce our four datasets (§II) and draws a first, observational comparison (§III). The core of the paper analyses the goodness of the CVSS global score as a test for exploitation (§IV), digs down over the submetrics (§V), and identifies trade-offs in the exploitation (§VI). Then we describe our randomized case-controlled analysis (§VII), and discuss the implication of our findings (§VIII) and threats to validity (§IX). We finally discuss related works (§X) and conclude (§XI).

II. Datasets

NVD is the reference database for disclosed vulnerabilities held by NIST. It has been widely used and analyzed in previous vulnerability studies [12], [24], [22]. Our NVD dataset contains data on 49599 vulnerabilities.

The Exploit-db[5] (EDB) includes information on proof-of-concept exploits also represented in the Open Source Vulnerability Database (OSVDB). Both OSVDB[6] and EDB[7] derive data from Metasploit Framework. EDB references exploited CVEs by each entry in the db. Most notable studies relying on either EDB or OSVDB are [24], [4]. EDB contains data on 8122 vulnerabilities for which a *proof-of-concept code* is reported.

TABLE I
Summary of our datasets

DB	Content	Collection method	#Entries
NVD	CVEs	XML parsing	49599
EDB	Publicly exploited CVEs	Download and web parsing to correlate with CVEs	8122
SYM	CVEs exploited in the wild	Web parsing to correlate with CVEs	1277
EKITS	CVEs in the black market	ad-hoc analysis + Contagio's Exploit table	114

EKITS is our dataset of vulnerabilities bundled in Exploit Kits[8] sold on the black market. EKITS is a substantial expansion on Contagio's Exploit Pack Table[9]. EKITS repots exploits 114 unique CVEs bundled in 90+ exploit kits. We cannot disclose the individual sources of the black-hat communities because this might hamper us from future studies.

In order to determine whether a vulnerability has been used in the wild we have collected exploited CVEs from Symantec's AttackSignature[10] and ThreatExplorer[11] public data. The SYM dataset contains 1277 CVEs identified in viruses (local threats) and remote attacks (network threats) by Symantec's commercial products. This has of course some limitation as direct attacks by individual motivated hackers against specific companies are not considered in this metric. Note that SYM does not report volumes of exploits, but only the binary information "evidence of an exploit in the wild for that CVE is reported" or "is not reported".

Table I summarizes the content of each dataset and the collection methodology. They are available upon request[12].

III. Exploratory analysis of datasets

As a starting point, we perform an exploratory analysis of our four datasets: *Given a dataset (NVD, EDB, EKITS), what is the likelihood that a vulnerability it contains is going to be exploited in the wild?* i.e. occurs also in SYM?

Table II reports the likelihood of a vulnerability being a threat if it is contained in one of our datasets. Each row represents a dataset from which the intersection with the smaller ones has been ruled out: this is to avoid data overlapping that would falsify the results. The vulnerabilities which exploits are sold in the market (EKITS) have 75.73% chances of being monitored as actively exploited. This percentage is much lower for EDB-EKITS and NVD-(EDB+EKITS).

Figure 1 is a Venn diagram of our datasets; size of the area is proportional to the number of vulnerabilities and the color is an indication of the CVSS score (a detailed analysis of the CVSS scores will follow up briefly).

[5] http://www.exploit-db.com/

[6] http://blog.osvdb.org/2012/08/15/august-2012-a-few-small-updates

[7] http://www.exploit-db.com/author/?a=3211&pg=1

[8] Exploit Kits are web sites that the attacker deploys on some public webserver he/she owns. When the victim is fooled in making an HTTP connection to the Exploit Kit, the latter checks for vulnerabilities on the user's system and, if any, tries to exploit them; eventually, it infects the victim machine with malware of some sort.

[9] http://contagiodump.blogspot.it/2010/06/overview-of-exploit-packs-update.html

[10] http://www.symantec.com/security_response/attacksignatures/

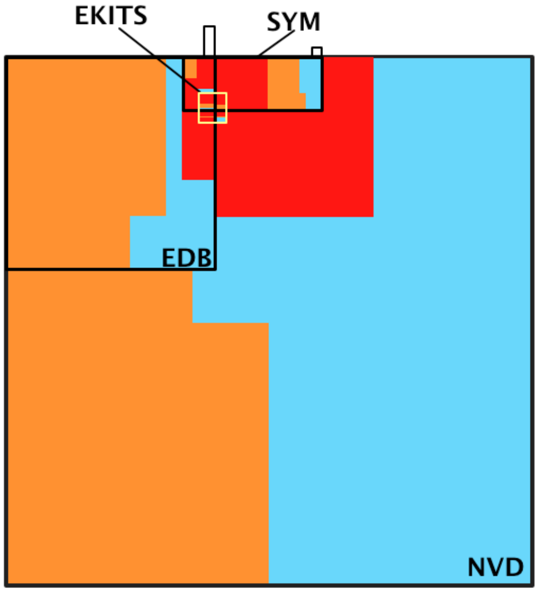
[11] http://www.symantec.com/security_response/threatexplorer/

[12] http://securitylab.disi.unitn.it/doku.php?id=datasets

TABLE II
CONDITIONAL PROB. OF VULN. FROM A DATASET BEING A THREAT

Conditional probability that a vulnerability v is listed by Symantec as threat knowing that it is contained in a dataset, i.e. $P(v \in \mathsf{SYM} \mid v \in dataset)$.

	vuln in SYM	vuln not in SYM
EKITS	75.73%	24.27%
EDB-EKITS	4.08%	95.92%
NVD-(EDB+EKITS)	2.10%	97.90%



Dimensions are proportional to data size. In red vulnerabilities with CVSS≥9 score. Medium score vulnerabilities are orange, and cyan represents vulnerability with CVSS lower than 6. The two small rectangles outside of NVDspace are vulnerabilities whose CVEs are not present in NVD.

Fig. 1. Relative Map of vulnerabilities per dataset

As one can see from the picture many vulnerabilities in the NVD are not exploited. The EDB is not overly better in terms or representativeness of actual exploitation in the wild: EDB and SYM share 393 vulnerabilities only. This means that EDB does not contain about 75% of the threats measured by Symantec in the wild. As a minor note, at the collection time NVD did not reference all vulnerabilities we found: the SYM and EDB datasets contain respectively 9 and 63 vulnerabilities that are not present in the NVD dataset. CVSS data on these vulnerabilities is therefore missing.

A rushing conclusion might be that, if one sees a vulnerability affecting his/her software in the black market, there is roughly a 75% chance that it is exploited in the wild. The same cannot be said about EDB and NVD, for which the percentages is less than 5%. However, a possible counter observation would be that EDB and NVD include many low CVSS score vulnerabilities and therefore better results could be obtained if we eliminate the vulnerabilities with little chances of being exploited.

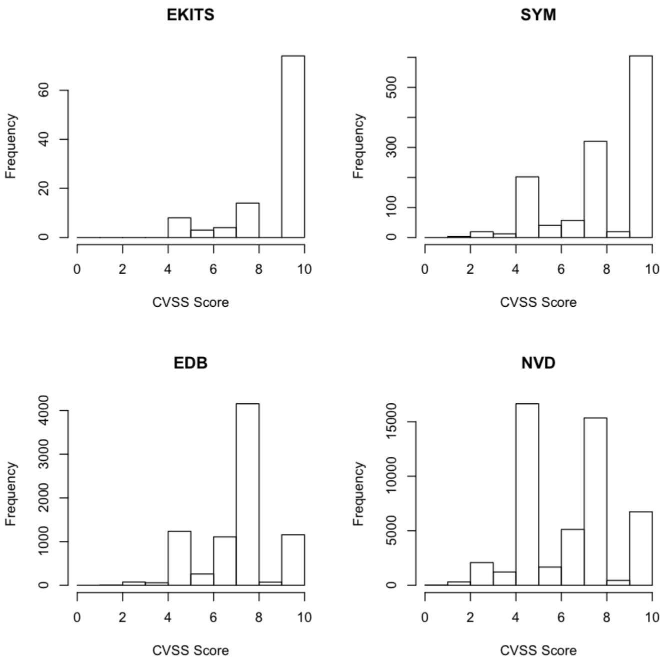


Fig. 2. Distribution of CVSS scores per dataset.

To analyze the CVSS score we report its histogram distribution in Figure 2. We identify three clusters of vulnerabilities throughout all our datasets and three corresponding categories of scores:

1) HIGH: CVSS $\geq$ 9
2) MEDIUM: 6 $\leq$ CVSS $<$ 9
3) LOW: CVSS $<$ 6

In Figure 1, red, orange and cyan areas represent HIGH, MEDIUM and LOW score vulnerabilities respectively. The NVD reports a large number of HIGH CVSS vulnerabilities that are not included in SYM. Similarly, while most of the intersection between EDB and SYM is covered by HIGH score CVEs, much of the red area for EDB is not included in SYM. By looking only at HIGH and MEDIUM score vulnerabilities in EDB one would deal with about 94% false positives (i.e. HIGH and MEDIUM score vulnerabilities not included in SYM). False positives decrease to 79% if one considers vulnerabilities with HIGH scores only. Table III reports distribution of HIGH, MEDIUM, LOW scores per each dataset. Looking at SYM, 52% of its vulnerabilities have a CVSS score strictly lower than 9 (665 out of 1277), and 21% are strictly lower than 6 (272): 1 out of 5 vulnerabilities exploited in the wild is ranked as a “low risk vulnerability”, and 1 out of 2 as “non-high risk”. From a first look, HIGH CVSS scores seem to be over-estimating the risk of exploitation for a large volume of vulnerabilities both in NVD and EDB, and not being representative of vulnerabilities in SYM.

However, this is only an observational analysis from which it is hard to make statistically sound conclusions: (a) HIGH, MEDIUM or LOW CVSS scores may only be loosely correlated to inclusion of a vulnerability in SYM. (b) These results are strongly influenced by the volume of the datasets: NVD

TABLE III
INCIDENCE OF CVSS SCORES PER DATASET

CVSS Score	EKITS	SYM	EDB	NVD
HIGH	86	612	1.209	7.026
MEDIUM	16	393	5.324	20.858
LOW	12	272	1.589	21.715
tot	**114**	**1.277**	**8.122**	**49.599**

TABLE IV
OBSERVATIONAL SPECIFICITY AND SENSITIVITY OF EACH DATASET.

Sensitivity is the probability of the CVSS score being medium or high for vulnerabilities actually exploited in the wild. Specificity is the probability of the CVSS score being low for vulnerability not exploited in the wild.

test(v.CVSS) = H v M — SYM	EKITS	EDB	NVD
Sensitivity	97.4%	94.4%	78.7%
Specificity	32.0%	20.3%	44.4%

contains almost 50.000 vulnerabilities, while those monitored in the wild are less than 1.300. In NVD and EDB there might be a lot of "noisy" vulnerabilities that are not reported in SYM because of other factors, such as age or software type (i.e. old or rare vulnerabilities that Symantec may not detect). To address (a) we measure the goodness of the CVSS score as a test for exploitation by means of two metrics, namely *sensitivity* and *specificity* (§IV). As for (b), we further explore the CVSS *sub*scores of vulnerabilities to underline technical peculiarities of vulnerabilities in SYM (§V) and use these as control variables to sample from EKITS, EDB, and NVD vulnerabilities representative of those reported in SYM (§VII).

IV. SENSITIVITY AND SPECIFICITY

In the medical domain, the sensitivity of a test is the conditional probability of the test giving positive results when the illness is present. The specificity of the test is the conditional probability of the test giving negative result when there is no illness. In our context, we want to assess to what degree our current test (the CVSS score) predicts the illness (the vulnerability being actually exploited in the wild and tracked in SYM).

Following the preliminary analysis in Section III we consider MEDIUM and HIGH CVSS scores as positive tests while LOW scores are negative tests. In formulae, Sensitivity=$Pr(v.score \geq 6 \mid v \in \mathsf{S}YM)$ while Specificity= $Pr(v.score < 6 \mid v \notin \mathsf{S}YM)$. Table IV reports the observational specificity and sensitivity for each dataset.

For the CVSS score to be a good indicator within a dataset, sensitivity and specificity should be both high, possibly over 90%. As shown in Table IV, EKITS is the dataset that performs the best in terms of sensitivity: out of 100 vulnerability exploited in the wild 97 are predicted to be dangerous (H or M CVSS score). EDB scores well in terms of sensitivity too: a proof-of-concept exploit *and* a HIGH or MEDIUM CVSS score may be a good test for exploitation. Differently for NVD, a HIGH or MEDIUM CVSS score is *not* a good indicator that an exploit will actually show off in the wild: 21 vulnerabilities out of 100 which are actually dangerous would fail to get the HIGH or MEDIUM score (79% sensitivity). Unfortunately, all databases show poor specificity: more than 1 out of 2 *not* dangerous vulnerabilities would be wrongly tagged with a HIGH or MEDIUM score. Loosely speaking, the CVSS test would generate a medical unnecessary panic among otherwise healthy individuals.

TABLE V
POSSIBLE VALUES FOR THE EXPLOITABILITY AND IMPACT SUBSCORES.

Impact subscore

Confidentiality	Integrity	Availability
Undefined	Undefined	Undefined
None	None	None
Partial	Partial	Partial
Complete	Complete	Complete

Exploitability subscore

Access Vector	Access complexity	Authentication
Undefined	Undefined	Undefined
Local	High	Multiple
Adjacent Net.	Medium	Single
Network	Low	None

However, this conclusion is only based on observational data: we report *all* data without random sampling and controlling for possible confounding variables that may influence the inclusion in SYM. Therefore, these results should be used to draw statistical conclusions with care. We will build a case-controlled experiment in a later section based on the results of our analysis on the Impact and Exploitability subscores.

V. THE IMPACT AND EXPLOITABILITY SUBSCORES

The general CVSS score takes into consideration two subscores: *Impact* and *Exploitability*. The former is a measure of the potential damage that the exploitation of the vulnerability could cause to the victim system; the latter attempts at measuring the likelihood-to-be-exploited of the vulnerability [4]. They are calculated on the basis of further variables that are reported in Table V.

The impact metric distribution is plotted in Figure 3. Somewhat surprisingly, high impact score vulnerabilities are not by default preferred by attackers: 20% of vulnerabilities in SYM have a LOW Impact score. This effect is much reduced for the EKITS dataset: only 8% of its vulnerabilities score LOW. As for EDB and NVD, the picture change completely: the greatest majority of vulnerabilities in EDB (5245, or 65%) have a medium score, and the remaining 35% is equally split between HIGH and LOW Impact vulnerabilities. This might explain the low specificity for EDB: many vulnerabilities that just have a proof-of-concept exploit are of little harm. In NVD 20% have HIGH Impact score, 40% are scored MEDIUM, and 40% LOW.

Looking in more detail into the Impact metric, Table VI shows the incidence of values of the Confidentiality, Integrity, Availability assessments for vulnerabilities in the SYM dataset. Negligible configurations are represented by a handful of vulnerabilities (e.g. the CCN case is represented by 1 vulnerability). Availability almost always assume the same value as Integrity, apart from the case where both Integrity and Confidentiality are set to "None". The average variation

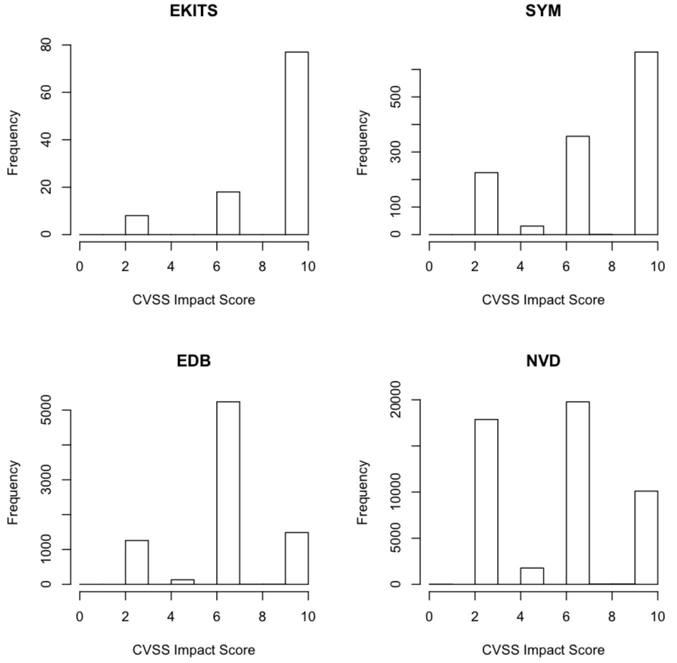


Fig. 3. Distribution of CVSS Impact subscores per dataset.

TABLE VI
INCIDENCE OF VALUES OF CIA TRIAD WITHIN THE SYM DATASET.

Confidentiality	Integrity	Availability	SYM	Negligible
C	C	C	51.53%	
C	C	N	0.08%	✓
C	N	C	0.08%	✓
C	N	N	0.23%	✓
P	P	P	26.16%	
P	P	N	1.64%	✓
P	N	P	0.16%	✓
P	N	N	7.67%	
N	C	C	0.23%	✓
N	P	C	0.08%	
N	P	P	0.63%	
N	P	N	3.68%	
N	N	C	1.57%	✓
N	N	P	6.26%	

TABLE VII
COMBINATIONS OF CONFIDENTIALITY AND INTEGRITY VALUES PER DATASET.

Confidentiality	Integrity	SYM	EKITS	EDB	NVD
C	C	51.61%	74.76%	18.11%	20.19%
C	P	0.00%	0.00%	0.02%	0.04%
C	N	0.31%	0.97%	0.71%	0.88%
P	C	0.00%	0.00%	0.01%	0.01%
P	P	27.80%	16.50%	63.52%	37.84%
P	N	7.83%	0.97%	5.61%	10.62%
N	C	0.23%	0.00%	0.18%	0.22%
N	P	4.39%	2.91%	5.07%	16.52%
N	N	7.83%	3.88%	6.75%	13.69%

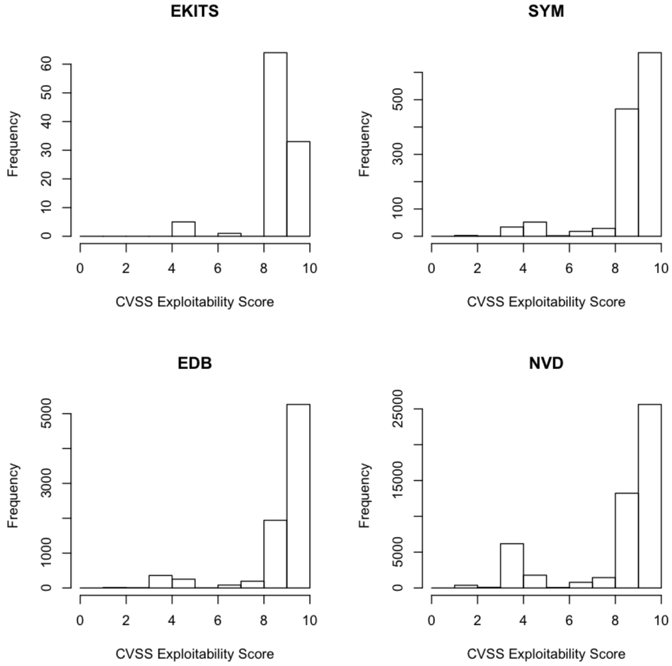


Fig. 4. Distribution of CVSS Exploitability subscores.

of the Impact score if Availability was not to be considered at all is less than 1%.

For the sake of readability, we therefore exclude the Availability assessment from the analysis, and proceed by looking at the two remaining Impact variables in the four datasets. The analysis is reported in Table VII. Most vulnerabilities in the NVD dataset score "partial" in the two Impact submetrics. This effect is enhanced in the EDB dataset, where almost 70% of vulnerabilities score partial in at least one of either Confidentiality, or Integrity. The scenario changes completely when looking at the SYM and EKITS datasets: most vulnerabilities (50%, 75%) score "Complete".

A. The Exploitability Subscore

Figure 4 shows the distribution of the Exploitability subscore per each dataset. Here the distinction between HIGH and MEDIUM Exploitability scores seem to be not very significant: numbers are qualitatively identical among all datasets. Almost all vulnerabilities, independently of the dataset, score between 8 and 10. These observations confirm Bozorgi et al.'s findings [4]: there is no direct relationship between Exploitability score and actual likelihood of exploitation. The Exploitability subscore looks therefore to be more a constant than a variable. This means that it has little to no influence on the variance of the final CVSS score, which may in turn affect the suitability of the CVSS as a risk metric.

Table VIII reports the total distribution of the exploitability variables. The greatest share of actual risk comes from vulnerabilities that can be remotely exploited; just 3% of vulnerabilities are only locally exploitable. Moreover, the great majority of discovered vulnerabilities is network-based (87.31%). Authentication is another essentially boolean variable: most exploited vulnerabilities do not require any authentication.

TABLE VIII
EXPLOITABILITY SUBFACTORS FOR EACH DATASET.

	metric	value	SYM	EKITS	EDB	NVD
Exploitability	Acc. Vec.	local	2.98%	0%	4.57%	13.18%
		adj.	0.23%	0%	0.12%	0.35%
		net	96.79%	100%	95.31%	87.31%
	Acc. Com.	high	4.23%	4.85%	3.37%	4.54%
		medium	38.35%	63.11%	25.49%	30.42%
		low	57.24%	32.04%	71.14%	65.68%
	Auth.	multiple	0%	0%	0.02%	0.05%
		single	3.92%	0.97%	3.71%	5.35%
		none	96.08%	99.03%	96.27%	95.45%

TABLE IX
RELATIONSHIP BETWEEN ACCESS COMPLEXITY, IMPACT AND ACTUAL EXPLOITATION

		Impact	SYM	EKITS	EDB	NVD
Access Complexity	High	HIGH	1.33%	2.91%	0.58%	0.92%
		MEDIUM	1.88%	1.94%	2.34%	1.89%
		LOW	1.02%	0.00%	0.46%	1.89%
	Medium	HIGH	32.50%	55.34%	8.84%	7.65%
		MEDIUM	3.60%	4.85%	11.35%	7.69%
		LOW	2.43%	2.91%	5.29%	14.83%
	Low	HIGH	18.09%	16.50%	8.89%	11.80%
		MEDIUM	22.55%	10.68%	50.89%	30.43%
		LOW	16.60%	4.85%	11.36%	22.90%

VI. EXPLOITATION TRADE-OFFS

Among all subscores, access complexity present some interesting results: the percentage of "very difficult" vulnerabilities is equal (and very low) among all datasets but the percentage of "medium-complexity" vulnerabilities in the SYM and EKITS datasets is much higher than in EDB. Medium-complexity vulnerabilities in the EKITS and SYM datasets are respectively 63.11% and 38.35% of the totals. As a comparison, only 25.49% of vulnerabilities in the EDB dataset have medium-complexity. Exploits in EDB seem to mostly capture easy vulnerabilities (71.14%).

To explain the higher average Complexity for vulnerabilities exploited in the wild we hypothesized a trade-off for the attacker: he/she is willing to put extra-effort in the exploitation only if it is worth it (i.e. the vulnerability has HIGH Impact). Table IX reports the results of the analysis. The trade-off is particularly evident in the medium-complexity range of vulnerabilities: if an attacker is going to exploit a medium complexity vulnerability, most likely this will be a HIGH impact one (32.50%). This trend is even more evident in the EKITS dataset, in which this percentage increases to 55.34%. This supports the hypothesis that the extra effort required to write an exploit for a more complex vulnerability is to be weighted with a corresponding "return on investment". Upon LOW Complexity vulnerabilities, on the other hand, there is no clear difference between HIGH, MEDIUM and LOW impacts: as long as exploitation is easy, the attacker may be willing of exploiting it regardless of the Impact score.

TABLE X
CASE-CONTROLLED CONDITIONAL PROBABILITY

Case-controlled distribution among dataset of CVSS scores (explanatory variable) vs actual exploit in the wild as reported by SYM (response variable).

EKITS'

	v $\in$ SYM	v $\notin$ SYM	p-value
CVSS H or M	379 (81.33%)	87 (18.67%)	$< 2.2 \exp -16$
CVSS L	20 (18.52%)	88 (81.48%)	

EDB'

	v $\in$ SYM	v $\notin$ SYM	p-value
CVSS H or M	248(20.84%)	942 (79.16%)	$3.55 \exp -2$
CVSS L	3 (1.20%)	248 (98.80%)	

NVD'

	v $\in$ SYM	v $\notin$ SYM	p-value
CVSS H or M	50 (4.98%)	954 (95.02%)	$5.66 \exp -4$
CVSS L	5 (1.77%)	277 (98.23%)	

VII. RANDOMIZED CASE-CONTROLLED STUDY

In order to obtain stronger statistical results on the suitability of the CVSS score as a risk metric for vulnerabilities, we use the distribution of CVSS characteristics (alongside with software and year of the vulnerability) to generate a case-controlled study where the cases are the vulnerabilities in SYM (loosely corresponding to cases of lung cancer), and NVD, EDB, and EKITS correspond to patients from various sources and medical conditions. We are looking for a control variable (like smoking) that could overwhelmingly explain exploitation (cancer). We identify as possible control variables for inclusion in SYM *confidentiality, integrity, availability, software and year*. We then generate three random samples of vulnerabilities from the EKITS, NVD and EDB datasets; because of the selection on the control variables, these samples contain vulnerabilities comparable to those present in the SYM database, and are therefore representative of the vulnerabilities detected by Symantec in the wild[13].

Table X shows the data for each sample where we consider as a (tentative) explanatory variable the value of the CVSS and as response variable the presence of the vulnerability in SYM. We run a Fisher's exact test (because data is not normal) for each of the datasets to check for statistical significance of the results. The p-values are reported in Table X. We recall that the p value does not measure the strength of an effect or an association (it is up to us to see it in the data), but only the certainty that the effect that we see in the data is not due to chance. A p value less than 0.05 is considered statistically significant because there is less than 5% chances that the data could exhibit the distribution by chance.

The test shows that the results are statistical significant for all the samples; however, the NVD' and EDB' are, in contrast to EKITS', not far from the $p < 0.05$ mark.

A. Rule-based policies for risk mitigation with CVSS

To understand wether a *HIGH CVSS* $\rightarrow$ *HIGH risk* policy is meaningful, we adopt an approach similar to that used by Evans in [6] to estimate the effectiveness of seat belts in preventing fatalities. In his case, the "effectiveness" was given by the difference in the probability of having a fatal car crash when wearing a seatbelt and when not. More formally, Pr(Death x Seat belt) - Pr(Death x not Seat belt). In our case,

[13] The sampling was performed with the statistical tool R-CRAN [21].

TABLE XI
RELATIVE RISK FOR CVSS SCORE

Relative risk (by difference or ratio of probabilities) for a vulnerability to be exploited depending on the CVSS score and the database.

$v \in SYM$ vs $v \notin SYM$	Pr(H+M) - Pr(L)	Pr(H+M) / Pr(L)
EKITS'	+62.81%	4.5x
EDB'	+19.64%	17.3x
NVD	+3.2%	2.8x

TABLE XII
CASE-CONTROLLED SPECIFICITY AND SENSITIVITY.

Case-controlled sensitivity and specificity of the CVSS score being medium or high and the vulnerability being actually exploited in the wild (i.e. in SYM). Data has been random sampled from EDB and NVD according SYM's distribution of values for CVSS subscores.

CVSS H v M — Exploit	EKITS	EDB'	NVD'
sensitivity	94.98%	97.60%	90.90%
specificity	50.28%	22.02%	22.72%

the effect we are interested in is the ability of the CVSS score (combined with the datasets) to predict the actual exploit in the wild (i.e. present in SYM). Table XI shows both the difference and the ratio of the probabilities. Either approach can be used to evaluate the strength of an association.

Each row in the table tells us the chances that a vulnerability with a MEDIUM or HIGH CVSS score is actually exploited in the while vs one with LOW scores.

For EKITS'we see that a HIGH-MEDIUM vulnerability has around +62% more chances of being exploited (difference) and more than 4.5 times the chances of being exploited than a vulnerability with LOW (ratio). Both methods tell that ending up in the black market is a bad sign. For EDB', the evidence is less strong. We only have +20% more chances albeit the ratio is 17 times higher. The reason for this conflicting result is the low prevalence rate of exploited vulnerabilities in EDB. Many of them are not exploited, even after controlling for SYM-like characteristics and this dominate the difference of probability. NVD' has even weaker association for the same reasons: we only have +3.2% increase in chances and a ratio of 3 times. For NVD' and EDB', patching HIGH or MEDIUM risk vulnerabilities would only diminish the overall risk of being actually attacked by 3% and 20% respectively.

As a consequence, getting rid of all HIGH CVSS vulnerabilities first may not be a good strategy, as otherwise suggested by rule-based policies [20]: this would results in a negligible reduction in relative risk.

B. Sensitivity and specificity for case-controlled study

The higher ratio of NVD' and EDB' determines new values for the specificity and sensitivity of the CVSS score, reported in Table XII. The sensitivity of the test is quite high among all the datasets. This result is interesting in particular with respect to EDB', for whom HIGH CVSS scores might be a good test for exploitation. Yet, the CVSS score has a dramatically poor specificity for all datasets. Sampling SYM-like characteristics does not help in scoring vulnerabilities as "non-dangerous" ones. In particular, if we consider a specificity of 25%, only 1 out of 4 non-attacked vulnerabilities are marked as LOW score. The remaining three are instead marked as MEDIUM or HIGH.

Given our results on case-controlled specificity and sensitivity of CVSS, we conclude that the CVSS score is not a reliable risk test for vulnerabilities; different results among different datasets evidence that its reliability varies depending on the reference dataset.

VIII. DISCUSSION AND IMPLICATIONS

Vulnerability assessment and patching has traditionally been a matter of great discussion within the community [5], [23], [24]. Here we summarize the main implications from our study.

Implication #1. Vulnerabilities exploited in the wild show specific patterns in the CVSS subscores; these observations can help to improve the sensitivity and specificity of the CVSS score. Some conclusions are more absolute (exceptions counted on one's fingers), while others are only statistically significant (hence the adverb "usually"), with a p-value lower than $< 2.2E-16$ for Fisher's exact test.

1) *Actually exploited vulnerabilities are remotely exploitable and do not require multiple authentication.* Despite SYM containing local threats, only 3% of vulnerabilities are assessed as "only locally exploitable". Vulnerabilities exploitable from an adjacent network are even less interesting. 4% of vulnerabilities require a single instance of authentication; none of them require multiple authentication.
2) *Availability impact is irrelevant.* The impact of more than 96% of vulnerabilities in SYM can still be accurately assessed without taking into consideration the value of Availability.
3) *Confidentiality and Integrity losses usually go hand-in-hand.* The overwhelming majority of vulnerabilities in SYM have complete or partial losses for both Confidentiality and Integrity: other combinations are less likely to be exploited.
4) *"Exploits" in EDB are usually for easy vulnerabilities.* Proof-of-concept exploits released in the EDB are for vulnerabilities easier to exploit than those actually exploited by attackers.
5) *Medium-complexity vulnerabilities are usually interesting only if they come along with a high impact.* Non-trivial to exploit vulnerabilities seem to be of interest for the attacker only if they come with a higher final impact on the vulnerable system. In contrast, Low-complexity vulnerabilities are exploited uniformly among all impact scores.

Implication #2. Rule-based policies based on CVSS score, like the US Government NIST SCAP protocol [20], may not make for an effective strategy: only a negligible number of low-risk vulnerabilities are ruled out, even after controlling for "significant" vulnerabilities. Security policies may require a major adjustment to meet these observations. In particular, while the CVSS score underlines interesting characteristics of exploited vulnerabilities, it may be not expressive enough to reliably represent exploitation. Other factors such as software popularity, presence of the exploit in the market and existence of easier vulnerabilities for that software are all "contextual factors" that might be worth exploring in future work.

Implication #3. The black market can be a good source to assess which vulnerabilities represent high risk. Exploits for vulnerabilities traded in the black market significantly overlap with those recorded in the wild, which may indicate that the presence of an exploit in the black markets can be a good indicator of the associated vulnerability risk.

IX. Threats to validity

We identify a number of threats to validity. [19].

Construct validity A number of issues we encountered while collecting SYM and EKITS may affect the soundness of the data collection. Because of the unstructured dataset of the original SYM dataset, to build SYM we needed to take some preliminary steps. We couldn't be sure about whether the collected CVEs were relevant to the threat. To address this issue, we proceeded in two steps. First, we manually analyzed a random selection of about 50 entries to check for the relevance of the CVE entries in the "description" and "additional references" sections of each entry. To double-check our evaluation, we questioned Symantec in an informal communication: our contact confirmed that the CVEs are indeed relevant. Another issue is what data from Symantec's attack-signature and threat-explorer datasets to use. Attack and infection dynamics are not always straightforward, and network and host-based threats often overlap. However, in this case, we are interested in a general evaluation of risk. Moreover, Exploit Kits enforce a drive-by download attack mechanism, therefore they are related to both the network and local threat scenario. We therefore can safely rely on both the datasets for our analyses.

Due to the shady nature of the tools, the list of exploited CVEs in EKITS may be incomplete and/or incorrect. We don't know any straightforward way to address this issue; to mitigate the problem, we crossed-referenced entries with knowledge from the security research community and from our direct observation of the black markets.

External validity is concerned with the applicability of our results to real-world scenarios. Symantec is a world-wide company and a leader in the security industry. We are therefore confident is considering their data representative sample of real-world scenarios. Yet, our conclusion cannot be generalized to the risk due to targeted attacks. Targeted attacks in the wild of a specific platform or system are less likely to generate an entry into a general anti-virus product, and therefore less likely to be represented in the SYM database.

X. Related works

Many studies before ours analyzed and modeled trends in vulnerabilities. Among all, Frei et al. [8] were maybe the first to link the idea of life-cycle of a vulnerability to the patching process. Their dataset was a composition of NVD, OSVDB and 'FVDB' (Frei's Vulnerability DataBase, obtained from the examination of security advisories for patches). The life-cycle of a vulnerability includes discovery time, exploitation time and patching time. They showed that, according to their data, exploits are often quicker to arrive than patches are. They were the first to look, in particular, at the difference in time between time of first "exploit" and time of disclosure of the vulnerability. This work have recently been extended by Shahzad et al. [24], which presented a comprehensive vulnerability study on NVD and OSVDB datasets (+ Frei's) that included vendors and software in the analysis. Many interesting trends on vulnerability patching and exploitation are presented, and support Frei's conclusion. However, they basically looked at the same data: looking at EDB or OSVDB may say little about actual threats and exploitation of vulnerabilities. The difference with our paper, here, is that we look at a *sample of actual attack data* (SYM) and underline differences in vulnerability characteristics with other datasets. For a thorough description of our datasets and a preliminary discussion on the data, see [3]. An analysis of the distribution of CVSS scores and subscores has been presented by Scarfone et al. in [22] and Gallon [9]. However, while including CVSS subscore analysis, their results are limited to data from NVD and do not provide any insight on vulnerability exploitation. In this sense, Bozorgi et al. [4] were probably the first in looking at CVSS subscores against exploitation. They showed that the "exploitability" metric, usually interpreted as "likelihood to exploit" did not match with data from EDB: their results were the first to show that the interpretation of CVSS metrics might not be entirely straightforward. We extended their first observation with a in-depth analysis of subscores and of actual exploitation data.

On a slightly different line of research are studies concerned with the discovery of vulnerabilities. In [5] Clark et. al. underlined the presence of a 'honeymoon effect' in the discovery of the first vulnerability for a software, that is related with the "familiarity" of the product. In other words, the more popular the software the smaller the gap between software release and first vulnerability disclosure.

Other studies focused on the modeling of the vulnerability discovery processes. Foundational in this sense are the works of Alhazmi et al. [2] and Ozment's [18]. The former fits 6 vulnerability models to vulnerability data of four major operative systems, and shows that Alhazmi's 'S shaped' model is the one that performs the better. However, as previously underlined by Ozment [18], vulnerability models often rely on unsound assumptions such as the independence of vulnerability discoveries. Current vulnerability discovery models are indeed not general enough to represent trends for all software [13]. Moreover, vulnerability disclosure and discovery are complex processes [17], and can be influenced by {black/white}-hat community activities [5], [8] and economics [15].

Our analysis of the vulnerabilities marketed in exploit-kits is also interesting because it confirms that the market for exploits is significantly different than the IRC markets for credit cards and other stolen goods. Indeed, dismantling some previous analysis [7], Herley et al. [10] shown that IRC markets feature all the characteristics of a typical "market for lemons" [1]: the vendor has no drawbacks in scamming the buyer because of the complete absence of a unique-ID and of a reputation system. Moreover, the buyer cannot in any way assess the quality of the good (e.g. the amount of credit available) beforehand.

In contrast, Savage et al. [16] analyzed the private messages exchanged in 6 underground forums. Most interestingly, their analysis shows that these markets feature the characteristics typical of a regular market: sellers do re-use the same ID, the transactions are moderated, and reputation systems are in place and seem to work properly. These observations coincide with our direct exploration of the black markets. The results reported in this paper show that by buying exploit kits one buys something that might actually work: the exploits in exploit kits are actually seen in the wild.

XI. Conclusion

In this paper we presented our four datasets of vulnerabilities (NVD), proof-of-concept exploits (EDB), exploits traded in the black market (EKITS), and exploits recorded in the wild(SYM). We showed that, in general, the CVSS score and its submetrics capture some interesting characteristics of the vulnerabilities whose exploits are recorded in the wild but it is not expressive enough to be used as a reliable test for exploitation (with both high sensitivity and high specificity).

Alas, the bottom-line answer to the question set out in the title of this paper is not entirely satisfactory. *You should surely worry in a few cases:*

- your vulnerability is listed by an exploit kit in the black market and have a medium-high CVSS score;
- your vulnerability has a proof of concept exploit (eg in EDB), requires no authentication, can be exploited over the network and have medium complexity but high-impact (with a medium-high CVSS score).

Unfortunately, nor CVSS subscores, nor the existence of exploits, nor the trading on the black market offer a statistically sound test for ruling out the 98% of the cases for which users at large shouldn't worry. However, it is not clear whether you should really *fix* it quick or you can wait: the chances of suffering from an attack do not increase much if you do not, even if the CVSS is high and the vulnerability is similar to others already exploited. You should at least check if a proof-of-concept exploit exist, but the overall risk status of the system will not remarkably diminish.

This makes CVSS rule-based policies not straightforward to implement: the compliance with current regulations is in contrast with measurably low gains in terms of actual security.

A robust claim can instead be made for the databases subject of this study: *using NVD, EDB (or consequently OVSDB) to assess software exploits in the wild is the wrong thing to do.* Without additional attention, those databases can only be used to assess the upper hand in the race between software vendors and security researchers.

Acknowledgements

This work was partly supported by the EU-SEC-CP-SECONOMICS and MIUR-PRIN-TENACE Projects. We would like to thank Tudor Dimitras and Viet H. Nguyen for the many useful discussions. Viet actually wrote the second script used for cross-checking the SYM dataset. No statement in this paper should be interpreted as endorsed by Symantec.

References

[1] G. A. Akerlof. The market for "lemons": Quality uncertainty and the market mechanism. *The Quarterly Jour. of Econ.*, 84:pp. 488–500, 1970.

[2] O. Alhazmi and Y. Malaiya. Application of vulnerability discovery models to major operating systems. *IEEE Trans. on Rel.*, 57(1):14 – 22, march 2008.

[3] L. Allodi and F. Massacci. A preliminary analysis of vulnerability scores for attacks in wild. In *ACM Proc. of CCS BADGERS'12*, 2012.

[4] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker. Beyond heuristics: learning to classify vulnerabilities and predict exploits. In *Proc. of SIGKDD'10*, pages 105–114. ACM, 2010.

[5] S. Clark, S. Frei, M. Blaze, and J. Smith. Familiarity breeds contempt: the honeymoon effect and the role of legacy code in zero-day vulnerabilities. In *Proc. of ACSAC'10*, pages 251–260, 2010.

[6] L. Evans. The effectiveness of safety belts in preventing fatalities. *Accident Anal. & Prev.*, 18(3):229–241, 1986.

[7] J. Franklin, V. Paxson, A. Perrig, and S. Savage. An inquiry into the nature and causes of the wealth of internet miscreants. In *Proc. of CCS'07*, pages 375–388, 2007.

[8] S. Frei, M. May, U. Fiedler, and B. Plattner. Large-scale vulnerability analysis. In *Proc. of LSAD'06*, pages 131–138. ACM, 2006.

[9] L. Gallon. Vulnerability discrimination using cvss framework. In *Proc. of NTMS'11*, pages 1–6, 2011.

[10] C. Herley and D. Florencio. Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy. *Springer Econ. of Inf. Sec. and Priv.*, 2010.

[11] M. Howard, J. Pincus, and J. Wing. Measuring relative attack surfaces. *Comp. Sec. in the 21st Century*, pages 109–137, 2005.

[12] F. Massacci, S. Neuhaus, and V. Nguyen. After-life vulnerabilities: A study on firefox evolution, its vulnerabilities, and fixes. In *Proc. of ESSoS'11*, LNCS, pages 195–208, 2011.

[13] F. Massacci and V. Nguyen. An independent validation of vulnerability discovery models. In *Proc. of ASIACCS'12*, 2012.

[14] P. Mell and K. Scarfone. *A Complete Guide to the Common Vulnerability Scoring System Version 2.0*. CMU, 2007.

[15] C. Miller. The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. In *Proc. of WEIS'07*, 2007.

[16] M. Motoyama, D. McCoy, S. Savage, and G. M. Voelker. An analysis of underground forums. In *Proc. of IMC'11*, 2011.

[17] A. Ozment. The likelihood of vulnerability rediscovery and the social utility of vulnerability hunting. In *Proc. of WEIS'05*, 2005.

[18] A. Ozment. Improving vulnerability discovery models. In *Proc. of QoP'07*, pages 6–11, 2007.

[19] D. E. Perry, A. A. Porter, and L. G. Votta. Empirical studies of software engineering: a roadmap. In *Proc. of ICSE'00*, pages 345–355. ACM, 2000.

[20] S. D. Quinn, K. A. Scarfone, M. Barrett, and C. S. Johnson. Sp 800-117. guide to adopting and using the security content automation protocol (scap) version 1.0. Technical report, 2010.

[21] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[22] K. Scarfone and P. Mell. An analysis of cvss version 2 vulnerability scoring. In *Proc. of ESEM'09*, pages 516–525, 2009.

[23] G. Schryen. Is open source security a myth? *Comm. ACM*, 54, 2011.

[24] M. Shahzad, M. Z. Shafiq, and A. X. Liu. A large scale exploratory analysis of software vulnerability life cycles. In *Proc. of ICSE'12*, pages 771–781. IEEE Press, 2012.

[25] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In *Proc. of DAS'08*, volume 5094 of *LNCS*, pages 283–296. Springer, 2008.