Poster: Analysis of Exploits in the Wild
Or: do Cybersecurity Standards Make Sense?

Luca Allodi
PhD Student
DISI - University of Trento (Italy)
Email: allodi@disi.unitn.it

Fabio Massacci
Full Professor
DISI - University of Trento (Italy)
Email: massacci@disi.unitn.it

I. Introduction

Vulnerability exploitation is a major threat vector for cyber attacks [2], making vulnerability assessment a crucial moment in the security management process. The "criticality" of a vulnerability is often expressed in the classic form $Risk = Impact \times Likelihood$ [6]. The more high-risk vulnerabilities affect a system, the higher its final risk assessment. In this manuscript, we identify and analyse two major shortcomings in current software risk assessment approaches.

Problem 1. Worrying about every vulnerability. Vulnerability assessment is founded on the classical view of security and is notoriously synthesised in Schneier's quote "security is only as strong as the weakest link" [1]. In turn, this derives directly from the classic model of the attacker, assumed to be very powerful [4]. Some variations to this model exist, but the baseline remains the same: if a vulnerability is in my system, then an attacker will, sooner or later, exploit it. However, this is in contrast with trends in attacks reported in literature. For example, cybercrime attack tools such as exploit kits [1] represent two thirds of the threats for the final user [8], and yet they feature about 10 vulnerabilities each, some of which are five years old. If this observation was to hold for most exploits, it may be that the typical attacker is not as powerful as currently assumed to be; this would radically affect current mitigation and remediation strategies.

Problem 2. Reliance on a never-checked assessment methodology. The CVSS scoring system [6] is the *standard-de-facto* framework for vulnerability assessment. For example, the U.S Government SCAP protocol [7] recommends to use it to optimise patching strategies. However, the CVSS score has never been properly validated against actual attack data, and could therefore be misleading as a risk measure. For example, CVSS measures the "likelihood" of exploitation in its Exploitability subscore [3], [6]. Unfortunately, according to this metric the greatest majority of vulnerabilities have very high exploitability [3], meaning that, by measure of the CVSS score, any vulnerability is equally likely to be exploited. This limitation may substantially affect the optimality of security investment and management.

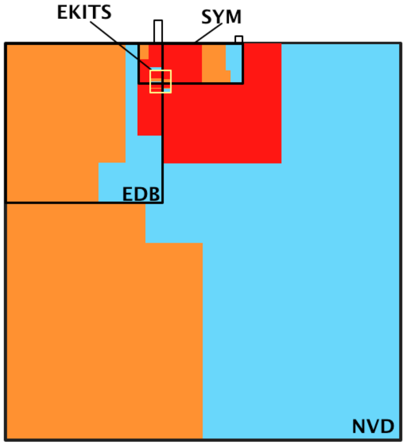

Fig. 1. Relative Map of vulnerabilities per dataset

II. Datasets

NVD: the universe of vulnerabilities. NVD contains all the vulnerabilities (CVEs) disclosed up to a certain moment in time. It reports a CVSS assessment for each vulnerability.
EDB: the white market for exploits. EDB reports the vulnerabilities for which a "proof-of-concept" exploit exists; this is *not* evidence of exploitation in the wild. However, many previous studies used EDB or OSVDB as reference databases for actual exploitation [5], [9].
SYM: records of exploits in the wild. Symantec keeps two public datasets of signatures for local and network threats: AttackSignature[2] and ThreatExplorer[3]. If a CVE is reported in SYM it means an exploit for it was observed in the wild.
EKITS: the black markets for exploits. In this dataset we track more than 90 attack tools traded in the cybercrime black markets and the vulnerabilities they exploit.

III. Preliminary results

Problem 1. Is everything exploited? Figure 1 is a Venn diagram representation of our datasets. Areas are proportional to volume of vulnerabilities and colours represent HIGH, MEDIUM and LOW score vulnerabilities. As one can see the greatest majority of vulnerabilities in the NVD are not

This work is partly supported by the projects EU-IST-NOE-NESSOS and EU-SEC-CP-SECONOMICS.

[1] http://www.schneier.com/blog/archives/2005/12/weakest_link_se.html

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

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

		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%

TABLE I
RELATIONSHIP BETWEEN ACCESS COMPLEXITY AND IMPACT

included nor in EDB nor in SYM. More interestingly, EDB covers SYM for about 25% of its surface, meaning that 75% of vulnerabilities exploited by attackers are never reported in EDB by security researchers. Moreover, 95% of exploits in EDB are not reported as exploited in the wild in SYM. The fact that SYM is not composed by a *random selection* of exploits from EDB evidences that the attacker is involved in an autonomous selection process. This is in sharp contrast with previous assumptions in literature [5], [9], [3]. Interestingly, our EKITS dataset overlaps with SYM about 80% of the time.

Conclusion 1. The attacker does **not** *exploit every vulnerability: not only most vulnerabilities in NVD are never exploited, but most exploits in EDB are of no interest for the real attacker. Differently, if a vulnerability is traded in the black markets, it is most likely going to be attacked.*

To further investigate differences among out datasets, we hypothesised that *hackers* are willing to exploit more difficult but powerful vulnerabilities than *security researchers* are: the former write exploits to attack systems, the latter to publish more exploit code to prove their skills. We therefore extended the analysis to the CVSS subscores Access complexity and Impact, representing the difficulty of exploitation of the vulnerability and the final impact of the exploitation to the system, respectively. Table I reports the results of the analysis. The trade-off is particularly evident in the medium-complexity range of vulnerabilities: most medium-complexity exploited vulnerabilities are HIGH impact ones (32.50%), while lower impact vulnerabilities are exploited only if very easy to. This trend is confirmed for the EKITS dataset as well. Differently, EDB presents mainly easy and medium-low impact vulnerabilities, evidencing that *security researchers* are not a reliable proxy for actual exploitation from the bad guys.

Conclusion 2. Current databases for vulnerabilities and exploits are **misleading** *with respect to what the bad guys are actually doing. Many conclusions drawn in previous studies [9], [5] should be taken with more than a grain of salt.*

Problem 2. Is CVSS a good predictor for exploitation? In order to understand if CVSS is a good predictor for risk, we tested its reliability as a *test* for exploitation [1]. In the medical domain, the sensitivity of a test is the conditional probability of the test giving positive results when the illness is present. Its specificity is the conditional probability of the test giving negative result when there is no illness. In our context, we assess to what degree the CVSS test predicts the illness ($v \in \mathsf{S}YM$). To make statistically sound conclusions, we sampled the NVD, EDB and EKITS datasets according to the distribution of the CVSS characteristics of the vulnerabilities in SYM (e.g. impact type, local or remote exploitability, etc.). For our experiment we consider CVSS scores higher than 6 to be HIGH, and those strictly lower than 6 to be LOW. In formulae, Sensitivity=$Pr(v.score \geq 6 \mid v \in \mathsf{SYM})$ while Specificity= $Pr(v.score < 6 \mid v \notin \mathsf{SYM})$. Results are reported in Table II. The sensitivity of our samples is quite high ($> 89\%$), meaning that the CVSS score does a good job in predicting which vulnerabilities are going to be exploited. On the other hand, the specificity is extremely low everywhere with a peak low in NVD and EDB at about 25%. This means that 3 times out of 4, a vulnerability or an exploit marked as HIGH risk is *not* going to be exploited: assuming linearity of cost per patch release, to prioritise patching for HIGH score vulnerabilities can be 300% more expensive than an optimal policy. These measures are supported by a strong statistical significance with $p < 2.2^{-16}$.

CVSS H v L — Exploit	EKITS	EDB	NVD
sensitivity	89.17%	98.14%	89.70%
specificity	49.73%	24.39%	22.22%

TABLE II
CASE-CONTROLLED SPECIFICITY AND SENSITIVITY.

Conclusion 3. The CVSS score is **not** *a good predictor for exploitation. Policies relying on it to build sound strategies, such as US NIST Standard for assessing Cybersecurity Risk [7], may be widely sub-optimal.*

IV. CONCLUSIONS AND FUTURE DIRECTIONS

We find that (a) very few exploits are interesting for the attacker and (b) current metrics to evaluate risk are misleading: current guidelines and policies are not well-grounded on reality. With our future work we aim at enhancing current risk models, and therefore allow for better risk management practices and policies for security.

REFERENCES

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

[2] W. Baker, M. Howard, A. Hutton, and C. D. Hylender. 2012 data breach investigation report. Technical report, Verizon, 2012.

[3] 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*, July 2010.

[4] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Trans. on Inf. Th.*, 29(2):198 – 208, mar 1983.

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

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

[7] 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.

[8] M. Rajab, L. Ballard, N. Jagpal, P. Mavrommatis, D. Nojiri, N. Provos, and L. Schmidt. Trends in circumventing web-malware detection. Technical report, Google, 2011.

[9] 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.