

Vulnerability valuation: security's inflation problem

Analyst: Wendy Nather

Pity the poor CISO, going through log reports: a million attacks foiled here, a million probes deflected there – and that's just in one day. While we're at it, let's buy a round of sympathy for the development team lead, which has just been handed a brand new scanner's 500-page PDF report, listing application vulnerabilities in the latest release. The reports are overwhelming in their scale. And yet at the same time – because nothing bad has actually happened (that anyone can tell) – the figures don't carry much meaning. Moreover, unless the vendor is doing it, someone in the enterprise has the daunting and monotonous task of weeding out false positives.

No matter where the practitioner looks, the vulnerability landscape is huge, and a portion of it is untrustworthy. The very people in a position to do something about these numbers are in danger of becoming numb to them.

Attacking (so to speak) the vulnerability counts

Most of the commercial vendors understand that a straight count (850 flaws in one application, 30,000 vulnerabilities in 2,000 desktops) is good for one dose of FUD – maybe two – but for security practitioners that have to deal with these numbers every day, it's friendlier to consolidate them (11 types of flaw in the same application; all the desktops need the last three OS patches). The lingua franca of event reporting, syslog, figured out summarizing early on ('last message repeated 25 times').

Other than scaring an executive who has never done the math, there's not much value in a raw count. It's more useful to describe the findings in terms of scope (60% of the systems are missing the same patch) or saturation (30 vulnerabilities in 2,000 lines of code, versus 300 vulnerabilities in 5,000 lines). Even so, the figures are depressingly high, especially for an organization that is running the discovery tool for the first time; the sheer amount of work that the findings represent can be so intimidating that the customer loses motivation to use the tool going forward. Nobody wants a security product to gather dust on the shelf.

Just as someone can make statistics mean anything, some reports from a network security device lend themselves to questionable statements of effectiveness. If a certain kind of probe is being successfully blocked, does it really matter whether there are 40 or 400,000? We have seen the marketing argument that 'even if only 1% of attacks slip through, that's 4,000 successful breaches,' and in some cases, it's misleading to the consumer. If a firewall drops all traffic on port 1337, then talking about how many millions of attacks it's prevented on that port isn't terribly impressive, nor is it possible that 1% of those same attacks could get through. Umbrella manufacturers don't brag about how many raindrops they stop.

Looking for a new denomination

Most IDS/IPS vendors have moved on to listing the number of attack signatures they support, their performance throughput, or their rate of false positives, the same as anti-malware sellers. However, the number of signatures has become mind numbing too – on the order of hundreds of thousands. It's a devalued metric that is going out of style, especially with anomaly detection as an alternative; in other words, some security vendors and practitioners are slowly backing away from counting vulnerabilities. Is this because the metric is less useful, or simply because the numbers involved are too big to manage?

There are fewer ways to enumerate problems in the application security field; but even there, a few vendors are starting to grade on a curve. They're ranking vulnerabilities by severity (and sometimes by exploitability), and setting target numbers for them at different levels depending upon the stated criticality of the application, so that an internal wiki with a SQL injection flaw gets a 'low' criticality rating while a health insurance application with the same flaw sets off all the risk red alerts. Vendors and customers have accepted that no code is flawless, and not all vulnerabilities are equal – but more than that, they're acknowledging the fact that the business executive won't always care about the numbers as much as the CISO does.

Security researchers are tackling both the problem of metrics overall and the problem of vulnerabilities in particular. It doesn't help that the terms 'vulnerability,' 'threat' and 'exploit' are sometimes used interchangeably in marketing and media. What many detection signatures look for are actually different ways of exploiting the same underlying vulnerability – for example, using the same buffer overflow to insert different attack code to achieve different results. A high number of malware signatures doesn't necessarily equal the same number of vulnerabilities, but it looks that way to the untrained eye. (And focusing on vulnerabilities as a sole measurement ignores the fact that malware doesn't always need to take advantage of a vulnerability, as email attachments and drive-by Web downloads will demonstrate.) Groups of security professionals, such as the ones that attend MetriCon, continue to try to develop meaningful models and numbers that will be better at capturing the state of security without being susceptible to 'irrational exuberance' by marketing mavens.

Avoiding the fool's gold standard

Another danger to security metrics lies in the 'easy standard': we tend to prefer standards that are concrete, small in number and unambiguous in execution. When an area is hard to quantify, such as security, businesses that have few resources to tackle it will go for the quick win, such as a PCI-DSS Report of Compliance. With a checklist, the question 'are we secure?' becomes reassuringly binary: either the checklist is completed, or it isn't (never mind that a checklist is just a very low-water-mark in the overall process of securing systems). When there is no checklist, something else in the industry will slowly be morphed into one: for an example, see the OWASP Top Ten list of security software flaws.

From one perspective, the OWASP Top 10 is brilliant: it distills many common coding mistakes into an understandable form, and allows us to corral very large numbers of vulnerabilities into categories. But it was never intended to be a complete list of everything a developer could get wrong in an application – for one thing, it's updated every year, and replaces the most common flaws with ones from the previous year that have simply dropped under the 'top 10' threshold. It doesn't mean those flaws have gone away, but developers looking for a guide may narrow their focus to the Top Ten and ignore the rest. When a bridge is being built, nobody stops at a top 10 list and decides that it's safe enough; we expect that it has undergone every test that engineers can devise for that design.

And the OWASP Top 10 has become the shorthand standard of vulnerability enumeration (along with the CWE Top 25, but that requires more column inches and more paper). When buyers need to express their application security requirements, they will reach for the most widely understood checklist, because that's easily inserted into a contract. The only other alternative today is to specify a requirement that the software pass a particular vulnerability scan, or that it pass a manual review – and small businesses that don't have money for a scan or expertise to review it are out of luck on both counts. The goal is to specify a binary condition that can be met or not, because that's how contracts are made enforceable.

The OWASP Top 10 is so lawyer-friendly that it has appeared in RFPs; when it starts showing up in state government procurement contracts (such as New Hampshire's Department of State RFP for a Web content management system this year), it's a good sign that at some point it will be suggested for inclusion in that state's legislation. For the same reasons that compliance is easier than effectiveness metrics to frame in legal language, the OWASP Top 10 is in danger of becoming the synecdoche of application security. Not even OWASP would see this as a good thing; the community's projects stretch way beyond a checklist to encompass much wider aspects of secure development. And this leaves vendors with no choice but to include the Top 10 in their own assessment results, at least as a starting point. We've even seen marketing departments refer to 'OWASP Top 10 compliance' – we're not sure they knew they were asserting compliance with a list of bad software mistakes.

Short-selling the vulnerability

Accidentally or not, the 'easy standards' are appearing to serve as a route to vulnerability consolidation – a revaluation of an impossibly large number of ways to describe how a system can be attacked. In many ways, the vulnerability is losing its value as a unit of security currency due to a flooded market. Although its ubiquity means that the state of Internet security overall is very troubling, we need better ways of describing the risk. Vulnerabilities are only the symptoms; we need informed sources to report on root causes, even if that type of data doesn't lend itself nicely to year-over-year trending. The discussion should move to topics such as where the most serious flaws are, whether they're in the system out of the starting gate, how quickly they're fixed, and how quickly they're replaced by new ones. We don't need to count the number of things wrong with a car to know it's a lemon; we just need to know what types of things to fix – or better yet, know how to spot a Rugged one and buy that instead.

Reproduced by permission of The 451 Group; copyright 2010-11. This report was originally published within The 451 Group's Market Insight Service.

For additional information on The 451 Group or to apply for trial access, go to:
www.the451group.com