# Buying Into the Bias: Why Vulnerability Statistics Suck

By Steve Christey (MITRE) and Brian Martin (Open Security Foundation)
July 11, 2013

Academic researchers, journalists, security vendors, software vendors, and professional analysts often analyze vulnerability statistics using large repositories of vulnerability data, such as “Common Vulnerabilities and Exposures” (CVE), the Open Sourced Vulnerability Database (OSVDB), and other sources of aggregated vulnerability information. These statistics are claimed to demonstrate trends in vulnerability disclosure, such as the number or type of vulnerabilities, or their relative severity. Worse, they are typically misused to compare competing products to assess which one offers the best security.

Most of these statistical analyses demonstrate a serious fault in methodology, or are pure speculation in the long run. They use the easily-available, but drastically misunderstood data to craft irrelevant questions based on wild assumptions, while never figuring out (or even asking the sources about) the limitations of the data. This leads to a wide variety of bias that typically goes unchallenged, that ultimately forms statistics that make headlines and, far worse, are used to justify security budget and spending.

As maintainers of two well-known vulnerability information repositories, we're sick of hearing about research that is quickly determined to be sloppy after it's been released and gained public attention. In almost every case, the research casts aside any logical approach to generating the statistics. They frequently do not release their methodology, and they rarely disclaim the serious pitfalls in their conclusions. This stems from their serious lack of understanding about the data source they use, and how it operates. In short, vulnerability databases (VDBs) are very different and very fickle creatures. They are constantly evolving and see the world of vulnerabilities through very different glasses.

This paper and its associated presentation introduce a framework in which vulnerability statistics can be judged and improved. The better we get about talking about the issues, the better the chances of truly improving how vulnerability statistics are generated and interpreted.

## Bias, We All Have It

Bias is inherent in everything humans do. Even the most rigorous and well-documented process can be affected by levels of bias that we simply do not understand are working against us. This is part of human nature. As with all things, bias is present in the creation of the VDBs, how the databases are populated with vulnerability data, and the subsequent analysis of that data. Not all bias is bad; for example, VDBs have a bias to avoid providing inaccurate information whenever possible, and each VDB effectively has a customer base whose needs directly drive what content is published.

Bias comes in many forms that we see as strongly influencing vulnerability statistics, via a number of actors involved in the process. It is important to remember that VDBs catalog the public disclosure of security vulnerabilities by a wide variety of people with vastly different skills and motivations. The disclosure process varies from person to person and introduces bias for sure, but even before the disclosure occurs, bias has already entered the picture.

Consider the general sequence of events that lead to a vulnerability being cataloged in a VDB.

1. A researcher chooses a piece of software to examine.
2. Each researcher operates with a different skill set and focus, using tools or techniques with varying strengths and weaknesses; these differences can impact which vulnerabilities are capable of being discovered.
3. During the process, the researcher will find at least one vulnerability, often more.
4. The researcher may or may not opt for vendor involvement in verifying or fixing the issue.
5. At some point, the researcher may choose to disclose the vulnerability. That disclosure will not be in a common format, may suffer from language barriers, may not be technically accurate, may leave out critical details that impact the severity of the vulnerability (e.g. administrator authentication required), may be a duplicate of prior research, or introduce a number of other problems.
6. Many VDBs attempt to catalog all public disclosures of information. This is a "best effort" activity, as there are simply too many sources for any one VDB to monitor, and accuracy problems can increase the expense of analyzing a single disclosure.
7. If the VDB maintainers see the disclosure mentioned above, they will add it to the database if it meets their criteria, which is not always public. If the VDB does not see it, they will not add it. If the VDB disagrees with the disclosure (i.e. believes it to be inaccurate), they may not add it.

By this point, there are a number of criteria that may prevent the disclosure from ever making it into a VDB. Without using the word, the above steps have introduced several types of bias that impact the process. These biases carry forward into any subsequent examination of the database in any manner.

## Types of Bias

Specific to the vulnerability disclosure aggregation process that VDBs go through every day, there are four primary types of bias that enter the picture. Note that while each of these can be seen in researchers, vendors, and VDBs, some are more common to one than the others. There are other types of bias that could also apply, but they are beyond the scope of this paper.

*Selection bias* covers what gets selected for study. In the case of disclosure, this refers to the researcher's bias in selecting software and the methodology used to test the software for vulnerabilities; for example, a researcher might only investigate software written in a specific language and only look for a handful of the most common vulnerability types. In the case of VDBs, this involves how the VDB discovers and handles vulnerability disclosures from researchers and vendors. Perhaps the largest influence on selection bias is that many VDBs monitor a limited source of disclosures. It is not necessary to argue what "limited" means. Suffice it to say, no VDB is remotely complete on monitoring every source of vulnerability data that is public on the net. Lack of resources - primarily the time of those working on the database - causes a VDB to prioritize sources of information. With an increasing number of regional or country-based CERT groups disclosing vulnerabilities in their native tongue, VDBs have a harder time processing the information. Each vulnerability that is disclosed but does not end up in the VDB, ultimately factors into statistics such as "there were X vulnerabilities disclosed last year".

*Publication bias* governs what portion of the research gets published. This ranges from "none", to sparse information, to incredible technical detail about every finding. Somewhere between selection and publication bias, the researcher will determine how much time they are spending on this particular product, what vulnerabilities they are interested in, and more. All of this folds into what gets published.

VDBs may discover a researcher's disclosure, but then decide not to publish the vulnerability due to other criteria.

*Abstraction bias* is a term that we crafted to explain the process that VDBs use to assign identifiers to vulnerabilities. Depending on the purpose and stated goal of the VDB, the same 10 vulnerabilities may be given a single identifier by one database, and 10 identifiers by a different one. This level of abstraction is an absolutely critical factor when analyzing the data to generate vulnerability statistics. This is also the most prevalent source of problems for analysis, as researchers rarely understand the concept of abstraction, why it varies, and how to overcome it as an obstacle in generating meaningful statistics. Researchers will use whichever abstraction is most appropriate or convenient for them; after all, there are many different consumers for a researcher advisory, not just VDBs. Abstraction bias is also frequently seen in vendors, and occasionally researchers in the way they disclose one vulnerability multiple times, as it affects different software that bundles additional vendor's software in it.

*Measurement bias* refers to potential errors in how a vulnerability is analyzed, verified, and catalogued. For example, with researchers, this bias might be in the form of failing to verify that a potential issue is actually a vulnerability, or in over-estimating the severity of the issue compared to how consumers might prioritize the issue. With vendors, measurement bias may affect how the vendor prioritizes an issue to be fixed, or in under-estimating the severity of the issue. With VDBs, measurement bias may also occur if analysts do not appropriately reflect the severity of the issue, or if inaccuracies are introduced while studying incomplete vulnerability disclosures, such as missing a version of the product that is affected by the vulnerability. It could be argued that abstraction bias is a certain type of measurement bias (since it involves using inconsistent "units of measurement"), but for the purposes of understanding vulnerability statistics, abstraction bias deserves special attention.

Measurement bias, as it affects statistics, is arguably the domain of VDBs, since most statistics are calculated using an underlying VDB instead of the original disclosures. As the primary sources of vulnerability data aggregation, several factors come into play when performing database updates.

## Why Bias Matters, in Detail

These forms of bias can work together to create interesting spikes in vulnerability disclosure trends. To the VDB worker, they are typically apparent and sometimes amusing. To an outsider just using a data set to generate statistics, they can be a serious pitfall.

In August, 2008, a single researcher using rudimentary, yet effective methods for finding `symlink` vulnerabilities single handedly caused a significant spike in `symlink` vulnerability disclosures over the past 10 years. Starting in 2012 and continuing up to the publication of this paper, a pair of researchers have significantly impacted the number of disclosures in a single product. Not only has this caused a huge spike for the vulnerability count related to the product, it has led to them being ranked as two of the top vulnerability disclosers since January, 2012. Later this year, we expect there to be articles written regarding the number of supervisory control and data acquisition (SCADA) vulnerabilities disclosed from 2012 to 2013. Those articles will be based purely on vulnerability counts as determined from VDBs, likely with no mention of why the numbers are skewed. One prominent researcher who published many SCADA flaws has changed his personal disclosure policy. Instead of publicly disclosing details, he now keeps them private as part of a competitive advantage of his new business.

Another popular place for vulnerability statistics to break down is related to vulnerability severity. Researchers and journalists like to mention the raw number of vulnerabilities in two products and try to compare their relative security. They frequently overlook the severity of the vulnerabilities and may not note that while one product had twice as many disclosures, a significant percentage of them were low severity. Further, they do not understand how the industry-standard CVSSv2 scoring system works, or the bias that can creep in when using it to score vulnerabilities. Considering that a vague disclosure that has little actionable details will frequently be scored for the worst possible impact, that also drastically skews the severity ratings.

## Conclusion

The forms of bias and how they may impact vulnerability statistics outlined in this paper are just the beginning. For each party involved, for each type of bias, there are many considerations that must be made. Accurate and meaningful vulnerability statistics are not impossible; they are just very difficult to accurately generate and disclaim.

Our 2013 BlackHat Briefings USA talk hopes to explore many of these points, outline the types of bias, and show concrete examples of misleading statistics. In addition, we will show how you can easily spot questionable statistics, and give some tips on generating and disclaiming good statistics.