Probably Don’t Rely on EPSS Yet

Author’s Note: This post was updated on June 9, 2022, to correct factual errors including references to Kenna Security instead of AlienVault and Fortinet. This post was updated on June 14, 2022, to edit content to reflect the publication of the EPSS FAQ on June 10, 2022.

Vulnerability management involves discovering, analyzing, and handling new or reported security vulnerabilities in information systems. The services provided by vulnerability management systems are essential to both computer and network security. This blog post evaluates the pros and cons of the Exploit Prediction Scoring System (EPSS), which is a data-driven model designed to estimate the probability that software vulnerabilities will be exploited in practice.

The EPSS model was initiated in 2019 in parallel with our criticisms of the Common Vulnerability Scoring System (CVSS) in 2018. EPSS was developed in parallel with our own attempt at improving CVSS, the Stakeholder-Specific Vulnerability Categorization (SSVC); 2019 also saw version 1 of SSVC. This post will focus on EPSS version 2, released in February 2022, and when it is and not appropriate to use the model. This latest release has created a lot of excitement around EPSS, especially since improvements to CVSS (version 4) are still being developed. Unfortunately, the applicability of EPSS is much narrower than people might expect. This post will provide my advice on how practitioners should and should not use EPSS in its current form.

This post assumes you know about the services comprising vulnerability management and why prioritization is important during analysis and response. Response includes remediation (patching or otherwise removing the problem) and mitigation (doing something to reduce exposure of vulnerable systems or reduce impact of exploitation). Within coordinated vulnerability disclosure roles, I’ll focus just on people who deploy systems. These are the folks most likely to have legitimate uses of EPSS, but even for many deployers this approach can lead to a short circuit rather than a shortcut if they’re not careful.

EPSS semi-formalized as a special interest group (SIG) at FIRST in 2020. I’ve participated on the SIG since its inception. I say this not to give myself any special authority, but rather to clarify why I’m posting this information here rather than integrating it into the EPSS website. The SIG has not prioritized publicizing the information in this post, and I think it is important information to consider when organizations decide if and how to adopt EPSS. A SIG at FIRST serves to “explore an area of interest or specific technology area, with a goal of collaborating and sharing expertise and experiences to address common challenges.” Basically, this means I’ve been on a lot of calls and email threads with people trying to improve EPSS. In general, I think everyone on the SIG has done a great job working within the constraints of donating their time and resources to a project, which was initially described by this 2020 paper.

However, I have a few concerns about EPSS that I’d like to highlight here. I have raised these concerns within the SIG, but the SIG has no formal voting process; so I can’t be sure whether my views represent a minority opinion.

Here are the two general spheres of problems I see: problems due to model opacity and problems stemming from the details of data provenance (elaborated below). EPSS cannot replace a vulnerability analysis or risk management process and should not be used by itself. However, EPSS v2 is currently useful in some restricted scenarios, which I’ll highlight below.

EPSS Opacity

The EPSS target audience, development process, and future governance are opaque.

EPSS uses machine learning to predict exploitation probabilities for each CVE ID (CVE IDs provide identifiers for vulnerabilities in IT products or protocols). This reliance on pre-existence of a CVE ID is one reason why EPSS is not useful to software suppliers, CSIRTs, and many bug bounty programs. Most of those stakeholders need to prioritize vulnerabilities that either do not have public CVE IDs (because, for example, the vendor is coordinating a fix prior to publication) or are types of vulnerabilities that never receive CVE IDs, such as misconfigurations. Furthermore, zero-day vulnerabilities may get a CVE ID upon publication and disclosure, but a zero day is almost always published because it is widely known to be exploited. The EPSS FAQ clarifies that vulnerabilities widely known to be exploited are out of scope for EPSS. That is, the target audience for EPSS is opaque. My understanding, based on these design decisions, is that EPSS is useful for some organizations that deploy software systems to prioritize application of software patches tied to CVE IDs. It is useful as long as the organization is mature enough that it can distinguish and has capacity to address vulnerabilities that are “just below the obvious” threats of widely exploited vulnerabilities and the EPSS data provenance matches the organization (see below). This is a big group of organizations that are worth helping. It can be complicated to determine whether you are in the target audience or not, so I recommend that you give the decision careful consideration.

EPSS calls itself an “open, data-driven effort”—but it is only open in the sense that anyone can come and ask questions during the meetings to the handful of people who actually have access to the code and data for producing the scores. SIG members generally do not have access to the code or the data. That handful of people are generally super nice and do their best to answer questions seriously within the constraints of the proprietary aspects of the data collection, training, and modeling. However, because salient operational details of the EPSS prediction mechanism are not open to the SIG generally, we can only rely on the metrics about them that are made available. These are fairly good metrics, because they include the performance metrics used to train the model. However, as a SIG member I have no special access to information beyond what any reader would have from going to the EPSS website. There is not a formal layer of governance and oversight that the SIG performs on the development of the model. That is, the process is opaque.

In addition, there is no guarantee that either the input data or the work to produce the predictions from the data will continue to be donated to the public indefinitely. It could go away at any time if just a couple key members of the SIG decide to stop or to charge FIRST for the data. Multiple vendors donating data would make the system more robust; multiple vendors would also address some, but not all, of the problems with the data discussed next.

This opacity makes the clear labeling of the outputs critically important, which is the topic of the next section.

EPSS Data and Outputs

EPSS outputs genuine probabilities. In the phrase “the probability that _____,” that blank needs to be filled in. On the first line of its website, EPSS purports to fill that blank as “probability that a software vulnerability will be exploited in the wild.” The EPSS SIG elaborates on this statement (e.g., explanations of how to interpret probabilities in general and the data sources that go in to the calculation of the probabilities). Nonetheless, even with understanding the elaborations, this statement is oversimplified enough that I think it is both misleading and wrong.
EPSS got here attempting to avoid one of our key criticisms of CVSS: CVSS vector elements are not actually numbers, just rankings, and so the whole idea of using mathematics to combine the CVSS vector elements into a final score is unjustified. EPSS takes in qualitative attributes, but the machine learning architecture treats all of these with the right kinds of mathematical formalisms and produces a genuine probability. These outputs still need the correctly specified event and timeframe. EPSS forecasts the probability that “a software vulnerability will be exploited in the wild in the next 30 days.” This statement appears to be well-defined, until we dig into what the inputs are and the implications this has for generalizability of the output data.

I’m worried about assumptions and connections that get introduced into the probability that we cannot capture with simple unit conversions or calculation of conditional probabilities. Here is the crux of the problem. As far as I know, the EPSS phrase “a software vulnerability will be exploited in the wild [in the next 30 days]” actually means the following:

- software vulnerability = a CVE ID in the National Vulnerability Database with a CVSSv3 vector string (see discussion of EPSS audience in relation to CVE ID dependencies above)
- exploited = an IDS signature triggered for an attempt to exploit the CVE ID over the network
- in the wild = a contributor to AlienVault or Fortinet whose network is instrumented with their IDS systems and their data is shared
- in the next 30 days = model training parameter window for analysis over past data

There are further important details that are not clear from the documentation. For example, only about 10 percent of the vulnerabilities with CVE IDs even have IDs signatures. So 90 percent of CVE IDs could never be detected to be actively exploited this way. Anyone who cares about vulnerabilities that are not exploitable over the network needs information in addition to EPSS.

Even for network-exploitable vulnerabilities, the way IDS signatures are created is complex. Moreover, the signature curators have their own priorities and performance aspects to optimize, which means the coverage for the signatures is probably much better than random as long as your environment is similar to the environment the IDS vendor is managing. The flip side is that your coverage is plausibly worse than random if your environment is a mismatch.

In some important way, EPSS is doing something smart. It’s saying, Hey, we saw IDS alerts for attempts to exploit these CVE IDs, and here are a handful of things we didn’t see alerts for but that seem similar. That’s great if you have an environment similar to the environments of AlienVault’s or Fortinet’s main and biggest customers. I don’t know where that is, but my guess is offices and other classic IT shops. They probably run mail and AD servers, databases, and Microsoft endpoints; are midsize; have employees who are English-speaking; are located primarily in North America; and are regular commercial-ish businesses.

The operational security of Fortinet and AlienVault mean they shouldn’t openly disclose the exactly location of their IDS sensors. Fortinet at least publishes vague data about where threats originate; as far as I know, AT&T says nothing about AlienVault’s shared content. How to adequately corroborate processes and conclusions in security to understand the extent of generalization that is justified is itself an open research question. We are working on it, but it’s a wicked problem.

Organizations should measure and validate the usefulness of EPSS in their environments. No organization should assume that its environment matches the data used to train EPSS. However, many organizations’ environments should be a near-enough match. It would help us solve this problem if organizations would tell the SIG how they validated it-environment and what the results were.

EPSS fairly consistently gives, for instance, low scores to IoT vulnerabilities that we know are being exploited. For example, there are several CVE IDs in CISA’s list with low EPSS scores, and there are plenty of CVE IDs with high EPSS scores not in that list. People seem to think that this discrepancy means one or the other is wrong. Actually, it probably does not inform rightness or wrongness about either. The discrepancy might be telling us that attackers use different methods to attack the organizations in CISA’s constituency than they use to attack AlienVault’s and Fortinet’s constituency. This interpretation would be consistent with the fact that we know attackers target victims using specific infrastructure. Perhaps, however, it is just the result of the expected error rate reported about the EPSS model. This result further suggests to me that organizations need to empirically validate that their environment fits well enough to the environments used to train EPSS.

How to Use EPSS Now

EPSS is great in that it is bringing attention to threat data. I agree 100 percent that paying attention to what attackers are exploiting is important in prioritizing vulnerabilities. The EPSS FAQ does not provide specific advice on where to start using EPSS scores; I’ll share my advice here. In summary, EPSS is not suited to software vendors, coordination CSIRTs, or PSIRTs and SOCs handling a large number of misconfigurations or other vulnerabilities without CVE IDs (common with bug bounty programs). EPSS is not good for protecting Operational Technology networks in infrastructure, healthcare, or manufacturing sectors. It is suited to teams doing patch management in mature organizations that already have good asset management and the surge capacity to handle emergencies posed by widely exploited vulnerabilities as an input to decisions about vulnerability management. EPSS is clear that “EPSS is not and should not be treated as a complete picture of risk.”

SSVC could use EPSS data and combine it with these other information items right now. CVSSv3 can also account for threat in the temporal metrics. I happen to not like that CVSSv3 implicitly assumes everything is being exploited (default worst case, temporal scores only reduce scores) even though we know from EPSS data and other sources that most vuls are not exploited; however, properly using the CVSS base, environmental, and temporal scores is probably better than using EPSS alone. When the EPSS website says EPSS is better than CVSSv3, it means CVSSv3 base scores. The CVSS SIG has made it clear you should not be using CVSS base scores by themselves to rank and sort vulnerabilities. EPSS is useful because it calls attention to that shortcoming with the way people have used CVSS base scores.

A high EPSS score is a signal that many people could pay attention to. If your environment resembles the environment that EPSS data comes from, you should use a high EPSS score to set values in SSVC or CVSSv3 temporal metrics related to public proof of concept or active exploitation. That would certainly be a win. To be clear, this is my recommendation on how to combine CVSSv3 with EPSS; there is no consensus on this topic.

One way to validate that your environment resembles the same starting point as the EPSS data is to try to measure how many false positive prioritizations and the number of misses of things you should care about. For stakeholder organizations that do not have the maturity to evaluate this question, improving your asset management system is probably a better time of your life than adopting EPSS.

You also might want to know how expensive it will be to remediate the CVE ID. I don’t know of anyone who has a good public system for this, but we know it’s something people need to be able to integrate into the decision.
Additional Resources


Read the SEI white paper “Historical Analysis of Exploit Availability Timelines,” which I coauthored with Allen D. Householder, Jeff Chrabaszcz (Govini), Trent Novelly, and David Warren.

The CERT Coordination Center Vulnerability Notes Database provides information about software vulnerabilities. Vulnerability notes include summaries, technical details, remediation information, and lists of affected vendors.