White Paper
Cisco Public

Prioritization to prediction, vol. 9

Role of the known exploited vulnerability catalog in risk-based vulnerability management

The bridge to possible

Table of Contents

Introduction & key findings

"Cybersecurity and Infrastructure Security Agency's Binding Operational Directive 22-01—Reducing the Significant Risk of Known Exploited Vulnerabilities" doesn't exactly roll off the tongue, does it? Despite the less-than-melodic name, it has become an important nexus of information about vulnerabilities and attacker activity. Frequently referred to as "the KEV" (Known Exploited Vulnerabilities), it's a list of vulnerabilities known to the Cybersecurity and Infrastructure Security Agency (CISA) to be the target of exploitation activity (more on what precisely that means later on).

If you are reading this report, it's probably obvious why we should pay attention to something like the KEV. Being nerdy data scientists, when we turn our attention to something like this, we do so with an eye toward statistics and placing the KEV in the context of other data we have available to us. This latest installment of the Prioritization to Prediction research series, created by the Cyentia Institute and sponsored by Cisco (formerly commissioned by Kenna Security), does just that: It explores the KEV and gives some context to what it means (and doesn't mean) for other organizations. Moreover, we demonstrate how the KEV can fit into any risk-based vulnerability management program. In fact, here are some key findings, but you'll really want to read the whole report to get the good stuff.

- The KEV is growing in fits and starts; however, it constitutes a very small fraction (only 0.5%) of all vulnerabilities.
- What's on the KEV differs from the larger population of vulnerabilities and known exploited vulnerabilities. They are more severe (1/3rd of the KEV is critical vs. just 15% of all vulns) and have different properties (more likely to be recent and from different vendors).
- Nearly every organization (98.3%) has detected a KEV vulnerability on its network at some point.
- The KEV offers a good signal regarding exploitation, but it doesn't include everything. 94% of CVEs that have exploitation activity aren't on the KEV.
- The KEV should be one of many data sources used in your risk-based vulnerability management strategy.

Exploring the KEV Catalog

The CISA KEV Catalog was initially published on November 3, 2021 along with the aforementioned compulsory directive. In this section, we're going to explore the size of the KEV, how it's grown, what vulnerabilities are in there, and which ones are cropping up a little more (or less) often than we'd expect. The first thing to clarify is exactly how a vulnerability comes to be included in the KEV. There are three criteria:

1. The vulnerability must have an assigned Common Vulnerabilities and Exposures (CVE) ID, and that CVE ID must be published (not reserved).
2. There must be evidence of attempts (successful or not) at active exploitation.
3. There must be clear action that the affected entity can take to remediate the vulnerability.

We'll begin by discussing the first and third criteria and spend a bit more time mincing words on "active exploitation" last. The first criteria–the fact that KEV keeps track of vulnerabilities the way just about everyone else does by using CVE ID[1]–will make our lives easier. This means that all the rich information contained in CVEs is at our disposal to slice, dice, and correlate with things that appear on the KEV. In addition to all that rich information we've collected over the years, CISA has decided to include two further pieces of information:

1. The suggested action needs to be taken to address that vulnerability.
2. The date by which those under the compulsory directive are required to remediate the vulnerability.

Given our prognostication concerning what vulnerabilities should be remediated (and even how quickly), this is actually pretty important and will allow us to do some very interesting analyses. As someone has pointed out in the past, publishing information about exploits where there is no remediation available likely increases everyone's risk. So, it's good that CISA makes available remediation its third criteria; in fact, this is the only reason a CVE would be removed from the KEV, if the remediation for some reason or another became untenable.

Having discussed the first and third criteria, the second concerns what people focus on when they hear about the KEV: The vulnerability is under "active exploitation" which is defined as: One for which there is reliable evidence that execution of malicious code was performed by an actor on a system without permission of the system owner.

In order to make it on the KEV, vulnerabilities must have a published CVE ID, have evidence of exploitation, and can be remediated.

CISA also makes it clear that this includes both successful and failed attempts at exploitation. Now, being the data-driven folks we are, the above raises some questions about exactly what is meant by "reliable evidence" but so far, CISA is not all that forthcoming. But it does say things like vulnerability scanning and proof of concept (PoC) code do not meet the evidentiary requirement. That suggests some form of actual exploitation for malicious purposes. But we aren't going to just take their word for it; we'll take a look at how the list correlates with other sources of active exploitation that we have.

With some history and definitions out of the way, let's dive in and start looking at the KEV from our unique data-driven perspective.

How big is the KEV Catalog?

On that opening day in November 2021, the KEV contained 287 CVEs from 84 vendors. Since then, it's more than tripled to 965 vulnerabilities as of July 1, 2023 and includes 199 vendors. This totals about 34 vulnerabilities per month, but if you think this is a steady pace of about one vulnerability per day, take a gander at Figure 1.

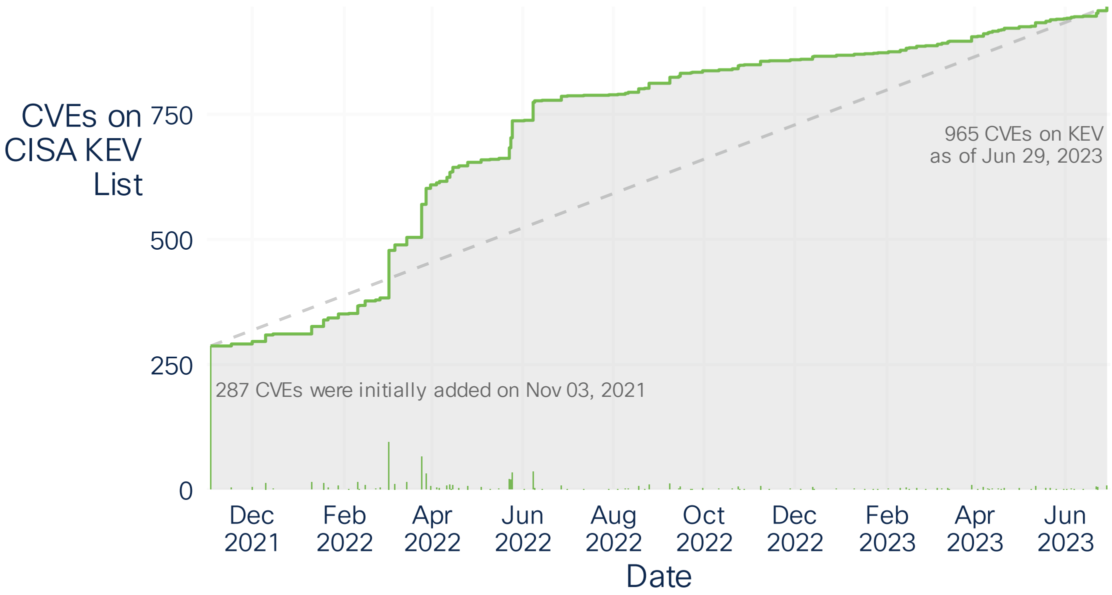

Figure 1. Growth rate of the KEV Catalog

The key takeaway here is that vulnerabilities are added to the KEV in fits and starts, with long periods between additions. The longest drought between sprinklings of new vulnerabilities to worry about was 26 days from December 15th, 2021 to January 10th, 2022. The largest addition, other than the initial 287, occurred in March 2022 with a dump of 95 vulnerabilities.

Now, tripling in size in a little over a year seems a tad scary given the nature of the vulnerabilities we are talking about here. Nevertheless, it turns out that there are a lot of vulnerabilities not on the KEV, with our 965 comprising just 0.47% of the ~206k CVEs published.

Takeaway: The KEV adds an average of 42 vulnerabilities per month (but it's bursty).

How comprehensive is the KEV Catalog?

You may interpret the previous result of the KEV as comprising a little less than half a percent of all CVEs as a pretty good thing. After all, if we only really have to worry about 1 in every 200 vulnerabilities, then maybe something like the 3x growth rate we saw in the last year ain't so bad. But this does beg the question about the comprehensiveness of the KEV Catalog. CISA claims it has evidence that these vulnerabilities are under active exploitation, but it makes no claim that these are all the vulnerabilities under active exploitation.

So, let's examine what we know about vulnerabilities exploited[2] in the wild from our sources. Specifically, Cisco Vulnerability Management tracks numerous public and private sources of intel on exploits, including Cyentia's Exploit Intelligence Service,[3] so let's compare the two sources in Figure 2.

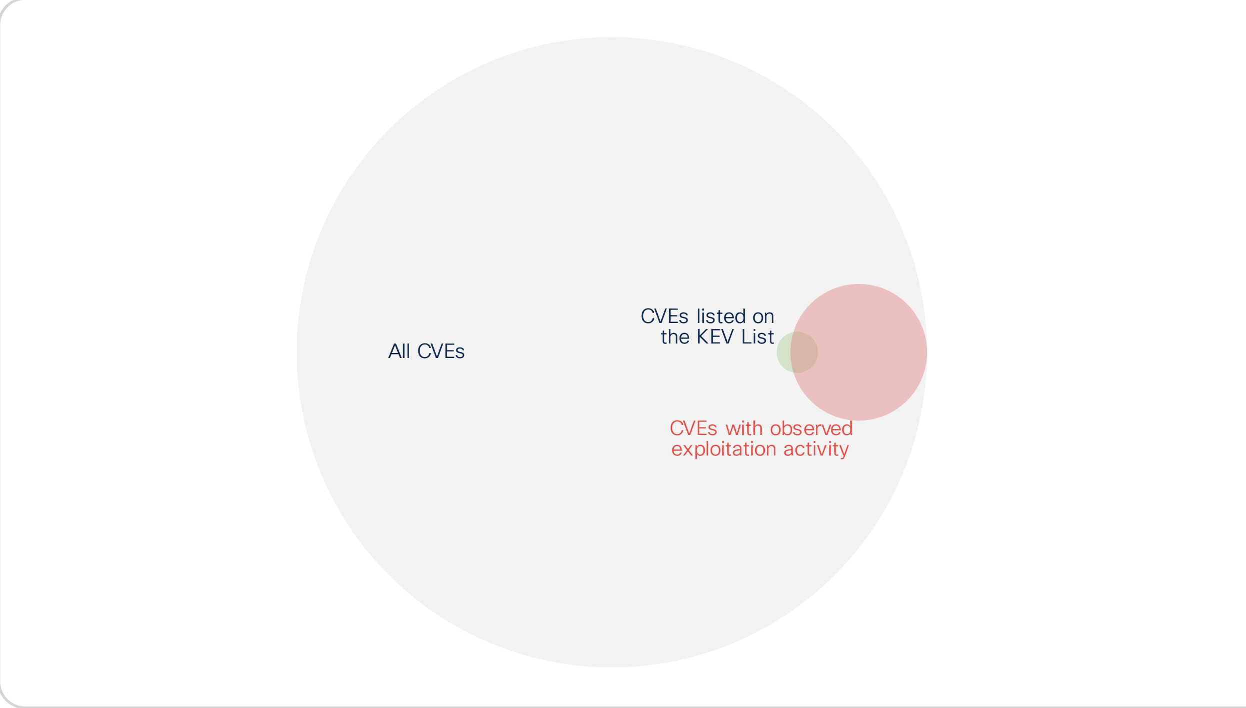
All CVEs
CVEs listed on the KEV List
CVEs with observed exploitation activity

Figure 2. Proportion of vulnerabilities exploited according to various sources

The pink circle shows us something that readers familiar with our work will know: roughly 5% of all vulnerabilities published have some exploitation activity according to our sources. The much smaller (10x smaller) green circle represents the KEV with just 0.45% of vulnerabilities listed. The substantial overlap indicates that a majority (around 2/3rds) of vulnerabilities on the KEV show up in our exploitation data. So, there's solid evidence for the "K" in KEV, and we have a high degree of confidence that threat actors are targeting (or at least, have targeted) these vulnerabilities. Practically speaking, that means if you remediate vulnerabilities on the KEV, you're not wasting your time. And that's a Very Good Thing in vulnerability management programs, where prioritizing remediation is…well, priority No.1.

But what about that vast swath of pink that doesn't overlap with the KEV? Roughly 94% of the vulnerabilities we are aware of that are being actively exploited in the wild don't show up on the KEV. So, why the discrepancy? It's mostly likely due to differences in the methodologies of collection, but it's tough to draw concrete conclusions without further information on what "actively exploited" means precisely to the KEV. One hypothesis we'll highlight in a bit is that the KEV's relative recency biases it towards newer vulnerabilities, meaning our tracking of older, though still actively exploited vulnerabilities are less likely to show up in the KEV. The key takeaway here is that the KEV is a good signal, and if you focus on this, you are reducing risk, but it isn't the be-all and end-all of exploitation activity.

The KEV focuses on vulnerabilities that are actively being exploited in the wild, but it's worth taking a quick look at exploit code. This matters because our prior research has shown a 15x increase in exploitation activity for vulnerabilities with published exploit code. Figure 3 is similar to the former Figure 2, except we are slicing by exploit code.

Why aren't all exploited CVEs in the KEV?

We mentioned the inclusion criteria previously, but even with those criteria combined with the acknowledgement that not all vulnerabilities that meet the criteria are in the KEV, it's worth asking, "Why this particular list and not more?" We (not being CISA) don't have a good answer, but that won't stop us from offering some thoughts. The following text is in a callout box to quarantine our speculation from, ya know, actual data analysis.

The primary purpose of the KEV is to identify what government agencies are required to fix within a specified time frame. This means that one implicit criterion (though not mentioned in the documentation) is that vulnerability affects software the U.S. government runs and cares about. As such, it's unlikely a vulnerability for Candy Crush will ever show up on the KEV, given that it is unlikely to be a supported piece of software at the federal level. This implicit criterion eliminates a certain proportion of "exploited-in-the-wild vulnerabilities." There may be other implicit criteria that we aren't privy to, but we don't want a visit from anyone in a dark suit from a three-letter agency, so we'll stop speculating here.

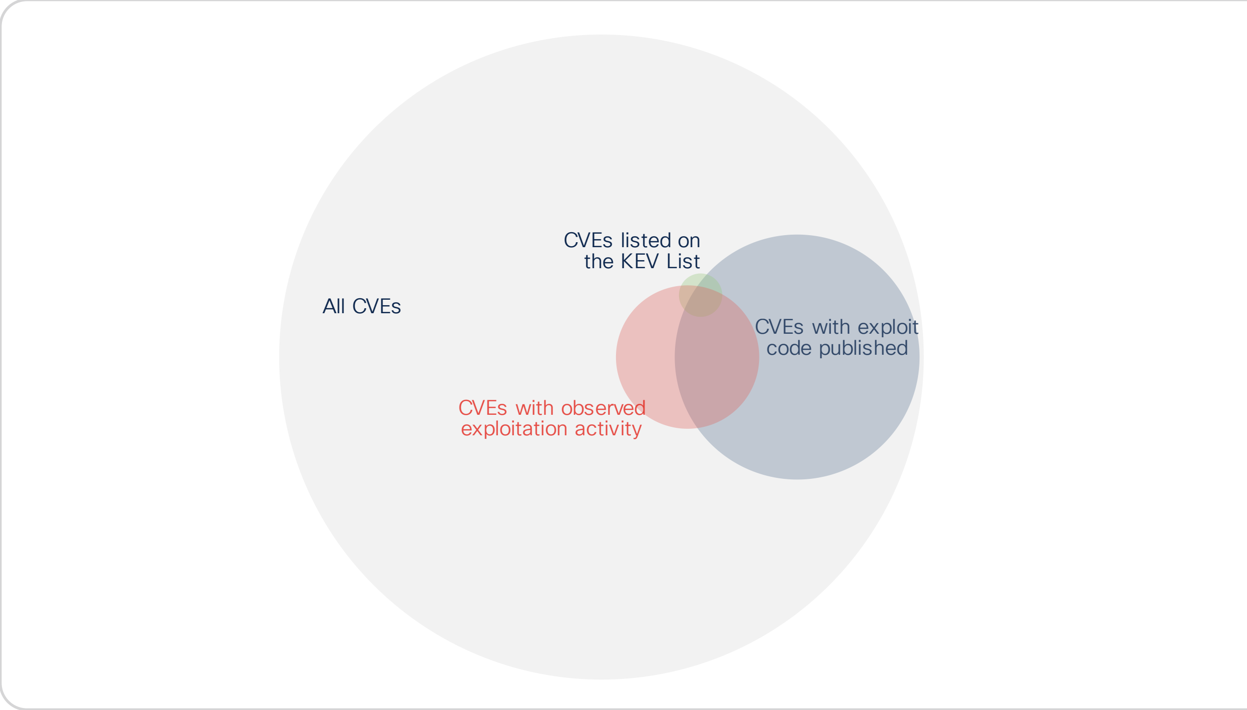
CVEs listed on the KEV List
All CVEs
CVEs with exploit code published
CVEs with observed exploitation activity

Figure 3. Overlap between vulnerabilities with exploit code and exploitation activity and appearance on the KEV

Over two-thirds (68%) of CVEs in the KEV have exploit code available, which is a testament to our previous findings. What's interesting is that the large gap we saw before (we see a lot of exploitation that the KEV doesn't) is not as pronounced here. About 55% of CVEs for which we have evidence of exploitation have exploit code publicly available. The implication here is that no matter how exploitation is measured, the likelihood of there being publicly available exploit code is (very roughly) the same.

Takeaway: The KEV offers a good start but cannot be the be-all and end-all of your risk-based vulnerability management strategy.

How old are the vulnerabilities in the KEV?

We've established that the KEV represents a good portion of the vulnerabilities that we also know are exploited as well as a few more. We speculated there might be some recency bias causing the mismatch, and now we can do a bit of poking around to find out if there is any evidence of that.

Figure 4 shows the distribution of vulnerabilities by publication date on the KEV (blue bars) and those for which we have evidence of exploitation (red bars). Though the KEV has been around for less than 1.5 years, it contains vulnerabilities that are much older, with the earliest being CVE-2002-0367, published on April 2nd, 2003. Among CVEs on the KEV, 5% were published prior to 2012 and 81% were published before the KEV's birthday on November 3, 2021.

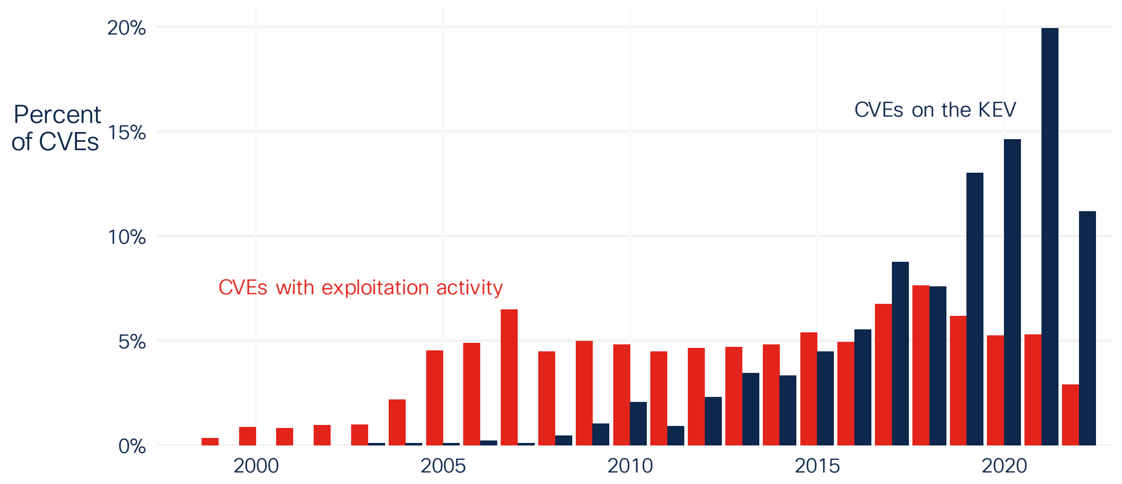

Figure 4. Distribution of publishing data for vulnerabilities with exploitation

It's interesting to contrast the publication date distribution of the KEV with that of the vulnerabilities for which we have exploitation activity. In order to account for recency bias in our own data, the red bars in Figure 4 only show vulnerabilities that had evidence of exploitation in the last 90 days. What's clear is that while the KEV is skewed towards recent vulnerabilities, there is a much more uniform distribution in what attackers are using right now. Some of those golden oldies CVEs are just too good for attackers to pass up.

Takeaway: The KEV has a recency bias. If you're prioritizing vulnerabilities based mainly on the KEV, you may overlook some oldies that hackers still consider goodies.

What vendors are represented?

So, which software vendors appear most frequently in the KEV? This tends to be top of mind when it comes to any list of vulnerabilities, and the KEV is no different. Plus, there's a good reason for that; vulnerability management processes are often product-centric (for example, Microsoft's Patch Tuesday).

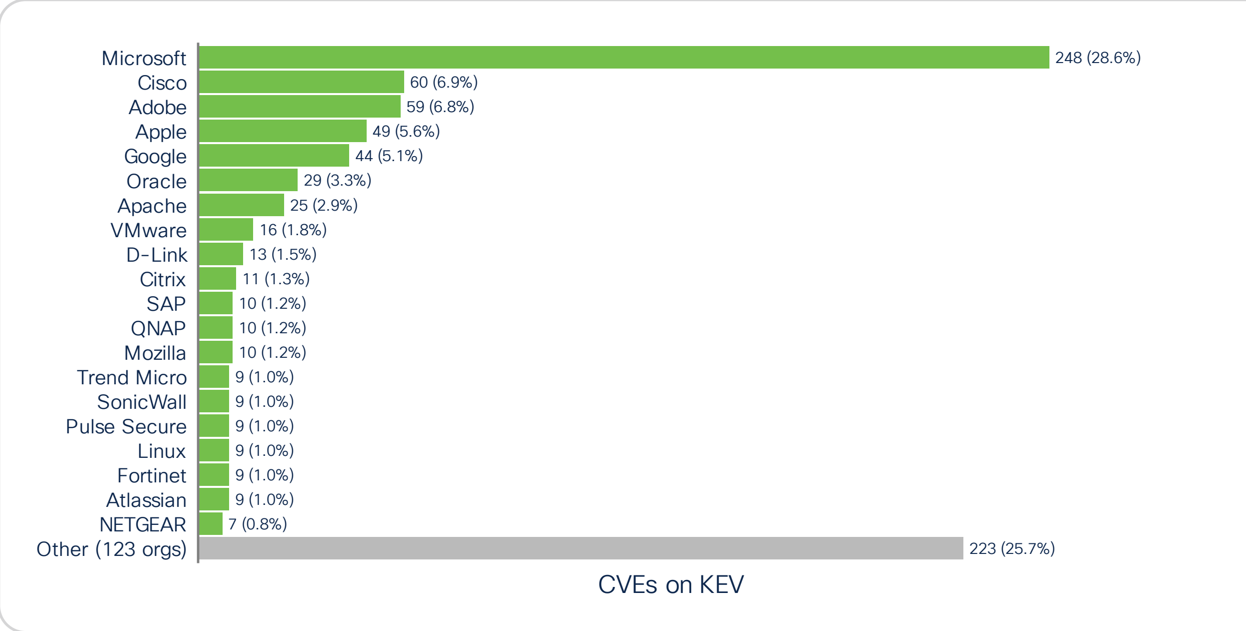

Figure 5. Count and percentage of CVEs on the KEV associated with each vendor

Figure 5 gives a breakdown of the top 20 vendors present on the KEV (plus a big lump of "others" who fall below 0.8%). Speaking of Microsoft, more vulnerabilities on the KEV affect its products than any other vendor by a long shot. We need to be careful about casting stones here though, because this doesn't mean Microsoft products are inherently more vulnerable than others. It means there are more "known" exploits against Microsoft products, and this has a lot to do with the fact that they have a ton of products with a huge install base. It's also likely a result of Microsoft's efforts to track and report exploits. Moreover, because of its efforts to make patches and security updates as painless as possible, Microsoft CVEs are more liable to meet the "must have a fix" criterion for inclusion in the KEV.

Given Microsoft's prominence across everything that would make it prone to vulnerabilities on the KEV, it's worth asking whether this prominence is out of proportion to its presence among all exploited CVEs. We answer that question with a resounding, "No. In fact, the opposite is true," in Figure 6.

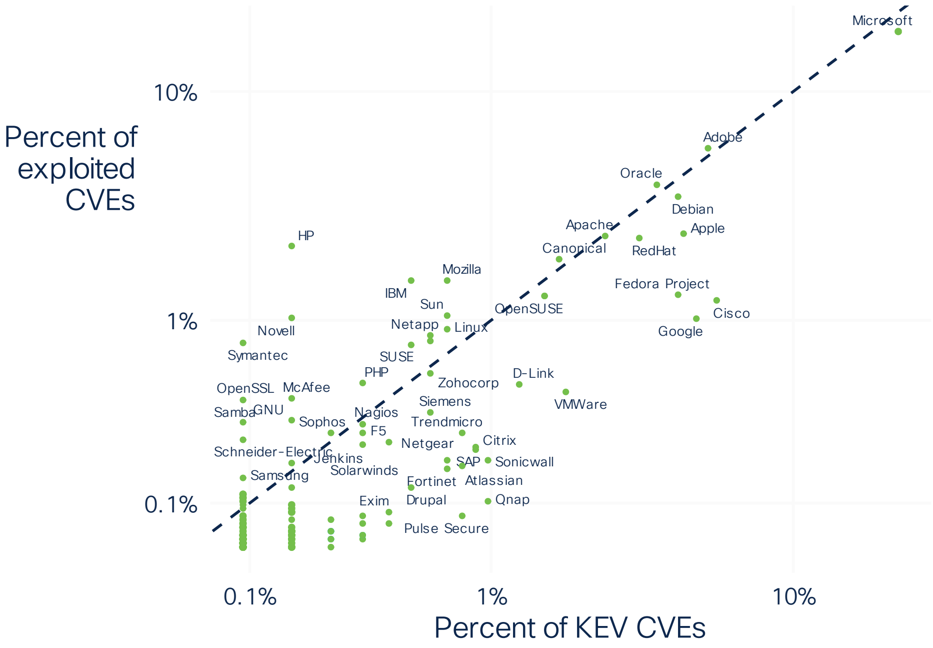

Figure 6. Proportion of CVEs in the KEV vs. all exploited CVEs by vendor

The important visual cue in Figure 6 is the diagonal dashed line. Vendors above and to the left of this line have a lower proportion of CVEs on the KEV relative to their prevalence in the general population of exploited CVEs, while those below and to the right are overrepresented on the KEV. So, Microsoft is actually a tiny bit overrepresented on the KEV, yeah, it's 22.2% of the KEV but it's 18.2% of all exploited CVEs. HP is an interesting case here accounting for 0.1% of KEV CVEs but a fairly high 2.1% of all exploited CVEs. This may be due to the fact that while HP used to be a market giant, it is not quite the mover and shaker it once was. Given the KEV has only been in existence for a little over a year, that recency bias we talked about may be influencing things.

Who is featured on the KEV way more than they should be? Google, Cisco, and Fedora are high up in both but overrepresented on the KEV. There is likely no overarching reason why for any of these vendors. For Cisco, it's likely the fact that it makes internet-facing devices that are widely used throughout enterprise environments. For Fedora, it could be because so much enterprise infrastructure runs on servers using some flavor of Red Hat Linux. As for Google, well, how many of you downloaded and read this report using Chrome or an Android phone?

Takeaway (more of an FYI, really): Some vendors are more or less represented on the KEV relative to the total number of exploited vulnerabilities.

Are vulnerabilities with certain properties more or less likely to appear on the KEV?

Who the responsible vendor is and when a vulnerability came out are just two small bits (Bytes? Nits?) of information about individual vulnerabilities. But it behooves us to talk about some of the other important features that make a vulnerability more or less likely to appear on the KEV. We must be careful here because categorizing and contextualizing vulnerabilities is challenging. There are many different data sources, many of which are in complex and difficult to parse formats (free-text descriptions and the hierarchical nature of CWEs come to mind, for example). This somewhat messy data makes it tough to say, “Vulnerabilities with this feature are KEV-prone.”

Nevertheless, in the interest of at least pulling back the curtain a bit, we examine two important properties: the Common Vulnerability Scoring System (CVSS) and vulnerability descriptions. CVSS is (for better or worse) the wet finger in the wind people use for, “How panicked I should be about this vulnerability,” so it’s worthwhile taking apart the vector a bit and having a look. We do that in Figure 7.

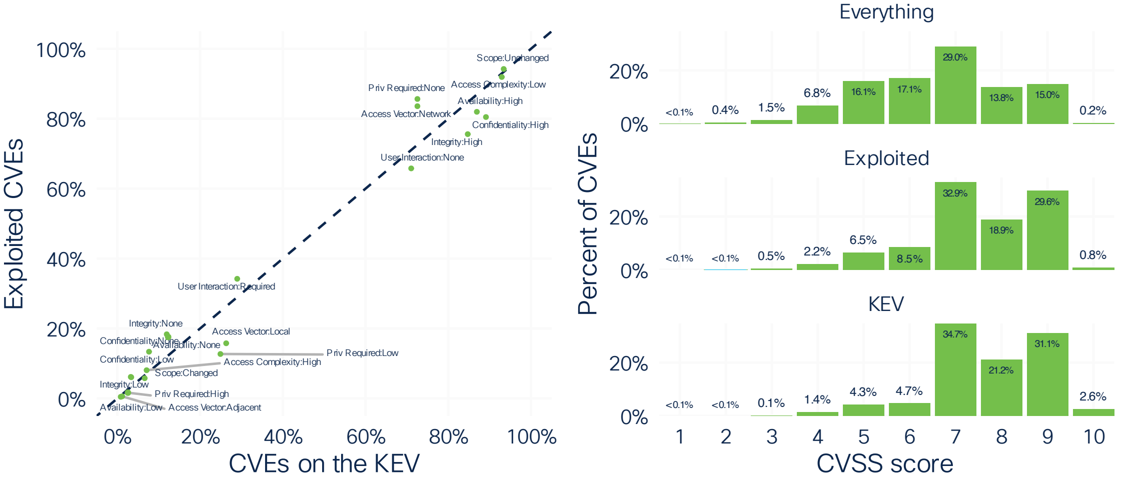

Figure 7: The left panel shows the relative prevalence of a metric value on the KEV vs. all exploited vulnerabilities, similar to Figure 6. The right side shows the distribution of the base scores for all CVEs, exploited vulnerabilities and the KEV

First, we can see that overall, the distribution of CVSSv3 vectors is more or less in agreement between exploited vulnerabilities and KEV vulnerabilities. This is unsurprising, as both lists are going to focus on high-value things for attackers. We’ll highlight a few potentially notable differences. Let’s start with the infamous CIA triad. We can see that for all three (confidentiality, integrity, and availability), “High” is overrepresented. This makes sense, as the KEV contains the most dangerous of the dangerous and so will lean hard into the things that can potentially do some damage. What’s rather counterintuitive is the underrepresentation (though ever so slight) of Privileges Required:None and Access Vector:Network. Chances are this is because the data on exploited vulnerabilities is in many (but not all) cases from IDS sensors biased towards those things that can be network exploitable with few secrets needed by the attacker.

The right-hand portion of Figure 7 is perhaps more straightforward. Simply stated, CVEs that are exploited in the wild generally have higher CVSSv3 Base Scores[4] than average, and those in the KEV have higher scores still. This makes sense, given the Base Score incorporates information about how easy a vulnerability is to exploit, and the easier it is to exploit, the more likely it is to appear in either our exploited list or the KEV.

If the combination of the CVSS vector and score constitutes the wet finger in the wind for how worried we should be about a vulnerability, the description is all the way down in the dirt. At times more or less lucid, these descriptions get to the nitty gritty of what a vulnerability is. However, giant blobs of text are not all that conducive to the forms of analysis we've been doing. We transform these text descriptions using some fancy natural language processing (NLP) to extract specific interesting keywords, and then in Figure 8, we draw the same type of comparisons as those in the previous two figures.

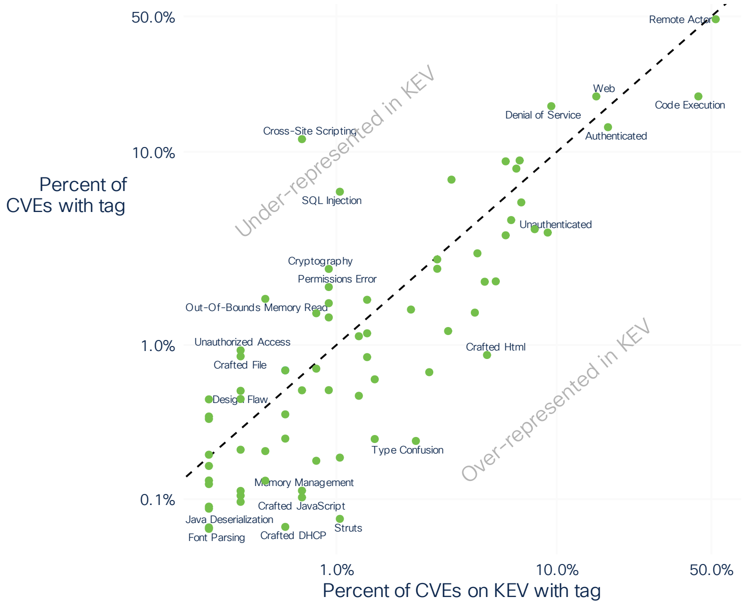

Figure 8. Proportion of CVEs in KEV with tags vs. all published CVEs

One interesting takeaway here is the overrepresentation on the KEV of "crafted" vulnerabilities; for example, "crafted HTML," "crafted javascript," and "crafted DHCP." This may be a reflection of the type of targeted crafted attacks that government organizations see, but we are getting a little close to speculative territory with that analysis.

A weird quirk of our technique is the overrepresentation of both "authenticated" and "unauthenticated" in the KEV CVE descriptions. This is not a contradiction, but rather a reflection of the fact that this can be used in two different ways in descriptions. It can mean the attacker needs to be authenticated or that the vulnerability allows an attacker to become authenticated. The overrepresentation means it's more likely the latter.

On the other side of the diagonal, we see things like cross-site scripting and SQL injection largely overrepresented. This likely goes back to the exploit data, which may be more prone to detect attackers using widespread "spray-and-pray" techniques to identify vulnerable systems.

Takeaway: Both exploited and KEV vulnerabilities are higher in severity than those in the general population of vulnerabilities.

KEVs in the enterprise

The vulnerabilities that are being actively exploited "somewhere" may not be all that impactful if they don't affect any software on computers on actual networks. This has been a common theme in the P2P series for a long time. The vulnerabilities you need to worry about are those that are already being exploited (or have PoC code available and so could be easily exploited) and that are actually in your environment. No need to worry about something that doesn't affect your network. With that in mind, we examined the 9.6M active assets that Cisco Vulnerability Management tracks to see how often the KEV CVEs show up among the 637.5M instances of vulnerabilities in those assets.

What percentage of KEVs affect active assets?

The title of this subsection is the most obvious question we can start with. In prior P2Ps, we've found that approximately two-thirds of all published CVEs have not been observed or detected by organizations. Not ones to rest on our laurels (and we recognize that the underlying data can evolve), we updated that particular stat based on the latest data, and it still holds true (the left two quadrants on the left-hand side of Figure 9).

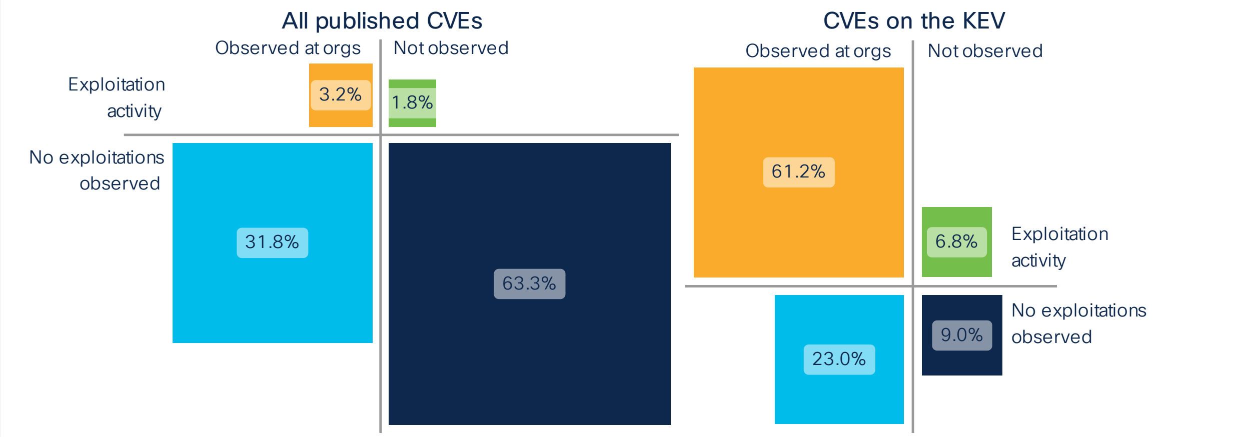

Figure 9. Proportion of vulnerabilities observed and/or exploited on the CVE vs. KEV List

Here's a quick refresher on how to read Figure 9, focusing on the left side:

- The horizontal line separates CVEs with known exploitation activity (about 5%).
- The vertical line separates CVEs that show up somewhere within organizations.
- Thus, when looking at all published CVEs, only 3.2% represent a real and present risk to organizations (both observed in an organization and exploited in the wild).
- We regularly advocate that vulnerability management programs prioritize their efforts on these high-risk vulnerabilities.

> "Only 3.2% of all published CVEs represent real and present risk, while 60% of the KEV warrant priority remediation."

So, if we only examine those CVEs in the KEV Catalog, how does it compare? The right chart reveals some pretty substantial differences. First, the lower-right, teal-colored quad is much smaller among KEV CVEs, indicating that far fewer of these can be easily deprioritized (not observed and not exploited). Conversely, there exists a much larger percentage of CVEs on the KEV that are both observed and exploited, meaning that 60% of the KEV should be a priority for organizations.

Fastidious readers will be drawn to the contrast between the right side of Figure 9 and the Venn diagrams in Figures 2 and 3. In combination, this set of figures indicates that the KEV is incredibly efficient, in that almost everything on there has been exploited in the wild. On the other hand, it achieves pretty poor coverage because there's a lot of stuff not on the KEV that is exploited in the wild. We'll hammer on this more in later sections, but good on you, eagle-eyed reader, for figuring it out early.

Takeaway: The majority of KEVs represent real risk to organizations and warrant prioritized remediation.

How prevalent are KEVs across organizations?

The next question is the obverse side of the coin to the previous one. We can have a single number, 86M, which represents the number of times a KEV vulnerability has shown up in an organization's asset. While this might make it seem as if there is a single number that sums things up, we should probably slice this across vulnerabilities, assets, and organizations in different ways, and Figure 10 does that.

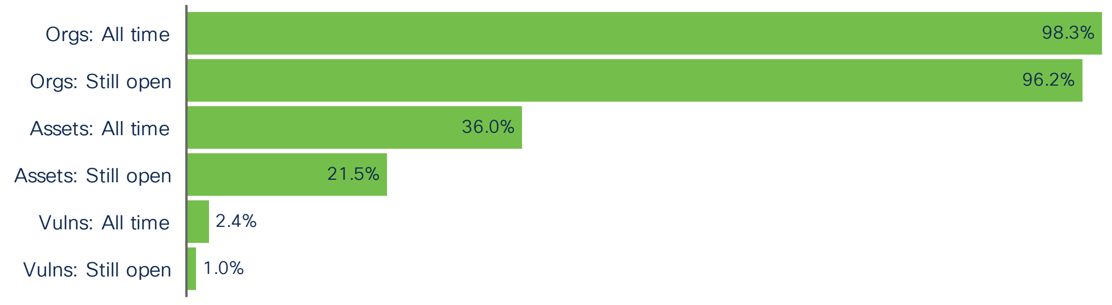

Figure 10. Prevalence of KEV CVEs in organizations by various measures

The funny thing about Figure 10 is we could give just about any answer you want. The percentage of organizations[5] that have ever seen a vulnerability on the KEV is almost all of them (98.3%), even if we restrict it to "still open" (96.2%). Among assets, that number drops to middle-of-the-road figures of 36% and 22% for those that have seen a KEV vulnerability or still have an open KEV vulnerability, respectively. Finally, we could say that they are rare if we consider that they only make up 2.4% of the 3.5B ever found in assets and 1.1% of the 815M vulnerabilities open at the time of the study.

This is not us trying to be cagey or hedge our bets on what will get the best headlines but rather just a result of doing this analysis at varying levels of aggregation. Let's dive a bit deeper to see how often individual KEV vulnerabilities show up in organizations. Figure 11 reveals that prevalence varies widely—something we often see in this space.

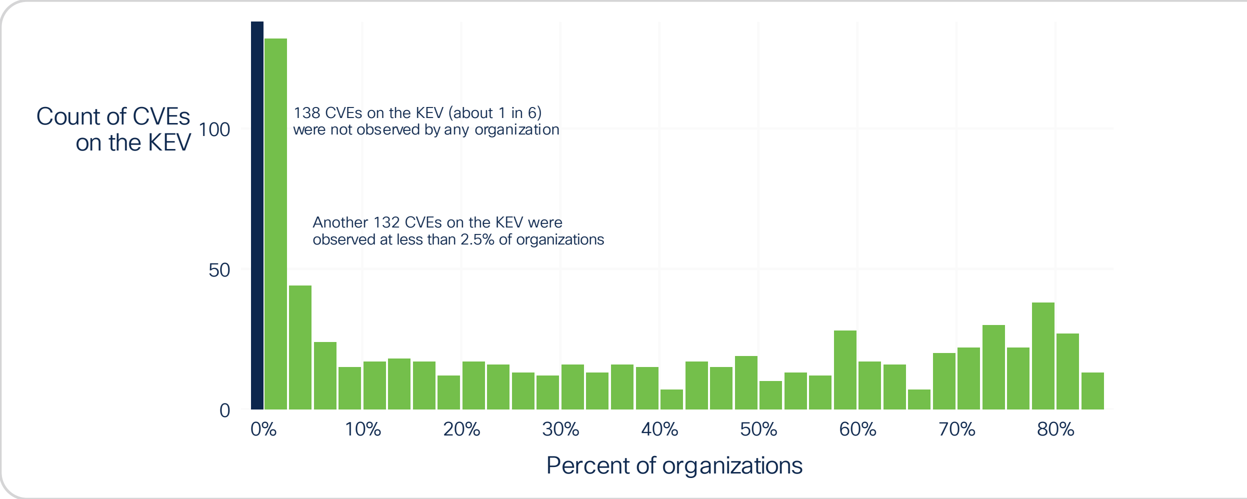

Figure 11. Distribution of KEV CVE prevalence in organizations

Notably, 16% (138 out of 866) of the KEV vulnerabilities have never appeared in an organization's vulnerability scans. Another 130 only crop up in less than 2.5% of organizations, which is little more than a quarter of the KEV. However, that tail extends quite a ways: 1 in 9 KEV CVEs are seen by more than 75% of organizations, while a number of the vulnerabilities on the KEV show up in as many as 85% of organizations, with CVE-2020-1147 topping the list appearing in 84% of the organization's scans. This is an infamous SharePoint remote code execution vulnerability that affected a great many Microsoft products, essentially guaranteeing that it will show up in a lot of organizations. In the next section, we'll dive into what other KEV vulnerabilities seem to crop up everywhere.

Takeaway: Vulnerabilities on the KEV are ubiquitous in organizations but constitute a small portion of the overall volume of vulnerabilities vying for remediation priority.

What are the most prevalent KEVs?

CVE-2020-1147 (the most prevalent vulnerability we observed) gives us a good excuse to get out our magnifying glass and start examining the commonalities among KEV CVEs at an individual level, albeit with charts, graphs, statistics, and such. Once again, there are multiple ways of measuring "common." We'll use the proportion of organizations in which a CVE was observed and the percentage of vulnerable assets affected and put those two measures in a scatterplot in Figure 12.

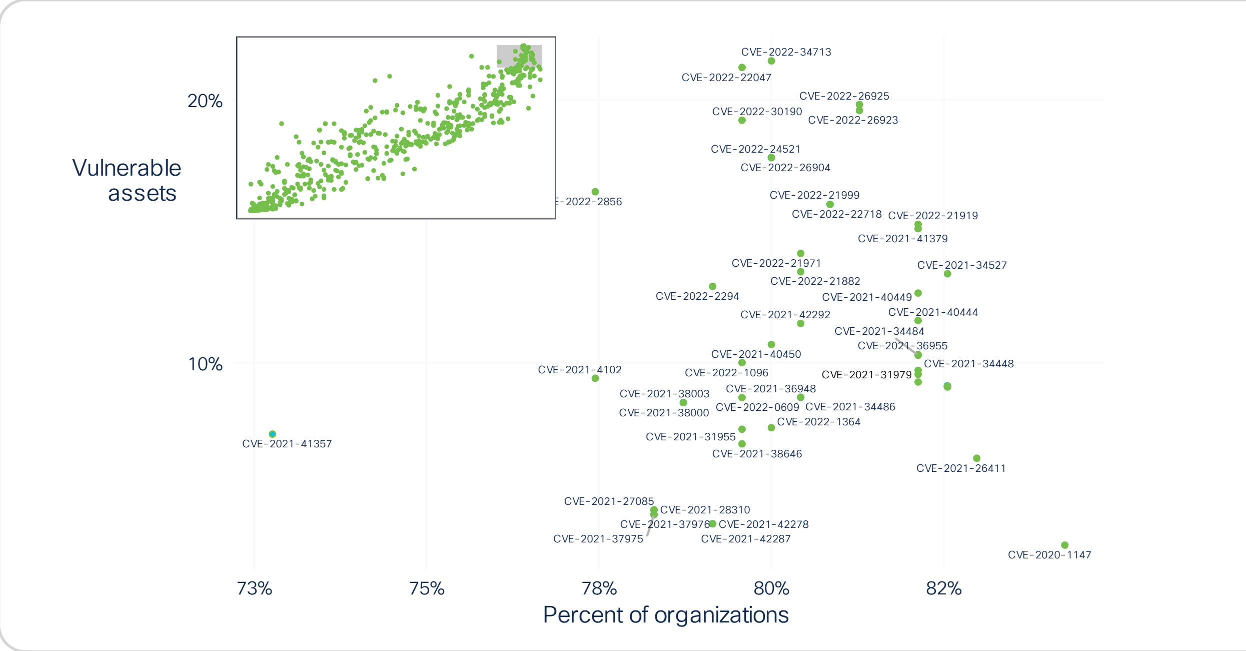

Figure 12. The most common KEV CVEs by percentage of organizations and of assets

The scatterplot shows CVEs run the gamut. But those at the top right represent the most common on both dimensions. The recency bias that we mentioned earlier is on display here through all the most prevalent CVEs with IDs from the 2020s. Another probable contributing factor is that organizations have had more time to remediate older vulnerabilities and are still working on the newer ones. We'll let the reader poke around Figure 12 to find their own favorites.

What percentage of vulnerabilities predate the KEV?

That last fact may bring our subsection title to mind, and given our data, it is easy to confirm that the majority of the 86M instances of KEV CVEs that appear in assets within our data (63%) were remediated before the KEV was even formed. This is noteworthy because their existence (or not) on the KEV would not influence remediation priority.

Of course, all those vulnerabilities on the KEV didn't just show up at once. Old assets could be scanned for the first time, new assets will come online, and vulnerable software versions installed. In Figure 13 below, we track a vulnerability's discovery relative to its addition to the KEV, and as a reference, its publication.

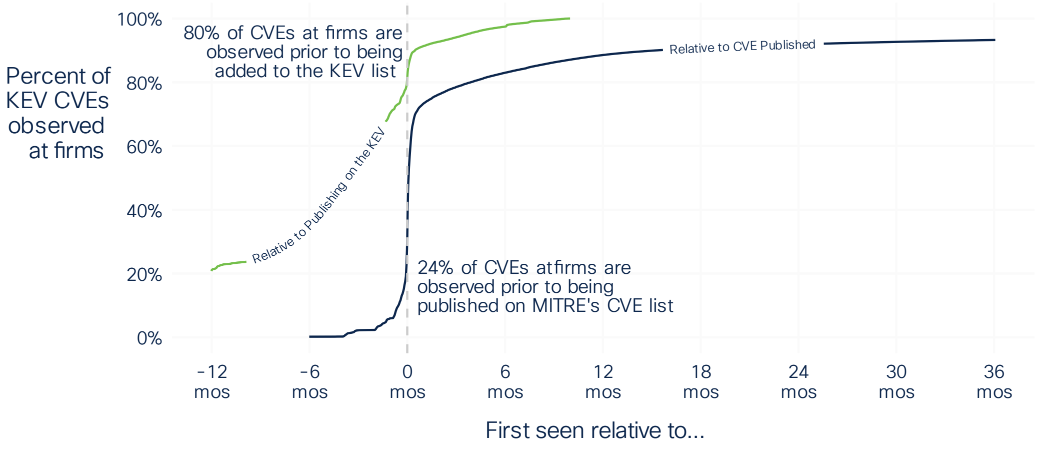

Figure 13. CVE first seen relative to its KEV addition and publication date

Most KEV vulnerabilities are not discovered until they are published, and that discovery spikes rapidly (dark blue line). The green line examines that same timeline except instead of being relative to the publication date, it's relative to a CVE's addition to the KEV, and the difference is striking: most are discovered before their publication on the KEV. This makes us wonder about the wisdom of publishing a "remediation deadline date" along with its vulnerability in the KEV Catalog. Only a small percentage of vulnerabilities are discovered around the date they are added. Does this mean those discovered after their addition should have a compressed remediation timeline, or are they instantly late? It seems more reasonable to publish a "fix this within x days of discovery" notice than a hard date, but nobody asked us.

What percentage of KEVs have been remediated?

The next step after discovery is remediation. Let's focus on the 37% that are open or discovered after their publication on the KEV: of these, 75% have been closed, but 25% remain open at the time of this study.

Given that the KEV includes a binding directive for U.S. government agencies to "FIX THESE vulnerabilities and FAST," an examination of how that 75% breaks down across industries seems like an interesting place to start. What we observe in Figure 14 isn't exactly a shining vision of that sector's reactivity.

> "25% of CVEs that were open or discovered after the KEV launched are still open."

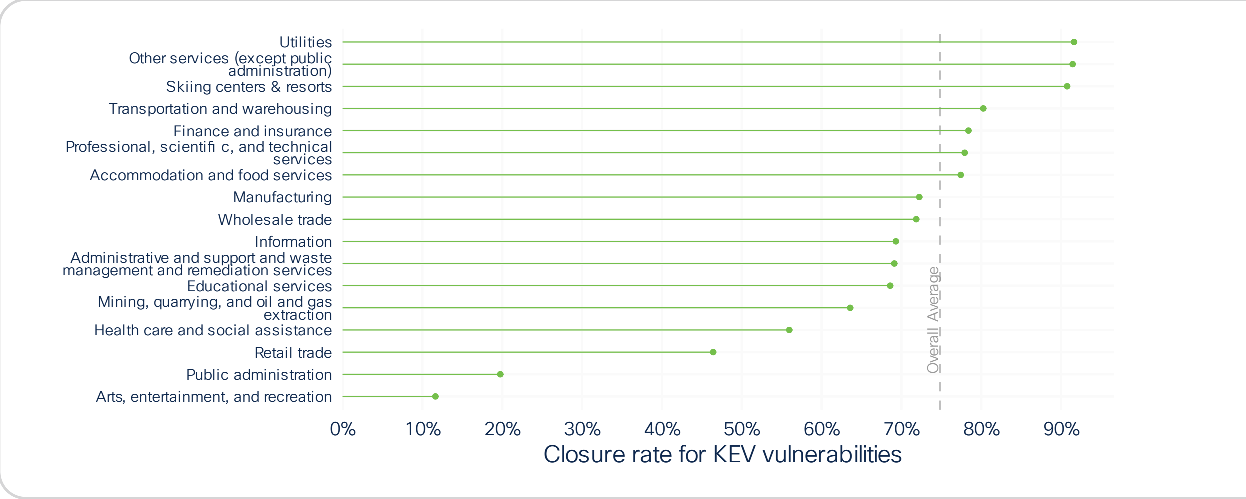

Figure 14. Percentages of closed KEV vulnerability instances discovered after their addition to the KEV

Note in the chart above, "Public Sector" includes U.S. government agencies that are supposed to fix KEV vulnerabilities with a quickness, but they've only closed about 20% of them, which is well below average. Now, one caveat: The public sector has 2.3 M vulnerabilities, and 58.4 K of these are KEV vulnerabilities. That represents 0.06% of the total in both categories, so it's a pretty small sample.

What proportion of KEVs are fixed by the deadline?

An interesting aspect of the KEV is the inclusion of a recommended "remediate by this date" statement, a yardstick that isn't included with most vulnerabilities. The time between the addition to the KEV and the remediation deadline falls into four categories (for the most part).

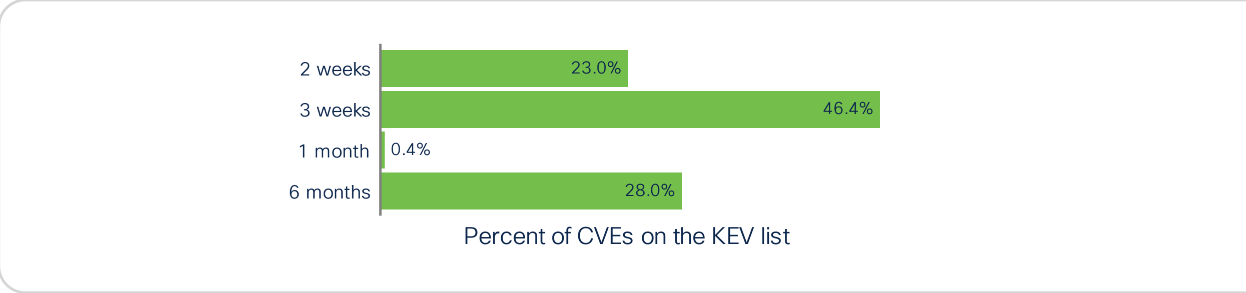

Figure 15. Prescribed remediation windows for CVEs on the KEV List

The numerically inclined will note that these four time periods only add up to 97.8%. The remaining 2.2% have due dates before the date they were added, while others have strange one-off time windows.

Now, generally, we'd reach into our statistical toolkit and pull out our trusty "survival analysis" hammer, but that's less meaningful here, as the majority were found and fixed before the KEV even existed, making it unlikely that their appearance on the KEV influenced their prioritization.

So, let's just examine those 37% that were open at or after their inclusion on the KEV and we have already mentioned that 75% was closed, but what proportion were fixed by the date prescribed on the KEV?

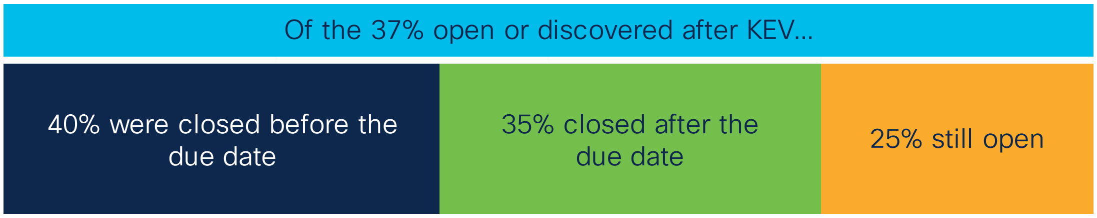

Figure 16. Percentage of KEV vulnerabilities fixed by the KEV-provided deadline

40% is not exactly a passing grade but with an additional 35% closed after the due date that's a solid (albeit belated) C. But remember most of the data is coming from non-government sources and they are not required to follow the KEV.

Remember, one of the requirements for inclusion on the KEV is that remediation is possible for the CVE. But, of course, different types of software have different amounts of… Friction… When it comes to remediation. For example, updating the firmware on a network appliance is a tad more difficult than letting your browser reboot in the background. In Figure 17, we look at what percentage of KEV vulnerabilities within some software categories are fixed by the deadline.

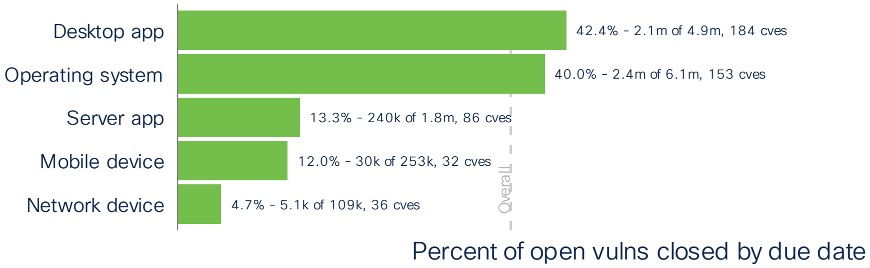

Figure 17. Percentage of KEV vulnerabilities fixed by the KEV-provided deadline, broken down by device type

As a reminder, this represents vulnerabilities that were open at the time of the CVEs' publication on the KEV. Figure 16 tells the exact story we speculated about in the previous paragraph (prescient, aren't we?). Desktop applications and operating systems do get closed by the deadline (or rather the deadline imposed by the KEV) a little over 40% of the time, while things that tend to be a little tougher to "turn off and on again" fare less well, with a paltry 4.7% of vulnerabilities on network devices meeting the deadline.

While we are three quarters of the way through a report focused on the KEV, this figure doesn't tell a uniquely KEV story. Specifically, this is a story about network hygiene and how people are bad at the things that are difficult and laborious, like keeping their network devices up to date. These are internet-facing assets that contain "exploited in the wild" vulnerabilities. Ensuring they are up to date is absolutely crucial to keeping organizations safe; however few do it well. Part of the reason for this is that updating software on these assets can be a pain in the neck; the process isn't as straightforward as clicking "Update Now" on your Windows machine, and the downtime incurred when doing a firmware update can impact an organization's day-to-day business. We'd encourage businesses to power through this pain and get this basic stuff taken care of.

Can organizations realistically fix all KEVs by the deadline?

In volume 7, we established that the typical organization fixes 15.5% of its total open vulnerabilities in a given month. We are aware that the KEV represents a small fraction of all vulnerabilities, but it's reasonable to wonder whether, by our definition of capacity, that 15.5% is enough to clear the board of KEV organizations. To address this, we need to ascertain what proportion of open vulnerabilities are on the KEV but on a month-by-month basis.

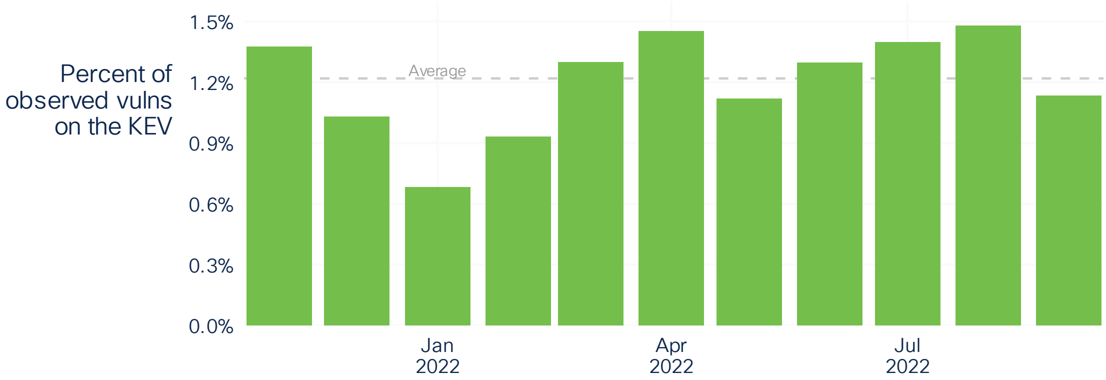

Figure 18. Percentage of open KEV vulnerabilities on a monthly basis

KEVs represent at most, 1.5% of vulnerabilities vying for remediation, which is less than the typical capacity of 15%. So yes, most organizations have ample ability to stay on top of KEVs and the capacity to focus on all those other vulnerabilities that are exploited in the wild.

Takeaway: The typical org has remediation capacity that's 10x greater than what would be required to keep up with the KEV (15% vs. 1.5%).

KEV for risk-based vulnerability management

We've provided some information on the size and shape of the KEV as well as how the catalog manifests itself within the enterprise. Without a doubt, there is some meaningful signal contained in the KEV that makes it useful to any risk-based vulnerability management program. But exactly how should the KEV be incorporated into such a strategy? After all, there is a large number of data sources one could leverage to reduce the risk from the myriad of vulnerabilities found in an enterprise system. In this final section, we'll explore some possibilities and demonstrate that the KEV is one piece of a much larger picture.

How do we determine what constitutes a good remediation strategy? We've got two main yardsticks we can use: coverage and efficiency.

- **Coverage**—Proportion of vulnerabilities with known exploitation activity that are remediated
- **Efficiency**—Proportion of all remediated vulnerabilities with known exploitation activity

Without belaboring the point too much here (this is ground already well-trodden by previous P2Ps), it's worth reviewing why these two concepts are crucial. Primarily, they represent the main tension in the vulnerability remediation space: organizations can't remediate everything, so they must pick and choose. They want to pick as many of the dangerous vulnerabilities as possible (coverage) and not waste too much effort on the non-dangerous stuff (efficiency).

So let's start with the hypothetical "What if you just patched vulnerabilities that show up on some list?" To answer this, we start with a dataset of 56.6K unique CVEs, manifesting as 584.6M open vulnerabilities in all assets across the organizations in our dataset. Then we consult with various data sources and imagine what would happen if we remediated all the vulnerabilities in each data source. For example, what would the coverage and efficiency be if everything scoring higher than the 90th percential according to the Exploit Prediction Scoring System (EPSS) was remediated? Or what about remote code execution? What if the vulnerability shows up within a module in Metasploit? We'll look at a dozen strategies, and Figure 19 plots the coverage and efficiency each remediation strategy would have if we only used that data source.

CISA KEV: Listed in the CISA KEV Catalog

Metasploit: Appear in a MetaSploit module

ExploitDB: Contained in ExploitDB

Github: Identified as exploit code in GitHub from the Cyentia's Exploit intelligence Service

Reversing Labs: Evidence of exploitation from RL

RCE: Exploitable via remote code execution

EPSS Critical: EPSS score in the 98th percentile

EPSS High: EPSS score in the 90th percentile

CVSS Med+: CVSSv3 base score of Medium and above (CVSS 4+)

CVSS High+: CVSSv3 base score of High (CVSS 7+)

CVSS Critical: CVSSv3 base score of Critical (CVSS 9+)

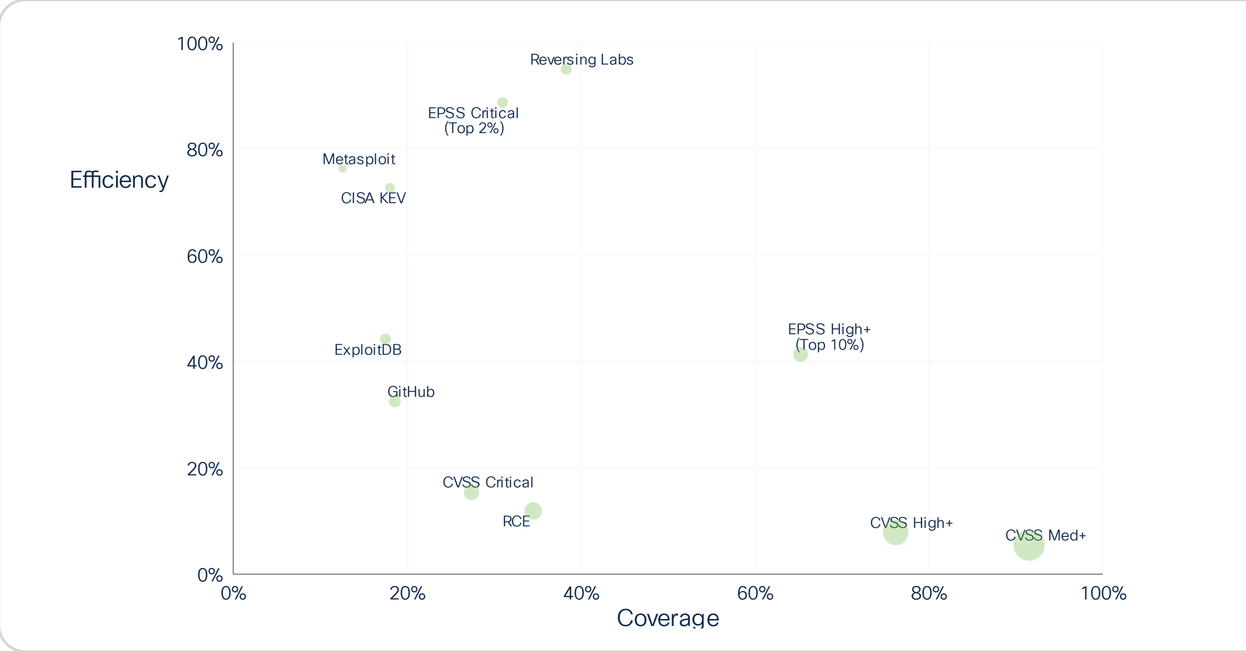

Figure 19. Performance of various remediation strategies based on single data sources (up and to the right is better)

As a general trend, Figure 19 does a good job of showing the trade-off between coverage and efficiency. Patch everything that could be bad (like everything CVSSv3 Medium and above), and you'll achieve great coverage at the expense of a lot of unnecessary work that will tax your security team. Meanwhile, KEV falls somewhere on the other extreme (upper left): if one only patches KEV vulnerabilities, there won't be much wasted effort, but you'll only get about 19% of what's actually being exploited in the wild.[6] We won't touch on all the points in that figure; we'll let the reader pick and ponder their favorite data source.

The single data sources in Figure 19 are a bit of a straw man though. No remediation organization is going to choose one bit of data to predicate their whole remediation strategy on. A reasonable strategy might be to ask, "What logical combination of the above gets me the furthest to the upper right?" There is an exceedingly large[7] number of ways you could combine the data sources to devise a remediation strategy ("If it's in Metasploit or EPSS Critical and CVSS High or RCE… But not in ExploitDB"), so we look at two approaches that represent opposite ends of the spectrum. First, imagine an efficiency-focused organization that is only going to remediate a vulnerabilities if it shows up in **all** of a subset of sources. Second, we imagine an organization focused on coverage that might pick a subset of data sources and remediate a vulnerability if it shows up in **any** of them. We plot these two strategies for all possible subsets of the above data sources in Figure 20.

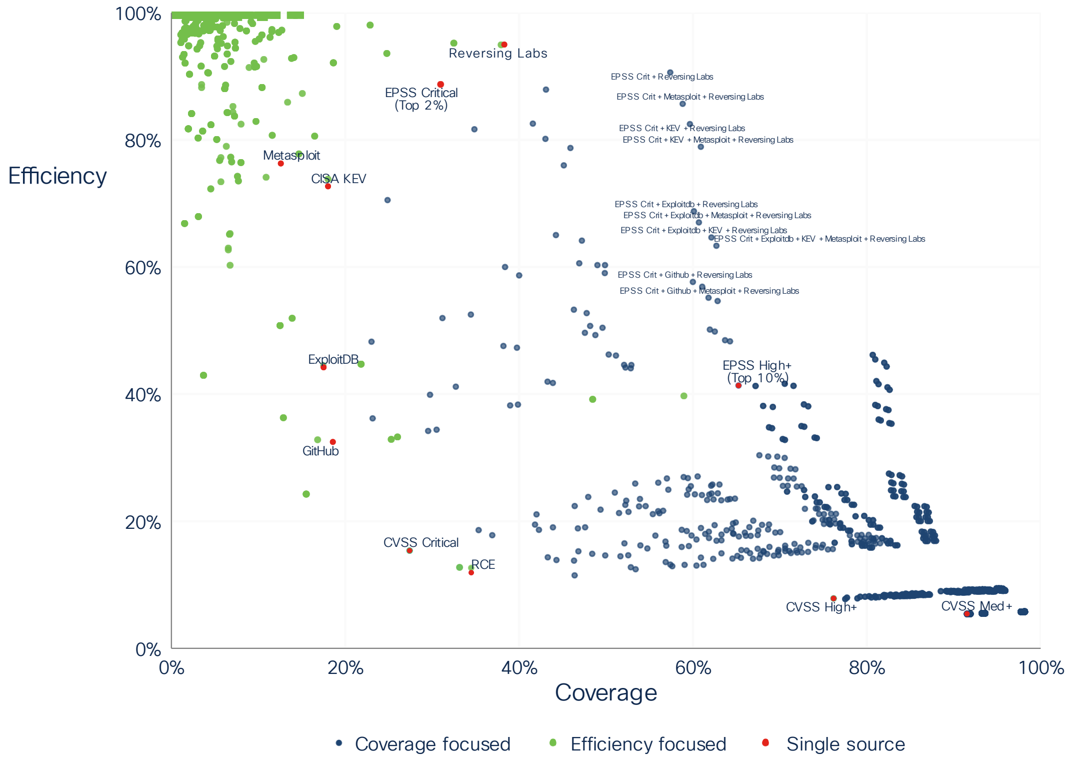

Figure 20. Various combinations of strategies from various data sources

The major takeaway here is that the best strategies select from a wide variety of sources and take action when a vulnerability shows up in any of them (not all). The green points in the upper left are highly efficient but have very little coverage. Little effort is wasted, but there is a lot of potential risk. In contrast, the dark blue points (some combination of sources, and we'll remediate if we see a CVE in any of them) exhibit a much wider spread. It's true that many are down in the lower right, denoting a lot of wasted effort but good coverage; however, a surprising amount have more than 50% coverage and 75% efficiency, which is quite the feat. We label a few of the top performers. In particular, where things that are in the top 98th percentile of EPSS are combined with some Cisco Vulnerability Management information and reversing labs, this seems to be a pretty great strategy. Moderate gains in coverage can be obtained by adding in the KEV and Metasploit but at the cost of some efficiency. We want to offer one caveat before people haul off and just pick one of the combos in the chart: this is across all organizations in our dataset and may not be right for you. Every organization is different; thus the combinations above may not be the best for your organization. You need a strategy and solution to get it right.

Conclusion, recommendations, and reflections

Through our data analysis, we've turned the KEV over in our metaphorical hands and scrutinized it under our metaphorical loupes to attempt to see where it fits in a vulnerability remediation program. The KEV represents a growing list of vulnerabilities that we can corroborate to have exploitation in the wild from other sources. They tend to be more severe than vulnerabilities that aren't on the KEV, and just about every organization has seen one pop up in its assets at least once. Moreover, it mirrors many of the trends we'd see in the larger vulnerability remediation space, particularly where vulnerabilities that are easy to remediate vulnerabilities get remediated fast, while those that are difficult to remediate take a bit more time. This is in spite of the fact that those hard-to-remediate vulnerabilities are often the most important with respect to network hygiene.

The most important takeaway is that it's a useful but incomplete signal for a vulnerability remediation program. While the majority of vulnerabilities on the KEV are additionally seen as "exploited" by other data sources, a large number of exploited vulnerabilities aren't on the KEV. This means that the KEV can't be the be-all and end-all for organizations, even those required to adhere to it, and all possible data sources should be adopted when building out a vulnerability management program.

To that end, there is no data source that acts as a silver bullet with respect to vulnerability remediation. Some are certainly better than others (see EPSS, reversing labs, and the KEV Catalog in Figure 20), but you really need a variety of sources to get an idea of the whole landscape. Furthermore, while the above chart gives us a good indication of the overall shape, it's likely that your individual organization could benefit from some other combination of data sources. Nevertheless, with the need for a variety of sources comes additional complexity, and having a platform to ingest and manage those vulnerabilities is a must. We are getting dangerously close to selling something here, so we'll stop, but you get the point.

Learn more: cisco.com/go/secure

Notes

1. Throughout this report, we'll use the term CVE ID, CVE, and vulnerability interchangeably. For our purposes here, they are the same thing.
2. Bear in mind that "exploited" can mean different things to different people. We try to be consistent in referring to vulnerabilities with known exploit code or PoCs vs. those that are actively being exploited in.
3. See the "How can you know…" callout section for more information.
4. We're not gonna dive into temporal scores, as these represent a little of what we are trying to measure anyway, namely exploitation, but also because they aren't widely available. We'll also ignore the environmental group of CVSS metrics because they are organization specific and not applicable across all organizations.
5. To make this analysis more relevant, we removed orgs with <100 active assets.
6. We made this point once already, but it's worth making again.
7. $2^{2047}-1$ to be exact with is ≈ $1.6*10^{616}$