KENNA Security

CYENTIA INSTITUTE

PRIORITIZATION TO PREDICTION

Volume 2: Getting Real About Remediation

PRIORITIZATION TO PREDICTION

VOLUME 2: GETTING REAL ABOUT REMEDIATION

KENNA Security

This research was commissioned by Kenna Security. Kenna collected and provided the datasets for this reserarch to the Cyentia Institute for independent analysis and drafting of this report.

Kenna Security is a leader in predictive cyber risk. The Kenna Security Platform enables organizations to work cross-functionally to determine and remediate cyber risks. Kenna leverages Cyber Risk Context Technology™ to track and predict real-world exploitations, focusing security teams on what matters most. Headquartered in San Francisco, Kenna counts among its customers many Fortune 100 companies, and serves nearly every major vertical. For more information, visit www.kennasecurity.com.

CYENTIA® INSTITUTE

Analysis for this report was provided by the Cyentia Institute. Cyentia seeks to advance cybersecurity knowledge and practice through data-driven research. We curate knowledge for the community, partner with vendors to create analytical reports like this one, and help enterprises gain insight from their data.

Find out more: www.cyentia.com.

Introduction

"Exploitation prediction models based on external vulnerabilities are useful. But what are organizations actually seeing and doing when it comes to remediating vulnerabilities inside their own environment?"

You may have asked a similar question when seeking to apply findings from the first volume of our Prioritization to Prediction report to the day-to-day battle against vulnerabilities impacting your organization. We certainly did, which is why we immediately began an "outside-in" transition for ongoing analysis of Kenna Security's amazing dataset.

As the name suggests, the first report sought to predict which of the thousands of vulnerabilities published each month were most likely to be exploited, and thus deserving of priority remediation. That effort, though valuable, was largely theoretical. It dealt with CVEs in the abstract and modeled potential outcomes achieved by various remediation strategies. In this new report, we seek to apply and test those theories in the real world using data extracted from hundreds of production environments inundated with billions of vulnerabilities. As you can imagine, that dataset holds many valuable lessons for vulnerability management programs looking to efficiently reduce risk. You'll find a preview of what we learned on the next page, but don't stop there. The devil, as they say, is in the details, and there's plenty of those you won't want to miss in the pages that follow!

A few of the questions we seek to answer in this report:

- What proportion of vulnerabilities are observed and open across 500+ organizations and 3+ billion assets?
- How comprehensive and efficient are organizational vulnerability remediation practices in reality?
- How long does it take to remediate vulnerabilities across the network? Does time-to-remediate differ among firms?

Some key terms used in this report

We've made an effort to use conventional terminology in this report, but some things we discuss don't have a clear or universally accepted term. In such cases, we chose a term that seemed appropriate and endeavored to use it consistently. This page should help keep us honest and avoid misinterpretation.

Exploited vulnerabilities:

In the last report, we made a distinction between an *exploit* (proof-of-concept or working code for exploiting a vulnerability) and *exploitation* (attacks targeting a vulnerability in the wild). We generally deal with those events in an either-or manner in this report, and decided to simply refer to such vulnerabilities as "exploited."

Observed vulnerabilities:

When we say a vulnerability was "observed" in this report, we mean that at least one instance of that vulnerability was detected by a scanner, discovered by a penetration test, or otherwise actually seen in an asset managed by an organization.

Open/Closed vulnerabilities:

Vulnerabilities observed in an environment are in an open or closed state. Closed means the vulnerability has been patched, fixed, or otherwise remediated. Open means it has not been closed and thus exposes the affected asset(s) to any exploits that exist now or in the future.

Key Findings

- About one-third of published CVEs are actually observed in live enterprise environments. That reduces the problem space, but...
- Only 5% of all CVEs are both observed within organizations AND known to be exploited. These are the ones that really matter!
- Some CVEs (about 3%) were observed across a million or more assets each. Kinda puts the trite "just patch it, dummy" advice in perspective, doesn't it?
- Does it surprise you to know that 40% of vulnerabilities observed in enterprise networks are still open (unremediated) today? Maybe this will: over 75% remain open a year after the CVE is published.
- Perhaps you're thinking "vulns left open must be low severity." Good thought, but it's wrong. Roughly one-third to one-half of observed vulnerabilities remain open for any given CVSS score.
- A mere three vendors are associated with more than two out of every three open vulnerabilities. Which ones? C'mon, it's an intro—RTFM (FM being "full manuscript," of course)!
- 15.6% of all open vulnerabilities observed across organizational assets in our sample have known exploits. That represents a sizable attack surface exposed to a wide range of adversaries.
- Organizations work hard to shrink this attack surface. Overall, they remediate 70% of higher-risk vulnerabilities. But the remediation efficiency rating is just 16%, indicating firms opt to fix many low-risk issues that could have been ignored or delayed.
- Coverage and efficiency vary greatly among firms—over 50% between top and bottom performers—indicating different remediation strategies lead to very different outcomes.
- We also looked at remediation timeframes. The median time to remediation for vulnerabilities is 90 days. Only about one-third of vulnerabilities is remediated in the first 30 days and the final one-third remain open after 180 days.
- All of the above suggests vulnerability management programs really do matter and a real, measurable improvement can be gained by making smarter remediation decisions.

Key Findings from Volume 1

23% of published vulnerabilities have associated exploit code.

2% of published vulnerabilities have observed exploits in the wild.

The chance of a vulnerability being exploited in the wild is 7X higher when exploit code exists.

50% of exploits publish within two weeks surrounding new vulnerabilities.

The volume of exploitation detections jumps five-fold upon release of exploit code.

Remediation strategies can be measured in terms of coverage (recall) and efficiency (precision).

Remediating vulnerabilities with CVSS 7+ achieves efficiency of 32% and coverage of 53%.

Remediating vulnerabilities for the top 20 vendors achieves efficiency of 12% and coverage of 22%.

The proposed prediction model performs 2X and 8X more efficiently (respectively) than the approaches above with equivalent coverage.

Revisiting the Case for Prioritization

If you recall, the first installment of our *Prioritization to Prediction* report began building the case for prioritization (and ultimately prediction) of vulnerabilities by establishing some basic facts about published and exploited CVEs. First, there's a lot of them—over 110,000 published—far more than any organization could realistically handle. Second, just over one-fifth of those CVEs have known exploit code and even fewer are known to have been exploited in the wild (~2%). This led us to the conclusion that successful vulnerability management requires a delicate balance of identifying and fixing what matters (more likely to be exploited) while not wasting resources on what matters less (not likely to be exploited). The numbers are not in favor of maintaining that balance, which is why we shifted focus to developing a prediction model fit to that purpose.

Some Critical Observations

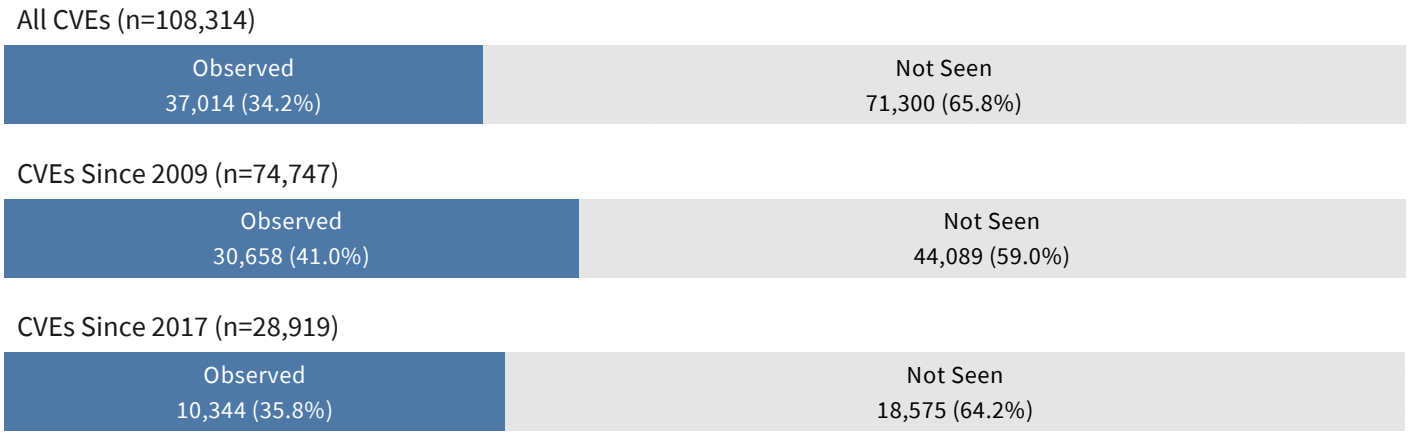
Let's revisit the case for prioritization we began building in the first report. Our evidence supporting this investigation is strong. It consists of more than 3 billion vulnerabilities managed across 500+ organizations and 55 sources of external intelligence. We will begin by comparing the picture we painted last time of "all externally published vulnerabilities" to a new view of "all internally observed vulnerabilities" affecting organizational assets. If we take a moment in time (late 2018) and look across organizations using Kenna to manage and prioritize vulnerabilities in their environment, we can start to see what's really going on.

First and foremost, only about one-third of all published CVEs are actually observed in live organizational environments. By "observed" here, we mean that at least one instance of that CVE was detected by a vulnerability scanner, discovered by a penetration test, or otherwise actually seen in an asset managed by a particular organization. In other words, it's reality rather than just theory.

This proportion of observed CVEs varies somewhat depending on how we change the aperture of scope. Looking across all time, we see that one-third statistic; 37k out of 108k of CVEs (34%) were observed by at least one organization. Narrowing to the last 10 years of published CVEs pushes that ratio up a bit to just over 40%. When we close in on just CVEs published since 2017, we find 36% of them observed within organizations. Figure 1 gives the details behind observed/published ratios for these different time slices.

FIGURE 1

Count and proportion of vulnerabilities observed in live organizational environments



Source: Kenna / Cyentia

Similar to the exploitation ratio from our first report, these statistics on observed vulnerabilities reinforce the need to prioritize what matters. Vulnerabilities observed in our own environment are surely more critical for remediation than those merely recorded in a public database. But why does the gap between observed and published CVEs exist at all? There are two obvious possibilities: 1) scanners can't detect them and 2) they simply aren't there. As with many things, the truth lies somewhere in the middle.

It's no secret or surprise that variability exists among scanners in terms of the vulnerabilities they detect. After all, keeping up with the ~300 new CVEs per week we saw in 2018 is no small task for tool developers. But because Kenna is scanner-agnostic across hundreds of customers, these results don't stand or fall on the capabilities of any one scanning vendor or platform. Ostensibly, we have strong representative coverage of vulnerabilities observable by scanners in the market.

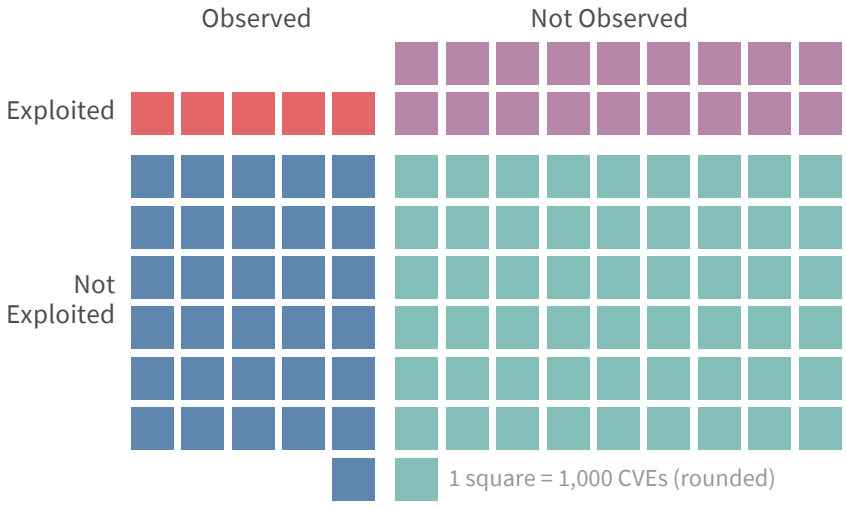
However, that does not imply the gap between observed and published CVEs has nothing to do with the tools we use to identify them. Most Android vulnerabilities, for example, were not observed, suggesting that scanners struggle to detect them, organizations aren't scanning for them, or, more likely, a combination of both and more. Regardless, it's safe to say the estimates of observed vulnerabilities presented in Figure 1 are on the lower end of reality.

Scanner coverage notwithstanding, the Occam's Razor explanation for the results in Figure 1 is that the majority of CVEs simply aren't there to be found. The affected asset(s) may not exist on the network. Maybe the vulnerabilities pertain to older software or hardware. Perhaps they only affect obscure or rarely used versions. Whatever the explanation, it leaves a large gap between all published CVEs and those that represent a real and present danger in your environment.

On Observations and Exploitations

Let's pull a little more on that "real and present danger" thread. Our "Prioritization to Prediction" report found the chance of a vulnerability being exploited in the wild is 7X higher when exploit code exists. So there's a "real" danger from exploited CVEs and we should track them. Now let's overlay what we know about exploitation with what we learned above about vulnerabilities actually observed ("present") in enterprise environments. Figure 2 offers a nifty way of viewing these interrelated categorizations.

FIGURE 2
Ratio of observed/not observed and exploited/not exploited for all published CVEs



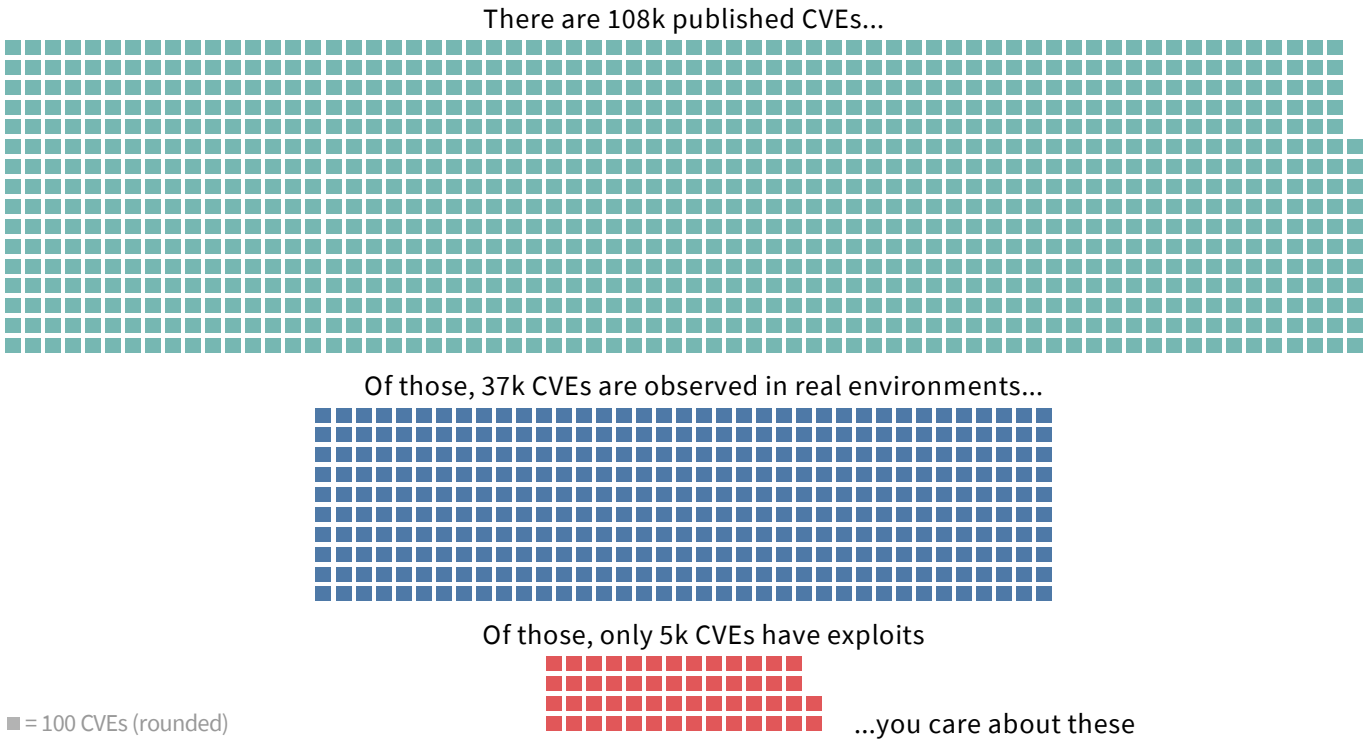
Source: Kenna / Cyentia

Those red squares represent vulns you really don't want in your network!

Does more recent history show a similar trend? To test that, we can break out the same distinctions using only CVEs published in the last two years (since 2017-01-01). While the proportions shift around a little (3% observed and exploited; 64% not observed and not exploited), the changes don't alter the message or warrant a second chart. Instead, we'll close this section with another view of the vulnerability priority funnel.

FIGURE 3

Proportion of all published CVEs that have been observed and exploited



Source: Kenna / Cyentia

Take note of the two extremes in Figure 2. Half of CVEs are neither exploited nor observed, while just 5% are both exploited and observed. Think about that for a moment—of the 108,000+ CVEs published, only 5% represent a "real and present" danger to most organizations. Granted, "only" may not be the best choice of words here since that still leaves us with 5,000+ vulnerabilities to track and remediate across myriad assets. But it's a far more manageable decision space than the one we started with for all published CVEs.

CHALLENGES TO VULNERABILITY DATA COLLECTION

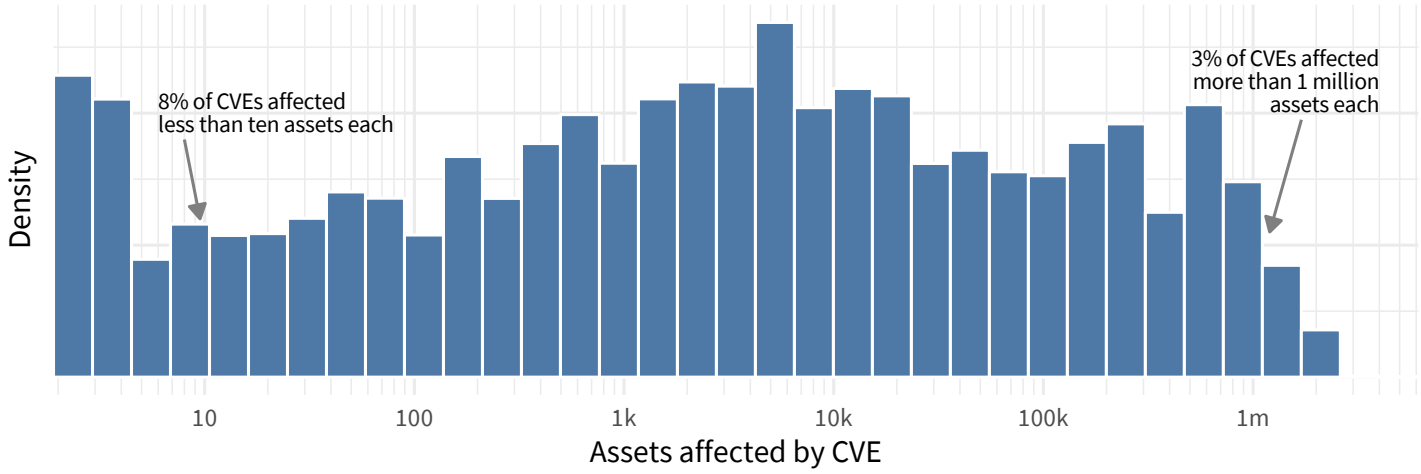
We'd be remiss if we didn't acknowledge and discuss some of the issues involved in studying vulnerabilities and exploits. One of the challenges with observing exploitation "in the wild" is that a detection signature has to be written and implemented across sensors. In some cases, such as SQL injection or XSS, the signatures and detection should work independent of the specific underlying vulnerability. But some CVEs are unique enough to require their own signature. Generally speaking, it's easier to create a signature using exploit code as a template, so detection of exploitation is often dependent on published exploits. Also, how the detection is done matters a lot. Using network-based detection mechanisms, we may never see host-specific exploits. Additionally, detecting the exploitation of something like CVE-2017-14937 (the detonation of passenger airbags in some cars), opens up a whole new realm of challenges. But think of these numbers as a snapshot in time pieced together from multiple perspectives. We may not have a full panoramic vista in fine detail, but we have enough to examine and improve upon the status quo as the practice of vulnerability management improves.

Open Talk on Vulnerabilties

The above figures and statements are true to the data, but they don't reveal the whole truth. We've been treating vulnerabilities as singular things, but in reality one CVE could be present across numerous assets in an organization, each one as exploitable as the next. So the decision is never really as simple as "do I fix this one vulnerability or not?" It gets much more complex at scale. Figure 4 gives a sense of that scale, tallying the number of instances of each CVE observed within live enterprise environments.

FIGURE 4

Distribution of the total number of assets affected by CVEs



Source: Kenna / Cyentia

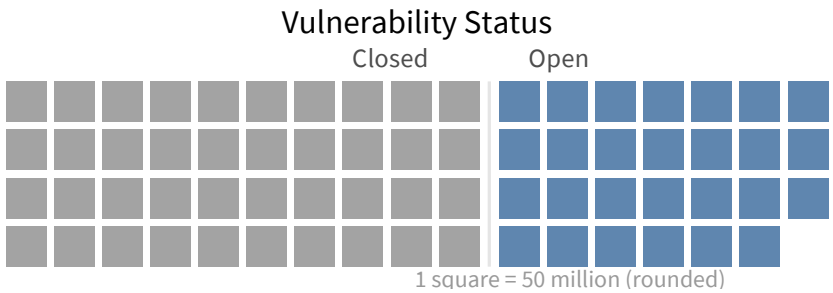
Just in case it's not clear, Figure 4 is a breakout of the shaded portion of Figure 1 (observed CVEs), enumerated by the number of assets exhibiting that vulnerability. It's easy to see that some vulnerabilities affect only a few systems, while others impact millions. That has a direct bearing on the difficulty/cost of remediation and the overall risk assessment process.

Figure 4 gets us closer to the truth, but we're still not quite there yet. Not all of these observed vulnerabilities (or all instances of them across all assets) will be open at any given time. Some will have been closed (patched or addressed in some way), and so get scoped out of current and future remediation decisions.

As those in vulnerability management programs are well aware, the number of open and closed vulnerabilities is constantly fluctuating across a complex and dynamic environment. New ones pop up, affect numerous systems, get remediated, and the process starts again somewhere else (and often in parallel). But if we just take a macro view of open to closed vulnerabilities over the last decade, we see the ratio depicted in Figure 5.

FIGURE 5

Overall ratio of open to closed vulnerabilities observed by organizations



Source: Kenna / Cyentia

> Open vulnerabilities may be overlooked, deprioritized, or couldn't be remediated.

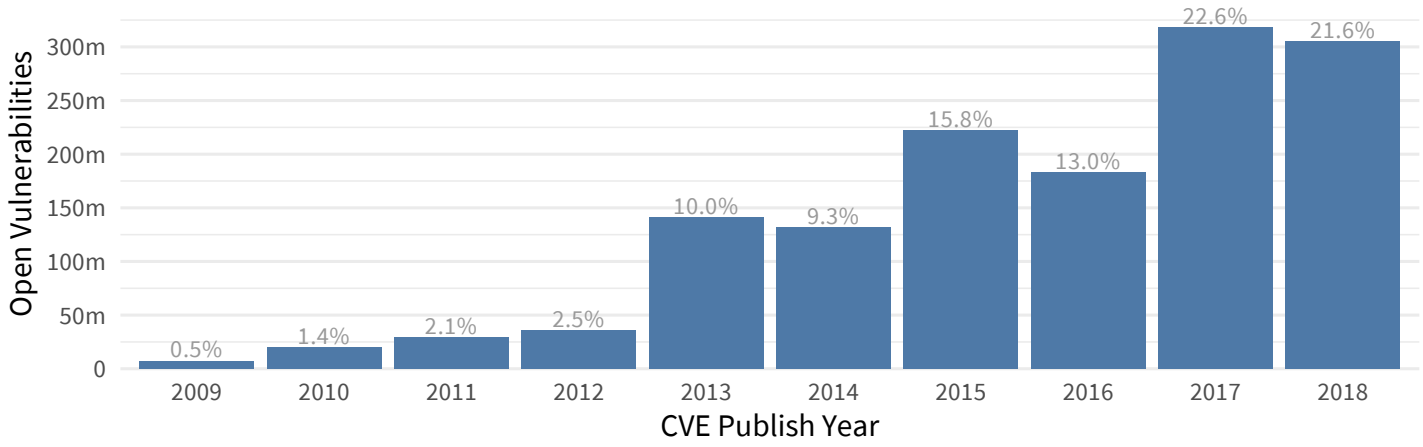
There's nothing visually exciting about this chart, but the lack of dramatic contrast is actually rather interesting. Does it surprise you to know that 40% of CVEs observed in enterprise networks are still open? Even more interesting is how this ratio of open to closed vulnerabilities changes over time, by severity, and by vendor. Let's go there now.

The Age of Open Vulnerabilities

Figure 6 offers a historical perspective covering millions of open vulnerabilities for CVEs published over the last decade. It reveals that about 22% (or 300+ million) of all open vulnerabilities observed by organizations in our dataset were associated with CVEs published in 2018. Not directly observable from the figure (unless you do the math) is that over 75% remain open at least one year after the associated CVE was published.

FIGURE 6

Count and proportion of open vulnerabilities by year



Source: Kenna / Cyentia

Depending on your perspective and experience, that may not seem like such a bad track record for closing vulnerabilities. But the other end of the chart does not inspire much hope—over 40% of all open vulnerabilities are tied to a CVE published in 2015 or earlier. But is time the only factor at work here? Is there a steady rate of decay for vulnerabilities in enterprise environments that we simply need to wait out? Obviously, there's more to this story than that.

The Severity of Open Vulnerabilities

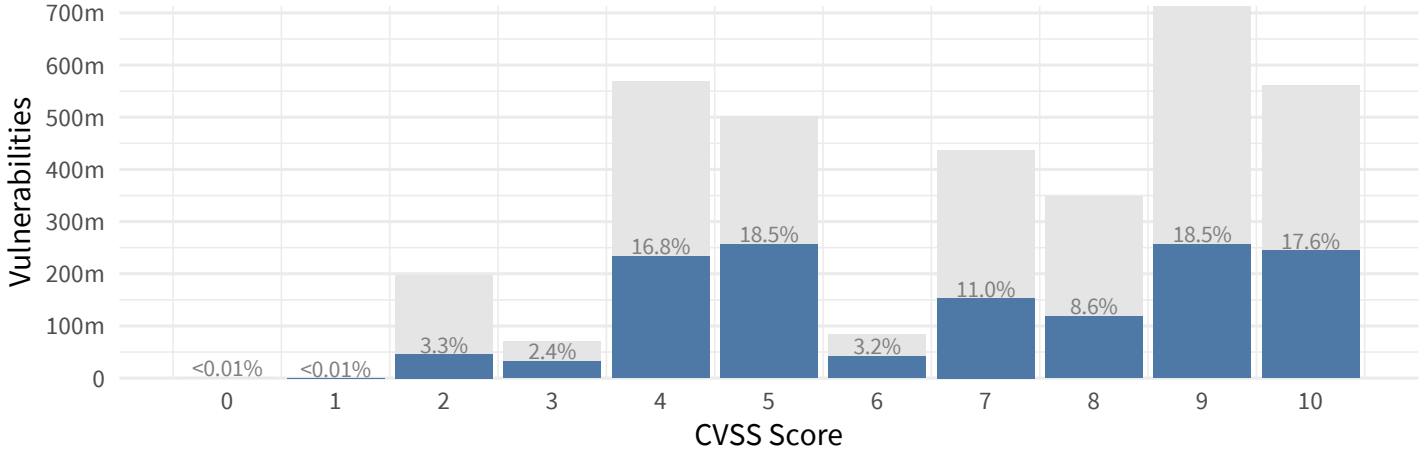
One possible version of this story is that organizations close higher-severity vulnerabilities with more urgency. Viewing observed vulnerabilities by CVSS score should offer evidence to either corroborate or refute this explanation. Figure 7 serves up that view and also offers a comparison of the open/closed ratio for each level of CVSS.

Ignore the gray portion of the bars in Figure 7 for the moment. The colored portions follow a format similar to Figure 6 and depict the proportion of open vulnerabilities corresponding to each CVSS score. From that, we see that ~18% of all open vulnerabilities observed by organizations in our sample earned the maximum CVSS score of 10. Another 17% scored a 9, and so on. It's evident that a very large number of higher-severity vulnerabilities exist in organizations. This probably has less to do with remediation strategies and more to do with the inherent properties of CVEs and CVSS scoring metrics (some scores are simply more common). Even so, it would be hard to back up the "higher severity = higher urgency" story from these results.

Now let's move on to the full height of the bars, which total all observed vulnerabilities across all assets for all severity levels. From that, we can see the ratio of open (colored segment) to closed (gray segment) vulnerabilities relative to each CVSS score. These proportions aren't labeled, but the main takeaway is fairly easy to suss out. Roughly one-third to one-half of observed vulnerabilities are open for any given score. Thus, we cannot conclude that firms are stamping out all instances of CVSS 10s and then progressively iterating through lower and lower severity levels until they're all closed.

FIGURE 7

Count and proportion of open vulnerabilities by CVSS score



Source: Kenna / Cyentia

The Sources of Open Vulnerabilities

If you're anything like us, you've been looking at the figures above wondering why millions of vulnerabilities are open regardless of age, severity, etc. The real answer is "it's complicated." But that's probably not very satisfying. We thought a better answer might be found from the source of these vulnerabilities—the vendors that introduced them in the first place.

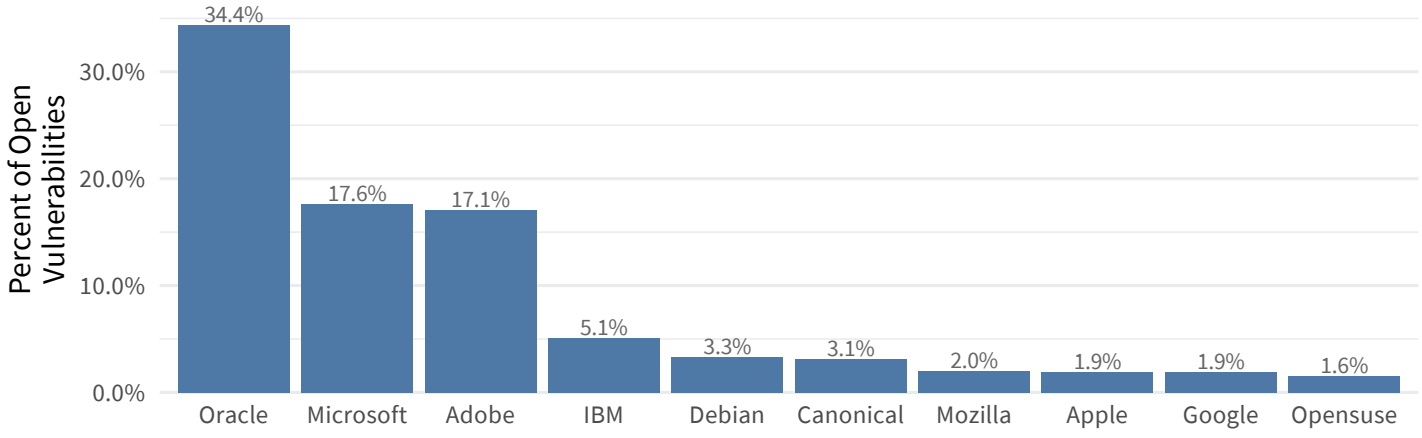
Using the Common Platform Enumeration (CPE) supplied by NVD, we can identify vendors associated with CVEs. This is not a one-to-one relationship because some CVEs identify multiple vendors and vice versa, but it allows us to construct a decent record of how organizations are remediating vulnerabilities from different vendors and products.

The Big Three

Let's just get these three out there: Oracle, Microsoft, and Adobe. Together, they are responsible for seven out of every 10 open vulnerabilities (69.1%) observed by customers. But this shouldn't be surprising given the dominating footprint they have across the modern enterprise technology landscape. So let's reserve any judgment against them and instead examine the data.

FIGURE 8

Proportion of open vulnerabilities by associated vendor



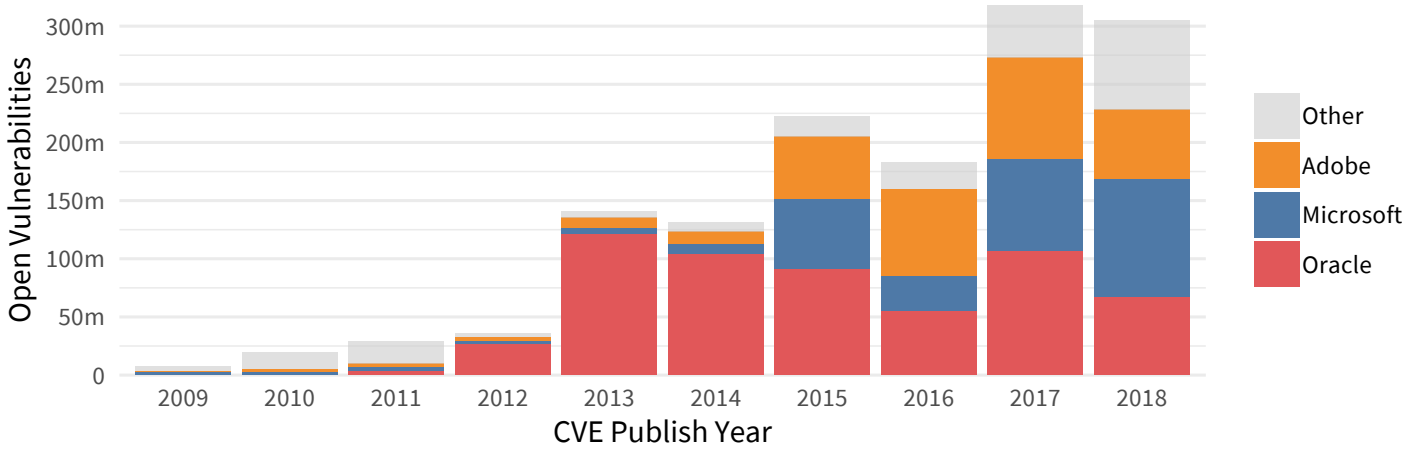
Source: Kenna / Cyentia

Full disclosure: We got a little tricky with the data in Figure 8. We merged Sun into Oracle (2010 acquisition) and Redhat into IBM (acquisition announced 2018), which strengthened Oracle's #1 ranking and pulled IBM into the #4 spot. We also had to deal with the challenge of attribution, since the CPE data doesn't consistently distinguish the responsible vendor from the affected product(s). So Vendor A may be associated with a CVE affecting Products A1 and A2, but Vendor B may be the source of and responsible for fixing the bug (e.g., An Adobe Acrobat bug that affects multiple versions of Windows). But such issues are not extensive enough to invalidate the message of Figure 8.

OK; so a large majority of more than a billion open vulnerabilities can be attributed to Oracle, Microsoft, and Adobe. How does that look over time? Figure 9 gives the answer, and we recommend taking a moment to ruminate on it.

FIGURE 9

Count of open vulnerabilities from Adobe, Microsoft, and Oracle over time



Source: Kenna / Cyentia

Notice, in particular, the difference between Oracle and Microsoft. Microsoft eats the largest slice of the vulnerability pie in 2018, but has only a tiny sliver before 2015. It's hard to see anything other than Oracle among CVEs from 2012-2014, but that predominance lessens over time. Adobe seems to borrow a page from both, expanding and then contracting over the last year. Finally, it's worth calling out that the proportion of vulnerabilities from "Other" vendors is significantly greater for both older (pre-2012) and newer (post-2017) CVEs. One explanation for this is fueled by the recent growth in Certified Numbering Authorities (CNAs), which facilitates a wider range of CVEs from a more diverse set of vendors.

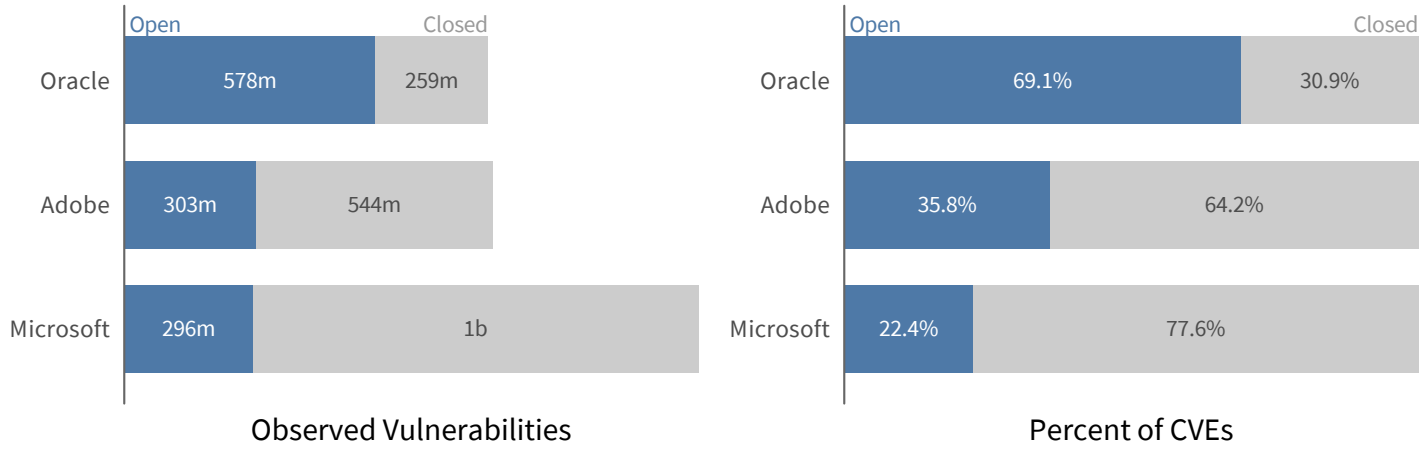
All of this begs the question of who or what is responsible for what we see in Figure 9. Is it the vendors who develop patches for vulnerabilities affecting their products? Is it the patches themselves—perhaps some are easier to deploy than others? Is it the organizations that deploy those patches or other remediations? "All of the above" is the easy answer, but it's probably the most correct one.

Figure 10 offers another view of the Big Three by comparing the ratio of open to closed vulnerabilities for each vendor. Microsoft is up first and should be easy to spot because it has the longest bar on the left side of the chart. Despite having the highest number of observed vulnerabilities, Microsoft has the lowest proportion of open vulnerabilities of the Big Three (22%). It's not a stretch to infer that organizations find it easier to patch or remediate Microsoft products, perhaps due partially to their ubiquity and partially to initiatives like Patch Tuesday.

Unfortunately for Oracle, organizations appear to struggle when it comes to remediating vulnerabilities in their products. A whopping 66% of Oracle vulnerabilities remain open, according to our data. Combine that statistic with the historical view from Figure 9, and it's apparent that not only are Oracle vulnerabilities not getting closed, they are getting older too. To be fair, Oracle inherited some...shall we say "challenges" when it took responsibility for maintaining the Java platform via the acquisition of Sun Microsystems. And they've been more or less steadily whittling down that stack of challenges ever since.

FIGURE 10

Ratio of open to closed vulnerabilities from Adobe, Microsoft, and Oracle



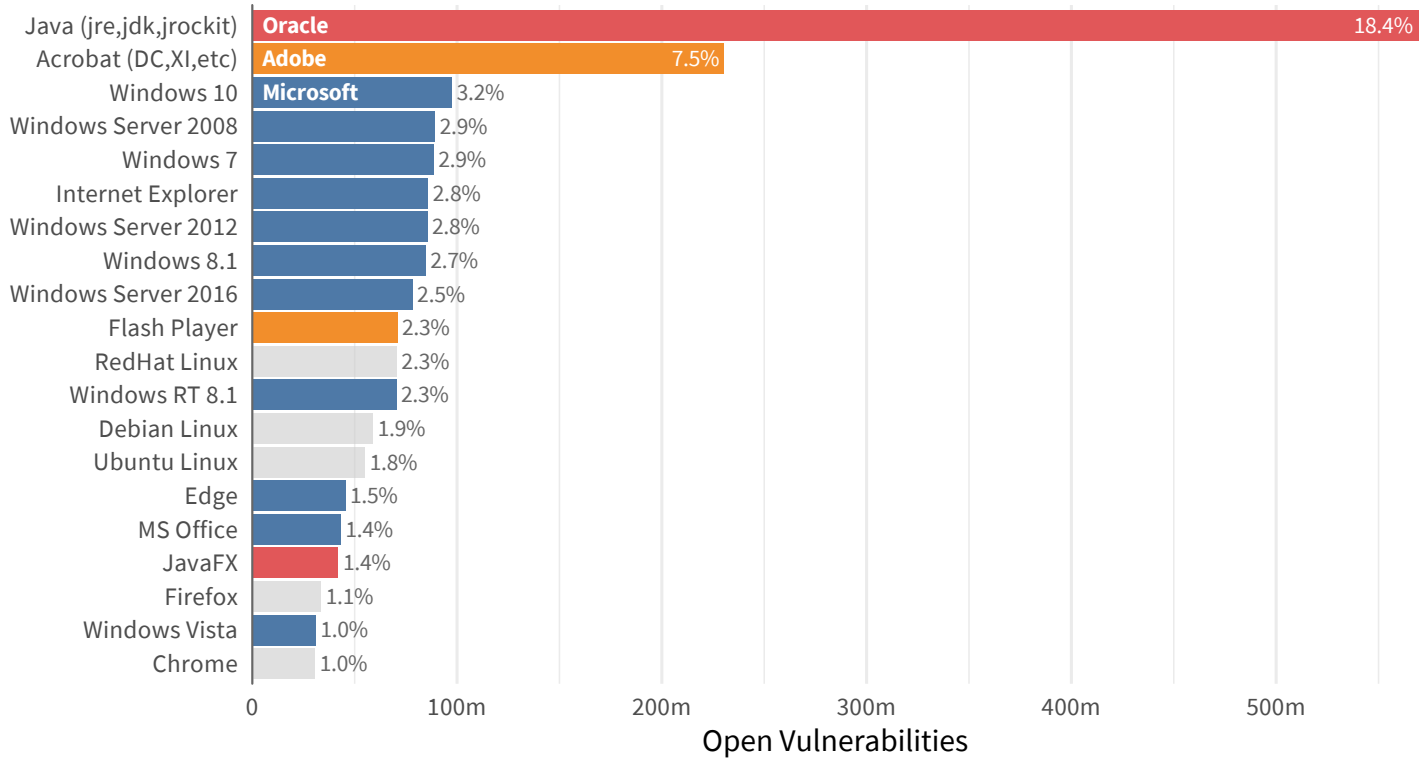
Source: Kenna / Cyentia

The Other Big Three

Speaking of "stack of challenges," Figure 11 ranks products according to the number of open vulnerabilities associated with each. There are three product families sitting squarely on top: Oracle's Java, Adobe's Acrobat, and everything else. Yes, Figure 11 shows a lot more than three products, but you get the point. It's rather top heavy. Beyond that, perhaps the most interesting takeaway is the lack of fuel for the "my OS is securer than yours" fire. Versions of Windows and Linux are both well represented and generally vary within a single percentage point. If anything, it might be seen as a feather in Microsoft's cap to achieve a similar open to close ratio given its much higher count of total vulnerabilities.

FIGURE 11

Proportion of open vulnerabilities by associated products, with top three vendors color-coded



Source: Kenna / Cyentia

Finding the Balance: Coverage and Efficiency

Deciding which vulnerabilities to remediate is a daunting task. In a perfect world, all vulnerabilities would be remediated as they were discovered. But unfortunately, that doesn't happen in the real world. With thousands of new vulnerabilities every year multiplied across disparate assets, reality necessitates prioritization. It comes down to choosing a subset of vulnerabilities to focus on first. But how can we measure the quality of prioritization?

There are many different measurement techniques for a binary decision (to remediate or not to remediate). It's tempting to go for overall accuracy—proportion decided correctly—but this can be a little misleading when so many vulnerabilities are never exploited. For example, a company choosing not to remediate anything will be "right" around 77% of the time. It might look like a good strategy on paper, but not so much in practice. Instead of decision model accuracy, we will focus on the two concepts of **COVERAGE** and **EFFICIENCY**.

COVERAGE MEASURES THE COMPLETENESS OF REMEDIATION. Of all vulnerabilities that should be remediated, what percentage was correctly identified for remediation? For example, if 100 vulnerabilities have existing exploits, and yet only 15 of those are remediated, the coverage of this prioritization strategy is 15%. The other 85% represents unremediated risk. Technically, coverage is the true positives divided by the sum of the true positives and false negatives.

EFFICIENCY MEASURES THE PRECISION OF REMEDIATION. Of all vulnerabilities identified for remediation, what percentage should have been remediated? For example, if we remediate 100 vulnerabilities, yet only 15 of those are ever exploited, the efficiency of this prioritization strategy is 15%. The other 85% represents resources that may have been more productive elsewhere. Technically, efficiency is the true positives divided by the sum of the true positives and false positives.

Ideally, we'd love a remediation strategy that achieves 100% coverage and 100% efficiency. But in reality, a direct trade-off exists between the two. A strategy that prioritizes only the "really bad" CVEs for remediation (i.e., CVSS 10) may have a good efficiency rating, but this comes at the cost of much lower coverage (many exploited vulnerabilities have a CVSS score of less than 10). Conversely, we could improve coverage by remediating CVSS 6 and above, but efficiency would drop due to chasing down CVEs that were never exploited.

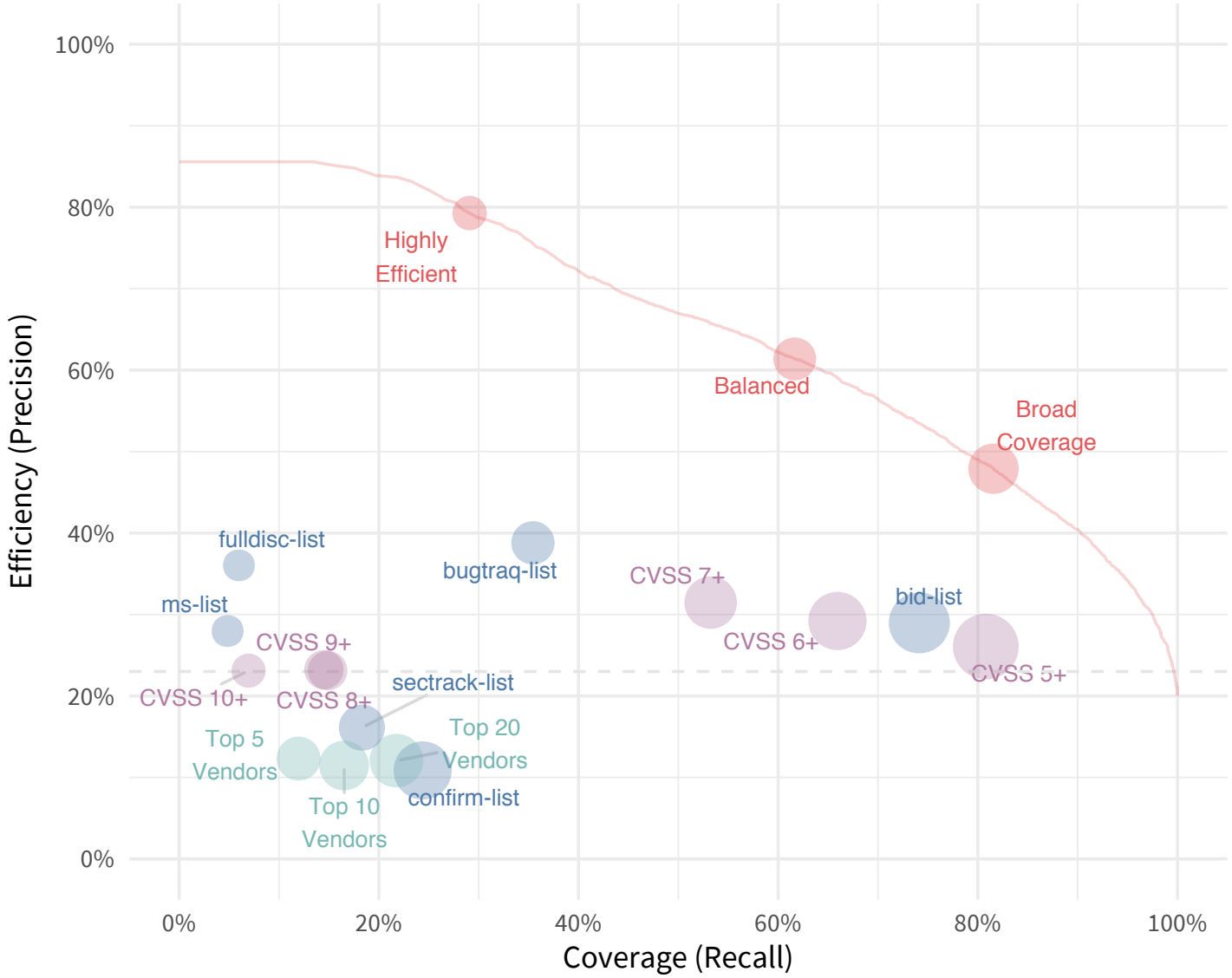
Realized Coverage & Efficiency

So, now we know a lot more about the scale, severity, and sources of open vulnerabilities in enterprise environments. But where does that leave us with respect to our goal of "getting real" about improving the status quo? We're headed there forthwith. But first, it will help to revisit the concepts of coverage and efficiency from our last report.

If the first *Prioritization to Prediction* report emphasized any one thing, it was that successful vulnerability management must balance the two opposing goals of coverage (fix everything that matters) and efficiency (delay/deprioritize what doesn't matter). The powerful thing about those goals is that they can be objectively measured and compared among different remediation strategies. Figure 12 is a flashback to the those results from the last report.

FIGURE 12

Coverage/Efficiency plot for various prioritization strategies and prediction model thresholds



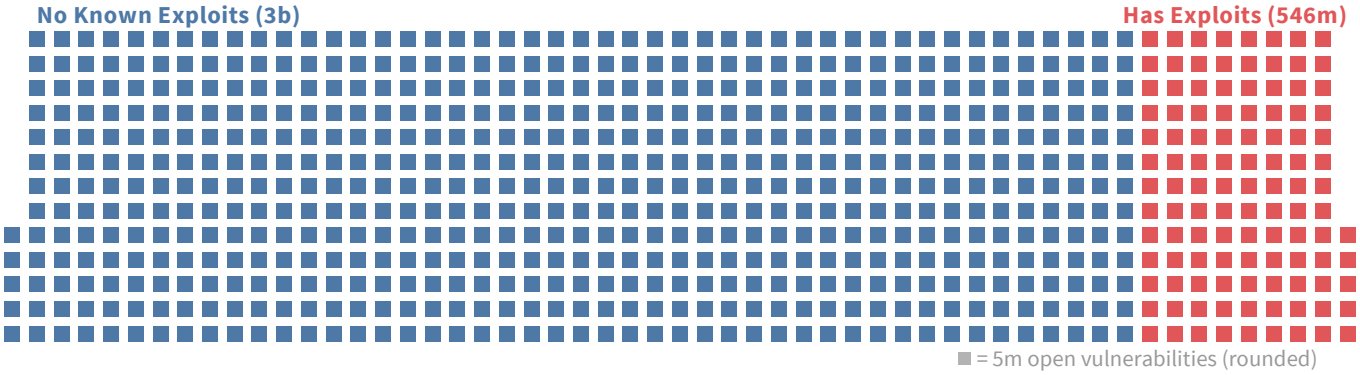
Source: Kenna / Cyentia

As a quick review, the points represent various remediation strategies and the size of those points corresponds to the total number of CVEs remediated by each. Those farther to the right achieve better coverage, while those higher up exhibit greater efficiency. No strategy nears perfect coverage and efficiency (hence the need for balance), but it is apparent that some demonstrably outperform others. Compared to a strategy of remediating all CVEs with CVSS score of 7 or more, the prediction model we developed in the last report achieved twice the efficiency (61% vs. 31%), half the effort (19K vs. 37K CVEs), one-third the false positives (7K vs. 25K CVEs), and a better level of coverage (62% vs. 53%). Pretty powerful support for remediation decisions, right?

But the approach shown in Figure 12 gets far more powerful when we incorporate the data we've been examining on observed and open vulnerabilities. We can now measure how organizations are actually doing in terms of efficiency and coverage metrics in their own environments. In other words, we're now ready to "get real" about improving vulnerability management. Let's start at a high level and talk about where we'd expect organizations to fall on the efficiency-coverage scale.

FIGURE 13

Overall ratio of open to closed vulnerabilities observed by organizations



Source: Kenna / Cyentia

Figure 13 distills billions of data points into a simple chart showing that 15.6% of all open vulnerabilities observed by organizations in our sample have known exploits. This means that if anyone were to randomly select vulnerabilities to prioritize for remediation, they could expect to be "right" about 16% of the time (15.6% efficiency rating). Coverage would start at zero and go up to 100% once they remediated everything (which never happens). But how are firms performing in reality? If we consider the "No Known Exploit" segment in Figure 13 as vulnerabilities we can safely ignore/delay and the "Has Exploits" segment as those we should prioritize, we can create the "Confusion Matrix" of organizational remediation efforts captured in Figure 14.

FIGURE 14

Confusion matrix of organizational remediation efforts

	Should Remediate	Delay Remediation
Did Remediate	381m (TP)	1.7b (FP)
Did Delay	163m (FN)	1.2b (TN)

Source: Kenna / Cyentia

Figure 14 enables us to calculate the overall coverage and efficiency across the entire dataset of observed vulnerabilities in live environments:

Coverage is 70%:

$$TP / (TP + FN)) = 381m / 544m = 70\%$$

Efficiency is 18.5%:

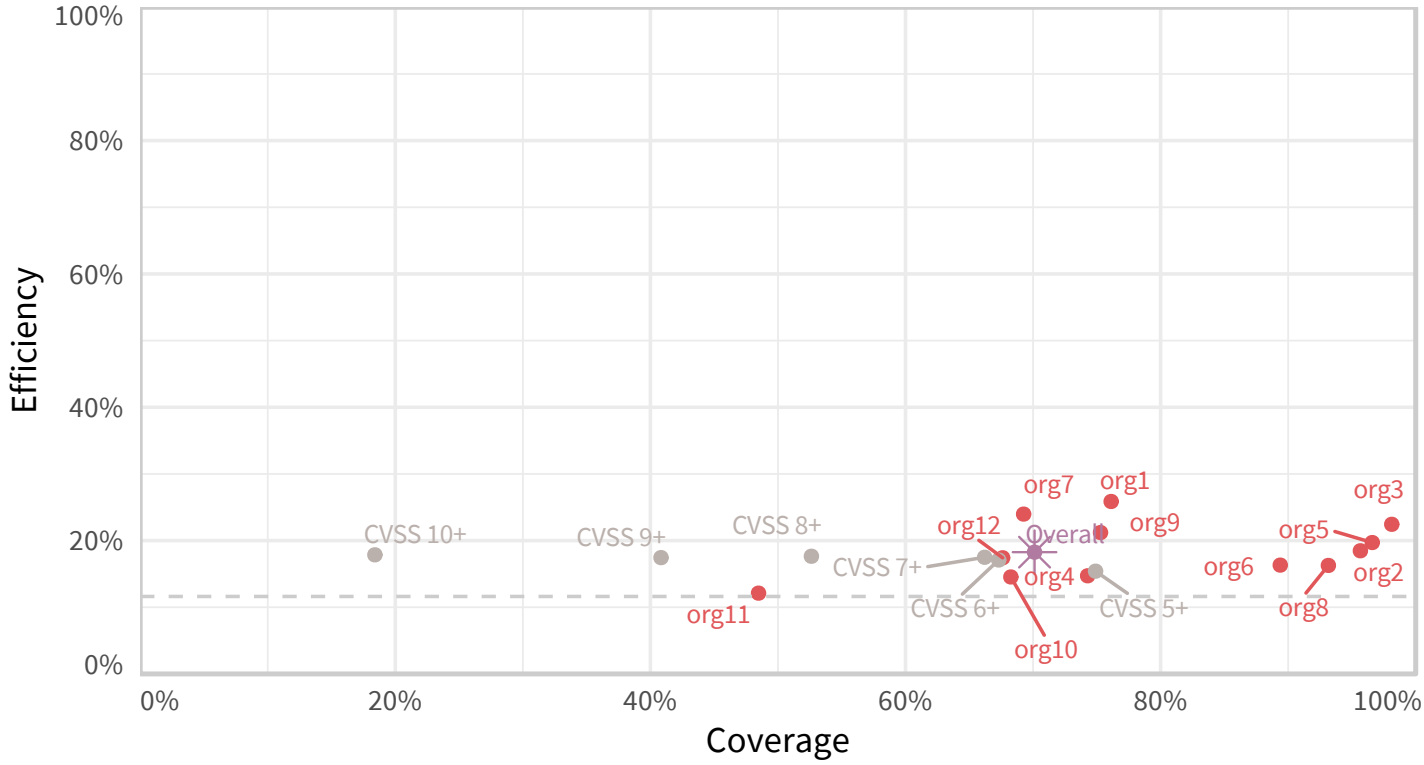
$$TP / (TP + FP) = 381m / 2.1b = 18.3\%$$

Figure 14 enables us to calculate the overall coverage and efficiency across the entire dataset of observed vulnerabilities in live environments. This is an interesting finding that reveals something about the prevailing vulnerability remediation strategy in practice today. Most organizations appear to sacrifice efficiency for the sake of broader coverage. Said differently, firms are addressing the majority of higher-risk issues (70% coverage), but paying a relatively high cost to achieve that goal. This is classic risk-averse behavior, and undoubtedly driven by the comparatively higher cost of getting it wrong and suffering a breach.

To make this even more real and relevant, we picked 12 organizations to use as concrete examples for this type of analysis. Together, they observed 190 million vulnerabilities across their networks, of which 73% (139m) had been closed at the time of this study. These organizations were selected from a range of sizes and industries to highlight variation in remediation strategies and outcomes. We applied the calculations outlined above to each organization and plotted them on the coverage-efficiency grid in Figure 15. We also included the outcome of several CVSS-based patching strategies for comparison.

FIGURE 15

Coverage/Efficiency plot for our 12 example organizations (plus estimates of CVSS-based scores for comparison)



Source: Kenna / Cyentia

You can think of Figure 15 as the "reality" to Figure 12's "theory" and there are some notable differences. Given our previous research, it shouldn't be terribly surprising that CVSS did not fare well as a prioritization strategy in the real world. Just keying on CVSS 10+ would address about a fifth of the vulnerabilities that should be fixed and four-fifths of that effort would be spent on vulnerabilities that represent lower risk and thus could/should have been delayed.

Turning attention to the 12 organizations, it's great to see several of them getting up near 100% on the coverage axis. That indicates they're doing a good job staying on top of the high-priority vulnerabilities. There's another cluster around the 75% coverage mark, which still isn't bad, especially when you consider where many possible strategies fell in Figure 12. One firm falls below the 50% line, meaning they've remediated less than half of vulnerabilities with known exploits. This spread of coverage levels is interesting, but the burning question from Figure 15 is why is efficiency so low?

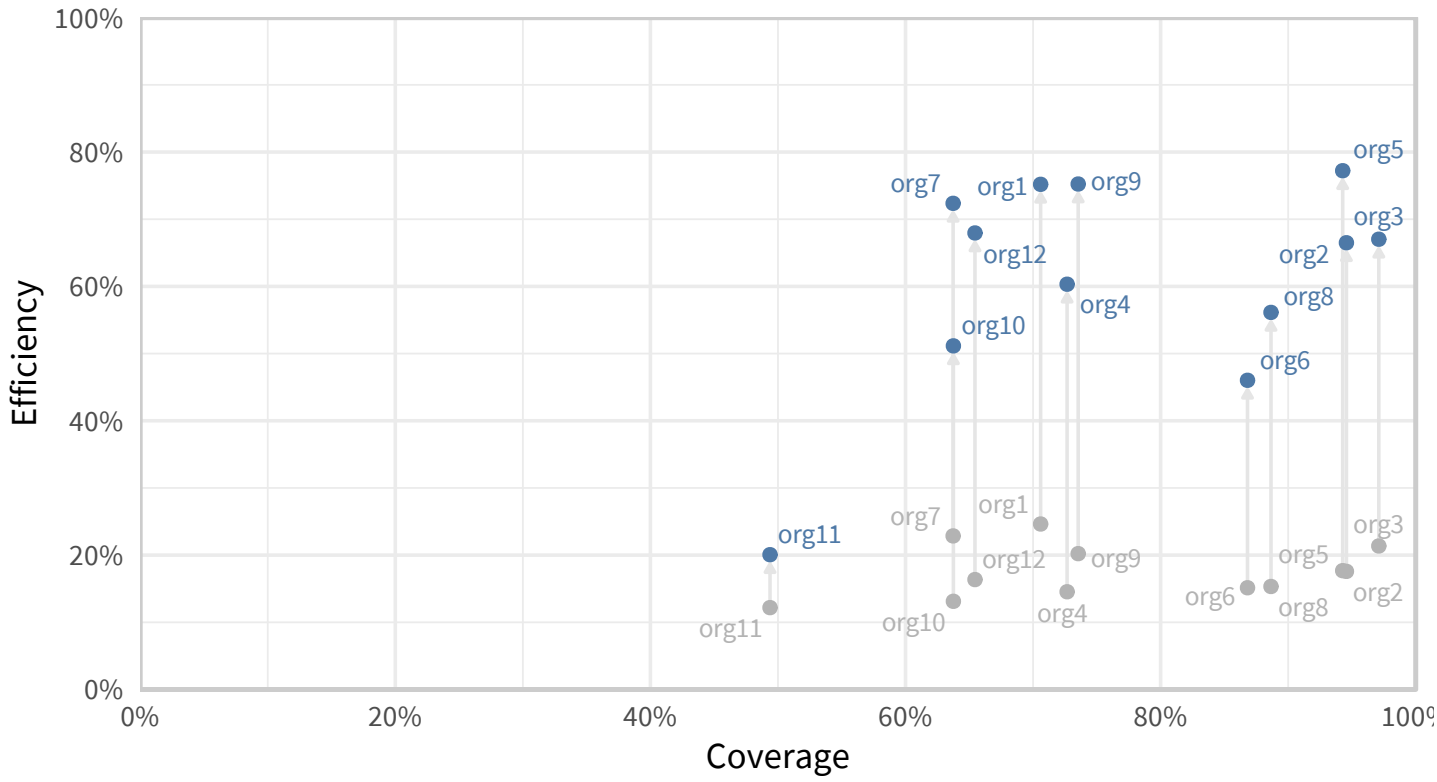
Patching Is Inefficient (But Not Really)

We'll spoil the punchline and give that answer right up front: patches are inefficient. At least, when it comes to unmodified calculations of coverage and efficiency. Allow us to explain and then we'll modify the math and share an updated Figure 15. While we often speak of remediating vulnerabilities, the actual decision is often whether or not to apply a patch. The relationship between vulnerabilities and patches is often not a simple one-to-one. In the hundreds of thousands of patches we've collected, about half fix two or more CVEs. Let's say the patch you deploy fixes five CVEs and only one of those is exploited. According to the raw efficiency calculation, you chose "wrong" four out of five times. But you really didn't choose to remediate those other four, so your efficiency metric is inherently penalized by patching. Hence the title of this section.

But we all know patching isn't really inefficient, so we have a challenge to calculate a cleaner efficiency rating that accounts for patches. We have the data to do this, starting with which CVEs have been observed and/or exploited. We also have data linking CVEs to observed vulnerabilities as well as CVEs to one or more patches. The result is Figure 16, which shows the jump in efficiency once we account for patches.

FIGURE 16

Coverage/Efficiency plot for our 12 example organizations modified to account for patching



Source: Kenna / Cyentia

The lift in efficiency via this alternate method that accounts for patches is readily apparent in Figure 16. We believe this offers a more realistic view of coverage and efficiency for the organizations depicted. It is very interesting to note that the degree of change on the efficiency axis isn't uniform. Some organizations are clearly making more efficient patch-centric remediation decisions than others. And it's not just an industry thing either; they intermingle substantially. Focusing on the alternate (upper) positioning of firms, we find the variation among them very intriguing. The range between the minimum and maximum is roughly 50% for both coverage and efficiency. This suggests vulnerability management programs really do matter and a real, measurable improvement can be gained by making smarter remediation decisions. That's music to our ears and we hope yours too!

Life & Death for Vulnerabilities

The previous section does a good job showing where organizations stand in terms of remediating vulnerabilities, but it doesn't show anything about how long it took them to get there. But not to worry, this section will fill in that critical missing element.

Think about the life of a vulnerability: Its birth is generally silent and it will remain dormant until discovered (hopefully by someone without malicious intent). Once discovered, any number of things may happen. If publicly disclosed (with or without a logo), it could enter the CVE process, which then triggers a flurry of acronyms (NVD, CVSS, CPE, etc.). It could also have an exploit developed, weaponized, and delivered in actual attacks. Defenders will generate signatures to detect such activity and vulnerability scanners will scour networks for its existence. For an individual instance of the vulnerability, the end can come swiftly in the form of a patch, upgrade, configuration change, or other workaround.

Organizations work hard to minimize the window of time between the birth (discovery) and death (remediation) of vulnerabilities in their environment. It's never a simple "patch everything" solution. Remediations must balance the characteristics of the vulnerability, criticality of associated threats, complexities of the environment, responsible parties, etc. As challenging as that sounds, the problem isn't just remediating that one vulnerability—it's dealing with the thousands of instances across every imaginable asset that can creep into an organization's network. All of this helps to explain why our data pegs the median time to remediate a vulnerability at 90 days.

A Primer on Survival Analysis

To derive that 90-day statistic (and others we will share momentarily), we employed a technique called survival analysis on the data from the same 12 organizations we studied in the previous sections. Survival analysis is a set of methods to understand the time element (duration) until an event. That event could be death in medical studies (hence "survival" analysis) or the failure of a component in a manufacturing process. In our sphere, the event of interest is the remediation (death) of a vulnerability. Here's a brief walkthrough of the basic principle and then we'll hop into the results.

If an organization observed 100 live/open vulnerabilities within its assets today (Day 0), some might be remediated before quitting time, but let's say for this example that 90 of them lived to see another day. The survivability rate on Day 0 would be 90%. As time passes and vulnerabilities continued to be killed remediated, that proportion will drop. Of course, subsequent scans/tests will identify new vulnerabilities and so there's a constant give and take over time. But survival analysis is mainly concerned with time-to-event measurements, and so the "survivability clock" for all vulnerabilities starts ticking down at Day 0, regardless of the actual date they were first observed. That should be a sufficient foundation; let's fast-forward through the (rather complex) process of survival analysis and look at Figure 17. It presents an aggregate view of survival rates for 190 million vulnerabilities observed across the 12 firms.

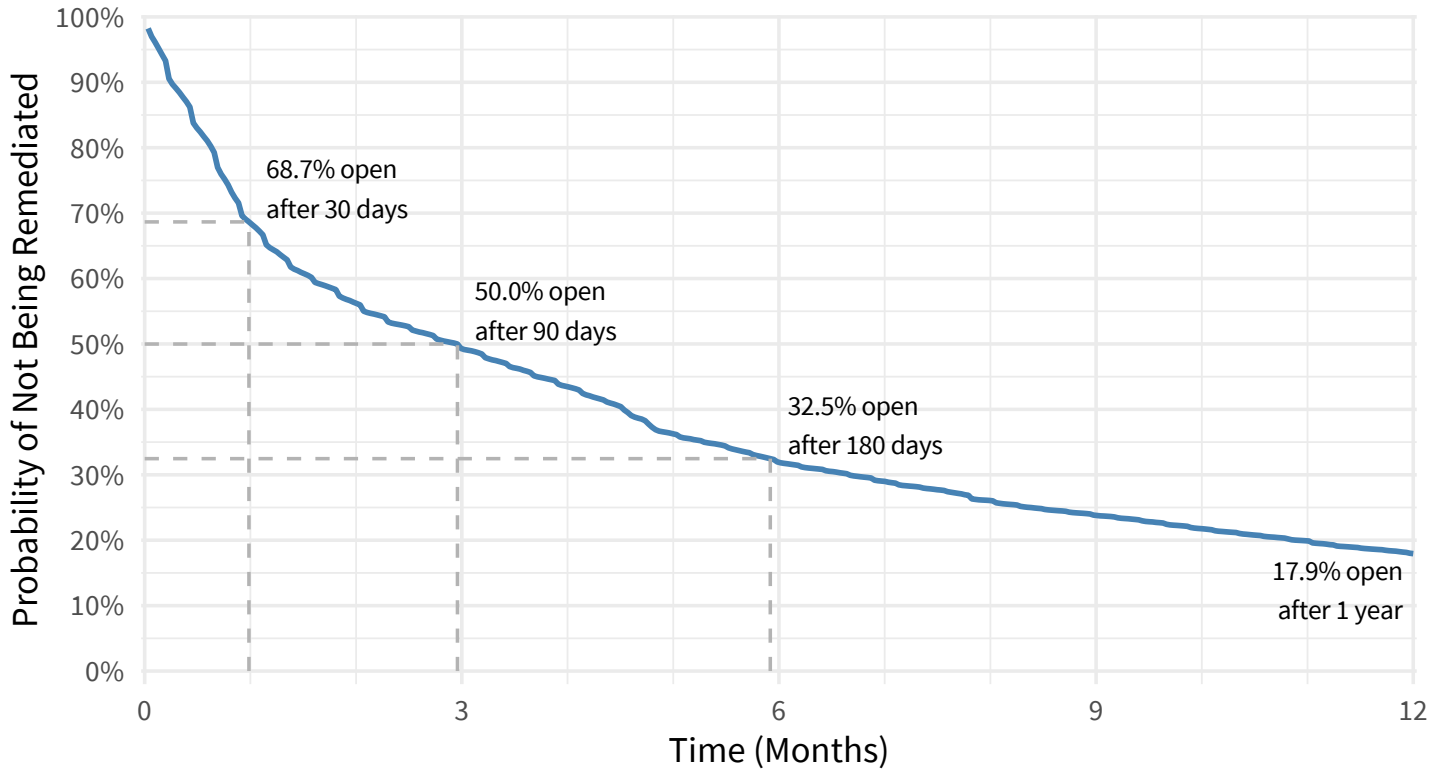
The Lifespan of Vulnerabilities

Starting at the top left of Figure 17, 100% of vulnerabilities exist at the time of discovery. Then there is a rush to remediate important assets and/or critical vulnerabilities and the line begins dropping at a rather steep pace in the first month. Just under 70% of all discovered vulnerabilities are remediated in those first 30 days. The slope of the line shifts noticeably after the 30-day point, indicating remediation efforts slow down a bit after that. Three months in, we see that 50% of vulnerabilities are remediated. To connect the dots, this is where we get the 90-day median time to remediation statistic.

It takes another three months to get a full two-thirds of all vulnerabilities remediated (we're now a total of six months after discovery). Things really slow down now, because the next six months remediate about half of what's left over (about 15% of total vulns) leaving about 18% of vulnerabilities still open after a year. It seems a hard-knock life for vulns and vuln programs alike.

FIGURE 17

Survival analysis on 190 million vulnerabilities across our 12 example firms[2]



Source: Kenna / Cyentia

None of These Are Like the Other

Figure 17 shows the overall trend of the life and death of vulnerabilities within firms. But it's an average, and the problem with averages is that they're so...average. Averages can help describe the whole, but they can be misleading at the individual level by reducing the spread of reality into a single number (or line). Just as nobody has exactly 2.5 children (the worldwide average), no two firms will follow the exact same vulnerability remediation path. But how closely do organizations track to the average? Figure 18 has the answer and one big message: There is a lot of variation in remediation strategies across organizations and no two organizations are alike.

Again, let's start at the upper left, where even in the first month we see a ton of variation from the average (gray dotted line) and also among organizations. Org8 managed to close 50% of its vulnerabilities in the first month. A more "cautious" approach to remediation is seen at the top of the cluster where Org1 remediated only 10% of its vulnerabilities in the first month.

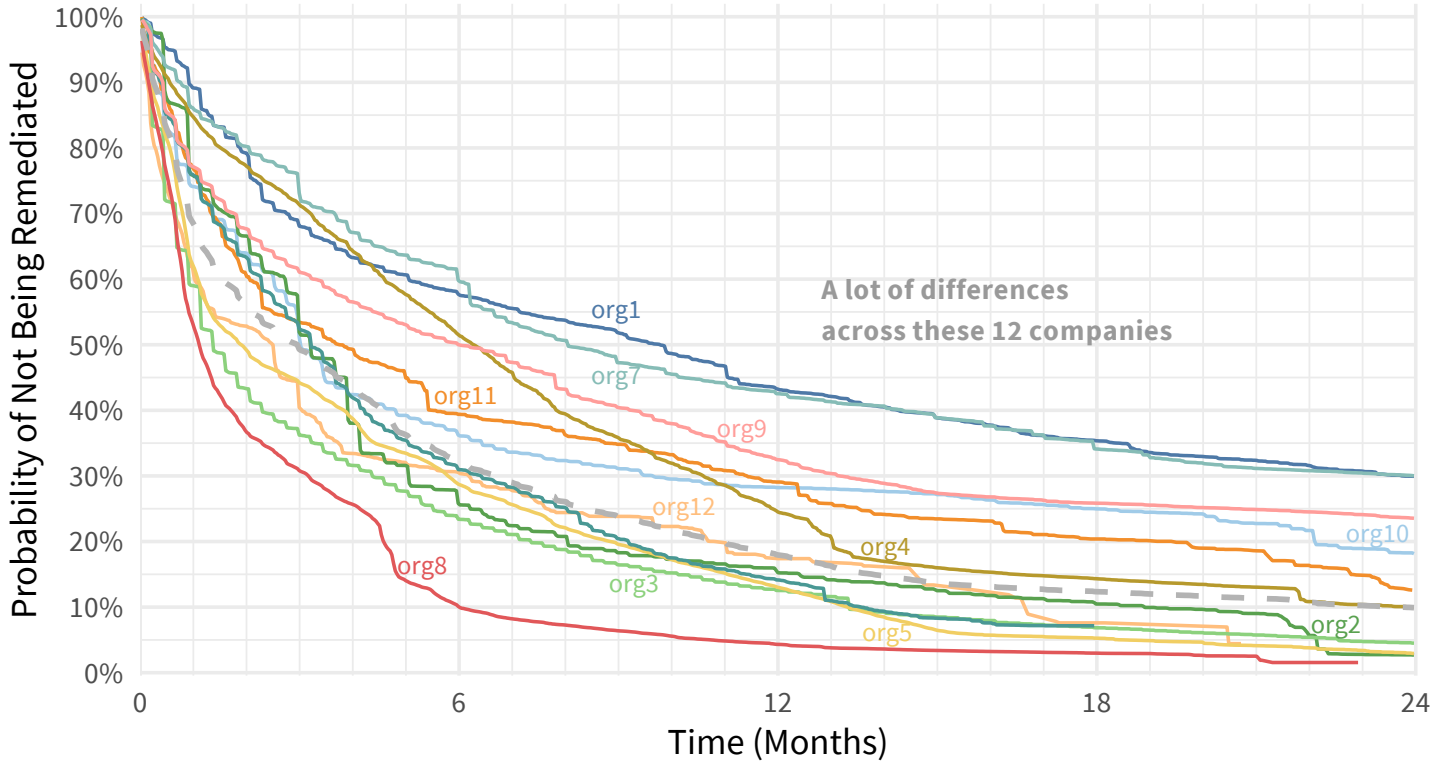
Going out to three months, Org8 closed out 70% of its vulnerabilities, while Org7 flips that ratio with a 30% closure rate (70% still open). Proceeding on to the 12-month mark, look at the spread. Only a scant 5% of vulnerabilities are still open in Org8 after a year, while several others are still struggling with 4X to 8X that proportion of open vulnerabilities.

[2] One thing that is interesting in this survival curve is the tiny little bumps in the line that are visible along the whole line. This is an amplification of a weekly scanning schedule used by the majority of organizations. Even though vulnerabilities may be remediated at any point, the scanners will report them closed on the schedule.

It's also interesting to note that the trajectory of remediation is not set in stone at the beginning. Org4, for instance, starts behind the global curve but more or less follows it in later stages. The opposite can be said of Org10. This chart can also be viewed left to right. For example, look at the 50% mark on the vertical axis. One organization remediates 50% of its vulns in just about a month, while on the far right we see another organization that takes about 10 months to get 50% closed.

FIGURE 18

Survival analysis on vulnerabilities observed by each of our 12 sample firms

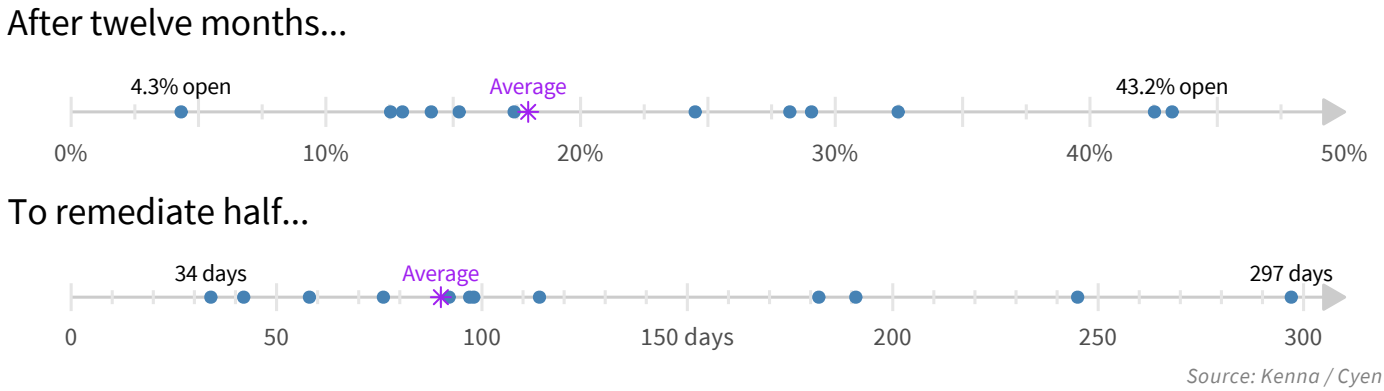


Source: Kenna / Cyentia

There's a lot going on in Figure 18, which may hinder simple or quick observations. We hope Figure 19 will meet that need. It compares the 12 organizations (the dots) on two comparisons of interest. The first shows the proportion of vulnerabilities open for each firm at the one year mark and the second shows the number of days required by each to remediate 50% of their issues. Once again, it underscores the wide variation in performance that exists among vulnerability management programs and the benefits that can be gained from an optimized strategy.

FIGURE 19

Simplified view of survival analysis on 190 million vulnerabilities across our 12 example firms



Source: Kenna / Cyentia

Conclusion

We started the *Prioritization to Prediction* research with Kenna in the spring of 2018, and this report is just the latest installment in what will be an ongoing series over the next couple years. Despite the natural doom and gloom we may feel studying millions of failures in our technology products, the future's actually looking good from our perspective. As we mature the way we evaluate and respond to vulnerabilities, programs developed to manage them will become more successful. And that will benefit us all.

We've already learned that studying the corpus of published CVEs can provide some interesting and useful insight (and is certainly better than not studying it at all). This approach can help us evaluate various remediation strategies and improve on our overall programs to manage risk associated with vulnerabilities across our networks. But only about one-third of all the published CVEs are ever seen in a live environment and, of those, only 14% (5% overall) have known exploits against them.

Organizations strive to remediate 100% of these easily exploitable and higher-risk vulnerabilities, but fall short. Overall, firms get to about 70% of these critical vulnerabilities. Depending on your perspective, that may either be seen as good or bad. The efficiency of these efforts is fairly low, but is greatly improved by models that factor patches into the equation. Multi-bug patches boost coverage and efficiency "for free" and significantly reduce the decision space (decisions are "do I deploy this patch?" rather than "do I fix this vuln?").

We also covered the time to remediation and showed that firms exhibit quite a bit of variation. The median amount of time to fully remediate all affected assets associated with a vulnerability is three months. One-third of vulnerabilities are closed within the first month, and one-third remain open after 6 months. As with coverage and efficiency ratings, we saw substantial differences among our small sample of organizations with respect to these remediation timelines and milestones.

All of the above suggests vulnerability management programs really do matter and a real, measurable improvement can be gained by making smarter remediation decisions. Clearly, there is more work to do here and we are excited to say "to be continued."

Putting It In Practice

It's always tempting to read something like this report, mentally flag a few findings for contemplation on the drive home, and then forget about them when the next day's task list overwrites those memory blocks. To help avoid that outcome, we offer the following action items for your consideration:

1. Try to reverse-engineer your strategy or rules for vulnerability remediation decisions. If you find it overly simple (e.g., "fix CVSS7+," overly complex, rife with special cases and dependencies, etc., it's a good sign they could be improved.

2. Try measuring your vulnerability remediation efforts using objective measures, such as coverage and recall. At the very least, go through the thought exercise of "could we measure this, why or why not, and what would it show if we did?"

3. Applying the technique of survival analysis to vulnerabilities across your environment is probably too much to bite off on your own. But you might be able to get a sense of what the survival curve looks like for your organization. Next time a vulnerability gets priority remediation attention (e.g., in an out-of-cycle patch), start the clock from the date of discovery, tally the number of systems affected, and track how that changes over time through scanning, pen tests, etc. If you can, try this with a few different sources/types of vulnerabilities to see if the curve varies.

Appendix A: Data Sources

This study focuses on the vulnerabilities described in MITRE's Common Vulnerabilities and Exposures (CVE) List. But in order to provide more context to the study and help measure the importance of remediating any specific CVE, we also leverage several other sources . We describe these sources and attributes in this section.

Common Vulnerabilities and Exposures (CVE)

We focus our research on discovered and disclosed vulnerabilities contained in the CVE List from MITRE. We do this primarily because CVEs are publicly tracked, readily available, extensive (although not exhaustive), and have become the de facto standard adopted by many other projects and products. It should be noted, however, that CVEs are neither comprehensive nor perfect. Many vulnerabilities are unknown, undisclosed, or otherwise have not been assigned a CVE ID. Furthermore, CVE listings are curated by humans, which makes them vulnerable to biases, errors, and omissions.[1] Despite these challenges, the CVE List is a valuable community resource that greatly assists the otherwise untenable task of vulnerability management.

Since its inception, well over 100,000 CVE entries have been created. Another 20,000+ are still in "reserved" status, meaning they have been allocated or reserved for use by a CNA or researcher, but the details have not yet been populated. Over 4,000 have been rejected for various reasons and another eight are split out or labeled as unverifiable. We chose to include the ~500 published CVEs currently in the "disputed" state since most describe weaknesses that would be useful to an attacker. A total of 108,910 CVEs from the CVE List were utilized in this research.

For all intents and purposes, each of these published CVEs represents a decision and potential action for vulnerability management programs. The criteria for those decisions may be simple in the singular case (e.g., "Does that exist in our environment?"), but prove to be quite difficult in the aggregate (e.g., "Where do we start?"). Figure 1 reinforces this challenge by demonstrating the increasing volume of reserved and published CVEs over time.

CVE Enrichment Projects

In addition to the basic CVE information produced by MITRE, this research also leverages the details added to each CVE by the National Vulnerability Database (NVD). NVD enriches the base CVE information with details leveraging other community projects, which include the following:

Common Vulnerability Scoring System (CVSS). This provides a process to capture the principal characteristics of a vulnerability and produce a numerical score that reflects its severity. The CVSS standard has very recently moved to version 3, but the majority of published CVEs were recorded using version 2, so we use version 2 in this report. CVSS was developed and is maintained by the Forum of Incident Response and Security Teams (FIRST).

Common Platform Enumeration (CPE). This provides a standard machine-readable format for encoding names of IT products, platforms, and vendors. It was developed at MITRE, but ongoing development and maintenance is now handled by NIST.

Common Weakness Enumeration (CWE). This provides a common language for describing software security weaknesses in architecture, design, or code. It was developed and is maintained by MITRE. We won't be discussing CWEs in this study.

Each piece of enrichment data offers potentially useful context for decisions. Basic remediation strategies may rely on CVSS alone, while others will factor in the type of vulnerability (CWE) along with the vendor and product and the exposure of the vulnerabilities across environments.

[1] *For discussion of these biases and other CVE-related issues, see 2013 BlackHat presentation titled "Buying into the Bias: Why Vulnerability Statistics Suck" from Brian Martin and Steve Christey.*

Exploit Code and Activity

Basic analysis of CVEs may stop at data from MITRE and NVD, but those miss an important part of the equation: What CVEs are attackers actually exploiting in the wild? Unfortunately, no universal source of exploit activity exists, so we have to collect this information through multiple direct and indirect ways. These include host and network-based detection systems as well as by reverse engineering the malware and tools used by attackers.

Sources used in this study to track which CVEs have been actively exploited include the SANS Internet Storm Center (monthly statistics from the ISC signature collection Honeynet Project), Secureworks CTU (active campaigns associated with CVEs), Alienvaults OSSIM metadata (reputation feed, collecting IDS signature hits across 100,000+ devices in 150+ countries) and Reversing Labs metadata.

But sometimes observing exploitation in the wild comes too late for risk-averse vulnerability remediation strategies. In such cases, published exploit code serves as a good indicator of exploitability because it enables attackers to easily weaponize a vulnerability. Roughly two out of every three CVEs with active exploit detections also have published exploit code. Tracking the publication of exploit code, therefore, is important to remediation prioritization.

Sources used in this study to track which CVE's have public exploit code include Exploit DB, several exploitation frameworks (Metasploit, D2 Security's Elliot Kit, and Canvas Exploitation Framework), the Contagio dump and data from Reversing Labs, and Secureworks CTU.

Not only are exploit code releases strongly correlated with active exploitations, but they also indicate something more: the characteristics of a vulnerability that exploit writers target. Even if we haven't seen a specific CVE with published exploit code, the exploited vulnerabilities tend to share similar characteristics and traits with written exploits.

Vulnerability Observations

Information shared on observed vulnerabilities (open or closed) is drawn from the Kenna Security Platform. Hundreds of organizations use this platform as part of their vulnerability management programs. Ingested into the platform is a rich dataset from a variety of internal sources:

- Findings from any vulnerability scanner
- Asset and network-specific data from configuration management database (CMDB) tools
- Penetration test or red team findings
- Bug bounty programs
- Static application testing
- Dynamic application testing

Kenna uses all of this data to get a full view into the potential impact of each vulnerability, including the volume and velocity of attacker activity, as well as how critical each threat could be given your specific environment. We used it to derive the findings and statistics we shared in this report.

PRIORITIZATION TO PREDICTION

VOLUME 2: GETTING REAL ABOUT REMEDIATION

KENNA Security

CYENTIA INSTITUTE

> "Realized coverage & efficiency vary greatly among firms—over 50% between top and bottom performers—indicating different remediation strategies lead to very different outcomes."

Where is your strategy leading?